TYPICAL QUESTIONS & ANSWERS

OBJECTIVE TYPE QUESTIONS

Each question carries 2 marks.

Choose the correct or best alternative in the following:

Q.1   In the relational modes, cardinality is termed as:
        (A) Number of tuples.   **(B)** Number of attributes.
        (C) Number of tables.   **(D)** Number of constraints.

        Ans: A

Q.2   Relational calculus is a
        (A) Procedural language.   **(B)** Non- Procedural language.
        (C) Data definition language.   **(D)** High level language.

        Ans: B

Q.3   The view of total database content is
        (A) Conceptual view.   **(B)** Internal view.
        (C) External view.   **(D)** Physical View.

        Ans: A

Q.4   Cartesian product in relational algebra is
        (A) a Unary operator.   **(B)** a Binary operator.
        (C) a Ternary operator.   **(D)** not defined.

        Ans: B   Cartesian product in relational algebra is a binary operator.
            (It requires two operands. e.g., P X Q)

Q.5   DML is provided for
        (A) Description of logical structure of database.
        (B) Addition of new structures in the database system.
        (C) Manipulation & processing of database.
        (D) Definition of physical structure of database system.

        Ans: C   DML is provided for manipulation & processing of database.
            (Data stored in the database is processed or manipulated using data manipulation
            language commands as its name)

Q.6   'AS' clause is used in SQL for
       (A) Selection operation.   **(B)** Rename operation.
       (C) Join operation.   **(D)** Projection operation.

       Ans: B   'AS' clause is used in SQL for rename operation.
               (e.g., SELECT ENO AS EMPLOYEE_NO FROM EMP)

Q.7   ODBC stands for
       (A) Object Database Connectivity.
       (B) Oral Database Connectivity.
       (C) Oracle Database Connectivity.
       (D) Open Database Connectivity.

       Ans: D

Q.8   Architecture of the database can be viewed as
       (A) two levels.   **(B)** four levels.
       (C) three levels.   **(D)** one level.

       Ans: C

Q.9   In a relational model, relations are termed as
       (A) Tuples.   **(B)** Attributes
       (C) Tables.   **(D)** Rows.

       Ans:

Q.10   The database schema is written in
       (A) HLL   **(B)** DML
       (C) DDL   **(D)** DCL

       Ans: C

Q.11   In the architecture of a database system external level is the
       (A) physical level.   **(B)** logical level.
       (C) conceptual level   **(D)** view level.

       Ans: D

Q.12   An entity set that does not have sufficient attributes to form a primary key is a
       (A) strong entity set.   **(B)** weak entity set.
       (C) simple entity set.   **(D)** primary entity set.

       Ans: B

Q.13   In a Hierarchical model records are organized as
       (A) Graph.   **(B)** List.
       (C) Links.   **(D)** Tree.

Ans: D

Q.14 In an E-R diagram attributes are represented by
(A) rectangle.   (B) square.
(C) ellipse.   (D) triangle.

Ans: C

Q.15 In case of entity integrity, the primary key may be
(A) not Null   (B) Null
(C) both Null & not Null.   (D) any value.

Ans: A

Q.16 In tuple relational calculus $P_1 \to P_2$ is equivalent to
(A) $\neg P_1 \vee P_2$ (B) $P_1 \vee P_2$
(C) $P_1 \wedge P_2$ (D) $P_1 \wedge \neg P_2$

Ans: A In tuple relational calculus $P_1 \to P_2$ is equivalent to $\neg P_1 \vee P_2$.
(The logical implication expression A → B, meaning if A then B,is equivalent to $\neg A \vee B$)

Q.17 The language used in application programs to request data from the DBMS is referred to as the
(A) DML   (B) DDL
(C) VDL   (D) SDL

Ans: A

Q.18 A logical schema
(A) is the entire database.
(B) is a standard way of organizing information into accessible parts.
(C) describes how data is actually stored on disk.
(D) both (A) and (C)

Ans: A

Q.19 Related fields in a database are grouped to form a
(A) data file.   (B) data record.
(C) menu.   (D) bank.

Ans: B Related data fields in a database are grouped to form a data record.
(A record is a collection of related fields)

Q.20 The database environment has all of the following components except:
(A) users.   (B) separate files.
(C) database.   (D) database administrator.

Ans: A

Q.21 The language which has recently become the defacto standard for interfacing application programs with relational database system is
(A) Oracle.
(B) SQL.
(C) DBase.
(D) 4GL.

Ans: B

Q.22 The way a particular application views the data from the database that the application uses is a
(A) module.
(B) relational model.
(C) schema.
(D) sub schema.

Ans: D

Q.23 In an E-R diagram an entity set is represent by a
(A) rectangle.
(B) ellipse.
(C) diamond box.
(D) circle.

Ans: A

Q.24 A report generator is used to
(A) update files.
(B) print files on paper.
(C) data entry.
(D) delete files.

Ans: B

Q.25 The property / properties of a database is / are :
(A) It is an integrated collection of logically related records.
(B) It consolidates separate files into a common pool of data records.
(C) Data stored in a database is independent of the application programs using it.
(D) All of the above.

Ans: D

Q.26 The DBMS language component which can be embedded in a program is
(A) The data definition language (DDL).
(B) The data manipulation language (DML).
(C) The database administrator (DBA).
(D) A query language.

Ans: B

Q.27 A relational database developer refers to a record as
(A) a criteria.
(B) a relation.
(C) a tuple.
(D) an attribute.

Ans: C

Q.28 The relational model feature is that there

(A) is no need for primary key data.
(B) is much more data independence than some other database models.
(C) are explicit relationships among records.
(D) are tables with many dimensions.

Ans: B

Q.29 Conceptual design
    (A) is a documentation technique.
    (B) needs data volume and processing frequencies to determine the size of the database.
    (C) involves modelling independent of the DBMS.
    (D) is designing the relational model.

    Ans:C

Q.30 The method in which records are physically stored in a specified order according to a key field in each record is
    (A) hash.   **(B)** direct.
    (C) sequential.   **(D)** all of the above.

    Ans: A A method in which records are physically stored in a specified order according to a key field in each record is hash.
    (In hash method, a hash function is performed on the key value to determine the unique physical address of the record to store or retrieve)

Q.31 A subschema expresses
    (A) the logical view.   **(B)** the physical view.
    (C) the external view.   **(D)** all of the above.

    Ans: C A subschema expresses the external view.
    (External schemas are called also called as subschemas)

Q.32 Count function in SQL returns the number of
    (A) values.   **(B)** distinct values.
    (C) groups.   **(D)** columns.

    Ans: A Count function in SQL returns the number of values.
    (Count function counts all the not null values in the specific column. If we want to count only distinct values than the DISTINCT keyword is also to be used)

Q.33 Which one of the following statements is false?
    (A) The data dictionary is normally maintained by the database administrator.
    (B) Data elements in the database can be modified by changing the data dictionary.
    (C) The data dictionary contains the name and description of each data element.
    (D) The data dictionary is a tool used exclusively by the database administrator.

    Ans: B

Q.34 An advantage of the database management approach is
      (A) data is dependent on programs.
      (B) data redundancy increases.
      (C) data is integrated and can be accessed by multiple programs.
      (D) none of the above.

      Ans: C

Q.35 A DBMS query language is designed to
      (A) support end users who use English-like commands.
      (B) support in the development of complex applications software.
      (C) specify the structure of a database.
      (D) all of the above.

      Ans: D

Q.36 Transaction processing is associated with everything below except
      (A) producing detail, summary, or exception reports.
      (B) recording a business activity.
      (C) confirming an action or triggering a response.
      (D) maintaining data.

      Ans: C

Q.37 It is possible to define a schema completely using
      (A) VDL and DDL.   **(B)** DDL and DML.
      (C) SDL and DDL.   **(D)** VDL and DML.

      Ans: B

Q.38 The method of access which uses key transformation is known as
      (A) direct.   **(B)** hash.
      (C) random.   **(D)** sequential.

      Ans: B

Q.39 Data independence means
      (A) data is defined separately and not included in programs.
      (B) programs are not dependent on the physical attributes of data.
      (C) programs are not dependent on the logical attributes of data.
      (D) both (B) and (C).

      Ans: D both (B) and (C)

Q.40 The statement in SQL which allows to change the definition of a table is
      (A) Alter.   **(B)** Update.
      (C) Create.   **(D)** select.

      Ans: A

Q.41   E-R model uses this symbol to represent weak entity set ?
             (A) Dotted rectangle.
             (B) Diamond
             (C) Doubly outlined rectangle
             (D) None of these

             Ans: C

Q.42   SET concept is used in :
             (A) Network Model
             (B) Hierarchical Model
             (C) Relational Model
             (D) None of these

             Ans: A

Q.43   Relational Algebra is
             (A) Data Definition Language **.**
             (B) Meta Language
             (C) Procedural query Language
             (D) None of the above

             Ans: C

Q.44   Key to represent relationship between tables is called
             (A) Primary key   **(B)** Secondary Key
             (C) Foreign Key   **(D)** None of these

             Ans: C

Q.45   _____ produces the relation that has attributes of R1 and R2
             (A) Cartesian product   **(B)** Difference
             (C) Intersection   **(D)** Product

             Ans: A

Q.46   The file organization that provides very fast access to any arbitrary record of a file is
             (A) Ordered file   **(B)** Unordered file
             (C) Hashed file   **(D)** B-tree

             Ans: C

Q.47   DBMS helps achieve
             (A) Data independence   **(B)** Centralized control of data
             (C) Neither (A) nor (B)   **(D)**   both (A) and (B)

             Ans: D

Q.48   Which of the following are the properties of entities?
             (A) Groups   **(B)** Table

(C) Attributes   **(D)** Switchboards

Ans: C

Q.49   In a relation
(A) Ordering of rows is immaterial
(B) No two rows are identical
(C) (A) and (B) both are true
(D) None of these**.**

Ans: C

Q.50   Which of the following is correct:
(A) a SQL query automatically eliminates duplicates.
(B) SQL permits attribute names to be repeated in the same relation.
(C) a SQL query will not work if there are no indexes on the relations
(D) None of these

Ans: D

Q.51   It is better to use files than a DBMS when there are
(A) Stringent real-time requirements.
(B) Multiple users wish to access the data.
(C) Complex relationships among data.
(D) All of the above.

Ans: B

Q.52   The conceptual model is
(A) dependent on hardware.
(B) dependent on software.
(C) dependent on both hardware and software   .
(D) independent of both hardware and software.

Ans: D

Q.53   What is a relationship called when it is maintained between two entities?
(A) Unary   **(B)** Binary
(C) Ternary   **(D)** Quaternary

Ans: B

Q.54   Which of the following operation is used if we are interested in only certain columns of a table?
(A) PROJECTION   **(B)** SELECTION
(C) UNION   **(D)** JOIN

Ans: A

Q.55   Which of the following is a valid SQL type?
             (A) CHARACTER   **(B)** NUMERIC
             (C) FLOAT   **(D)** All of the above

             Ans: D

Q.56   The RDBMS terminology for a row is
             (A) tuple.   **(B)** relation.
             (C) attribute.   **(D)** degree.

             Ans: A

Q.57   Which of the following operations need the participating relations to be union compatible?
             (A) UNION   **(B)** INTERSECTION
             (C) DIFFERENCE   **(D)** All of the above

             Ans: D

Q.58   The full form of DDL is
             (A Dynamic Data Language   **(B)** Detailed Data Language
             (C) Data Definition Language   **(D)** Data Derivation Language

             Ans: C

Q.59   Which of the following is an advantage of view?
             (A) Data security   **(B)** Derived columns
             (C) Hiding of complex queries   **(D)** All of the above

             Ans: D

Q.60   Which of the following is a legal expression in SQL?
             (A) SELECT NULL FROM EMPLOYEE;
             (B) SELECT NAME FROM EMPLOYEE;
             (C) SELECT NAME FROM EMPLOYEE WHERE SALARY = NULL;
             (D) None of the above

             Ans: B

Q.61   The users who use easy-to-use menu are called
             (A) Sophisticated end users.   **(B)** Naïve users.
             (C) Stand-alone users.   **(D)** Casual end users.

             Ans: B

Q.62   Which database level is closest to the users?
             (A) External   **(B)** Internal
             (C) Physical   **(D)** Conceptual

             Ans: A

Q.63 Which are the two ways in which entities can participate in a relationship?
(A) Passive and active   (B) Total and partial
(C) Simple and Complex   (D) All of the above

Ans: B

Q.64 The result of the UNION operation between R1 and R2 is a relation that includes
(A) all the tuples of R1
(B) all the tuples of R2
(C) all the tuples of R1 and R2
(D) all the tuples of R1 and R2 which have common columns

Ans: D

Q.65 Which of the following is a comparison operator in SQL?
(A) =   (B) LIKE
(C) BETWEEN   (D) All of the above

Ans: D

Q.66 A set of possible data values is called
(A) attribute.   (B) degree.
(C) tuple.   (D) domain.

Ans: D

Q.67 Which of the operations constitute a basic set of operations for manipulating relational data?
(A) Predicate calculus   (B) Relational calculus
(C) Relational algebra   (D) None of the above

Ans:C

Q.68 Which of the following is another name for weak entity?
(A) Child   (B) Owner
(C) Dominant   (D) All of the above

Ans: A

Q.69 Which of the following database object does not physically exist?
(A) base table   (B) index
(C) view   (D) none of the above

Ans: C

Q.70 NULL is
(A) the same as 0 for integer
(B) the same as blank for character
(C) the same as 0 for integer and blank for character

(D) not a value

Ans: D

Q.71 Which of the following is record based logical model?
(A) Network Model   **(B)** Object oriented model
(C) E-R Model   **(D)** None of these

Ans: A

Q.72 A data dictionary is a special file that contains:
(A) The name of all fields in all files.
(B) The width of all fields in all files.
(C) The data type of all fields in all files.
(D) All of the above.

Ans: D

Q.73 A file manipulation command that extracts some of the records from a file is called
(A) SELECT   **(B)** PROJECT
(C) JOIN   **(D)** PRODUCT

Ans: A

Q.74 The physical location of a record is determined by a mathematical formula that transforms a file key into a record location is :
(A) B-Tree File   **(B)** Hashed File
(C) Indexed File   **(D)** Sequential file.

Ans: B

Q.75 Using Relational Algebra the query that finds customers, who have a balance of over 1000 is
(A) $\Pi_{\text{Customer\_name}}(\sigma_{\text{balance} > 1000}(\text{Deposit}))$
(B) $\sigma_{\text{Customer\_name}}(\Pi_{\text{balance} > 1000}(\text{Deposit}))$
(C) $\Pi_{\text{Customer\_name}}(\sigma_{\text{balance} > 1000}(\text{Borrow}))$
(D) $\sigma_{\text{Customer\_name}}(\Pi_{\text{balance} > 1000}(\text{Borrow}))$

Ans: **A**

Q.76 A primary key is combined with a foreign key creates
(A) Parent-Child relation ship between the tables that connect them.
(B) Many to many relationship between the tables that connect them.
(C) Network model between the tables that connect them.
(D) None of the above.

Ans: A

Q.77 In E-R Diagram derived attribute are represented by

(A) Ellipse   (B) Dashed ellipse
(C) Rectangle   (D) Triangle

Ans B

Q.78   Cross Product is a:
(A)   Unary Operator   (B) Ternary Operator
(C)   Binary Operator   (D) Not an operator

Ans: C

Q.79   An instance of relational schema R (A, B, C) has distinct values of A including NULL
values. Which one of the following is true?
(A) A is a candidate key   (B)   A is not a candidate key
(C) A is a primary Key   (D)   Both (A) and (C)

Ans: B

Q.80   Consider the join of a relation R with relation S. If R has m tuples and S has n tuples,
then the maximum size of join is:
(A)   mn   (B) m+n
(C)   (m+n)/2   (D) 2(m+n)

Ans: A

Q.81   The natural join is equal to :
(A)   Cartesian Product
(B)   Combination of Union and Cartesian product
(C) Combination of selection and Cartesian product
(D) Combination of projection and Cartesian product

Ans: D

Q.82   Which one of the following is not true for a view:
(A)   View is derived from other tables.
(B)   View is a virtual table.
(C) A view definition is permanently stored as part of the database.
(D) View never contains derived columns.

Ans: C

Q.83   A primary key if combined with a foreign key creates
(A)   Parent-Child relationship between the tables that connect them.
(B)   Many to many relationship between the tables that connect them.
(C)   Network model between the tables that connect them.
(D)   None of the above.

Ans: A

Q.84 In E-R Diagram relationship type is represented by
(A) Ellipse (B) Dashed ellipse
(C) Rectangle (D) Diamond

Ans: D

Q.85 Hierarchical model is also called
(A) Tree structure (B) Plex Structure
(C) Normalize Structure (D) Table Structure

Ans: A

Q.86 To delete a particular column in a relation the command used is:
(A) UPDATE (B) DROP
(C) ALTER (D) DELETE

Ans: C

Q.87 The _____ operator is used to compare a value to a list of literals values that have been specified.
(A) BETWEEN (B) ANY
(C) IN (D) ALL

Ans: A

Q.88 A logical schema
A) is the entire database
B) is a standard way of organizing information into a accessible part
C) describe how data is actually stored on disk
D) none of these

Ans: D

Q.89 A B-tree of order m has maximum of _____ children
(A) m (B) m+1
(C) m-1 (D) m/2

Ans: A

Q.90 _____ function divides one numeric expression by another and returns the remainder.
(A) POWER (B) MOD
(C) ROUND (D) REMAINDER

Ans: B

Q.91 A data manipulation command the combines the records from one or more tables is called
(A) SELECT (B) PROJECT
(C) JOIN (D) PRODUCT

Ans: C

Q.92   In E-R diagram generalization is represented by
            (A)   Ellipse   **(B)**   Dashed ellipse
            (C)   Rectangle   **(D)**   Triangle

            Ans: D

Q.93   _____ is a virtual table that draws its data from the result of an SQL
            SELECT statement.
            (A)   View   **(B)**   Synonym
            (C)   Sequence   **(D)**   Transaction

            Ans: A

Q.94   The method of access which uses key transformation is known as
            (A) Direct   **(B)**   Hash
            (C) Random   **(D)**   Sequential

            Ans: B

Q.95   A table joined with itself is called
            (A) Join   **(B)** Self Join
            (C) Outer Join   **(D)** Equi Join

            Ans: B

Q.96   _____ data type can store unstructured data
            (A)   RAW   **(B)**   CHAR
            (C)   NUMERIC   **(D)**   VARCHAR

            Ans: A

Q.97   Which two files are used during operation of the DBMS
            (A)   Query languages and utilities
            (B)   DML and query language
            (C)   Data dictionary and transaction log
            (D)   Data dictionary and query language

            Ans: C

DESCRIPTIVES

Q.1   What is a database? Describe the advantages and disadvantages of using of DBMS.   **(7)**

Ans: Database – A database is a collection of related data and/or information stored so that it is available to many users for different purposes.

Advantages Of DBMS

1. **Centralized Management and Control** - One of the main advantages of using a database system is that the organization can exert, via the DBA, centralized management and control over the data.
2. **Reduction of Redundancies and Inconsistencies -** Centralized control avoids unnecessary duplication of data and effectively reduces the total amount of data storage required. Removing redundancy eliminates inconsistencies.
3. **Data Sharing** - A database allows the sharing of data under its control by any number of application programs or users.
4. **Data Integrity** - Data integrity means that the data contained in the database is both accurate and consistent. Centralized control can also ensure that adequate checks are incorporated in the DBMS to provide data integrity.
5. **Data Security** - Data is of vital importance to an organization and may be confidential. Such confidential data must not be accessed by unauthorized persons. The DBA who has the ultimate responsibility for the data in the DBMS can ensure that proper access procedures are followed. Different levels of security could be implemented for various types of data and operations.
6. **Data Independence** - Data independence is the capacity to change the schema at one level of a database system without having to change the schema at the next level. It is usually considered from two points of view: physical data independence and logical data independence. Physical data independence is the capacity to change the internal schema without having to change conceptual schema. Logical data independence is the capacity to change the conceptual schema without having to change external schemas or application programs.
7. **Providing Storage Structures for Efficient Query Processing -** Database systems provide capabilities for efficiently executing queries and updates. Auxiliary files called *indexes* are used for this purpose.
8. **Backup and Recovery -** These facilities are provided to recover databases from hardware and/or software failures.

Some other advantages are:
   - Reduced Application Development Time
   - Flexibility
   - Availability of up-to-date Information

Disadvantages Of DBMS

1. **Cost of Software/Hardware and Migration** - A significant disadvantage of the DBMS system is cost.

2. **Reduced Response and Throughput** - The processing overhead introduced by the DBMS to implement security, integrity, and sharing of the data causes a degradation of the response and throughput times.
3. **Problem with Centralization** - Centralization also means that the data is accessible from a single source namely the database. This increases the potential of security breaches and disruption of the operation of the organization because of downtimes and failures.

Q.2   Explain five duties of Database Administrator.   **(7)**

Ans:
1. DBA administers the three levels of the database and, in consultation with the overall user community, sets up the definition of the global view or conceptual level of the database.
2. Mappings between the internal and the conceptual levels, as well as between the conceptual and external levels, are also defined by the DBA.
3. DBA ensures that appropriate measures are in place to maintain the integrity of the database and that the database is not accessible to unauthorized users.
4. DBA is responsible for granting permission to the users of the database and stores the profile of each user in the database.
5. DBA is responsible for defining procedures to recover the database from failures with minimal loss of data.

Q.3   Explain the terms primary key, candidate key and foreign key. Give an example for each.   **(7)**

Ans: **Primary Key** – Primary key is one of the candidate keys that uniquely identifies each row in the relation.
Candidate Key – A candidate key of an entity set is a minimal superkey, that uniquely identifies each row in the relation.
Foreign Key – Let there are two relations (tables) *R* and *S*. Any candidate key of the relation *R* which is referred in the relation *S* is called the *foreign key* in the relation *S* and *referenced key* in the relation *R*. The relation *R* is also called as *parent table* and relation *S* is also called as *child table*.
For example:

STUDENT

Enrl No	Roll No	Name	City	*Mobile*
11	17	Ankit Vats	Delhi	9891663808
15	16	Vivek Rajput	Meerut	9891468487
6	6	Vanita	Punjab	
33	75	Bhavya	Delhi	9810618396

GRADE

Roll No	Course	*Grade*
6	C	A
17	VB	C
75	VB	A
6	DBMS	B
16	C	B

- Roll No is the primary key in the relation STUDENT and Roll No + Course is the primary key of the relation GRADE.
- Enrl No and Roll No are the candidate keys of the relation STUDENT.
- Roll No in the relation GRADE is a foreign key whose values must be one of those of the relation STUDENT.

Q.4 Differentiate between logical database design and physical database design. Show how this separation leads to data independence. **(7)**

Ans:

Basis	Logical Database Design	Physical Database Design
Task	Maps or transforms the conceptual schema (or an ER schema) from the high-level data model into a relational database schema.	The specifications for the stored database in terms of physical storage structures, record placement, and indexes are designed.
Choice of criteria	The mapping can proceed in two stages: - System-independent mapping but data model-dependent - Tailoring the schemas to a specific DBMS	The following criteria are often used to guide the choice of physical database design options: - Response Time - Space Utilization - Transaction Throughput
Result	DDL statements in the language of the chosen DBMS that specify the conceptual and external level schemas of the database system. But if the DDL statements include some physical design parameters, a complete DDL specification must wait until after the physical database design phase is completed.	An initial determination of storage structures and the access paths for the database files. This corresponds to defining the internal schema in terms of Data Storage Definition Language.

The database design is divided into several phases. The logical database design and physical database design are two of them. This separation is generally based on the concept of three-level architecture of DBMS, which provides the data independence. Therefore, we can say that this separation leads to data independence because the output of the logical database design is the conceptual and external level schemas of the database

system which is independent from the output of the physical database design that is internal schema.

Q.5 Consider the following relation schemes: **(2×7=14)**
Project (Project#, Project_name, chief_architect)
Employee (Emp#, Empname)
Assigned_To (Project#, Emp#)
Give expression in Tuple calculus and Domain calculus for each of the queries below:
(i) Get the employee numbers of employees who work on all projects.
(ii) Get the employee numbers of employees who do not work on the COMP123 project.

Ans:

(i) Tuple Calculus:
{t[Emp#] | t ∈ ASSIGNED_TO ∧ ∀p (p ∈ PROJECT → ∃u (u ∈ ASSIGNED_TO ∧ p[Project#] = u[Project#] ∧ t[Emp#] = u[Emp#]))}
Domain Calculus:
    {e | ∃p (<p, e> ∈ ASSIGNED_TO ∧ ∀p_1 (<p_1, n_1, c_1> ∈ PROJECT
        → <p_1, e> ∈ ASSIGNED_TO))}
(ii) Tuple Calculus:
{t[Emp#] | t ∈ ASSIGNED_TO ∧ ¬∃u (u ∈ ASSIGNED_TO
                ∧ u[Project#] = 'COMP123' ∧ t[Emp#] = u[Emp#])}
Domain Calculus:
{e | ∃p (<p, e> ∈ ASSIGNED_TO ∧ ∀p_1, e_1 (<p_1, e_1> ∉ ASSIGNED_TO
∨ p_1 ≠ 'COMP123' ∨ e_1 ≠ e))}

Q.6 What is ODBC? How does Oracle act as ODBC and give examples of front end uses with ODBC. **(7)**

Ans:
ODBC – Open DataBase Connectivity (ODBC) enable the integration of SQL with a general-purpose programming language. ODBC expose database capabilities in a standardized way to the application programmer through an application programming interface (API). Using ODBC, an application can access not just one DBMS but several different ones simultaneously.
ODBC achieve portability at the level of the executable by introducing an extra level of indirection. All direct interaction with a specific DBMS happens through a DBMS-specific driver. A driver is a software program that translates the ODBC calls into DBMS-specific calls. Drivers are loaded dynamically on demand since the DBMSs the application is going to access are known only at run-time. Available drivers are registered with a driver manager. The Oracle database driver translates the SQL commands from the application into equivalent commands that the Oracle DBMS understands and takes the result from the DBMS and translate into equivalent form for the application.
Example: Let there be a DSN named EMPLOYEE through, which we want to access the Oracle database in Visual Basic.
Dim CN As New ADODB.Connection
Dim RS As New ADODB.Recordset
CN.Open "DSN=employee", "scott", "tiger"

RS.Open "Select * From Emp", CN

Q.7 Define the five basic operators of relational algebra with an example each. **(7)**

Ans: Five basic operators of relational algebra are:

1. **Union (∪) -** Selects tuples that are in either P or Q or in both of them. *The duplicate tuples are eliminated.*

$R = P \cup Q$

2. **Minus (–) -** Removes common tuples from the first relation.

$R = P - Q$

3. **Cartesian Product or Cross Product (✕) -** The cartesian product of two relations is the concatenation of tuples belonging to the two relations and consisting of all possible combination of the tuples**.**

$R = P \times Q$

For Example:

P:

ID	Name
101	Jones
103	Smith
104	Lalonde

Q:

ID	Name
100	John
104	Lalonde

R = P ∪ Q

ID	Name
100	John
101	Jones
103	Smith
104	Lalonde

R = P – Q

ID	Name
101	Jones
103	Smith

R = P ✕ Q

P.ID	P.Name	Q.ID	Q.Name
101	Jones	100	John
101	Jones	104	Lalonde
103	Smith	100	John
103	Smith	104	Lalonde
104	Lalonde	100	John
104	Lalonde	104	Lalonde

4. **Projection (π) -** The projection of a relation is defined as a projection of all its tuples over some set of attributes, i.e., it yields a *vertical subset* of the relation. It is used to either *reduce* the number of attributes (degree) in the resultant relation or to *reorder* attributes. The projection of a relation T on the attribute A is denoted by $\pi_A(T)$.

5. **Selection (σ) -** Selects only some of the tuples, those satisfy given criteria, from the relation. It yields a *horizontal subset* of a given relation, i.e., the action is defined over a complete set of attribute names but only a subset of the tuples are included in the result.

$R = \sigma_B(P)$

For Example:

EMPLOYEE:

Id	Name
101	Jones
103	Smith
104	Lalonde
106	Byron

→

Name
Jones
Smith
Lalonde
Byron

Projection of relation EMPLOYEE over attribute Name

EMPLOYEE:

Id	Name
101	Jones
103	Smith
104	Lalonde
106	Byron

→

Result of Selection

Id	Name
104	Lalonde
106	Byron

Result of Selection over **EMPLOYEE** for **ID > 103**

Q.8   Explain entity integrity and referential integrity rules in relational model. Show how these are realized in SQL.   **(7)**

Ans:
Entity Integrity Rule – No primary key value can be null.
Referential Integrity Rule – In referential integrity, it is ensured that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.

In SQL, entity integrity and referential integrity rules are implemented as constraints on the relation called as primary key constraint and reference key constraint respectively. These constraints can be specified with relation at the time of creation of the relations or after the creation of the relations by altering the definition of the relations. For example:

CREATE TABLE DEPT

```
(DEPTNO    NUMBER PRIMARY KEY,
 DNAME     VARCHAR2(15));
```

CREATE TABLE EMP

```
(EMPNO     NUMBER PRIMARY KEY,
```

        ENAME   VARCHAR2(15),
        JOB   VARCHAR2(10),
        DEPTNO   NUMBER REFERENCES DEPT(DEPTNO));

Q.9   What are the advantages of embedded query language? Give an example of a embedded
        SQL query.   **(7)**

        Ans:
        Embedded query language – SQL can be implemented in two ways. It can be used
        interactively or embedded in a host language or by using API. The use of SQL commands
        within a host language (e.g., C, Java, etc.) program is called embedded query language or
        Embedded SQL. Although similar capabilities are supported for a variety of host
        languages, the syntax sometimes varies. Some of the advantages of embedded SQL are:
        ▪ SQL statements can be used wherever a statement in the host language is allowed.
        ▪ It combines the strengths of two programming environments, the procedural features of
          host languages and non-procedural features of SQL.
        ▪ SQL statements can refer to variables (must be prefixed by a colon in SQL statements)
          defined in the host program.
        ▪ Special program variables (called null indicators) are used to assign and retrieve the
          NULL values to and from the database.
        ▪ The facilities available through the interactive query language are also automatically
          available to the host programs.
        ▪ Embedded SQL along with host languages can be used to accomplish very complex and
          complicated data access and manipulation tasks.
        Example: The following Embedded SQL statement in C inserts a row, whose column
        values are based on the values of the host language variables contained in it**.**
        EXEC SQL
        INSERT INTO Sailors VALUES (:c_sname, :c_sid, :c_rating, :c_age);

Q.10   Consider the following relations: **(3.5 x 2=7)**
                S (S#, SNAME, STATUS, CITY)
                SP (S#, P#, QTY)
                P (P#, PNAME, COLOR, WEIGHT, CITY)
        Give an expression in SQL for each of queries below:
        (i)   Get supplier names for supplier who supply at least one red part
        (ii)   Get supplier names for supplier who do not supply part P2.

        Ans:(i)
          SELECT SNAME FROM S
                WHERE S# IN (SELECT S# FROM SP
                        WHERE P# IN (SELECT P# FROM P
                                WHERE COLOR = RED'))
        (ii) SELECT SNAME FROM S
        WHERE S# NOT IN (SELECT S# FROM SP WHERE P# = 'P2')

Q.11   Define a view and a trigger. Construct a view for the above relations which has the
        information about suppliers and the parts they supply. The view contains the S#,
        SNAME, P# , PNAME renamed as SNO, NAME, PNO, PNAME.   **(7)**

Ans:
View – A view is a virtual table which is based on the one or more physical tables and/or views. In other words, a view is a named table that is represented, not by its own physically separate stored data, but by its definition in terms of other named tables (base tables or views).

Trigger – A trigger is a procedure that is automatically invoked by the DBMS in the response to specified changes to the database. Triggers may be used to supplement declarative referential integrity, to enforce complex business rules or to audit changes to data.

Command:

CREATE VIEW SUP_PART (SNO, NAME, PNO, PNAME) AS
           SELECT S.S#, SNAME, P.P#, PNAME
             FROM S, SP, P
             WHERE S.S# = SP.S# AND P.P# = SP.P#

Q.12   Differentiate between the following: **(10)**

(i)   Theta Join. (ii) Equi Join. (iii) Natural Join
(iv)   Outer Join.

Ans:(i) Theta Join – The theta join operation is an extension to the natural-join operation that allows us to combine selection and a Cartesian product into a single operation. Consider relations $r(R)$ and $s(S)$, and let θ be a predicate on attributes in the schema $R \cup S$. The theta join operation $r \bowtie_\theta s$ is defined as follows:
$$r \bowtie_\theta s = \sigma_\theta (r \times s)$$
(ii) Equi Join – It produces all the combinations of tuples from two relations that satisfy a join condition with only equality comparison (=).
(iii)   Natural Join - Same as equi-join except that the join attributes (having same names) are not included in the resulting relation. Only one sets of domain compatible attributes involved in the natural join are present.
(iv)   Outer Join - If there are any values in one table that do not have corresponding value(s) in the other, in an equi-join that will not be selected. Such rows can be forcefully selected by using the outer join. The corresponding columns for that row will have NULLs. There are actually three forms of the outer-join operation: left outer join (⎯X), right outer join (X⎯) and full outer join (⎯X⎯).

Q.13   What are temporary tables? When are they useful? Justify with an example.   **(4)**

Ans:
Temporary tables exists solely for a particular session, or whose data persists for the duration of the transaction. The temporary tables are generally used to support specialized rollups or specific application processing requirements. Unlike a permanent table, a space is not allocated to a temporary table when it is created. Space will be dynamically allocated for the table as rows are inserted. The CREATE GLOBAL TEMPORARY TABLE command is used to create a temporary table in Oracle.
           CREATE GLOBAL TEMPORARY TABLE <table_name> (

                            <columns details>
) ON COMMIT {PRESERVE|DELETE} ROWS;

Q.14   Draw and explain the three level architecture of the database system.   **(7)**

Ans:
A DBMS provides three levels of data is said to follow three-level architecture. The goal of the three-schema architecture is to separate the user applications and the physical database. The view at each of these levels is described by a schema. The processes of transforming requests and results between levels are called *mappings*. In this architecture, schemas can be defined at the following three levels:
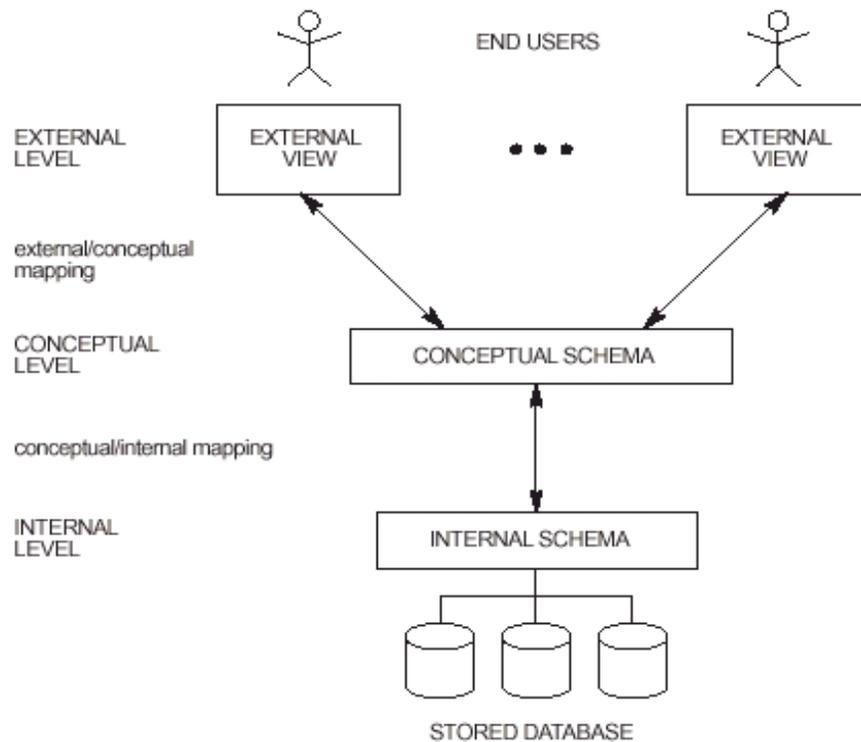
- **External Level or Subschema –** It is the highest level of database abstraction where only those portions of the database of concern to a user or application program are included. Any number of user views (some of which may be identical) may exist for a given global or conceptual view. Each external view is described by means of a schema called an *external schema* or *subschema*.
- **Conceptual Level or Conceptual Schema -** At this level of database abstraction all the database entities and the relationships among them are included. One conceptual view represents the entire database. This conceptual view is defined by the *conceptual schema*. There is only one conceptual schema per database. The description of data at this level is in a format independent of its physical representation. It also includes features that specify the checks to retain data consistency and integrity.
- **Internal Level or Physical Schema –** It is closest to the physical storage method used. It indicates how the data will be stored and describes the data structures and access methods to be used by the database. The internal view is expressed by the *internal schema*.

Q.15 Explain (a) Heap file (b) Sorted file. Also discuss their advantages and disadvantages.

Ans: Heap File is an unordered set of records, stored on a set of pages. This class provides basic support for inserting, selecting, updating, and deleting records. Temporary heap files are used for external sorting and in other relational operators. A sequential scan of a heap file (via the Scan class) is the most basic access method.

Sorted file The *sort* utility shall perform one of the following functions:

1. Sort lines of all the named files together and write the result to the specified output.

2. Merge lines of all the named (presorted) files together and write the result to the specified output.

3. Check that a single input file is correctly presorted.

Comparisons shall be based on one or more sort keys extracted from each line of input (or, if no sort keys are specified, the entire line up to, but not including, the terminating <newline>), and shall be performed using the collating sequence of the current locale**.**

Q.16 Describe a method for direct search? Explain how data is stored in a file so that direct searching can be performed.

Ans: For a file of unordered fixed length records using unspanned blocks and contiguous allocation, it is straight forward to access any record by its position in the file. If the file records are numbered 0,1,2,---,r-1 and the records in each block are numbered 0,1,---bfr-1; where bfr is the blocking factor, then ith record of the file is located in block [(i/bfr)] and is the (I mod bfr)th record in that block. Such a file is often called a relative or direct file because records can easily be accessed directly by their relative positions. Accessing a record based on a search condition; however, it facilitates the construction of access paths on the file, such as the indexes.

Q17 Explain the integrity constraints: Not Null, Unique, Primary Key with an example each. Is the combination 'Not Null, Primary Key' a valid combination. Justify. **(7)**

Ans: Not Null – Should contain valid values and cannot be NULL.
Unique – An attribute or a combination of two or more attributes must have a unique value in each row. The unique key can have NULL values.

Primary Key – It is same as unique key but cannot have NULL values. A table can have at most one primary key in it.

For example:

STUDENT

Roll No	Name	City	*Mobile*
17	Ankit Vats	Delhi	9891663808
16	Vivek Rajput	Meerut	9891468487
6	Vanita	Punjab	NULL
75	Bhavya	Delhi	9810618396

- Roll No is a primary key.

- Name is defined with NOT NULL, means each student must have a name.

- Mobile is unique**.**

'*Not Null, Primary Key*' is a valid combination. Primary key constraint already includes 'Not Null' constraint in it but we can also add 'Not Null' constraint with it. The use of 'Not Null' with 'Primary Key' will not have any effect. It is same as if we are using just 'Primary Key'.

Q.18   Explain the followings :
(i)   Nested Queries.
(ii) Cursors in SQL.
(iii) RDBMS.
(iv) View
(v) Application Programming Interface   **(14)**

Ans: (i) Nested Queries – A SELECT query can have subquery(s) in it. When a SELECT query having another SELECT query in it, is called as nested query. Some operations cannot be performed with single SELECT command or with join operation. There are some operations which can be performed with the help of nested queries (also referred to as subqueries). For example, we want to compute the second highest salary:
SELECT MAX(SAL) FROM EMP WHERE SAL < (SELECT MAX(SAL) FROM EMP)
Some operations can be performed both by Join and subqueries. The Join operation is costlier in terms of time and space. Therefore, the solution based on subqueries is preferred.
(ii)   **Cursors in SQL** – An object used to store the output of a query for row-by-row processing by the application programs. Cursors are constructs that enable the user to name a private memory area to hold a specific statement for access at a later time. Cursors are used to process multi-row result sets one row at a time. Additionally, cursors keep track of which row is currently being accessed, which allows for interactive processing of the active set.
(iii)   **RDBMS –** RDBMS is a database management system (DBMS) that stores data in the form of relations. Relational databases are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in many different ways. An important feature of

relational system is that a single database can be spread across several tables. This differs from flat-file databases, in which each database is self-contained in a single table.

(iv)   View – A view is a relation (virtual rather than base) and can be used in query expressions, that is, queries can be written using the view as a relation. In other words, a view is a named table that is represented, not by its own physically separate stored data, but by its definition in terms of other named tables (base tables or views). The base relations on which a view is based are sometimes called the existing relations. The definition of a view in a create view statement is stored in the system catalog. The syntax to create a view is:CREATE [OR REPLACE] VIEW <view_name> [(<aliases>)] AS <query> WITH   {READ   ONLY|CHECK   OPTION   [CONSTRAINT <constraint_name>]};

(v)   Application Programming Interface – Commercial SQL implementations take one of the two basic techniques for including SQL in a programming language – embedded SQL and application program interface (API). In the application program interface approach, the program communicates with the RDBMS using a set of functions called the Application Program Interface (API). The program passes the SQL statements to the RDBMS using API calls and uses API calls to retrieve the results. In this method, the precompiler is not required.

Q.19   Consider the following relational schema: **(7)**
              PERSON (SS#, NAME, ADDRESS)
              CAR (REGISTRATION_NUMBER, YEAR, MODEL)
              ACCIDENT (DATE, DRIVER, CAR_REG_NO)
              OWNS (SS#, LICENSE)
              Construct the following relational algebra queries:
(i)   Find the names of persons who are involved in an accident.
(ii)   Find the registration number of cars which were not involved in any accident.

          Ans: (i) $\pi_{NAME}(PERSON) \cap \pi_{DRIVER}(ACCIDENT)$
              (ii) $\pi_{REGISTRATION\_NUMBER}(CAR) - \pi_{CAR\_REG\_NO}(ACCIDENT)$

Q.20   What is a key? Explain Candidate Key, Alternate Key and Foreign Key.   **(7)**

          Ans:
          Key – A single attribute or a combination of two or more attributes of an entity set that is used to identify one or more instances (rows) of the set (table) is called as key.
          Candidate Key – A candidate key is a minimal superkey, which can be used to uniquely identify a tuple in the relation.
          Alternate Key – All the *candidate keys except primary key* are called as alternate keys.
          Foreign Key – Let there are two relations (tables) *R* and *S*. Any candidate key of the relation *R* which is referred in the relation *S* is called the *foreign key* in the relation *S* and *referenced key* in the relation *R*. The relation *R* is also called as *parent table* and relation *S* is also called as *child table*.

Q.21   What is data independence? Explain the difference between physical and logical data independence.   **(7)**

          Ans: Data independence is the capacity to change the schema at one level of a database system without having to change the schema at the next level. The three-schema architecture allows the feature of data independence. Data independence occurs

because when the schema is changed at some level, the schema at the next level remains unchanged; only the *mapping* between the two levels is changed. Types of data independence are:

- **Physical Data Independence –** It is capacity to change the internal schema without having to change conceptual schema. Hence, the external schemas need not be changed as well. Changes to the internal schema may be needed because some physical files had to be reorganized to improve the performance of retrieval or update. If the same data as before remains in the database, the conceptual schema needs not be changed.

- **Logical Data Independence -** It is the capacity to change the conceptual schema without having to change external schemas or application programs. The conceptual schema may be changed to expand the database (by adding a record type or data item), to change constraints, or to reduce the database (by removing a record type or data item). Only the view definition and the mappings need be changed in a DBMS that supports logical data independence. Changes to constraints can be applied to the conceptual schema without affecting the external schemas or application programs.

Q.22   Write short notes on:
            (i) Weak and strong entity sets.
            (ii) Types of attributes.
            (iii) Oracle Instance.
            (iv) Mid square method of hashing.   **(4 x 4 = 16)**

Ans: (i)   **Weak and Strong entity sets:** A strong entity set has a primary key. All tuples in the set are distinguishable by that key. A weak entity set has no primary key unless attributes of the strong entity set on which it depends are included. Tuples in a weak entity set are partitioned according to their relationship with tuples in a strong entity set. Tuples within each partition are distinguishable by a discriminator, which is a set of attributes. A strong entity set has a primary key. All tuples in the set are distinguishable by that key. A weak entity set has no primary key unless attributes of the strong entity set on which it depends are included. Tuples in a weak entity set are partitioned according to their relationship with tuples in a strong entity set. Tuples within each partition are distinguishable by a discriminator, which is a set of attributes.

(ii) Types of attributes: An attribute's type determines the kind of values that are allowed in the attribute. For example, the value *version 1* is not valid for an attribute defined as an integer, but the value *1* is valid. Numeric types (such as integer or real) can also be limited to a predefined range by their attribute definition.

Choice : An attribute with a list of predefined values.

ID Reference: An attribute with a value that is a Unique ID value from another element. It is typically used for element-based cross-references.

ID References: An attribute with a value of one or more Unique ID values from another element.

Integer: An attribute with a whole number value (no decimal parts). Examples of valid integers are 22, -22, and +322. An integer can be defined to fall within a range.

Integers: An attribute with a value of one or more integers. Enter each number on a separate line in the Attribute Value text box.

Real   An attribute with a real number value, with or without a decimal part (the value can also be expressed in scientific notation). Examples of valid real numbers are 2, 22.4, -0.22, and $2.3e^{-1}$. A real number can be defined to fall within a range.

Reals: An attribute with a value of one or more real numbers. Enter each number on a separate line in the Attribute Value text box.

String: An attribute with a value of a series of characters (text).

Strings: An attribute with a value of one or more strings. Enter each string on a separate line in the Attribute Value text box.

Unique ID: An attribute with a value of a unique text string. An element can have only one ID attribute (which can be of type Unique ID or Unique IDs). All ID values must be unique in the document or book. An element with a Unique ID attribute can be the source for an element-based cross-reference.

Unique IDs: An attribute with a value of one or more unique text strings. Enter each string on a separate line in the Attribute Value text box.

(iii) Oracle Instances: An instance is the (executed) Oracle software and the memory they use. It is the instance that manipulates the data stored in the database. It can be started independent of any database. It consists of:

1) A shared memory area that provides the communication between various processes.

2) Upto five background processes which handled various tasks.

Whenever an oracle instance starts, the file 'INIT.ORA' is executed.

(iv) Mid square method of hashing: In midsquare hashing, the key is squared and the address selected from the middle of the squared number.

Mid square method

* Square K.

* Strip predetermined digits from front and rear.

* e.g., use thousands and ten thousands places.

Q.23 Consider the following relational schemas:

      EMPLOYEE (EMPLOYEE_NAME, STREET, CITY)
      WORKS (EMPLOYEE_NAME, COMPANYNAME, SALARY)
      COMPANY (COMPANY_NAME, CITY)

Specify the table definitions in SQL. **(5)**

Ans:
```
CREATE TABLE EMPLOYEE
( EMPLOYEE_NAME       VARCHAR2(20) PRIMARY KEY,
  STREET              VARCHAR2(20),
  CITY              VARCHAR2(15));
CREATE TABLE COMPANY
( COMPANY_NAME       VARCHAR2(50) PRIMARY KEY,
  CITY              VARCHAR2(15));
CREATE TABLE WORKS
( EMPLOYEE_NAME       VARCHAR2(20)
      REFERENCES EMPLOYEE(EMPLOYEE_NAME,
  COMPANYNAME         VARCHAR2(50)
      REFERENCES COMPANY(COMPANY_NAME,
  SALARY              NUMBER(6),
 CONSTRAINT WORKS_PK PRIMARY KEY(EMPLOYEE_NAME,
      COMPANY_NAME));
```

Q.24 Give an expression in SQL for each of queries below: **(9)**

(i) Find the names of all employees who work for first Bank Corporation.
(ii) Find the names and company names of all employees sorted in ascending order of company name and descending order of employee names of that company.
(iii) Change the city of First Bank Corporation to 'New Delhi'

Ans:

(i) SELECT EMPLOYEE_NAME
    FROM WORKS
    WHERE COMPANYNAME = 'First Bank Corporation';
(ii) SELECT EMPLOYEE_NAME, COMPANYNAME
    FROM WORKS
    ORDER BY COMPANYNAME, EMPLOYEE_NAME DESC;
(iii) UPDATE COMPANY
    SET CITY = 'New Delhi'
    WHERE COMPANY_NAME = 'First Bank Corporation';

Q.25 Discuss the correspondence between the E-R model construct and the relation model construct. Show how each E-R model construct can be mapped to the relational model using the suitable example?

Ans: **An entity-relationship model (ERM):** An entity-relationship model (ERM) is an abstract conceptual representation of structured data. Entity-relationship modeling is a relational schema database modeling method, used in software engineering to produce a type of conceptual data model (or semantic data model) of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created using this process are called *entity-relationship diagrams*, or *ER diagrams* or *ERDs* for short.
ER-to-Relational Mapping Algorithm:
1) Step 1: Mapping of regular entity types: For each strong entity type E, create a relation T that includes all the simple attributes of a composite attribute.
2) Step2: Mapping of weak entity types: For each weak entity type W with owner entity type E, create relation R and include all simple attributes (or simple components of composite attributes) of W as attributed of R. In addition, include as foreign key attributes of R, the primary key attribute (s) of relation(s) that correspond to the owner(s) and the partial key of the weak entity type W, if any.
3) Mapping of relationship types: form a relation R, for relationship with primary keys of participating relations A and B as foreign keys in R. In addition to this, any attributes of relationship become an attribute of R also.
 4) Mapping of multivalued attributes: For each multilvalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus primary key attribute K-as a foreign key in R-of the relation that represents the entity type or relationship type that has A as an attribute.

Q.26 Explain the concepts of relational data model. Also discuss its advantages and disadvantages. **(7)**

Ans:
Relational Data Model – The relational model was first introduced by Prof. E.F. Codd of the IBM Research in 1970 and attracted immediate attention due to its simplicity and

mathematical foundation. The model uses the concept of a mathematical relation (like a table of values) as its basic building block, and has its theoretical basis in set theory and first-order predicate logic. The relational model represents the database as a
collection of *relations*. The relational model like all other models consists of three basic components:
- a set of domains and a set of relations
- operation on relations
- integrity rules

Advantages
- **Ease of use –** The revision of any information as tables consisting of rows and columns is quite natural and therefore even first time users find it attractive.
- **Flexibility –** Different tables from which information has to be linked and extracted can be easily manipulated by operators such as project and join to give information in the form in which it is desired.
- **Security –** Security control and authorization can also be implemented more easily by moving sensitive attributes in a given table into a separate relation with its own authorization controls. If authorization requirement permits, a particular attribute could be joined back with others to enable full information retrieval.
- **Data Independence –** Data independence is achieved more easily with normalization structure used in a relational database than in the more complicated tree or network structure. It also frees the users from details of storage structure and access methods.
- **Data Manipulation Language –** The possibility of responding to ad-hoc query by means of a language based on relational algebra and relational calculus is easy in the relational database approach. Provides simplicity in the data organization and the availability of reasonably simple to very powerful query languages.

Disadvantages
- **Performance** – If the number of tables between which relationships to be established are large and the tables themselves are voluminous, the performance in responding to queries is definitely degraded.
- **Unsuitable for Hierarchies** – While the relational database approach is a logically attractive, commercially feasible approach, but if the data is for example naturally organized in a hierarchical manner and stored as such, the hierarchical approach may give better results.

Q.27   Consider the following relational schema: **(14)**
                 Doctor(DName,Reg_no)
                 Patient(Pname, Disease)
                 Assigned_To (Pname,Dname)
Give expression in both Tuple calculus and Domain calculus for each of the queries:
(i)   Get the names of patients who are assigned to more than one doctor.
(ii)   Get the names of doctors who are treating patients with 'Polio'.

Ans:
(i)Tuple Calculus:
{p[PName] | p $\in$ PATIENT $\land$ $\exists a_1, a_2$ ($a_1 \in$ ASSIGNED_TO $\land$ $a_2 \in$ ASSIGNED_TO $\land$
p[PName] = a_1[PName] $\land$ a_1[PName] = a_2[PName] $\land$ a_1[DName] $\neq$ a_2[DName])}
Domain Calculus:

{p | ∃p₁, d₁, p₂, d₂ (<p, s> ∈ PATIENT ∧ <p₁, d₁> ∈ ASSIGNED_TO ∧ <p₂, d₂> ∈ ASSIGNED_TO ∧ p₁ = p₂ ∧ d₁ ≠ d₂)}

(ii)Tuple Calculus:

{u[Dname] | u ∈ ASSIGNED_TO ∧ ∃t (t ∈ PATIENT ∧ t[Disease] = 'Polio' ∧ t[PName] = u[PName])}

  Domain Calculus:

  {d | ∃p₁, p₂, s₂ (<p₁, d> ∈ ASSIGNED_TO ∧ <p₂, s₂> ∈ PATIENT
              ∧ p₁ = p₂ ∧ s₂ = 'Polio')}

Q.28   What are the features of embedded SQL? Explain.   **(7)**

Ans: Embedded SQL – SQL can be implemented in two ways. It can be used interactively or embedded in a host language or by using API. The use of SQL commands within a host language (e.g., C, Java, etc.) program is called embedded query language or Embedded SQL. Although similar capabilities are supported for a variety of host languages, the syntax sometimes varies. Some of the features of embedded SQL are:

- SQL statements can be used wherever a statement in the host language is allowed.
- It combines the strengths of two programming environments, the procedural features of host languages and non-procedural features of SQL.
- SQL statements can refer to variables (must be prefixed by a colon in SQL statements) defined in the host program.
- Special program variables (called null indicators) are used to assign and retrieve the NULL values to and from the database.
- The facilities available through the interactive query language are also automatically available to the host programs.
- Embedded SQL along with host languages can be used to accomplish very complex and complicated data access and manipulation tasks.

Q.29   What is the purpose of tables, private synonyms and public synonyms? If there are multiple objects of same name on an Oracle database, which order are they accessed in?

Ans: The purpose of table is to store data. If we use the PUBLIC keyword (or no keyword at all), anyone who has access to the database can use our synonym. If the database is not ANSI-compliant, a user does not need to know the name of the owner of a public synonym. Any synonym in a database that is not ANSI-compliant **and** was created in an Informix database server is a public synonym. In an ANSI-compliant database, all synonyms are private. If you use the PUBLIC or PRIVATE keywords, the database server issues a syntax error. If you use the PRIVATE keyword to declare a synonym in a database that is not ANSI-compliant, the unqualified synonym can be used by its owner. Other users must qualify the synonym with the name of the owner.

Q.30   Explain the followings: **(14)**
             (i)   Temporary Tables
             (ii) Integrity Constraints.

             Ans:

(i)   **Temporary Tables –** Temporary tables exists solely for a particular session, or whose data persists for the duration of the transaction. The temporary tables are generally used to support specialized rollups or specific application processing requirements. Unlike a permanent table, a temporary table does not automatically allocate space when it is created. Space will be dynamically allocated for the table as rows are inserted.

The CREATE GLOBAL TEMPORARY TABLE command is used to create a temporary table in Oracle.

CREATE GLOBAL TEMPORARY TABLE <table_name>
  (
        <columns details>
)
  ON COMMIT {PRESERVE|DELETE} ROWS;

(ii)Integrity Constraints – A database is only as good as the information stored in it, and a DBMS must therefore help prevent the entry of incorrect information. An *integrity constraint* is a condition specified on a database schema and restricts the data that can be stored in an instance of the database. If a database instance satisfies all the integrity constraints specified on the database schema, it is a legal instance. A DBMS enforces integrity constraints, in that it permits only legal instances to be stored in the database. Integrity constraints are specified and enforced at different times:

- When the DBA or end user defines a database schema, he or she specifies the integrity constraints that must hold on any instance of this database.
- When a database application is run, the DBMS checks for violations and disallows changes to the data that violate the specified integrity constraints.

Many kinds of integrity constraints can be specified in the relational model, such as, Not Null, Check, Unique, Primary Key, etc.

Q.31   Explain different types of failures that occur in Oracle database.   **(7)**

Ans: **Types of Failures –** In Oracle database following types of failures can occurred:
- Statement Failure
  - Bad data type
  - Insufficient space
- Insufficient Privileges (e.g., object privileges to a role)
- User Process Failure
  - User performed an abnormal disconnect
  - User's session was abnormally terminated
  - User's program raised an address exception
- User Error
  - User drops a table
  - User damages data by modification
- Instance Failure
- Media Failure
  - User drops a table
  - User damages data by modification
- Alert Logs
  - Records informational and error messages
  - All Instance startups and shutdowns are recorded in the log

- Every Create, Alter, or Drop operation on a rollback segment, tablespace, or database is record in the log
  - Recovery Views
  - DB Verify
    - Used to insure that a datafile is valid before a restore

Q.32   What is ODBC? What are the uses of ODBC? Under what circumstances we use this technology?   **(7)**

Ans:
ODBC – Open DataBase Connectivity (ODBC) enable the integration of SQL with a general-purpose programming language. ODBC expose database capabilities in a standardized way to the application programmer through an application programming interface (API). In contrast to Embedded SQL, ODBC allows a single executable to access different DBMSs without recompilation. Thus, while Embedded SQL is DBMS-independent only at the source code level, applications using ODBC are DBMS-independent at the source code level and at the level of the executable.
All direct interaction with a specific DBMS happens through a DBMS-specific driver. A driver is a software program that translates the ODBC calls into DBMS-specific calls. Drivers are loaded dynamically on demand since the DBMSs the application is going to access are known only at run-time. Available drivers are registered with a driver manager. The driver translates the SQL commands from the application into equivalent commands that the DBMS understands. An application that interacts with a data source through ODBC selects a data source, dynamically loads the corresponding driver, and establishes a connection with the data source. ODBC achieves portability at the level of the executable by introducing an extra level of indirection. In addition, using ODBC, an application can access not just one DBMS but several different ones simultaneously.

Q.33   List any two significant differences between a file processing system and a DBMS.   **(4)**

Ans:
File Processing System vs. DBMS
Data Independence - Data independence is the capacity to change the schema at one level of a database system without having to change the schema at the next level. In file processing systems the data and applications are generally interdependent, but DBMS provides the feature of data independence.
Data Redundancy – Data redundancy means unnecessary duplication of data. In file processing systems there is redundancy of data, but in DBMS we can reduce data redundancy by means of normalization process without affecting the original data. If we do so in file processing system, it becomes too complex.
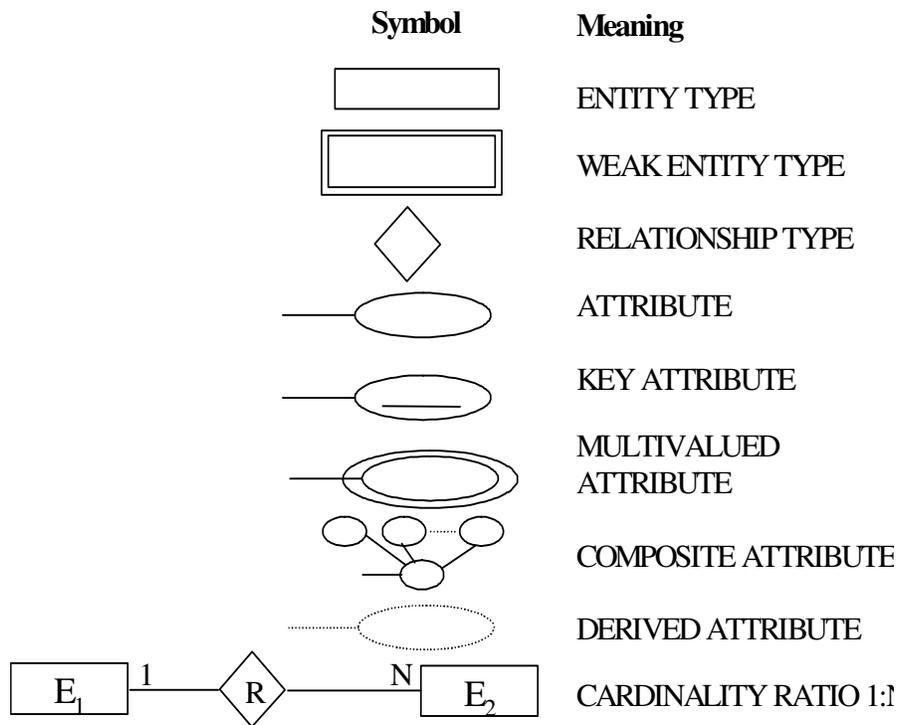
Q.34   Differentiate between various levels of data abstraction.   **(5)**

Ans: Data Abstraction – Abstraction is the process to hide the irrelevant things from the users and represent the relevant things to the user. Database systems are often used by non-computer professionals so that the complexity must be hidden from database system users. This is done by defining levels of abstract as which the database may be viewed, there are logical view or external view, conceptual view and internal view or physical view.

- **External View** – This is the highest level of abstraction as seen by a user. It describes only the part of entire database, which is relevant to a particular user.
- **Conceptual View –** This is the next higher level of abstraction which is the sum total of Database Management System user's views. It describes what data are actually stored in the database. It contains information about entire database in terms of a small number of relatively simple structure.
- **Internal View –** This is the lowest level of abstraction. It describes how the data are physically stored

Q.35   What are the various symbols used to draw an E-R diagram? Explain with the help of an example how weak entity sets are represented in an E-R diagram.   **(6)**

Ans: Various symbols used to draw an E-R diagram

Symbol	**Meaning**
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED ATTRIBUTE
	COMPOSITE ATTRIBUTE
	DERIVED ATTRIBUTE
E_1 —1— R —N— E_2	CARDINALITY RATIO 1:N

Weak Entity Sets - An entity set that does not have a key attribute is called weak entity set. A weak entity must participate in an identifying relationship type with an owner or identifying entity type Entities are identified by the combination of:
- A partial key of the weak entity type
- The primary key of the identifying entity type

Example:
Suppose that a DEPENDENT entity is identified by the dependent's first name and birhtdate, *and* the specific EMPLOYEE that the dependent is related to. DEPENDENT is a weak entity type with EMPLOYEE as its identifying entity type via the identifying relationship type DEPENDENT_OF
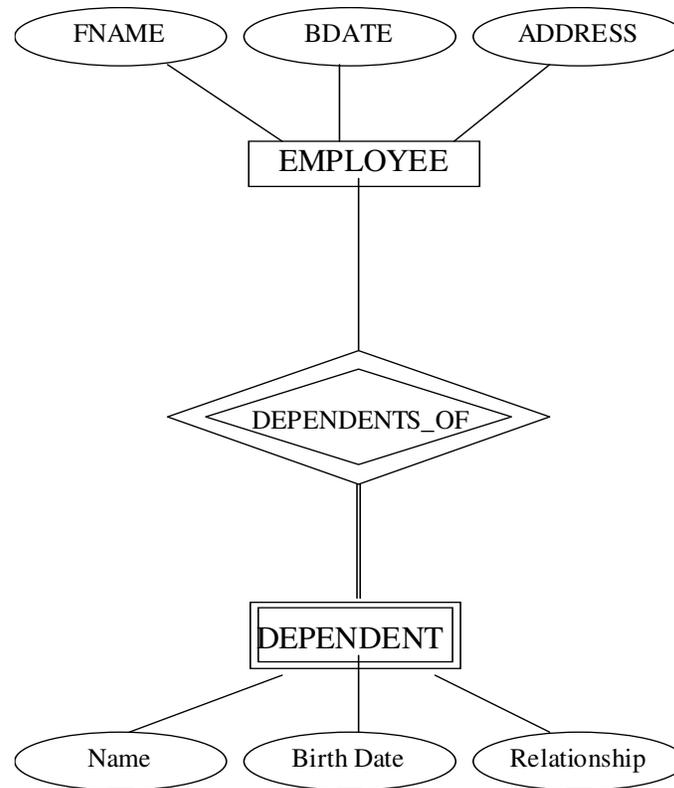
Q.36 Define the following terms: **(8)**
    a) Primary key. **b)** DML **c)** Multivalued attribute **d)** Relationship instance

    Ans: Primary Key – Primary key is one of the candidate keys. It should be chosen such that its attribute values are never, or very rarely, changed.
    b) **Data Manipulation Language (DML)** – A data manipulation language is a language that enables users to access or manipulate data as organized by the appropriate data model.
    c) **Multivalued Attribute** – Multivalued attribute may have more than one value for an entity. For example, PreviousDegrees of a STUDENT.
    d) **Relationship Instance –** A relationship is an association among two or more entities. An instance of relationship set is a set of relationships.

Q.37 Define a table in SQL called Client, which is used to store information about the clients. Define CLIENT_NO as the primary key whose first letter must start with 'C'. Also ensure that the column 'NAME' should not allow NULL values.

Column name	Data type	Size
CLIENT_NO	Varchar2	6
NAME	Varchar2	20
ADDRESS1	Varchar2	30
ADDRESS2	Varchar2	30
CITY	Varchar2	15
STATE	Varchar2	15
PINCODE	Number	6
BAL_DUE	Number	10, 2

(7)

Ans:
  CREATE TABLE CLIENT
  (CLIENT_NO   VARCHAR2(6) PRIMARY KEY CHECK (CLIENT_NO LIKE
  'C%'),
              NAME   VARCHAR2(20) NOT NULL,
              ADDRESS1   VARCHAR2(30),
              ADDRESS2   VARCHAR2(30),
              CITY   VARCHAR2(15),
              STATE   VARCHAR2(15),
              PINCODE NUMBER(6),
              BAL_DUE NUMBER(10,2))

Q.38   An orchestra database consists of the following relations: **(3.5 x 2=7)**
              CONDUCTS (conductor, composition)
              REQUIRES (composition, Instrument)
              PLAYS (Player, Instrument)
              Give the relational calculus queries for the following:
              (i)   List the compositions and the players.
              (ii)   List the compositions which require the 'violin' and the 'congo'

Ans: (i) Tuple Calculus:
{r[Composition] || p[Player] | r ∈ REQUIRES ∧ p ∈ PLAYS
∧ r[Instrument] = p[Instrument]}
Domain Calculus:
{c || p | ∃i_1, i_2 (<c, i_1> ∈ REQUIRES ∧ <p, i_2> ∈ PLAYS ∧ i_1 = i_2)}
(ii)   Tuple Calculus:
{r[Composition] | r ∈ REQUIRES ∧ ∃u (u ∈ REQUIRES
 ∧ r[Composition] = u[Composition] ∧ r[Instrument] = 'violin'
 ∧ u[Instrument] = 'congo')}
Domain Calculus:
{c | ∃i_1, c_2, i_2 (<c, i_1> ∈ REQUIRES ∧ <c_2, i_2> ∈ REQUIRES
∧ c_1 = c ∧ i_1 = 'violin' ∧ i_2 = 'congo')}

Q.39   Perform the following with syntax and a suitable example
              (i) Create a table from existing table.
              (ii) Insert data in your table from another table.

              Ans: (i) Create table <table-name> as <select-statement>
              e.g, To create a new table 'N_emp' with employee names and their identification
                numbers only from employee table, statement is   to create table N_emp as select
              empname, empid from employee
              (ii) insert into <table-name > <select-statement>
              e.g, To insert tuples from employee into N_emp created above, use following
              statement
                      Insert into N_emp select empname, empid from employee

Q.40   What is an INDEX as defined in ORACLE? Write the syntax of creating an INDEX.
              Create an index for the table Client, field CLIENT_NO of Q.   **(2+2+3)**

Ans: Indexes in Oracle – Index is typically a listing of keywords accompanied by the location of information on a subject. In other words, An index can be viewed as an auxiliary table which contains two fields: the key and the location of the record of that key. Indexes are used to improve the performance of the search operation. Indexes are not strictly necessary to running Oracle, they do speed the process.

Syntax of Creating an Index:
CREATE [BITMAP] [UNIQUE] INDEX <index_name> ON
<table_name>(<column_name1> [, <column_name2>] . . .);

Command:
CREATE INDEX client_client_no ON client(client_no);

Q.41   Consider the following relational database:
STUDENT (name, student#, class, major)
COURSE (course name, course#, credit hours, department)
SECTION (section identifier, course#, semester, year, instructor)
GRADE_REPORT (student#, section identifier, grade)
PREREQUISITE (course#, presequisite#)
Specify the following queries in SQL on the above database schema.   **(3.5 x 4=14)**
(i)   Retrieve the names of all students majoring in 'CS' (Computer Science).
(ii)   Retrieve the names of all courses taught by Professor King in 1998
(iii)   Delete the record for the student whose name is 'Smith' and whose student number is 17.
(iv)   Insert a new course <'Knowledge Engineering', 'CS4390', 3, 'CS'>

Ans: (i) SELECT NAME FROM STUDENT WHERE MAJOR = 'CS'
(ii)   SELECT COURSE_NAME FROM COURSE C, SECTION S
               WHERE C.COURSE# = S.COURSE#
                  AND INSTRUCTOR = 'KING' AND YEAR = 1998
                           OR
             SELECT COURSE_NAME FROM COURSE
               WHERE COURSE# IN (SELECT COURSE# FROM SECTION
               WHERE INSTRUCTOR = 'KING' AND YEAR = 1998)
(iii) DELETE FROM STUDENT WHERE NAME = 'Smith' AND STUDENT# = 17
(iv)   INSERT INTO COURSE
              VALUES('Knowledge Engineering', 'CS4390', 3, 'CS')

Q.42   Explain the concept of a data model. What data models are used in
            database management systems?   **(7)**
Ans:
Data Model – Model is an abstraction process that hides irrelevant details while highlighting details relevant to the applications at hand. Similarly, a data model is a collection of concepts that can be used to describe structure of a database and provides the necessary means to achieve this abstraction. Structure of database means the data types, relationships, and constraints that should hold for the data. In general a data model consists of two elements:
- A mathematical notation for expressing data and relationships.
- Operations on the data that serve to express queries and other manipulations of the data.

Data Models used in DBMSs:

- **Hierarchical Model -** It was developed to model many types of hierarchical organizations that exist in the real world. It uses tree structures to represent relationship among records. In hierarchical model, no dependent record can occur without its parent record occurrence and no dependent record occurrence may be connected to more than one parent record occurrence.

- **Network Model -** It was formalised in the late 1960s by the Database Task Group of the Conference on Data System Language (DBTG/CODASYL). It uses two different data structures to represent the database entities and relationships between the entities, namely *record type* and *set type*. In the network model, the relationships as well as the navigation through the database are predefined at database creation time.

- **Relational Model -** The relational model was first introduced by E.F. Codd of the IBM Research in 1970. The model uses the concept of a mathematical relation (like a table of values) as its basic building block, and has its theoretical basis in set theory and first-order predicate logic. The relational model represents the database as a collection of *relations*.

- **Object Oriented Model –** This model is based on the object-oriented programming language paradigm. It includes the features of OOP like inheritance, object-identity, encapsulation, etc. It also supports a rich type system, including structured and collection types.

- **Object Relational Model –** This model combines the features of both relational model and object oriented model. It extends the traditional relational model with a variety of features such as structured and collection types.

Q.43   Briefly explain the differences between a stand alone query language, embedded query language and a data manipulation language.   **(7)**

          Ans: Stand alone Query Language – The query language which can be used interactively is called stand alone query language. It does not need the support of a host language.

          Embedded Query Language – A query language (e.g., SQL) can be implemented in two ways. It can be used interactively or embedded in a host language. The use of query language commands within a host language (e.g., C, Java, etc.) program is called embedded query language. Although similar capabilities are supported for a variety of host languages, the syntax sometimes varies.

          Data Manipulation Language (DML) – A data manipulation language is a language that enables users to access or manipulate data as organized by the appropriate data model.

Q.44   Consider the following relations for a database that keeps track of business trips of salespersons in a sales office:

          SALESPERSON (SSN, Name, start_year, Dept_no)
          TRIP (SSN, From_city, To_city, Departure_Date, Return_Date, Trip_ID)
          EXPENSE(TripID, Account#, Amount)

      Specify the following queries in relational algebra: **(4x3 =12)**
      (i)   Give the details (all attributes of TRIP) for trips that exceeded $2000 in expenses.
      (ii)   Print the SSN of salesman who took trips to 'Honolulu'

(iii) Print the trip expenses incurred by the salesman with SSN= '234-56-7890'.Note that the salesman may have gone on more than one trip. List them individually

Ans: (i) $\pi_{TRIP.*}$ ($\sigma_{amount > 2000}$ (TRIP $\bowtie$ EXPENSE))

(ii) π_{SSN} ($\sigma_{to\_city = \text{'Honolulu'}}$ (TRIP))

(iii) $\pi_{EXPENSE.tripid, amount}$ ($\sigma_{SSN = \text{'234-56-7890'}}$ (TRIP $\bowtie$ EXPENSE))

Q.45   What is the difference between a key and a superkey?   **(2)**

Ans: Key – A key a single attribute or a combination of two or more attributes of an entity set that is used to identify one or more instances (rows) of the set (table). It is a minimal combination of attributes.
Super Key – A super key is a set of one or more attributes that, taken collectively, allows us to identify uniquely a tuple in the relation.

Q.46   Why are cursors necessary in embedded SQL?   **(2)**

Ans: A cursor is an object used to store the output of a query for row-by-row processing by the application programs. SQL statements operate on a set of data and return a set of data. On other hand, host language programs operate on a row at a time. The cursors are used to navigate through a set of rows returned by an embedded SQL SELECT statement. A cursor can be compared to a pointer.

Q.47   Write a program in embedded SQL to retrieve the total trip expenses of the salesman named 'John' for the relations of Q. 44   **(6)**

Ans:
EXEC SQL BEGIN DECLARE SECTION;
long total_expenses;
EXEC SQL END DECLARE SECTION;
EXEC SQL
SELECT SUM(AMOUNT) INTO :total_expenses FROM EXPENSE WHERE TRIPID IN (SELECT TRIP_ID FROM TRIP
WHERE SSN = (SELECT SSN FROM SALEPERSON WHERE NAME = 'John'));
printf("\nThe total trip expenses of the salesman John is: %ld", total_expenses);

Q.48   What are views? Explain how views are different from tables.   **(6)**

Ans:
A view in SQL terminology is a single table that is derived from other tables. These other tables could be base tables or previously defined views. A view does not necessarily exist in physical form; it is considered a virtual table, in contrast to base tables, whose tuples are actually stored in the database. This limits the possible update operations that can be applied to views, but it does not provide any limitations on querying a view. A view represents a different perspective of a base relation(s). The definition of a view in a create view statement is stored in the system catalog. Any attribute in the view can be updated as long as the attribute is simple and not derived from a computation involving two or more

base relation attribute. View that involve a join may or may not be updatable. Such views are not updatable if they do not include the primary keys of the base relations.

Q.49   What do you mean by integrity constraints? Explain the two constraints, check and foreign key in SQL with an example for each. Give the syntax.   **(8)**

       Ans:Integrity Constraints –An *integrity constraint* is a condition specified on a database schema and restricts the data that can be stored in an instance of the database. If a database instance satisfies all the integrity constraints specified on the database schema, it is a legal instance. A DBMS enforces integrity constraints, in that it permits only legal instances to be stored in the database.
       CHECK constraint – CHECK constraint specifies an expression that must always be true for every row in the table. It can't refer to values in other rows.
       Syntax:
       ALTER TABLE <table_name>
       ADD CONSTRAINT <constraint_name> CHECK(<expression>);
       FOREIGN KEY constraint – A foreign key is a combination of columns with values based on the primary key values from another table. A foreign key constraint, also known as referential integrity constraint, specifies that the values of the foreign key correspond to actual values of the primary or unique key in other table. One can refer to a primary or unique key in the same table also.
       Syntax:
         ALTER TABLE <table_name>
           ADD CONSTRAINT <constraint_name> FOREIGN KEY(<column_name(s)>)
           REFERENCES <base_table>(<column_name>) ON {DELETE | UPDATE}
           CASCADE;

Q.50   Define the following constraints for the table client of Q.37   **(6)**
       (i)   BAL_DUE must be at least 1000.
       (ii)   NAME is a unique key.

       Ans: (i)ALTER TABLE CLIENT
           ADD CONSTRAINT CLIENT_BAL_DUE_C1 CHECK(BAL_DUE < 1000);
       (ii)   ALTER TABLE CLIENT
           ADD CONSTRAINT CLIENT_NAME_U UNIQUE(NAME);

Q.51   What are the different types of database end users? Discuss the main activities of each.
                                                                     (7)

       Ans:
       End-Users – End-users are the people whose jobs require access to the database for querying, updating, and generating reports; the database primarily exists for their use. The different types of end-users are:
- **Casual end-users –** occasionally access the database, need different information each time
- **Naive or Parametric end-users –** includes tellers, clerks, etc., make up a sizable portion of database end-users, main job function revolves around constantly querying and updating the database

- **Sophisticated end-users –** includes engineers, scientists, business analyst, etc., use for their complex requirements
- **Stand-alone users –** maintain personal databases by using ready-made program packages, provide easy-to-use menu-based or graphics-based interfaces

Q.52   Discuss the typical user friendly interfaces and the types of users who use each.   **(7)**

Ans:
User-friendly interfaces provided by a DBMS may include the following:

- **Menu-Based Interfaces for Web Clients or Browsing –** These interfaces present the user with lists of options, called menus, that lead the user through the formulation of a request. Pull-down menus are a very popular techniques in Web-based user interfaces. They are also used in browsing interfaces, which allow a user to look through the contents of a database in an exploratory and unstructured manner.
- **Forms-Based Interfaces –** A forms-based interface displays a form to each user. Forms are usually designed and programmed for naive users and interfaces to canned transactions. Many DBMSs have forms specification languages.
- **Graphical User Interfaces (GUIs) –** A GUI typically displays a schema to the user in diagrammatic form. The user can then specify a query by manipulating the diagram. In many cases, GUIs utilizes both menus and forms. Most GUIs use a pointing device to pick certain parts of the displayed schema diagram.
- **Natural Language Interfaces –** These interfaces accept requests written in English or some other language and attempt to "understand" them. A natural language interface usually has its own "schema," which is similar to the database conceptual schema, as well as a dictionary of important words.
- **Interfaces for Parametric Users –** Parametric users, such as bank tellers, often have a small set of operations that they must perform repeatedly. The interfaces for these users usually have a small set of abbreviated commands with the goal of minimizing the number of keystrokes required for each request.
- **Interfaces for the DBA –** Most database systems contain privileged commands that can be used only by the DBA's staff. These include commands for creating accounts, setting system parameters, granting account authorization, changing a schema, and a reorganizing the storage structures of a database.

Q.53   With the help of an example show how records can be deleted and updated in QBE. **(5)**
        (i)   Increase Pay_Rate of employees with the skill of 'cook' by 10%.
        (ii)   Delete employee record for EMP# 123459

Ans: (i)

EMPLOYEE	Emp#	Name	Skill	Pay_Rate
	EX		Cook	PX
U.	EX			PX * 1.1

(ii)

EMPLOYEE	Emp#	Name	Skill	Pay_Rate
D.	123459			

Q.54   Describe cardinality ratios and participation constraints for relationship types.   **(4)**

Ans:
Cardinality Ratios – The cardinality ratios for a relationship type specifies the *maximum* number of relationship instances that an entity can participate in. The possible cardinality ratios for relationship types are one-to-one (1:1), one-to-many or many-to-one (1:M or M:1), and many-to-many (M:N).
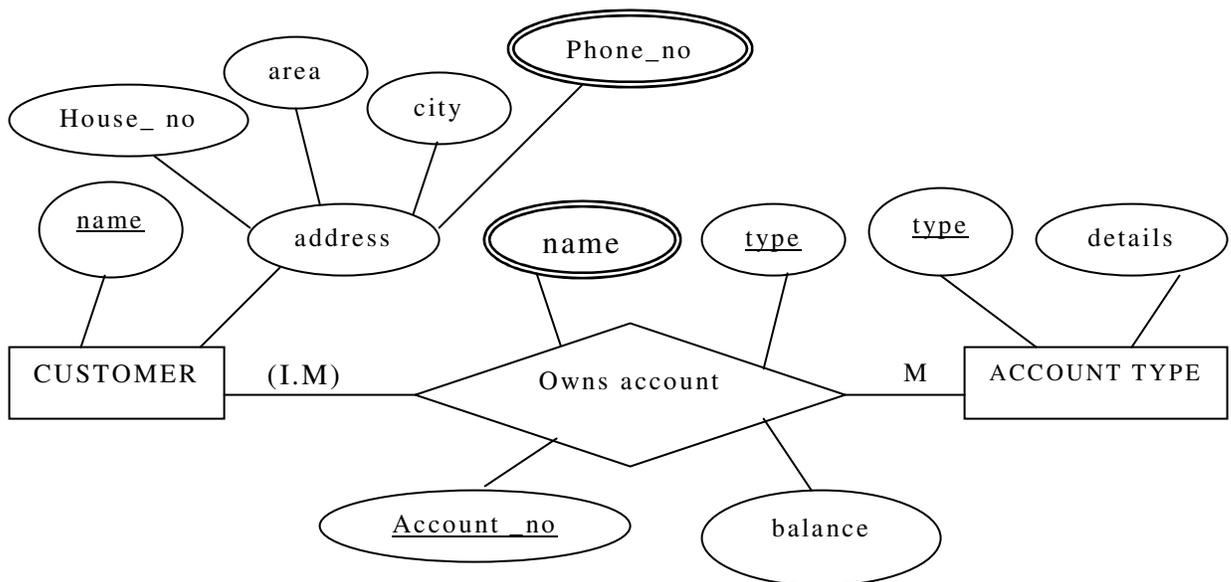**Participation Constraints –** The participation constraint specifies whether the existence of an entity depends on its being related to another entity via the relationship type. This constraint specifies the *minimum* number of relationship instances that each entity can participate in. It is sometimes called the minimum cardinality constraint. There are two types of participation constraints – total and partial.

Q.55   Information about a bank is about customers and their account. Customer has a name, address which consists of house number, area and city, and one or more phone numbers. Account has number, type and balance. We need to record customers who own an account. Account can be held individually or jointly. An account cannot exist without a customer.
Arrive at an E-R diagram. Clearly indicate attributes, keys, the cardinality ratios and participation constraints.   **(10)**
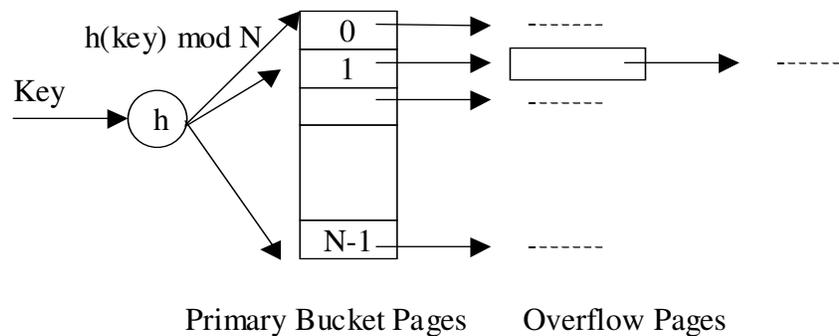
Ans:

Q.56   Describe the static hash file with buckets and chaining and show how insertion, deletion and modification of a record can be performed.   **(9)**
Ans:
In static hash file organization, the term bucket is used to denote a unit storage that can store one or more records.A file consists of buckets 0 through N-1, with one primary page per bucket initially and additional overflow pages chained with bucket, if required later. Buckets contain data entries (or data records). In hashing scheme, a hash function, h, is performed on the key of the record to identify the bucket to which data record belongs to. The hash function is an important component of the hashing approach. The main problem with static hash file is that the number of buckets is fixed.

Insertion of a record – To insert a data entry, the hash function is used to identify the

Primary Bucket Pages   Overflow Pages

Static Hash File

correct bucket and then put the data entry there. If there is no space for this data entry, a

new overflow page will be allocated, put the data entry on this page, and the page to the overflow chain of the bucket.

Deletion of a record – To delete a data entry, the hash function is used to identify the correct bucket, locate the data entry by searching the bucket, and then remove it. If the data entry is the last in an overflow page, the overflow page is removed from the overflow chain of the bucket and added to a list of free pages.

Modification of a record – To modify a data entry, the hash function is used to identify the correct bucket, locate the data entry by searching the bucket and get it, modify the data entry, and then rewrite the modified data entry on it.

Q.57   What are the reasons for having variable length records? What types of separator characters are needed for each?   **(5)**

Ans:
Variable-Length Records – Variable-length records are those records, which are of different sizes. A file may contain variable-length records in any of the following situations:
• **Records having variable length fields –** In this case, the end-of-field symbol along with end-of-record symbol can be used as the separator characters.
• **Records having repeating fields** – In this case, the number of repetitions of repeating fields can be used with end-of-record symbol as the separator characters.
• **Records having optional fields –** In this case, the end-of-field symbol along with end-of-record symbol can be used as the separator characters.
• **File containing records of different record types -** In this case, a special field (or control field) can be prefixed with each record and end-of-record symbol can be used as the separator characters.

In each of the above situations either we can use end-of-record symbol or the length of each record can be prefixed at the beginning of the record for the separation of the variable-length records.

Q.58   Define the following terms   **(7 x 2 = 14)**

(i)   Derived and stored attribute.

         (ii)   Distributed system.
         (iii)   Interblock gap
         (iv)   Degree of a relation.
         (v)   Catalog
         (vi)   Conceptual schema
         (vii)   DDL and SDL.

Ans: (i) Derived and Stored Attribute - In some cases, two or more attribute values are related, for example, Age and BirthDate attributes of a person. For particular person entity, the value of Age can be determined from the current date and the value of that person's BirthDate. Hence, the attribute Age is called as derived attribute and the attribute BirthDate is called as stored attribute.

(ii) Distributed System – A distributed system consists of a number of processing elements that are interconnected by a computer network and that cooperate in performing certain assigned tasks.

(iii) Interblock Gap – A track of a disk is divided into equal-sized disk blocks. Blocks are separated by fixed-size gaps, called as interblock gaps, which include specially coded control information written during disk initialization.

(iv) Degree of a Relation – The degree or arity of a relation is the number of attributes *n* of its relation schema.

(v) Catalog – A relational DBMS maintains information about every table and index that it contains. A catalog is a collection of special tables, which stores the descriptive information of every table and index.

(vi) Conceptual Schema – Conceptual schema describes the structure of the whole database for a community of users. It hides the details of physical storage structures and concentrates on describing entities, data types, relationships, and constraints.

(vii) DDL and SDL – The data definition language (DDL) is used by DBA and database designers to define conceptual schema, internal schema, and mappings between these two. In some DBMSs, a clear separation is maintained between conceptual schema and internal schema. In that case, DDL is used to specify the conceptual schema only. Another language, storage definition language (SDL) is used to specify the internal schema. The mappings between the two schemas may be specified in either one of these languages.

Q.59   Define a relation.   **(2)**

Ans: Relation – A relations is a named two-dimensional table of data. Mathematically, a relation can be defined as a subset of the cartesian product of a list of domains. Each relation consists of a set of named columns and an arbitrary number of rows. The columns correspond to the fields describing each tuple in the table or relation. The rows correspond to each instance of the entity described by the table or relation.

Q.60   Describe entity integrity and referential integrity. Give an example of each.   **(6)**

Ans:
Entity Integrity Rule – If the attribute A of relation R is a prime attribute of R then A cannot accept null values.

Referential Integrity Rule – In referential integrity, it is ensured that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.

For example:

STUDENT

Enrl No	Roll No	Name	City	*Mobile*
11	17	Ankit Vats	Delhi	9891663808
15	16	Vivek Rajput	Meerut	9891468487
6	6	Vanita	Punjab	
33	75	Bhavya	Delhi	9810618396

GRADE

Roll No	Course	*Grade*
6	C	A
17	VB	C
75	VB	A
6	DBMS	B
16	C	B

- **Roll No** is the primary key in the relation STUDENT and Roll No + Course is the primary key of the relation GRADE. (Entity Integrity)
- **Roll No** in the relation GRADE (child table) is a foreign key, which is referenced from the relation STUDENT (parent table). (Referential Integrity).

Q.61   Consider the two relations given below

R

A	B	C
A1	b1	c1
Null	b2	null
a1	b1	c1

S

D	A	F
d1	a1	f1
d1	a2	null

Given that A is the primary key of R, D is the primary key of S and there is a referential integrity between S.A and R.A, discuss all integrity constraints that are violated. **(6)**

Ans: (i) Primary key of R contains the 'null' value and the value 'a1' is duplicated, hence it violates the entity integrity constraint in the relation R.

(ii) In primary key of S, the value 'd1' is duplicated, hence it violates the entity integrity constraint in the relation S.

(iii) The foreign key S.A contains the value 'a2', which is not available in the parent key R.A, hence it violates the referential integrity constraint in the relation S.

Q.62   Given the following relations
  TRAIN (NAME, START, DEST)
  TICKET (PNRNO., START, DEST, FARE)
  PASSENGER (NAME, ADDRESS, PNRNO.)
Write SQL expressions for the following queries: **(3.5 x 4 = 14)**
Note: Assume NAME of Train is a column of Ticket.

(i) List the names of passengers who are travelling from the start to the destination station of the train.

(ii) List the names of passengers who have a return journey ticket.

(iii) Insert a new Shatabti train from Delhi to Bangalore.

(iv) Cancel the ticket of Tintin.

Ans

(i) SELECT P.NAME FROM TRAIN T, TICKET I, PASSENGER P
WHERE P.PNRNO = I.PNRNO AND T.NAME = I.NAME
AND T.START = I.START AND T.DEST = I.DEST

(ii) SELECT NAME FROM PASSENGER
WHERE PNRNO IN (SELECT DISTINCT A.PNRNO
FROM TICKET A, TICKET B WHERE A.PNRNO = B.PNRNO
AND A.START = B.DEST AND A.DEST = B.START)

(iii) INSERT INTO TRAIN
VALUES('Shatabdi', 'Delhi', 'Banglore'

(iv) DELETE FROM TICKET
WHERE PNRNO = (SELECT PNRNO FROM PASSENGER
WHERE NAME = 'Tintin')

Q.63 Define outer union operation of the relational algebra. Compute the outer union for the relations R and S given below. **(2 x 3)**

R

A	B	C
a1	b1	c1
a3	b2	c2

S

D	A	F
d1	a1	f1
d1	a2	null

Ans:
Outer Join - If there are any values in one table that do not have corresponding value(s) in the other, in an equi-join that will not be selected. Such rows can be forcefully selected by using the *outer join. The corresponding columns for that row will have NULLs.* There are actually three forms of the outer-join operation: left outer join ($\overline{}$X), right outer join (X$\overline{}$) and full outer join ($\overline{}$X$\overline{}$).

R.A	B	C	D	S.A	F
a1	b1	c1	d1	a1	f1
a3	b2	c2	Null	Null	Null
Null	Null	Null	d1	a2	Null

Q.64 Given the following relations **(3 x 3)**
         Vehicle (Reg_no, make, colour)
         Person(eno, name, address)
         Owner(eno, reg_no)
Write expressions in the relational algebra to answer the following queries:-
(i) List the reg_no of vehicles owned by John.

(ii) List the names of persons who own maruti cars.

(iii) List all the red coloured vehicle.

Ans: (i) π_{reg_no} ($\sigma_{name='John'}$ (PERSON $\bowtie$ OWNER))

(ii) π_{name} ($\sigma_{make='maruti'}$ (PERSON $\bowtie$ OWNER $\bowtie$ VEHICAL))

(iii) $\sigma_{colour='red'}$ (VEHICAL)

Q.65   Describe DROP TABLE command of SQL with both the options – CASCADE and RESTRICT.   **(5)**

Ans:
DROP TABLE command – This command drops the specified table. Dropping a table also drops indexes and grants associated with it. Objects built on dropped tables are marked invalid and cease to work.
CASCADE and RESTRICT options – The DROP TABLE with RESTRICT option destroy the table unless some view or integrity constraint refer to the given table; if so, the command fails. But with the CASCADE option, any referencing views or integrity constraints are (recursively) dropped as well.

Q.66   With respect to Oracle describe the following: **(3 x 3)**

(i) Data Block.

(ii) Data dictionary.

(iii) Segments.

Ans:Data Block – In oracle the data blocks are referred to as tablespaces. A tablespace is an area of disk consisting of one or more disk files. A tablespace can contain many tables, indexes, or clusters.
(ii) **Data Dictionary** – Data dictionaries are the system tables that contain descriptions of the database objects and how they are structured.
(iii) **Segments –** Each table has single area of disk space, called segment, set aside for it in the tablespace. Segments consists of contiguous sections called extents.

Q.67   Describe the responsibilities of the DBA and the database designer.   **(7)**

Ans: The responsibilities of DBA and database designer are:
1. Planning for the database's future storage requirements
2. Defining database availability and fault management architecture
3. Defining and creating environments for development and new release installation
4. Creating physical database storage structures after developers have designed an application
5. Constructing the database
6. Determining and setting the size and physical locations of data files
7. Evaluating new hardware and software purchase
8. Researching, testing, and recommending tools for Oracle development, modeling, database administration, and backup and recovery implementation, as well as planning for the future
9. Providing database design and implementation

10. Understanding and employing the optimal flexible architecture
    to ease administration, allow flexibility in managing I/O, and to increase the
    capability to scale the system

Q.68   What are the four main characteristics of the database approach?
                                                             (7)

     Ans: The four main characteristics of the database approach are:
         1. Self-describing nature of a database system.
         2. Insulation between programs and data, and data abstraction.
         3. Support of multiple views of the data.
         4. Sharing of data and multi-user transaction processing.

Q.69   Differentiate between DDL and DML.   **(4)**

     Ans: **DDL - Data Definition Language:** statements used to define the database
structure or schema. Some examples:
- CREATE - to create objects in the database
- ALTER - alters the structure of the database
- DROP - delete objects from the database
- TRUNCATE - remove all records from a table, including all spaces allocated for
  the records are removed
- COMMENT - add comments to the data dictionary
- RENAME - rename an object

DML - Data Manipulation Language: statements used for managing data within
schema objects. Some examples:
- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain
- MERGE - UPSERT operation (insert or update)
- CALL - call a PL/SQL or Java subprogram
- EXPLAIN PLAN - explain access path to data

Q.70   List any two disadvantages of a database system.   **(3)**

     Ans: **The disadvantages of database system are:**
- Database systems are complex, difficult, and time-consuming to design.
- Substantial hardware and software start-up costs.
- Damage to database affects virtually all applications programs.
- Extensive conversion costs in moving form a file-based system to a database system.
- Initial training required for all programmers and users.

Q.71   **Explain the utilities that help the DBA to manage the database.**   **(7)**

     Ans: Every DBA uses database utilities to manage and control their databases. But
there is a lot of confusion in the field as to what, exactly, is a database utility. There are a

lot of definitions floating around out there. DBAs constantly refer to utilities, tools, solutions, and suites.

So, first of all, let's be clear on what a utility is and what is a "tool" or "solution." A utility is generally a single purpose program for moving and/or verifying database pages; examples include LOAD, UNLOAD, REORG, CHECK, COPY, and RECOVER. A database tool is a multi-functioned program designed to simplify database monitoring, management, and/or administrative tasks. A solution is a synergistic group of tools and utilities designed to work together to address a customer's business issue. A suite is a group of tools that are sold together, but are not necessarily integrated to work with each other in any way. Of course, these are just my definitions. But there are useful definitions that make it easier to discuss DBA products and programs

Q.72   Differentiate between

(i)   WHERE and HAVING clause in SQL.
(ii)   Strong entity set and weak entity set.
(iii)   Spanned and unspanned organisation.   **(3.5   x   3)**

Ans: (i) WHERE and HAVING clause in SQL
The WHERE clause is basically used for implementing conditions on every tuple of the relation.
The HAVING clause is used in combination with the GROUP BY clause. It can be used in a SELECT statement to filter groups of the records that a GROUP BY returns.
The syntax for the HAVING clause is:
SELECT column1, column2, ... column_n, aggregate_function (expression)
FROM tables
WHERE predicates
GROUP BY column1, column2, ... column_n
HAVING condition1 ... condition_n;
Aggregate_function can be a function such as SUM, COUNT, MIN, or MAX.
(ii)   Strong entity set and weak entity set: A strong entity set has a primary key. All tuples in the set are distinguishable by that key. A weak entity set has no primary key unless attributes of the strong entity set on which it depends are included. Tuples in a weak entity set are partitioned according to their relationship with tuples in a strong entity set. Tuples within each partition are distinguishable by a discriminator, which is a set of attributes. A strong entity set has a primary key. All tuples in the set are distinguishable by that key. A weak entity set has no primary key unless attributes of the strong entity set on which it depends are included. Tuples in a weak entity set are partitioned according to their relationship with tuples in a strong entity set. Tuples within each partition are distinguishable by a discriminator, which is a set of attributes.
(iii) Spanned and unspanned organization: If records are not allowed to cross block boundaries then the organisation **is** called unspanned *record organisation* otherwise it is called spanned *record organisation*

Q.73   Discuss with examples about various types of attributes present in the ER model.   **(6)**

Ans: Types of Attributes are:
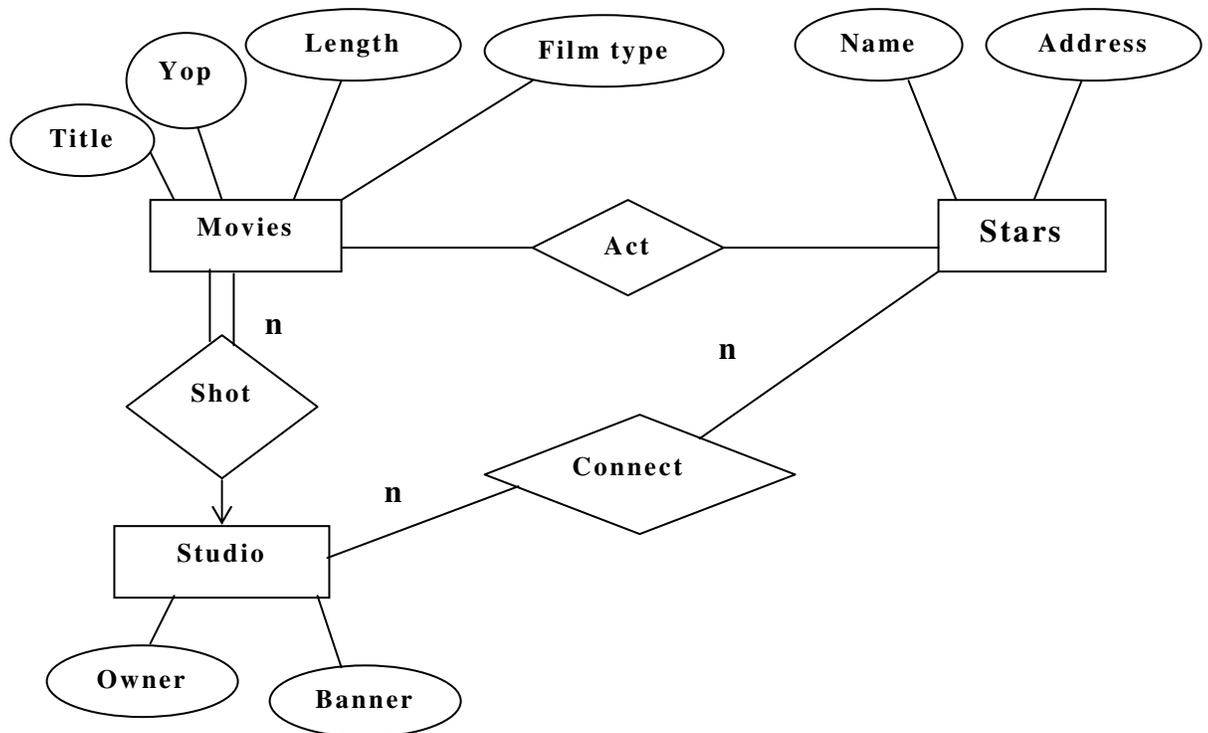   • SIMPLE attributes are attributes that are drawn from the atomic value domains

E.g. Name = {John} ; Age = {23}

- COMPOSITE attributes: Attributes that consist of a hierarchy of attributes
  E.g. Address may consists of "Number", "Street" and "Suburb" → Address = {59 + 'Meek Street' + 'Kingsford'}
- SINGLE VALUED attributes: Attributes that have only one value for each entity
  E.g. Name, Age for EMPLOYEE
- MULTIVALUED attributes: Attributes that have a set of values for each entity
  E.g. Degrees of a person: ' BSc' , 'MIT', 'PhD'
- DERIVED attributes: Attributes Contain values that are calculated from other attributes
  Eg. Age can be derived from attribute DateOfBirth. In this situation, DateOfBirth might be called Stored Attribute

Q.74 Information about films contains information about movies, stars and studios. Movies have a title, year of production, length and the film type. Stars have a name and address. Studios have a owner and a banner. Movies are shot in studios which own them. A movie is shot in only one studio. Stars are connected to one or more studios but can act in any film which may or may not be owned by the studio. Arrive at an E-R diagram. Clearly indicate attributes, keys, the cardinality ratios and participation constraints. **(8)**

Ans:

Q.75 What is the main goal of RAID technology? Describe the levels 1 through 5. (6)

Ans: **RAID** stands for **R**edundant **A**rray of **I**nexpensive (or sometimes "Independent")**D**isks.
RAID is a method of combining several hard disk drives into one logical unit (two or

more disks grouped together to appear as a single device to the host system). RAID technology was developed to address the fault-tolerance and performance limitations of conventional disk storage. It can offer fault tolerance and higher throughput levels than a single hard drive or group of independent hard drives. While arrays were once considered complex and relatively specialized storage solutions, today they are easy to use and essential for a broad spectrum of client/server applications.

RAID 1 : Mirrored set without parity. Provides fault tolerance from disk errors and failure of all but one of the drives. Increased read performance occurs when using a multi-threaded operating system that supports split seeks, very small performance reduction when writing. Array continues to operate so long as at least one drive is functioning. Using RAID 1 with a separate controller for each disk is sometimes called *duplexing*. SNIA definition.

RAID 2: Redundancy through Hamming code. Disks are synchronised and striped in very small stripes, often in single bytes/words. Hamming codes error correction is calculated across corresponding bits on disks, and is stored on multiple parity disks. SNIA definition.

RAID 3: Striped set with dedicated parity/Bit interleaved parity. This mechanism provides an improved performance and fault tolerance similar to RAID 5, but with a dedicated parity disk rather than rotated parity stripes. The single parity disk is a bottle-neck for writing since every write requires updating the parity data. One minor benefit is the dedicated parity disk allows the parity drive to fail and operation will continue without parity or performance penalty.

RAID 4: Block level parity. Identical to RAID 3, but does block-level striping instead of byte-level striping. In this setup, files can be distributed between multiple disks. Each disk operates independently which allows I/O requests to be performed in parallel, though data transfer speeds can suffer due to the type of parity. The error detection is achieved through dedicated parity and is stored in a separate, single disk unit.

RAID 5: It distributes data and parity information across all disks.

Q.76   An employee record has the following structure
             struct employee {
                 int   eno;
                 char   name[22];
                 float   salary;
                 char   dept[10];}};

   (i)   Calculate the record size R in bytes.
   (ii)   If the file has 500 records, calculate the blocking factor bfr and the number of blocks b, assuming an unspanned organization with block size B = 512 bytes.
   (iii)   What is the unused space in each block and in the last block?   **(8)**

Ans: (i)   2 + 22+ 4 + 10 =38 bytes
      (ii) bfr = floor(B/R) = [512/38] =13
         Number of blocks b = ceiling (500/13) =39
      (iii) Unused space in each block
        = 512 – 38 *13 = 512 – 494= 18 bytes
       Unused space in the last block
        = (39 *13 – 500) * 38 bytes = (507 -500) * 38 bytes
        = 7 * 38 =266 bytes

Q.77 Define the following terms
      (i) Hashing   **(ii)** Specialization
      (iii) Value set.   **(iv)** DBMS.
      (v) Host language.   **(vi)** Database state.
      (vii) Trigger.   **(2 x 7)**

Ans: **(i)** **Hashing:** Hashing is a method to store data in an array so that storing, searching, inserting and deleting data is fast (in theory it's O(1)). For this every record needs an unique key.

The basic idea is not to search for the correct position of a record with comparisons but to compute the position within the array. The function that returns the position is called the 'hash function' and the array is called a 'hash table'.

(ii) Specialization: Specialization allows you to define new kinds of information (new structural types or new domains of information), while reusing as much of existing design and code as possible, and minimizing or eliminating the costs of interchange, migration, and maintenance.

(iii) **Value set:** It is a set of values. You do not always want a user to enter junk free text into all the fields. Hence, Oracle Apps uses value set to validate that correct data is being entered in the fields in screen.

(iv) DBMS: database management system (DBMS) is computer software designed for the purpose of managing databases based on a variety of data models.

(v) **Host Language:** You can write applications with SQL statements embedded within a host language. The SQL statements provide the database interface, while the host language provides the remaining support needed for the application to execute.

(vi) **Database state:** The actual data in the database at a particular moment in time is called a database state.

(vii) Trigger: A database trigger is procedural code that is automatically executed in response to certain events on a particular table in a database. Triggers can restrict access to specific data, perform logging, or audit data modifications.

Q.78 Differentiate between natural join and outer join.

                                                         (2)

Ans: Natural join is a binary operator that is written as ($R * S$) where R and S are relations. The result of the natural join is the set of all combinations of tuples in R and S that are equal on their common attribute names. In this only one column out of columns having same name attributes is retained.

An **Outer join** contains those tuples and additionally some tuples formed by extending an unmatched tuple in one of the operands by "fill" values for each of the attributes of the other operand.

Q.79 For the relations R and S given below:

R

	A	B	C
1		2	3
4		5	6
7		8	9

S

B	C	D
2	3	10
2	3	11
6	7	12

Compute
(i) $\Pi_{A,C}(R)$ **(ii)** $\sigma_{B=2}(S)$
(iii) natural join   **(iv)** outer join   **(12)**

Ans: (i)

A	C
1	3
4	6
7	9

(ii)

B	C	D
2	3	10
2	3	11

(iii)

A	B	C	D
1	2	3	10
1	2	3	11

(iv)   Assuming left outer join

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	NULL
7	8	9	NULL

Q.80   Define union compatibility? Explain why INTERSECTION of two relations can not be performed if they are not union compatible?   **(4)**

Ans: Two relations $R(A1, A2,-------, A_n)$ and $S(B1, B2,-----, B_n)$ are said t union compatible if they have the same degree n and if dom (A_i) = dom(B_i) for all i=1---n
Intersection requires two relations to be union compatible since otherwise it is not possible to formulate the criterion as to on what basis will attributes be considered as common.

Q.81   What is a view in SQL? When can views be updated?   **(4)**

Ans: A view is a virtual table that consists of columns from one or more tables. Though it is similar to a table, it is stored in the database. It is a query stored as an object. Hence, a view is an object that derives its data from one or more tables. These tables are referred to as base or underlying tables.
Views can be updated if they are defined on a single table without any aggregate functions can be mapped to an update on the underlying base table under certain conditions**.**

Q.82   Using SQL create a view RS for the relations R and S of Q79. The view consists of the columns A and D renamed as X and Y respectively. Insert a tuple <10, 15> into it. Show the contents of the view.   **(6)**

Ans: create view RS as select A as X , D as Y from Q79;
                   insert into RS values (10, 15);

Q.83   Consider the relations defined below:
            PHYSICIAN (regno, name, telno, city)

PATIENT (pname, street, city)

VISIT (pname, regno, date_of_visit, fee)

Where the regno and pname identify the physician and the patient uniquely respectively. Express queries (i) to (iii) in SQL.

       (i)   Get the name and regno of physicians who are in Delhi.

       (ii)   Find the name and city of patient(s) who visited a physician on 31 August 2004.

       (iii)   Get the name of the physician and the total number of patients who have visited her.

       (iv)   What does the following SQL query answer

       SELECT DISTINCT name

           FROM PHYSICIAN P

          WHERE NOT EXISTS

              (SELECT *

                FROM VISIT

                WHERE regno = p.regno)   **(3.5 x 4)**

Ans: **(i)** Select name, regno from PHYSICIAN where city = 'Delhi';

(ii)   Select pname, city from PATIENT,VISIT where PATIENT.pname=VISIT.pname and date_of_visit = '31-Aug-04';

(iii) select name, count(*) from PHYSICIAN, VISIT where PHYSICIAN.regno = VISIT.regno group by Physician . regno;

(iv)   This will give the name of physicians who have not visited any patient.

Q.84   Write short notes on

       (i)   Data models.

       (ii)   Oracle database structure.

       (iii)   Group By clause in SQL.

       (iv)   Retrieval in QBE.   **(3.5 x 4)**

Ans: (i)   **Data models:** A data model is an abstract model that describes how data is represented and accessed.

The term data model has two generally accepted meanings:

A data model *theory*, i.e. a formal description of how data may be structured and accessed.

A data model *instance*, i.e. applying a data model *theory* to create a practical data model *instance* for some particular application.

(ii)   **Oracle database structure:**

The relational model has three major aspects:

Structures: Structures are well-defined objects that store the data of a database. Structures and the data contained within them can be manipulated by operations.

Operations: Operations are clearly defined actions that allow users to manipulate the data and structures of a database. The operations on a database must adhere to a pre-defined set of integrity rules.

Integrity Rule: Integrity rules are the laws that govern which operations are allowed on the data and structures of a database. Integrity rules protect the data and the structures of a database.

An ORACLE database has both a physical and a logical structure. By separating physical and logical database structure, the physical storage of data can be managed without affecting the access to logical storage structures.

(iii)   **Group By clause in SQL:** The GROUP BY clause can be used in a SELECT statement to collect data across multiple records and group the results by one or more columns.

The syntax for the GROUP BY clause is:

SELECT column1, column2, ... column_n, aggregate_function (expression)

FROM tables

WHERE predicates

GROUP BY column1, column2, ... column_n;

aggregate_function can be a function such as SUM, COUNT, MIN, or MAX.

(iv)   **Retrieval in QBE :** A "GUI" for expressing queries.

Based on the Domain Relational Calculus (DRC)

Actually invented before GUIs.

Very convenient for simple queries.

Awkward for complex queries.

QBE and IBM trademark.

But has influenced many projects

Especially PC Databases: Paradox, Access, etc.

Q.85   Explain the disadvantages of file oriented approach.   **(6)**

Ans: Applications are designed in isolation. Design of application is optimized for one application. Independently developed applications leads to data redundancy. Wasted storage space due to redundancy.

     a.   There is loss of data integrity because integrity checking is not automated.

     b.   Difficulty in accessing data.

     c.   Information is available only in reports.

     d.   Data Isolation.

     e.   Inadequate security

     f.   Limited Flexibility

     g.   High Maintenance cost.

     h.   Each data file of an application is a separate entity.

Q.86   Explain the three data models namely relational, network and hierarchical and compare their relative advantages and disadvantages.

                                                                 (10)

Ans: **Hierarchical Model:** In hierarchical model, data elements are connected to one another through links. Records are arranged in a top-down structure that resembles a tree or genealogy chart. The top node is called the root, the bottom nodes are called leaves, and intermediate nodes have one parent node and several child nodes. The root can have any number of child nodes but a child node can have only one parent node. Data are related in a nested, one-to-many set of relationships, while many-to-many relationship cannot be directly expressed.

     A child record occurrence must have a parent record occurrence; deleting a parent record occurrence requires deleting all its child record occurrences.

A network data model can be regarded as an extended form of the hierarchical model; the principle distinction between the two being that in a hierarchical model, a child record

has exactly one parent whereas in network model, a child record can have any number of parents. It may have zero also.

Data in the network model is represented by collection of records and relationship among data is represented by links, which can be viewed, as pointers. The records in the database are organized as collection of arbitrary graphs, which allows to have one-to-many as well as many-to-many relationship is a collection of data items which can be retrieved from a database, or which can be stored in a database as an undivided object. Thus, A DBMS may STORE, DELETE or MODIFY records within a database. In this way, a number of records within a network database are dynamically changed. The network model can be graphically represented as follows:

A labeled rectangle represents the corresponding entity or record type. An arrow represents the set type, which denotes the relationship between the owner record type and member record. The arrow direction is from the owner record type to the member record type.

A labeled rectangle represents the corresponding entity or record type. An arrow represents the set type, which denotes the relationship between the owner record type and member record. The arrow direction is from the owner record type to the member record type.

Each many to many relationship is handled by introducing a new record type to represent the relationship wherein the attributes, if any, of the relationship are stored. We when create two symmetrical 1:M sets with the member in each of the sets being the newly introduced record type. In this model, the relationships as well as the navigation through the database are predefined at database creation time.

In relational model the data and the relations among them are represented by a collection of tables. A tables is a collection of records and each record in a table contains the same fields. The attractiveness of the relational approach arouses from the simplicity in the data organization and the availability of ably simple to very powerful query languages. The relational model is based on a technique called "Normalization" proposed by E.F. Codd. This model reduces the complexity of the Network and Hierarchical Models. This model uses the certain mathematical operations from relational algebra and relational calculus on the relation such as projection, union and joins etc. where fields in two different tables take values from the same set, a join operation can be performed to select related records in the two tables by matching values in those fields. A description of data in terms of a data model is called a schema. In relation model, the schema for a relation specifies its name, the name of each field and the type of each field.

Navigation through relations the represent an M:N relationship is just as simple as through a 1:M relationship. This leads us to conclude that it is easier to specify how to manipulate a relational database than a network or hierarchical one. This in turn leads to a query language for the relational model that is correct, clear, and effective in specifying the required operations. Unfortunately, the join operation is inherently inefficient and demands a considerable amount of processing and retrieval of unnecessary data. The structure for the network and hierarchical model can be implemented efficiently. Such an implementation would mean that navigating through these databases, though awkward, requires the retrieval of relatively little unnecessary data.

Q.87   In an organization several projects are undertaken. Each projects can employ one or more employees. Each employee can work on one or more projects. Each project is undertaken on the required of client. A client can request for several projects. Each
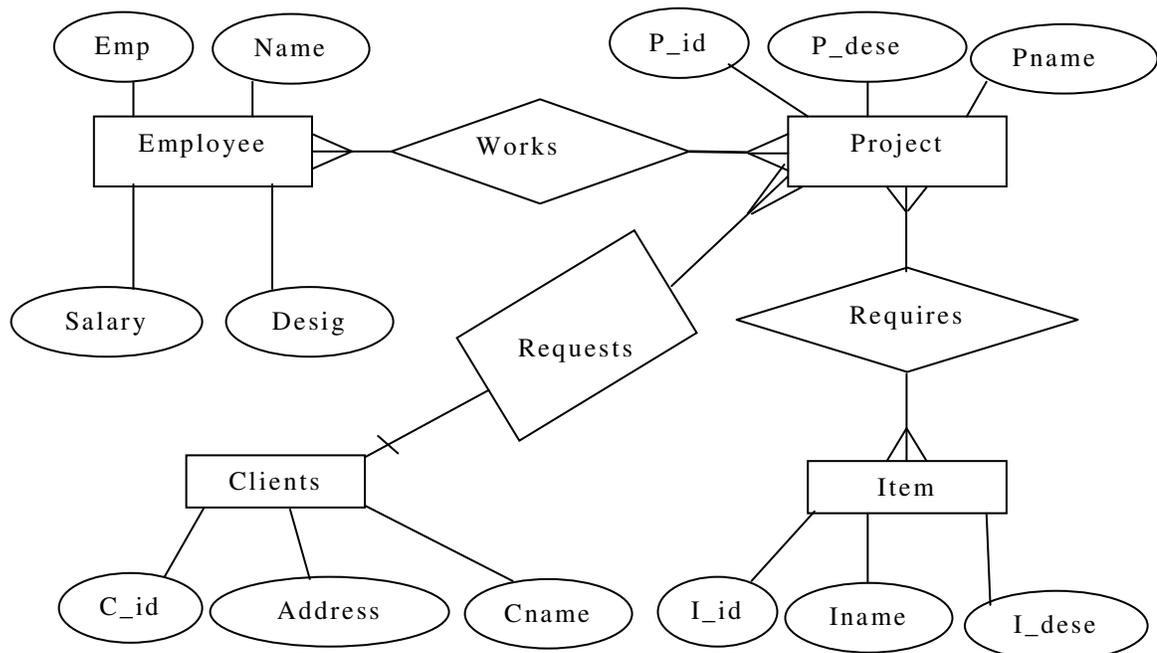
project have only one client. A project can use a number of items and a item may be used by several projects. Draw an E-R diagram and convert it to a relational schema.

(10)

Ans: The ER Diagram can be converted into the following relational schema.
Entity Sets

      (i)   Employee (Empno, Name, Salary, Desig)
      (ii)   Project (P_id, Pname, P-desc)
      (iii)   Item (I_id, Iname, I-desc)
      (iv)   Clients (C_id,Cname, Address)
      (v)   Works_on (Empno, P_id)
      (vi)   Requests (P_id, C_id)
      (vii)   Requires (P_id, I_id)

Q.88   Define : (i) Identifying relationship (ii) Specialisation / generalization
               (iii) Aggregation.   **(6)**

     Ans: (i) Identifying relationship : It is relationship between strong entity and a weak entity it is represented by a doubly outlined diamond in Chen notation.
     (ii)   **Specialisation /generalization :** This is a top down / bottom up approach to the design of database. It shows the IS_A relationship between a super class (parent) and associated sub-classes. It is represented by inverted triangle in the Chen notation.
     (iii)   **Aggregation :** This is used whenever we intend to show a relationship between an entity and the relationship. There is no provision for this in the ER Model. This is shown by placing the relationship between two entities alongwith the entities in a box and setting up a relationship between this box and another entity.

Q.89   A well-maintained relational DBMS has a high level of data integrity. What features of a relational DBMS contribute towards this level of integrity? **(6)**

Ans: Relational DBMS provides high level of data integrity, by using the following rules:

Entity Integrity: Entity integrity says that a prime attribute in a relation can not accept null values.

Referential Integrity: The referential integrity rule is related with the foreign key concept. Let R1 and R2 are two relations where R1 is having an attribute(s) with primary key. Let R2 be having a foreign key, which refers to relation R1 via the same set of attributes. Then the value of the foreign key in a tuple in R2 relation must either be equal to primary key of a tuple in a relation R1 relation or be entirely NULL.

In a more general way, the referential integrity rule states that every foreign key value must match a primary key value in an associated table. Referential integrity ensures that we can correctly navigate between related entities> Referential Integrity is a state where each foreign key value has matching primary key value. In other words, it guarantees that data in a dependent table has matching records in the table(s) on which it depends. With modern databases, referential integrity is enforced with the use of primary and foreign keys, which will not allow insertions, deletions, or changes to data that violate the rules of integrity configured by the database administrator.

Student_id	X (6)
Subject_code	X(6)
Semester_code	X(6)
Subject_name	X(20)
Lecturer_id	X(8)
Lecturer-name	X(20)
Grade	X(6)

PRIMARY KEY (student_id, subject_code)
FOREIGN KEY (subject_code) REFERENCES TABLE subjects.

The student_grade table stores the information of the grade obtained by all the students in their respective subjects taught by a lecturer in a semester. The design of this table is not normalized and it would lead to various anomalies.

Every time a grade for a student is inserted for some subject taught by a lecturer, subject_name and lecturer_name have also to be inserted. So these field will be repeated for all the students for a particular subject taught by a teacher. For example if there are five students who studied Mathematics subject taught by Mr. Ashok then these two values along with their Ids will be repeated in the relation. This will lead to redundancy. If a change is required in subject_name then it will be reqired to modify the data in both the tables student_grade and subjects, otherwise leading to a **inconsistent** state. For example if we wish to change the subject name of Maths to Mathematics in subjects table then this change has to be made in student_grade also.

If we declare a record in student_grade table, which have only one entry for a particular lecturer, then it will loose the information of that lecturer. Moreover, if a new lecturer joins, his/her information can not be inserted in the relation unless he/she teaches a subject to the students of any semester.

Q.90 Explain the relevance of Data Dictionary in a Database System. **(10)**

Ans: Data dictionary is a database in its own right residing on the disk which consist of Meta data which is = Data about all entity sets + attributes + relationships among entity sets + constraints. It consist of compiled form of definitions, structure and usage information on data stored, design decisions, usage standards, application programme descriptions, user information. It is consulted by DBMS before DML operation and by user to learn what each piece of data and various synonymous of data fields mean. Data dictionary can be integrated system where it is part of DBMS or add ons to DBMS. In integrated system data dictionary contains information concerning external, conceptual and internal level of data base. Both in source and object form. It contains source code of each data field value, frequency of its use, audit trail concerning updates and cross reference information. Present system are all add ons standards do not exist for iintegrity data dictionary with DBMS. Data dictionary should be integrated in database it defines and thus include its own definition so that it can be queried with the same language use for queering database.

Q.91 Differentiate between **(3 x 3)**
(i) Procedural and non procedural languages.
(ii) Key and superkey
(iii) Primary and secondary storage

Ans: (i) Procedural and non procedural languages - A procedural language specifies the operations to be performed on the existing data to derive the results. It also specifies the sequence of operations in which they will be performed. But, a non procedural language specifies only the result or information required not how it is obtained.
(ii) Key and superkey - A key a single attribute or a combination of two or more attributes of an entity set that is used to identify one or more instances (rows) of the set (table). If we add some additional attributes to a primary key then that *augmented key* is called as super key. Therefore, *the primary key is the minimum super key*.
(iii) Primary and secondary storage – Primary storage device stores the data temporarily. Primary storage is generally used by the processing unit to temporary store the data, intermediate results, and the final results before storing to the secondary storage because the secondary storage devices are not directly accessible by the CPU. But, if we want to store data permanently then the secondary storage devices are required. Secondary storage devices are slower than the primary storage devices.

Q.92 Consider the following relations:
BRANCH(bno, street, area, city, pcode, Tel_no, Fax_no)
STAFF(Sno, Fname, Lname, address, position , salary, bno)
Express the following queries in SQL:
(i) List the staff who work in the branch at '163 main street'
(ii) Find staff whose salary is larger than the salary of every member of staff at branch B3.

Ans: **(i)** Select Fname, Lname from STAFF, BRANCH where STAFF.bno = BRANCH.bno and street = '163 main street'

(ii)   Select Fname, Lname from STAFF where salary > (select max (salary) from Staff where bno='B3'

Q.93   **(i)**   Consider employee (e_no, e_name, e_salary, d_code), dept (d_code, d_name) and dependent (depndt_name, e_no, relation).
Show the names of employees in purchase and accounts departments with at least one dependent.

(ii)   Consider student (std_id, std_name, date_of_birth, phone, dept_name). Put the data for a student with student id200, name arun, birth date 1st February, 1985, phone number (01110 32818 and dept name English in the student table.

(iii)   A constraint named less_than_20 was defined on the field date_of_birth of table student. Delete this constraint.

(iv)   Consider the table student and list names of students in the departments other than maths and computer**.**

(v)   Consider employee table of (i) and list names of department(s) for which average salary for department is more than 10,000.

(vi)   Create role named role_table that allows a user to create tables. Using role_table allow users kripa and reena to create tables.

(vii)   Create a view emp_dep containing e_name and number of dependents from the tables employee and dependent of (i)   **(2x7)**

    Ans:(i)   SELECT e_name FROM employee, dependent, dept
         WHERE employee.d_code=dept.d_code
           AND employee.e_no=dependant.e_no
            AND d_name IN ('ACCOUNTS', 'PURCHASE')
             GROUP BY e_name
             HAVING COUNT (e_no)>=1

(ii)   INSERT INTO student VALUES (200,'arun,''01-FEB-85',
         '(0111)32818','English');

(iii)   ALTER TABLE student DROP CONSTRAINT less_than_20;

(iv)   SELECT std_name FRM student WHERE dept_name NOT IN
         ('Maths','Computer);

(v)   (SELECT d_name FROM dept
          WHERE d_code IN (SELECT d_code
           FROM emp GROUP BY d_code
         HAVING AVG(sal) ? 10000);

(vi)   CREATE ROLE role_table;
         GRANT CREATE ANY TABLE TO role_table;
           GRANT role_table TO kripa, reena;

(vii)   CREATE VIEW emp_dept AS SELECT ename,
         COUNT(*) FROM employee, dependent
   WHERE employee.e_no = dependant.e_no GROUP BY e_name;

Q.94   Mention the kind of constraints we can specify in the CREATE command of DDL.
                                               (2)

   Ans: **The kind of constraints which can be specified are as follows:-**
PRIMARY KEY, NOT NULL, UNIQUE, FOREIGN KEY, CHECK**.**

Q.95   What is the difference between a primary index and a secondary index? What are the advantages of using an index and what are its disadvantages. **(10)**

       Ans: **Primary Index:** A primary index is an ordered file whose records are of fixed length with two fields. The first field is the ordering key field-called primary key-of the data file, and the second field is a pointer to a disk block. There is one index entry in the index file for each block in the data file. Each index entry has the value of the primary key field for the first record in a block and a pointer to that block as its two field values. A major problem with a primary index is insertion and deletion of records. If we attempt to insert a record in it's correct positioning the data file, we have to not only move records to make space for the new record but also change some index entries.

Secondary Index: A secondary index is also an ordered file with two fields. The first field is non-ordering field of the data file that is an indexing field. The second field is either a block pointer or a record pointer. A secondary index on a candidate key looks just like a dense primary index, except that the records pointed to by successive values in the index are not stored sequentially.

In contrast, if the search key of a secondary index is not a candidate key, it is not enough to point to just the first record with each search-key value. The remaining records with the same search – key value could be anywhere in the file, since the records are ordered by the search key of the primary index, rather than by the search key if the secondary index. Therefore, a secondary index must contain pointers to all the records. Secondary indices improve the performance of queries that use keys other than the search key of the primary index. However, they impose a significant overhead on modification of the database. The designer of a database decides which secondary indices are desirable on an estimate of the relative frequency of query's and modifications.

Some of the advantages of using an index are:

(i)   Indexes speed up search on the indexed attributes(s). Without an index either a sequential search or some sort of binary search would be needed.

(ii)   Indexes can also speed up sequential processing of the file when the file is not stored as a sequential file.

Some of the disadvantages of using an index are :

(i)   An index requires additional storage. This additional storage can be significant when a number of indexes are being used on a file.

(ii)   Insertion, deletion and updates on a file with indexes takes more time than on a file without any indexes.

Q.96   What are the causes of bucket overflow in a hash file organization? What can be done to reduce the occurrence of bucket overflow? **(6)**

       Ans: When a record is inserted, the bucket to which it is mapped has space to store the record. If the bucket does not have enough space, a **bucket** overflow is said to occur. Bucket overflow can occur for several reasons:

Insufficient buckets : The number of buckets, which we denote n_b, must be chosen such than $n_b > n_r/f_r$, where n_r, denotes the total number of records that will be stored, and f_r, denotes the number of records that will fit in a bucket. This designation, of course, assumes that the total number of records is known when the hash function is chosen.

Skew : Some buckets are assigned more records than are others, so a bucket may overflow even when other buckets still have space. This situation is called bucket skew. Skew can occur for two reasons:

1.   Multiple records may have the same search key.
2.   The chosen hash function may result in non-uniform distribution **of** search keys.

So, that the probability of bucket overflow is reduced, the number of buckets is chosen to be $(n_r/f_r)*(1+d)$, where d is a fudge factor typically around 0.2. Some space is wasted: About 20 percent o the space in the buckets will be empty. But the benefit is that the probability of overflow is reduced.

Despite allocation of a few more buckets than required, bucket overflow can still occur. We handle bucket overflow by using overflow buckets. If a record must be inserted into a bucket b, and b is already full, the system provides an overflow bucket for b, and inserts the record into the overflow bucket, and so on.

All the overflow buckets of a given bucket are chained together in a linked list. Overflow handling using such linked list is called overflow chaining.

Q.97   Explain the concept of QBE   **(8)**

Ans: Query-by-example represents a visual/graphical approach for accessing information in a database through the use of query templates called as skeleton tables. It is used by entering example values directly into a query template to represent what is to be achieved. QBE was developed at IBMs. T.J. Watson Research Centre and it is used by many database systems for personal computers. QBE is a very powerful facility that gives the user the capability to access the information a user wants without the knowledge of any programming language.

Queries in QBE are expressed by skeleton tables. QBE has two distinct features :
-     QBE has two-dimensional syntax : Queries look like tables.
-     QBE queries are expressed "by example". Instead of giving a procedure for obtaining the desired answer, the user gives an example of what is desired.

An example of QBE is given as

EMPNO	NAME	SALARY	DESIG	DEPTNO
	P..x		"Manager"	

This query tells the system to look for tuples in EMP relation that have designation Manager. For each such tuple, the system assigns the value of the Ename attribute to the variable x and prints it.

Q.98   Describe the function of each of the following types of keys:
Primary, alternative, secondary and foreign.   **(8)**

Ans: **Primary Key :** The primary key is an attribute or a set of attributes that uniquely identify a specific instance of an entity. Every entity in the data model must have a primary key whose values uniquely identify instances of the entity.

To qualify as a primary key for an entity, an attribute must have the following properties :
*   It must have a non-null value for each instance of the entity.
*   The value must be unique for each instance of an entity
*   The values must not change or become null during the life of each entity instance.

Candidate Key and Alternate Key : In some instances, an entity will have more than one attribute that can serve as a primary key. Any key or minimum set of keys that could be a primary key is called a candidate key. Ones candidate keys are identified, one of

them is chosen as primary key. The choice of Primary key is based on guaranteed uniqueness and minimalism.

Candidate keys which are not chosen as the primary key are known as alternate keys.

Foreign Key : The primary key of one file or table which is implanted in another file or table to implement the relationships between them. Foreign keys are used to implement some types of relationships. Foreign keys do not exist in information models.

Q.99   Explain the structural components of a DBMS.   **(10)**

Ans: DML compiler, Embedded DML compiler, DDL Interpreter, Query Evaluation Engine, Authorization and integrity Manager, Transaction Manager, File Manager, Buffer Manager.

DML Compiler : Translates DML statements in a query language into low level instructions understandable by the query evaluation engine. Attempts to transforms users request into an equivalent and more efficient from for executing the query understandable by Data Manager, Interprets DDL statements and records them in a set of tables containing Meta data in a form that can be used by other components of a DBMS.

Query Evaluation engine : Vehicle executes low-level instructions generated by the DML compiler.

The Storage Manager components provide interface between the low level data stored in the database and application programs and queries submitted to the system. It is central software component of DBMS and called database control system also.

Authorization and Integrity Manager : Tests for the satisfaction of integrity constraints and checks the authority of users to access data.

Transaction Manager : Ensure the database remains in a constant (correct) state despite system failure, and that concurrent transaction executions proceed without conflicting.

File Manager : Manager allocation of space on disk storage and the data structures used to represent into stored on disk. It is also responsible for –

- Locating block of target record
- Requesting block from disk Manager
- Transmitting required record to data Manager

This file Manager can be implemented using an interface to the existing file subsystem provided by the O.S. of the host computer or it can include a file subsystem written especially for DBMS.

Disk Manager : Responsible for fetching data regulated by file Manager from disk storage into Main memory and deciding what data to cache in memory.

It is a part of the O.S. of host computer and all physical input/output operations are performed by it. It provides disk abstraction to file Manager s.t. file Manager need not be concerned with the physical characteristics of the underlying storage media.

Functions of Data Manager :
- Convert operations in users queries coming directly via query processor or indirectly in an application program from users logical view to a physical file system.
- It interfaces with file system enforces constraints to maintain consistency. And integrity of the data as well as its security. (Authorization and Integrity Manager).
- Synchronizing simultaneous operations performed by concurrent users. (Transaction Manager)
- Back up and recovery operations (Transaction Manager)

Q.100   Describe the GRANT function and explain, how it relates to security. What types of privileges may be granted? How are they revoked?   **(6)**

Ans: Since more than one user can access one database, there exists the need to restrict user from using entire database in any way. Setting up access privilege can do this. Any privilege can be granted to or revoked from user with taken of the GRANT and REVOKE statements. Each privilege can be granted or revoked individually or in a group. Such group of rights is called ROLE and rights can be granted or revoked into in the same way as to the user. Access privileges include SELECT, INSERT, UPDATE and DELETE.

GRANT {statement|role_name} ON {object_name} TO {user|role_name_2} [WITH GRANT OPTION]

statement	A statement to be granted or revoked.
role_name	The name of a role to be granted or revoked.
object_name	The name of an object (table, view or role) on which the rights will be changed.
user	The name of a user, the rights will be granted to or revoked from. All users are represented by the keyword PUBLIC.
role_name_2	The name of a role, the rights will be granted to or revoked from.
WITH GRANT	Gives the administration right on object.
OPTION	right can grant rights, revoke rights, create role, change role and delete role.

An example of granting permission to user B for reading EMP table is:
GRANT SELECT ON emp to B;
The REVOKE statement is used to revoke the permissions from the user.
The syntax for REVOKE statement is given as:
            REVOKE {statement|role_name} FROM {user|role_name};

Q.101   Define the following terms
       (i)   Catalog and meta data   (ii)   Parametric end users
       (iii) DBA   (iv)   Controlled redundancy
       (v)   Snapshot   (vi)   Data sublanguage
       (vii) High level DML   (viii) Data abstraction   **(16)**

Ans: i)   Catalog and meta data
A catalog is a compilation of records describing the contents of a particular collection or group of collections.Metadata (meta data, or sometimes metainformation) is "data about data", of any sort in any media. An item of metadata may describe an individual datum, or content item, or a collection of data including multiple content items and hierarchical levels, for example a database schema.
(ii)   Parametric end users may be given update access, but are generally not allowed to change the structure of data.
(iii)   DBA: A database administrator (DBA) is a person who is responsible for the environmental aspects of a database. In general, these include:
    ▪ Recoverability - Creating and testing Backups
    ▪ Integrity - Verifying or helping to verify data integrity

- Security - Defining and/or implementing access controls to the data
- Availability - Ensuring maximum uptime
- Performance - Ensuring maximum performance
- Development and testing support - Helping programmers and engineers to efficiently utilize the database

(iv) Controlled redundancy The replication of data within the data warehouse for the purposes of improved data access or understand ability.

(v) Snapshot A record of the state of an entry fetched from a persistent store at the time is it fetched

(vi) Data sublanguage In relational database theory, the term sublanguage, first used for this purpose by E. F. Codd in 1970, refers to a computer language used to define or manipulate the structure and contents of a relational database management system (RDBMS). Typical sublanguages associated with modern RDBMS's are QBE (Query by Example) and SQL (Structured Query Language).

(vii) High level DML A high-level or non-procedural DML allows the user to specify what data is required without specifying how it is to be obtained. Many DBMSs allow high-level DML statements either to be entered interactively from a terminal or to be embedded in a general-purpose programming language.

(viii) Data abstraction Data abstraction is a methodology that enables us to isolate how a compound data object is used from the details of how it is constructed from more primitive data objects.

Q.102   Differentiate between
        (i)   Procedural and non procedural DML
        (ii)   Forms based and graphical interface
        (iii)   Internal and external schema   **(4 x 3 = 12)**

Ans (i) Procedural and non procedural DML Procedural DML specify what data is needed and how to get those data. Non procedural DML specify what data is needed without specifying how to get those data.

(ii) Forms based and graphical interface: graphical user interface (GUI**) is a** type of user interface which allows people to interact with electronic devices like computers, hand-held devices (MP3 Players, Portable Media Players, and Gaming devices), household appliances and office equipment. A *GUI* offers graphical icons, and visual indicators as opposed to text-based interfaces, typed command labels or text navigation to fully represent the information and actions available to a user. The actions are usually performed through direct manipulation of the graphical elements.
Forms-based applications are inherently less usable and less functional than traditional GUI-based applications.

(iii) Internal and external schema: Internal Schema
- Describes the physical storage structure
- Uses a physical data model

External Schema
- Includes a number user views
- Uses a conceptual or an implementation data model

Q.103   Explain the conventions for displaying an ER schema as an ER diagram.   **(8)**

Ans: An E-R Diagram scheme may define certain constraints to which the contents of a database must conform.

Mapping Cardinalities: express the number of entities to which another entity can be associated via a relationship. For binary relationship sets between entity sets A and B, the mapping cardinality must be one of:
1. **One-to-one:** An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A. (Figure 2.3)
2. **One-to-many:** An entity in A is associated with any number in B. An entity in B is associated with at most one entity in A. (Figure 2.4)
3. **Many-to-one:** An entity in A is associated with at most one entity in B. An entity in B is associated with any number in A. (Figure 2.5)
4. **Many-to-many:** Entities in A and B are associated with any number from each other.

The appropriate mapping cardinality for a particular relationship set depends on the real world being modeled. Cardinality ratios for binary relationships are represented on ER diagram by displaying 1,M, N etc. on the diamonds.

Existence Dependencies: if the existence of entity X depends on the existence of entity Y, then X is said to be existence dependent on Y

Q.104   Describe the two alternatives for specifying structural constraints on   relationship types.
(8)

Ans: Relationship types have certain constraints that limit the possible combination of entities that may participate in relationship.
An example of a constraint is that if we have the entities Doctor and Patient, the organization may have a rule that a patient cannot be seen by more than one doctor. This constraint needs to be described in the schema.There are two main types of structural relationship constraints on: cardinality ratio, and participation

Q.105   Discuss the techniques for a hash file to expand and shrink dynamically. What are the advantages and disadvantages of each?   **(8)**

Ans:
The hashing techniques that allow dyanamic file expansion are:
 (i) Extendible hashing
 (ii) Linear hashing
The main advantage of extendible hashing that makes it attractive is that performance of the file does not degrade as the file grows. Also, no space is allocated in extendible hashing for future growth, but additional buckets can be allocated dynamically as needed. A disadvantage is that the directory must be searched before accessing the buckets themselves, resulting in two blocks accesses instead of one in static hashing.

Q.106   What are the reasons for having variable length records? What are the various ways to store variable length records?   **(4)**

Ans: **Variable-length records arise in a database in several ways:**
(i)Storage of multiple items in a file.
(ii)Record types allowing variable field size
(iii)Record types allowing repeating fields

Various ways of storing variable length records:
(i)   In fixed form by allocating every record, maximum amount of space that can be occupied a record and padding smaller sized records with nulls.
(ii) Special separator characters between the records to delineate variable length records.

Q107   Discuss the mechanism to read data from and write to a disk.   **(4)**

Ans: Disk read/write heads are mechanisms that read data from or write data to disk drives. The heads have gone through a number of changes over the years.

In a hard drive, the heads 'fly' above the disk surface with clearance of as little as 3 nanometres. The "flying height" is constantly decreasing to enable higher area density. The flying height of the head is controlled by the design of an air-bearing etched onto the disk-facing surface of the *slider*. The role of the air bearing is to maintain the flying height constant as the head moves over the surface of the disk. If the head hits the disk's surface, a catastrophic head crash can result.

The heads themselves started out similar to the heads in tape recorders—simple devices made out of a tiny C-shaped piece of highly magnetizable material called ferrite wrapped in a fine wire coil. When writing, the coil is energized, a strong magnetic field forms in the gap of the C, and the recording surface adjacent to the gap is magnetized. When reading, the magnetized material rotates past the heads, the ferrite core concentrates the field, and a current is generated in the coil. The gap where the field is very strong is quite narrow. That gap is roughly equal to the thickness of the magnetic media on the recording surface. The gap determines the minimum size of a recorded area on the disk. Ferrite heads are large, and write fairly large features.

Q.108   Consider the following relations
RENTER(rno, fname, lname, address, tel_no, pref_type, max_rent)
VIEWING(rno, pno, date, comment)
PROPERTY_FOR_RENT(pno, street, area ,city, pcode, type, rooms, rent)
Express the following queries in relational algebra.
(i) List the name and comments of all renters who have viewed a property.
(ii) Identify all renters who have viewed all properties with three rooms.

Ans: (i) $\Pi_{fname,lname,comment}$(RENTER $\bowtie$ VIEWING)

   (ii) $\Pi_{fname,lname}$ ($\sigma_{rooms\,=3}$ (PROPERTY_FOR_RENT)$\bowtie$RENTER
           $\bowtie$ VIEWING)

Q.109   Consider the relations
City (city_name, state)
Hotel (name, address)
City_hotel (hotel_name, city_name, owner)
Answer the following queries in relational algebra
(i)   Find the names and address of hotels in Agra.
(ii)   List the names of cities which have no hotel.
(iii)   List the names of the hotels owned by 'Taj Group'.   **(9)**

Ans: (i) $\Pi_{name,\ address}$ ($\sigma_{city\_name=\ 'Agra'}$ (Hotel$_{name=city\_name}$ $\bowtie$ City_hotel))
(ii) Π_{city_name} (City) – Π_{city_name}(City_hotel)

(iii) $\Pi_{\text{hotel\_name}} (\sigma_{\text{owner='Taj Group'}} (\text{City\_hotel}))$

Q.110   Explain the difference between using functions with and without grouping attributes in relational algebra. Give examples.   **(6)**

Ans: Group functions are used to group data of similar type.
GAMMA = grouping and aggregation
Applying $\text{GAMMA}_L(R)$
   • Group R according to all the grouping attributes on list L.
            o   That is: form one group for each distinct list of values for those attributes in R.
   • Within each group, compute AGG(A) for each aggregation on list L.
   • Result has one tuple for each group:
            o   The grouping attributes and
            o   Their group's aggregations.
Example: Grouping /Aggregation
R = (A B C)
      1 2 3
      4 5 6
      1 2 5
$\text{GAMMA}_{A,B,\text{AVG}(C)} (R) = ?$
First, group R by A and B :
      A B C
      1 2 3
      1 2 5
      4 5 6
Then, average C within groups:
A B AVG(C)
      1 2 4
      4 5 6

Q.111   Define the following with respect to SQL
(i)   Specifying alias   (ii) UNIQUE function
(iii) ORDER BY clause   (iv) LIKE predicate
(v) Asterisk (*)   **(10)**

Ans:(i) Specifying alias: is used to rename column or attribute in a table.
(ii)   UNIQUE function: If **UNIQUE** is specified then only **unique** values are used to calculate the mean.
(iii) ORDER BY clause: The **ORDER BY clause** allows you to sort the records in your result set. The **ORDER BY clause** can only be used in select statements.
(iv) LIKE predicate: The **LIKE predicate** searches for strings that have a certain pattern.
(v)   Asterisk (*): **In SQL**, the columns for all the tables and views in the FROM clause will be displayed.

Q.112   Consider the relations given below
Borrower (id_no, name)

Book (accno., title, author, borrower_idno)

(a) Define the above relations as tables in SQL making real world assumptions about the type of the fields. Define the primary keys and the foreign keys. **(7)**

(b) For the above relations answer the following queries in SQL

What are the titles of the books borrowed by the borrower whose id-no in 365.

  (i) Find the numbers and names of borrowers who have borrowed books on DBMS in ascending order in id_no.

  (ii) List the names of borrowers who have borrowed at least two books. **(9)**

Ans: **a)** Create table Book
      (Accno int Primary Key,
      title char(30),
      author char(30),
      borrow-idno int references Borrower_id.no);
      Create table borrower
        (id-no int Primary Key,
      name char(30));

b) **(i)** Select title from Book, Borrower where Borrower.id_no =Book.borrower-idno and borrower.id_no = 365

    (ii) Select id_no, name from Borrower, Book where Borrower.id_no= Book.borrower_idno and title= 'DBMS' order by id_no asc;

    (iii) Select name from Borrower, Book where Borrower.id_no = book.borrower_id_no having count (*) > 2

Q.113 Describe the structure of Oracle data dictionary. **(4)**

Ans: Generally, the data dictionary consists of base tables and user-accessible views. The base tables contain all database information that is dynamically updated by Oracle RDBMS. Oracle strictly discourages using those tables even for selects; the database users normally have no access to them, and even DBAs do not typically query these tables directly. The information stored in the base tables is cryptic and difficult to understand. The user-accessible views summarize and display the information stored in the base tables; they display the information from the base tables in readable and/or simplified form using joins, column aliases, and so on. Different Oracle users can have SELECT privileges on different database views.

Q.114 In Oracle what is system global area and how is it organized? **(6)**

Ans: The SGA is a collection of shared memory areas that along with the Oracle processes constitute an Oracle instance. All SGA components allocate and deallocate space in units of granules. Granule size is determined by total SGA size. On most platforms, the size of a granule is 4 MB if the total SGA size is less than 1 GB, and granule size is 16MB for larger SGAs. Some platform dependencies arise. For example, on 32-bit Windows, the granule size is 8M for SGAs larger than 1GB

Q.115 Write a short note on QBE**. (6)**

Ans: Stands for "Query By Example." QBE is a feature included with various database applications that provides a user-friendly method of running database queries.

Typically without QBE, a user must write input commands using correct SQL (Structured Query Language) syntax. This is a standard language that nearly all database programs support. However, if the syntax is slightly incorrect the query may return the wrong results or may not run at all. The Query By Example feature provides a simple interface for a user to enter queries. Instead of writing an entire SQL command, the user can just fill in blanks or select items to define the query she wants to perform. For example, a user may want to select an entry from a table called "Table1" with an ID of 123. Using SQL, the user would need to input the command, "SELECT * FROM Table1 WHERE ID = 123". The QBE interface may allow the user to just click on Table1, type in "123" in the ID field and click "Search." QBE is offered with most database programs, though the interface is often different between applications. For example, Microsoft Access has a QBE interface known as "Query Design View" that is completely graphical. The phpMyAdmin application used with MySQL, offers a Web-based interface where users can select a query operator and fill in blanks with search terms. Whatever QBE implementation is provided with a program, the purpose is the same – to make it easier to run database queries and to avoid the frustrations of SQL errors

Q.116   What three main types of actions involve databases? Briefly discuss these.   **(6)**

Ans: The three main types of actions involving databases are:
(i)   **Defining:** It involves specifying the data types, structures and constraints of data types, structures and constraints the data to be stored in the database.
(ii)   **Manipulating:** It includes functions such as querying the database to retrieve specific data updating the database to reflect changes in the miniworld and generating reports form data.
(iii) Sharing: It allows multiple users and programs to access the database simultaneously.

Q.117   What are the situations when DBMS should not be used?   **(6)**

Ans: DBMS should not be used in situations like:
       (i)   No need of security.
       (ii)   Not difficulty to access the data
       (iii)   No redundancy and inconsistency etc.

Q.118   What is the difference between a database schema and a database state?
                                                                   (4)
Ans: The collection of information stored in database at particular moment in time is called database state while the overall design of database is called database schema.

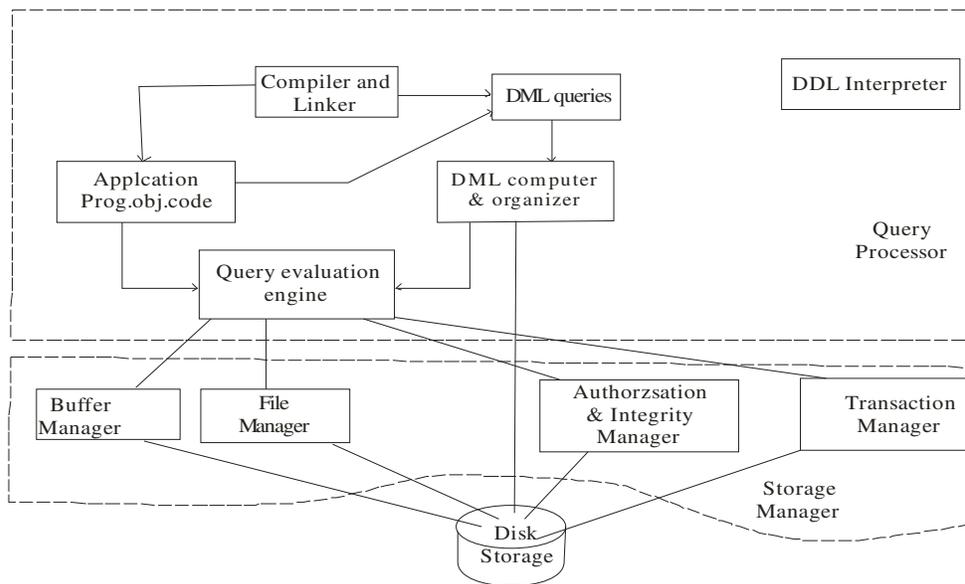**Q.119**   With the help of a diagram describe the typical component modules of a DBMS.   **(8)**

Ans: The functional components of a database system can be broadly divided   into:
1) Storage manager   and   **2)** query processor

1) Storage manager components includes:

(i) Authorization and integrity manager to test integrity constraints and authorized access to data.

(ii) Transaction manager to ensure that database remains in a   consistent state.

(iii) File manager to manage disk storage space.

(iv) Buffer manager is responsible for buffering data.

2)   **The query processor component includes:**

(i) DDL interpreter to interpret DDL statements

(ii)   DML compiler which translation DML queries into query evaluation plan for query evaluation engine.

(iii)   Query evaluation engine execute low level instructions generated by DML compiler.

Q.120   Define the following:

     (i) Participation role   **(ii)** Recursive relationship type

     (iii) Composite attributes   **(iv)** Entity

     Ans: (i) Participation role: roles are indicating in E-R diagram by labelling the lines that connect diamonds to rectangles.

     (ii) **Recursive relationship type:** A recursive relationship is one in which the same entity participates more than once in the relationship

     (iii) **Composite attribute:** A composite attribute has multiple components, each of which is atomic or composite.

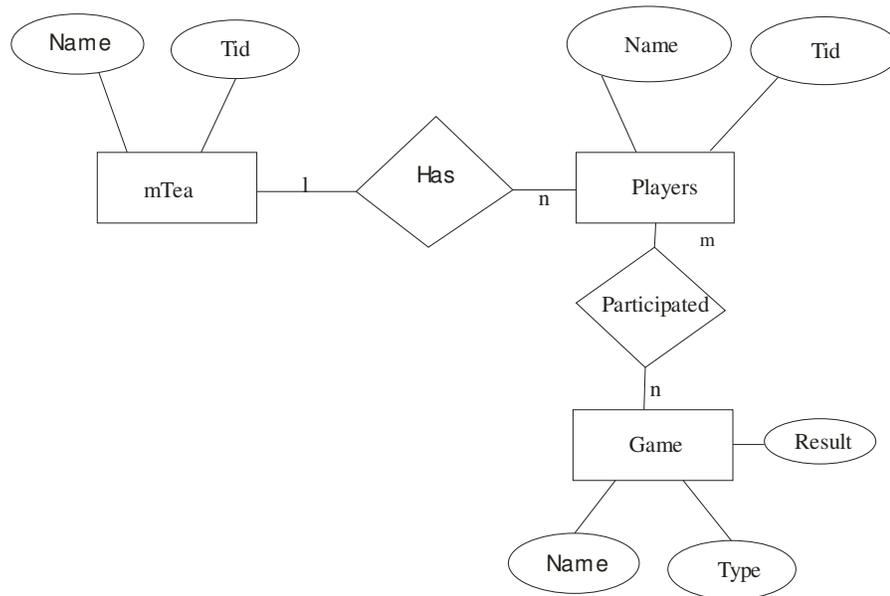   **(iv)** **Entity:** A DBMS entity is *either* a thing in the modeled world *or* a drawing element in an ERD.

Q.121   A database is to be constructed to keep track of the teams and games of a sport league. A team has a number of players, not all of whom participate in each game. It is desired to keep track of the players participating in each game of each team and the result of the game.

Create an ER diagram, completely with attributes, keys and constraints, for the above description. State any assumptions that you make. **(8)**

Ans:

Q.122   Discuss the types of integrity constraints that must be checked for the update operations – Insert and Delete. Give examples. **(8)**

Ans: Insert operation can violet any of the following four constraints:
1) Domain constraints can be violated if given attribute value does not appear in corresponding domain.
2) Key constraints can be violated if given attribute value does not appear in corresponding domain.
3) Entity integrity can be violated if the primary key of the new tuple t is NULL.
4) Referential integrity can be violated if value of any foreign key in t refers to a tuple that does not exist in referenced relation.

Delete operation can violate only referential integrity constraints, if the tuple being deleted is referenced by the foreign keys from other tuples in the database.

Q.123   Differentiate between the following giving advantages and disadvantages of each.
        (i)   Primary and secondary storage.
        (ii)   Open addressing and chaining for collision resolution.
        (iii)   Unordered and ordered file. **(12)**

Ans: i) Primary and secondary storage
*Computer storage is classified into primary(main) memory and secondary(peripheral) storage.
a) Primary storage is usually RAM; ~10ns access time
b) Secondary storage is usually hard disk drives; ~10ms access time
c) Secondary storage is a lot cheaper than primary, 3¢/Mb vs.$1/Mb.
    • Secondary storage is persistent

a)   Databases are stored in secondary memory, and large databases are manipulated in secondary storage.

b)   We need to minimise disk accesses when accessing and manipulating databases.

(ii)   Open addressing and chaining for collision resolution

In open addressing, proceeding from the occupied position specified by hash address the program checks the subsequent positions in order until an unused (empty) position is found.

Chaining: In this method various overflow locations are kept, usually by extending the way with a number of overflow positions. Additionally, a pointer field is added to each record location. A collision is resolved by placing the new record in an unused overflow location and setting the pointer of occupied hash address location to the address of that overflow location. A linked list of overflow records for each hash address is thus maintained.

(iii)   Unordered and ordered file

Unordered file do no has any sequence while ordered file has arranged in some sequence and data are assigned in ordered format.

Q.124   What is recursive closure? Why is it not possible to define this operation in relational algebra?   **(4)**

Ans: Recursive closure is applied to recursive relationship. An example of recursive operation is to retrieve all SUPERVISEES of an Employee e at all levels- that is , all EMPLOYEE e directly supervised by e; all employees e" directly supervised by each employee e" and so on

Q.125   Consider the two tables R1 (A,B,C) and R2 (P,Q,R) shown below

R1

A	B	C
a	15	10
a	25	8
b	15	5

R2

P	Q	R
b	15	5
c	10	8
b	5	6

Show the results of the following operations:

(i) $\Pi_{A,B}(R1)$

(ii) $F_{COUNT\ P, AVERAGE\ Q}(R2)$ (where F is the aggregate function)

(iii) $R_1 \times R_{1.B} > R_{2.Q} R_2$

(iv)   R1 X R2   **(12)**

Ans: (i)

A	B
a	15
a	25
b	15

(ii)

Count p	Average Q
3	10

(iii) R_1 x $R_{1.B} > R_{2.Q}$ R_2

A	B	C	P	Q	R
a	15	10	c	10	8
A	15	10	b	5	6
a	25	8	b	15	5
a	25	8	c	10	8
a	25	8	b	5	6
b	15	5	c	10	8
b	15	5	b	5	6

(iv) R_1 x R_2

A	B	C	P	Q	R
a	15	10	b	15	10
a	15	10	c	10	8
a	15	10	b	5	6
a	25	8	b	15	5
a	25	8	c	10	8
a	25	8	b	5	6
b	15	5	b	15	5
b	15	5	c	10	8
b	15	5	b	5	6

Q.126   Explain the EXISTS and UNIQUE functions of SQL. Give an example for each.

           (6)

Ans: EXISTS: The `EXISTS` function takes one parameter which is a SQL statement. If any records exist that match the criteria it returns true, otherwise it returns false. This gives you a clean, efficient way to write a stored procedure that does either an insert or update.
UNIQUE: If **UNIQUE** is specified then only **unique** values are used to calculate the mean.

Q.127   What is NULL? Give an example to illustrate testing for NULL in SQL.

           (4)

    Ans: The NULL SQL keyword is used to represent either a missing value or a value that is not applicable in a relational table.
           Consider there is a relation:
            Person(id, name, address, phone)
    Now to find ids and names of person who do not have a phone is:

Select id, name
from Person
where phone is null

Q.128   Describe substring comparison in SQL. For the relation Person(name, address), write a
SQL query which retrieves the names of people whose name begins with 'A' and address
contains 'Bangalore'.   **(6)**

Ans: SUBSTR is used to extract a set of characters from a string by specifying the
character starting position and end position and length of characters to be fetched.
example substr('hello',2,3) will return 'ell'
Select name
from Person
where name like 'A%' and address = 'Bangalore'

Q.129   Consider the following relations with keys underlined
Street (<u>name</u>, location, city)
House (<u>number</u>, street_name)
Lives (<u>name</u>, house_number)
Define the above relations as tables in SQL making real world assumptions about the
type of the fields. Define the primary keys and the foreign keys.   **(7)**

Ans: Create table street (name character(30) primary key, location character(30), city
character(30));
Create table house(number integer primary key, street_name character(30) references
street(name));
Create table lives (name character(30) primary key, house number integer references
house(number));

Q.130   For the relations given in Q129 answer the following queries in SQL

(i)   Get the names of persons who live in the street named 'Mahatma Gandhi'.
(ii)   Get the house numbers street wise.
(iii) Get the numbers of houses which are not occupied.   **(9)**

Ans: **(i)**   Select lives.name from Street, House, Lives where Street.name =
House.street_name and House. Number = Lives. House_number and street_name =
'Mahatma Gandhi';
(ii)   Select number from House
order by street_name;
(iii) Select number from house
minus
Select house_number from lives;

Q.131   How is the database organised in Oracle?   **(8)**

Ans: Database organized in Oracle in terms of table and table contains
attributes and values.

75

A database consists of one or more logical units called table spaces. Each table space in turn, consists of one or more physical structures called data files. These may be either, files managed by the operating system or raw devices.

Oracle database consists of following table spaces

(i) System **(ii)** Table spaces to store user data **(iii)** Temporary table spaces.

The space in table space is divided into units, called segments, that each contain data for a specific data structure. These are four types of segments:

(i)   Data segments

(ii)   Index segments

(iii) Temporary segments

(iv)   Rollback segments

Q.132   Explain the following with respect to QBE

(i).   CNT

(ii)   I

(iii) P

(iv)   D   **(8)**

    Ans: (i) CNT: It is one of the aggregation commands provide in QBE. It is used to identify number of tuples

(ii)   I : It is the command in QBE which is used to insert tuple (s) into the   database

(iii)   P: It is the command that is used to print (logically display) the value of the attribute in whose cell it is written. To display the entire relation, P has to be written in every field or by placing a single P in the column headed by the   relation   name   also displays entire relation.

(iv)   D: It is the command in QBE to delete tuple(s) from the relation.

Q.133   Discuss the differences between the candidate keys and the primary key of a relation. Give example to illustrate your answer.

    Ans: A candidate key is one which can be used as primary key that is not null and unique constraint both holding true. In short all primary keys are definitely candidate keys. That is one of the candidate keys is chosen as primary key.

Q.134   What are the DBMS languages? Briefly explain.   **(6)**

    Ans: Data Definition language (DDL):A database schema is specified by a set of definitions expressed by a special   language   called   data   definition   language. For example, the following statement in the SQL defines the account table

Create table account (account_number char(10), balance integer)

Execution of the above DDL statement creates the account table. DDL allows a user to specify storage structure and access methods used by the database system as well as specify certain consistency constraints like balance of an account should on fall below Rs1000.

Data Manipulation language (DML): A data-manipulation language (DML) is a language that enables users to access or manipulate data as organized by the appropriate data model.

There are basically two types:
1)   Procedural DMLs require a user to specify what data are needed and how to get those data.
2)   Declarative DMLs (also referred to as nonprocedural DMLs) require a user to specify what data are needed without specifying how to get those data.

Q.135   Explain the terms primary key, candidate key, alternate key and secondary key. In the given table identify each key.
STUDENT(SID,Regno, Name, City)   **(8)**

Ans: Primary Key: The primary key of a relational table uniquely identifies each record in the table. It can either be a normal attribute (or set of attributes) that is guaranteed to be unique (such as Social Security Number in a table with no more than one record per person) or it can be generated by the DBMS (such as a globally unique identifier, or GUID, in Microsoft SQL Server).
Candidate Key: A candidate key is a combination of attributes that can be uniquely used to identify a database record without any extraneous data. Each   table may have one or more candidate keys**.** One of these candidate keys is selected as the table primary key.
Alternate Key: An alternate key (**or secondary key**) is any candidate key which is not selected to be the primary key (PK).
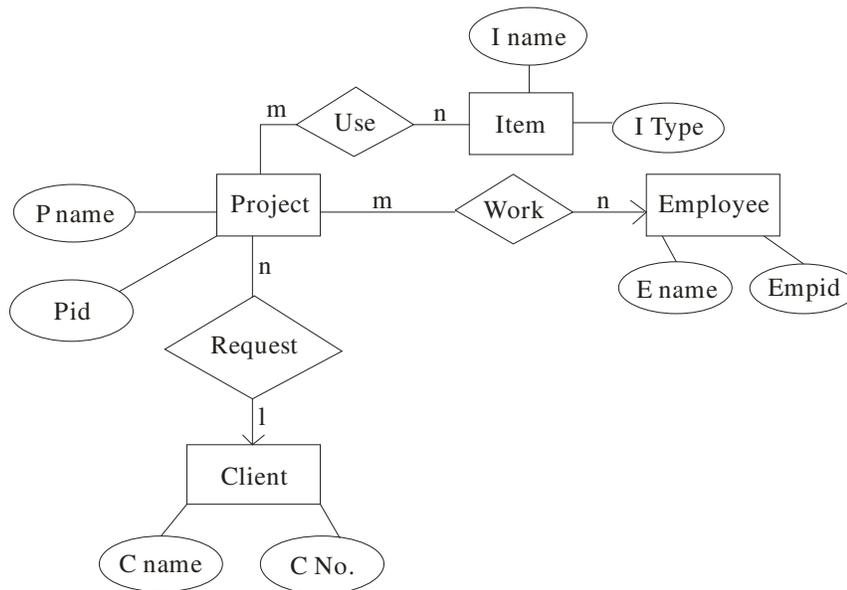For example, a relational database with a table "employee" could have attributes like "employee_id", "bank_acct_no", and so on. In this case, both "employee_id" and "bank_acct_no" serve as unique identifiers for a given employee, and could thus arguably be used for a primary key. Hence, both of them are called "candidate keys".   If, **for** example, "bank_acct_no" was chosen as the primary key, "employee_id" would become the alternate key
Secondary Key: Same as above
In above table SID is a primary key, SID and Regno are candidate keys and SIDor Regno are alternate or secondary key.

Q.136   In an organisation several projects are undertaken. Each projects can employ one or more employees. Each employee can work on one or more projects.Each project is undertaken on the request of client. A client can request for several projects. Each project has only one client. A project can use a number of items and a item may be used by several projects. Draw an E-R diagram and convert it to a relational schema.   **(10)**

Ans:

The relational schema has following tables
1)   Project (Pid, Pname)
2)   Employee (Empid, Ename)
3)   Client (Cno, Cname)
4)   Item (I type, I name)
5)   Use (Pid, I type)
6)   Work (Pid, Empid)
7)   Request (Pid, Cno)

Q.137   Define
        (i)   Identifying relationship.
        (ii)   Specialisation / generalization.
        (iii) Aggregation.   **(6)**

 Ans: **(i)**   **Identifying relationship:** An identifying relationship means that the child table cannot be uniquely identified without the parent. For example, you have this situation in the intersection table used to resolve a many-to-many relationship where the intersecting table's Primary Key is a composite of the left and right (parents) table's Primary Keys
    (ii)   **Specialisation / generalization:** Generalization/Specialization represents the is a relationship set, an essential element of the object oriented paradigm. The main idea in Generalization/Specialization is that one object class (the specialization) is a *subset* of another (the generalization).
    (iii) **Aggregation:** Aggregation refers to an abstraction in which a relationship between objects is regarded as a higher-level object.

Q.138   What is hash file organization? What are the causes of bucket overflow in a hash file organization? What can be done to reduce the occurrence of bucket overflow?   **(8)**

Ans: Hashing involves computing the address of a data item by computing a function on the search key value.

A hash function h is a function from the set of all search key values K to the set of all bucket addresses B.

We choose a number of buckets to correspond to the number of search key values we will have stored in the database.

To perform a lookup on a search key value K_i, we compute $h(K_{i)}$, and search the bucket with that address.

If two search keys i and j map to the same address, because, $h(K_i)=h(K_j)$ then the bucket at the address obtained will contain records with both search key values.

The hash functions selected should be such that it should result in uniform distribution of search keys.

Q.139 What do you understand by RAID? Explain RAID Level 4 and Level5.

(8)

Ans: RAID — which stands for Redundant Array of Inexpensive Disks (as named by the inventor), or alternatively Redundant Array of Independent Disks (a less relative name, and thus now the generally accepted one) — is a technology that employs the simultaneous use of two or more hard disk drives to achieve greater levels of performance, reliability, and/or larger data volume sizes.

RAID Level 4: In this setup, files can be distributed between multiple disks. Each disk operates independently which allows I/O requests to be performed in parallel, though data transfer speeds can suffer due to the type of parity. The error detection is achieved through dedicated parity and is stored in a separate, single disk unit.

RAID Level 5: Distributed parity requires all drives but one to be present to operate; drive failure requires replacement, but the array is not destroyed by a single drive failure. Upon drive failure, any subsequent reads can be calculated from the distributed parity such that the drive failure is masked from the end user. The array will have data loss in the event of a second drive failure and is vulnerable until the data that was on the failed drive is rebuilt onto a replacement drive

Q.140 Consider the relations:
PROJECT(proj#,proj_name,chief_architect)
EMPLOYEE(emp#,emp_name)
ASSIGNED(proj#,emp#)
Use relational algebra to express the following queries:
    (i)   Get details of employees working on project COMP33.
    (ii)   Get the employee number of employees who work on all projects.
    (iii)   Get details of project on which employee with name 'RAM' is working. **(9)**

Ans:(i) EMPLOYEE$\bowtie \Pi_{Emp\#}(\sigma_{project\#='comp33'}$(ASSIGNED))

(ii) $\Pi_{Emp\#}$(EMPLOYEE)-(ASSIGNED $\div \Pi_{project\#}$(PROJECT))

(iii)   PROJECT$\bowtie(\Pi_{project\#}(\sigma_{Emp\_name='RAM'}$(EMPLOYEE)$\bowtie$ASSIGNED))

Q.141 Differentiate between join and outer join.   **(4)**

Ans:Outer joins return all rows from at least one of the tables or views mentioned in the FROM clause, as long as those rows meet any WHERE or HAVING search conditions.

All rows are retrieved from the left table referenced with a left outer join, and all rows from the right table referenced in a right outer join. All rows from both tables are returned in a full outer join.

A table can be joined to itself in a **self-join.**

Q.142 Consider a table student (std_id, std_name, date_of_birth, percent_marks, dept_name). Write a QBE query to display names of Computer Science department students who have scored more than 80%.   **(3)**

Ans:

Stud_id	Stud_name	date_of_birth	Percent_marks	Dept_name
	P._S		>80	Computer Science Department

Q.143 Consider the relations
EMP(ENO,ENAME,AGE,BASIC_SALARY)
WORK_IN(ENO,DNO)
DEPT(DNO,DNAME,CITY)
Express the following queries in SQL

(i)   Find names of employees who work in a deptt. in Delhi.
(ii) Get the deptt. number in which more than one employee is working.
(iii) Find name of employee who earns highest salary in 'HR' department.

                                                                            (9)

Ans:(i) select ENAME from EMP, WORK_IN, DEPT where EMP.ENo= WORK_IN.ENO and WORK_IN.DNO= DEPT.DNO and CITY=   'Delhi';
(ii)   select DNO from WORK_IN group by DNO having count(*) >1
(iii)   select ENAME from EMP e where BASIC_SALARY >= (select max(BASIC_SALARY) from DEPT,WORK_IN where DNAME = 'HR' and e.ENO = WORK_IN.ENO and WORK_IN. DNO= DEPT.DNO)

Q.144 Explain various kinds of constraints that can be specified using CREATE TABLE command. Explain CASCADE and RESTRICT clauses of DROP SCHEMA command.
                                                                            (7)

Ans:

Constraint	Description
PRIMARY KEY	Determines which column(s) uniquely identifies each record. The primary key cannot be NULL, and the data value(s) must be unique.
FOREIGN KEY	In a one-to-many relationship, the constraint is added to the "many" table. The constraint ensures that if a value is entered into a specified column, it must already exist in the "one" table, or the record is not added.
UNIQUE	Ensures that all data values stored in a specified column are unique. The UNIQUE constraint differs from the PRIMARY KEY constraint in that it allows NULL values.
CHECK	Ensures that a specified condition is true before the data value is added to a table. For example, an order's ship date cannot be earlier than its order date.
NOT NULL	Ensures that a specified column cannon contain a NULL value. The NOT NULL constraint can only be created with the column-level approach to table creation.

Prevents the DROP from taking place if any dependent objects exists (RESTRICT) or causes all dependent objects do also be dropped (CASCADE).

Q.145   Write short notes on following:
   (i)   Extension and Intension.
   (ii) Weak and strong entity type.
   (iii) Views in SQL.
   (iv) Built in function in QBE.

(4 × 4 = 16)

Ans: (i)   In any data model, it is important to distinguish between the description of the database and the database itself the description of a database is called database schema, which is specified during database design and is not expected to change frequently. The actual data in a database is database itself. May change quite frequently.

The schema is sometimes called the intension, and a database state is called an extension of the schema.

(ii)   Weak and Strong entity type: An entity set that does not have a primary key is referred to as a weak entity set. The existence of a weak entity set depends on the existence of a strong entity set; it must relate to the strong set via a on-to-many relationship set. A discriminator of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set. The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence depends, plus the weak entity set's discriminator.

(iii) We define a view in SQL by using the create view command. To define a view, we must give the view a name and must state the query that computers the view. The form of the create view command is

Create view z as <query expression> where z is the view name

< query expression > is any legal query expression.

(iv) Built in function in QBE: QBE includes several built in functions like AVG, MAX,MIN,SUM and CNT. We must prefix these operators with ALL to create a multiset on which the aggregate operation is evaluated. The ALL operator ensures that duplicates

are not eliminated. We can use UNQ to specify that we want duplicates eliminated. QBE also offers the ability to compute functions on groups of tuples using the G operator, which is analogous to SQL's group by construct.

Q.146   What is Oracle Process? Explain any four processes started by Oracle.   **(8)**

Ans: Oracle Process is as follows:
SMON: The System Monitor carries out a crash recovery when a crashed instance is started up again. It also cleans temporary segments.
PMON: The Process Monitor checks if a user process fails and if so, does all cleaning up of resources that the user process has aquired.
DBWR : The Database Writer writes *dirty blocks* from the *database* buffer to the datafiles. How many DBWn Processes are started is determined by the initialization parameter DB_WRITER_PROCESSES. DBWR also writes the *actual SCN* with the Block.
LGWR : The Log Writer writes the redo log buffer from the SGA to the *online redo log file*.

Q.147   What is DBMS and what are functions of DBMS   **(10)**
Ans: DBMS consist of collection of integrated data and set of program to access those data.
The functions performed by a typical DBMS are the following:
- **Data Definition**: The DBMS provides functions to define the structure of the data in the application. These include defining and modifying the record structure, the type and size of fields and the various constraints/conditions to be satisfied by the data in each field.
- **Data Manipulation:** Once the data structure is defined, data needs to be inserted, modified or deleted. The functions which perform these operations are also part of the DBMS. These functions can handle planned and unplanned data manipulation needs. Planned queries are those which form part of the application. Unplanned queries are ad-hoc queries which are performed on a need basis.
- **Data Security & Integrity:** The DBMS contains functions which handle the security and integrity of data in the application. These can be Thus the DBMS provides an environment that is both convenient and efficient to use when there is a large volume of data and many transactions to be proved

Q.148   How many types of users works on database?   **(6)**

Ans: Users are differentiated by the way they expect to interact with the system
(i)   Application programmers–interact with system through DML calls
(ii)   Sophisticated users–form requests in a database query language
(iii)   Specialized users–write specialized database applications that do not fit into the traditional data processing framework
(iv)   Naive users–invoke one of the permanent application programs that have been written previously Examples, people accessing database over the web, bank tellers, clerical staff

Q.149   Describes the various relationship constraints by giving suitable example.   **(8)**

Ans: Constraints on relationships: There are two types of constraints on relationships.
1) Mapping cardinalities or cardinality ratios express the number of entities to which another entity can be associated via a relationship set. For a binary relationship set between entities A and B. The mapping cardinality may be any one of the following:
a) **One to One:** An entity in A is associated with at most one entity in B and an entity in B is associated with at most one entity in A.
b) **One to Many:** An entity in A can be associated with any number in B but an entity in B is associated with at most one in A.
c) **Many to one:** An entity in A is associated with at most one in B but an entity in B is associated with any number in A.
d) **Many to Many:** An entity in A and B can be associated with any number of entities in the other entity set.
2) **Participation constraints:** The participation of an entity set E in a   relationship set R is said to be total if every entity in E participates in at least one relationship in R. If only some entities in E participation of entity set E in relationship R is said to be partial.

Q.150   What is an E-R model? Draw an E-R Diagram for the company database with following Descriptions:
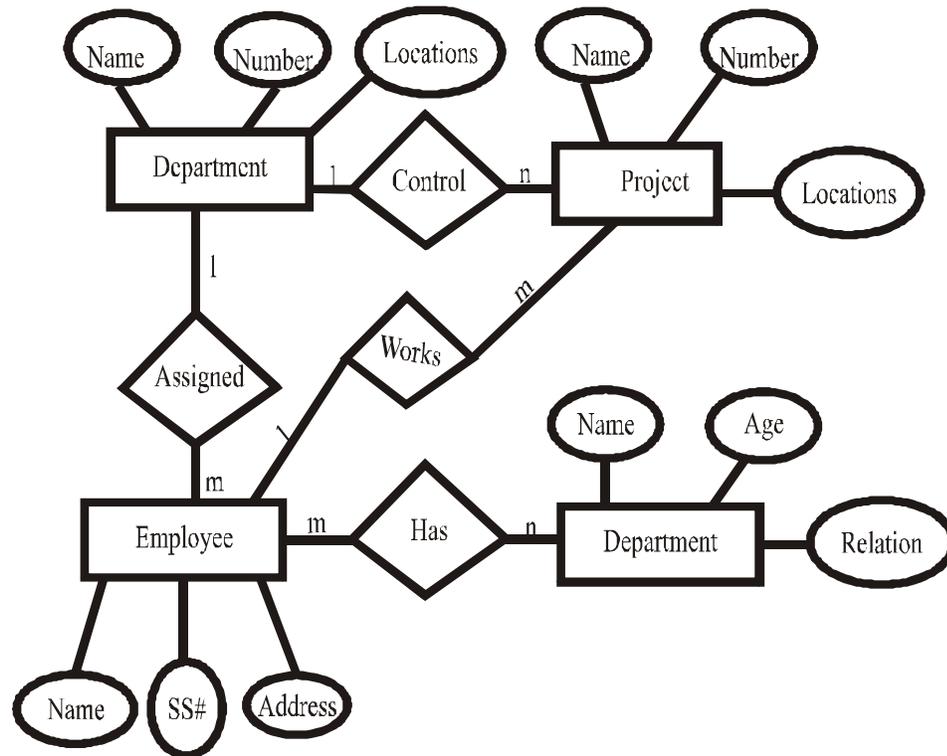The company is organized into departments. Each department has a unique name and a unique number with several locations.
A department controls a number of projects, each of which has a unique name, unique number and a single location.
We store each employees name, social security number, address, and salary. An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same departments.
We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's name, age and relationship to the employee.   **(8)**

Ans: An entity-relationship model (ERM) is an abstract conceptual representation of structured data. Entity-relationship modeling is a relational schema database modeling method, used in software engineering to produce a type of conceptual data model (or semantic data model) of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created using this process are called *entity-relationship diagrams*, or *ER diagrams* or *ERDs* for short.

Q.151 Define the following

    (i) Record-Based Logical Models
    (ii) Data Independence   **(8)**

    Ans:(i)   **Record-Based logical Models:** Also describe data at the conceptual and view levels. Unlike object-oriented models, are used to specify overall logical structure of the database, and provide a higher-level description of the implementation. Named so because the database is structured in fixed-format records of several types. Each record type defines a fixed number of fields, or attributes. Each field is usually of a fixed length (this simplifies the implementation). Record-based models do not include a mechanism for direct representation of code in the database. Separate languages associated with the model are used to express database queries and updates. The three most widely-accepted models are the relational, network, and hierarchical.

(**ii**)   **Data Independence:** Techniques that allow data to be changed without affecting the applications that process it. There are two kinds of data independence. The first type is data independence for data, which is accomplished in a database management system (DBMS). It allows the database to be structurally changed without affecting most existing programs. Programs access data in a DBMS by field and are concerned with only the data fields they use, not the format of the complete record. Thus, when the record layout is updated (fields added, deleted or changed in size), the only programs that must be changed are those that use those new fields.

Q.152   What is bucket overflow and how bucket overflow is handled by Over Flow Chaining or Closed Hashing?   **(8)**

Ans: Bucket: unit of storage containing records. Bucket is a disk block or contiguous block .Bucket contains multiple records .
            Hash Function: maps records to bucket numbers Function h from set of all search key values K to the set of all bucket addresses(numbers) B .Records with different search key values may be mapped same bucket. Entire bucket has to be searched to locate record.If bucket is full need *overflow buckets* and pointers
How to handle bucket overflow:
   • Buckets overflow is handled by using overflow buckets
   ○ *Closed Hashing (Chaining):* The overflow buckets of a given bucket are chained together in a linked list
   ○ *Open Hashing:* place in next available bucket (not suitable for database applications)
   ○ *Multiple Hashing :* use a second hash function

Q.153   What are the Constituents of File? Also explain all the possible file operations.   **(8)**

Ans: A file is organised logically as sequence of record. These records are mapped onto disk blocks. Although blocks are of fixed size determined by the physical properties of the disk and by the operating system, record sizes vary. In a relational database, tuples of distinct relations are generally of different sizes.
Possible operations on a file:
(i)   Create: a file can be created
(ii)   Delete: a file can be deleted
(iii)   Changing attributes of a file: The attributes of a file say read only, author's name etc. can be changed.
(iv)   Contents of the file can be altered.

Q.154   Express the following queries in SQL assumes that the data is stored in EMPLOYEE table with relevant fields.
(i)   Display name, job, salary, and hire date of employee who are hired between May 10, 1975 and December 20, 1980. Order the query in ascending order of hire date.
(ii)   Display name and hire date of employee who are employed after employee 'RAGHAV'.   **(6)**

Ans: (i) Select name, job, salary, hire date
from employee where hire date between
'May-10-1975' and 'Dec-20-1980'
order by hire date;
(ii)   Select name, hire date
        from employee
Where hire date> (select hire date from employee where name= 'Raghav');

Q.155   What are the various types of the update operations on relations? Also explain the constraints on these update operation. Give examples in support of your answer. **(10)**

Ans: There are three basic update separation on relations:
   (i) **Insert :** It is used to insert a new tuple or tuples in a relation. Insert can violate any of the four tuples of constraints : Domain constraints, key constraints , entity integrity and referential integrity
   (ii) **Delete :** It is used to delete tuples. The delete operation can violate only referential integrity, if the tuple being deleted is referenced by the foreign keys from other tuples in the database
   (iii) **Modify:** It is used to change value of some attributes in existing tuples. Modify can also violate any of the four constraints as specified in insert operation.
   Examples:
   (1) Inserting a tuple having null value for primary key violates entity integrity
   (2) Deleting a tuple in a table for which tuples exist other tables that are dependent on those tuples will violate referential key integrity.
   (3) Modifying a primary key attribute to any other such that is aheady exists in the table will violate entity integrity.

Q.156 Consider the relations

EMPLOYEE(emp#, name)
ASSIGNED_TO(project#, emp#)
PROJECT(project#, project_name, chief)
Express the following queries in Relational Algebra
(i) Get details of employee working on both comp354 and comp345 project numbers.
(ii) Find the employee number of employee who do not work on project comp678 **(8)**

Ans:(i) EMPLOYEE $\bowtie$ (ASSIGNED TO $\div \Pi_{project\ \#}$ ($\sigma_{(project\#='comp351'\ \vee\ project\#='comp\ 345}$
ASSIGNED _TO)))

(ii) $\Pi_{Emp\#}$(EMPLOYEE)-$\Pi_{Emp\#}$ $(_{project\#='c\ 678'}$ ASSIGNED_TO))

Q.157 Explain the SQL operators BETWEEN-AND, IN, LIKE and IS_NULL by taking suitable examples. **(8)**

Ans:

Comparison Operators	Description
LIKE	column value is similar to specified character(s). % and – are two special characters to match any substring and character respectively in a string. Eg, 'Perry%' matches any sting with Perry.
IN	column value is equal to any one of a specified set of values. Eg, SQL query to find all customers who are borrowers also from a bank as well as account holders in the bank is : Select cust# from borrower where cust# in(Select cust# in depositor)
BETWEEN...AND	column value is between two values, including the end values specified in the range. Eg, SQL query to find all employees hiredate between 1-JAN-2008 to 31-DEC-2008: Select * from employee where hiredate between 1-JAN-2008' and '31-DEC-2008'
IS NULL	column value does not exist. Eg, SQL statement to find all salesperson who are not earning any commission: Select * from employee where job= 'Sales' and commission is null.

Q.158   Write short note on followings:

(i)   Relational Constraints
(ii)   Disadvantages of Relational Approach
(iii)   Instances and Schemas

(4 × 4 = 16)

Ans: **(i)**   **Relational Constraints are:**
  ▪ NOT NULL
  ▪ Unique
  ▪ Primary key
  ▪ Foreign key
  ▪ Table check
(ii)   **Disadvantages of relational approach:**
  • Substantial hardware and system software overhead
  • May not fit all business models
  • Can facilitate poor design and implementation

87

• May promote "islands of information" problems
(iii) Instances and schemas: Databases changes over time as the information is inserted and deleted. The collection of information stored in database at a particular moment in time is called Instances and the overall design of database is called schemas.

Q.159 What are the key features of Oracle? **(8)**

Ans: Key features of Oracle are as follows:
• Read Consistency
• Concurrency
• Locking Mechanisms
• Quiesce Database
• RAC
• Portability

Q.160 Explain the following functions of Oracle with suitable examples:
(i) To_Char( ) (ii) Count( )
(iii) Trim( ) (iv) Length( )

(8)
Ans: (i) To_char( ): The TO_CHAR function converts a DATETIME, number, or NTEXT expression to a TEXT expression in a specified format. This function is typically used to format output data.
(ii) Count( ): The COUNT function returns the number of rows in a query.
The syntax for the COUNT function is:
SELECT COUNT (*expression*)
FROM tables
WHERE predicates;
(iii) Trim( ): In Oracle/PLSQL, the trim function removes all specified characters either from the beginning or the ending of a string.
The syntax for the trim function is:
trim([leading | trailing | both [trim_character]]   string1)
leading - remove *trim_string* from the front of *string1*.
trailing - remove *trim_string* from the end of *string1*.
both - remove *trim_string* from the front and end of *string1*.
(iv) Length( ): In Oracle/PLSQL, the length function returns the length of the specified string.
The syntax for the length function is:
length(string1)
string1 is the string to return the length for. If *string1* is NULL, then the function returns NULL.

Q.161 Explain the disadvantages of a file processing system.

Ans: Disadvantages of File Processing Systems include:
1) Data Redundancy
2) Data Inconsistency
3) Difficult to access data
4) Data isolation
5) Atomicity problem

6) Concurrent access anomalies
7) Security problem

Q.162   What is data model? Explain object based and record based data models.

      Ans: A data model is an abstract model that describes how data is represented and accessed.
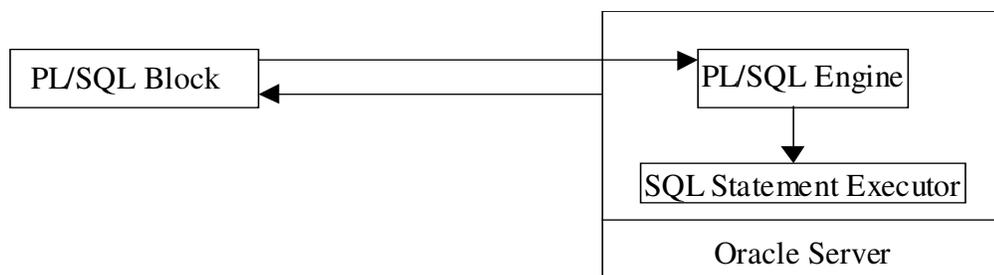**(i)**   **Object based data models:** Similar to a relational database model, but objects, classes and inheritance are directly supported in database schemas and in the query language
(ii)   **Record based data models:** is a database model based on first-order predicate logic. Its core idea is to describe a database as a collection of predicates over a finite set of predicate variables, describing constraints on the possible values and combinations of values.

Q.163   How is Oracle used in PL/SQL? Define the features of procedures and how they are defined.   **(7)**

      Ans:
PL/SQL – PL/SQL is Oracle's procedural language (PL) superset of the Structured Query Language (SQL). PL/SQL is block-structured language that enables developers to code procedures, functions and unnamed blocks that combine SQL with procedural statements. PL/SQL has its own processing engine that executes PL/SQL blocks and subprograms. If the host program does not have the PL/SQL engine, the blocks of code are sent to the Oracle server for processing. The following figure shows the PL/SQL engine as an integrated component of Oracle server. The engine executes procedure statements and passes SQL statements to the SQL statement executor in the Oracle server.

PL/SQL Engine

Procedures – Sophisticated business rules and application logic can be stored as procedures within Oracle. Stored procedures – groups of SQL and PL/SQL statements – allow you to move code that enforces business rules from the application to the database. As a result the code will be stored once for all applications to use. Procedures can be defined or created with the CREATE PROCEDURE command. The syntax in Oracle to define a procedure is:
CREATE [OR REPLACE] PROCEDURE [<user>.]<procedure_name>
   [(<arg1> [IN|OUT|IN OUT] <data type>
       [,<arg2> [IN|OUT|IN OUT] <data type>] ...)] {IS|AS}
       [<declaration>]
   BEGIN

```
<body>
[EXCEPTION
        WHEN <exception_name> THEN
                <error handling statements>
        [WHEN <exception_name> THEN
                <error handling statements>

        ...
        ...]
        [WHEN OTHERS THEN
                <error handling statements>]]
END;
```
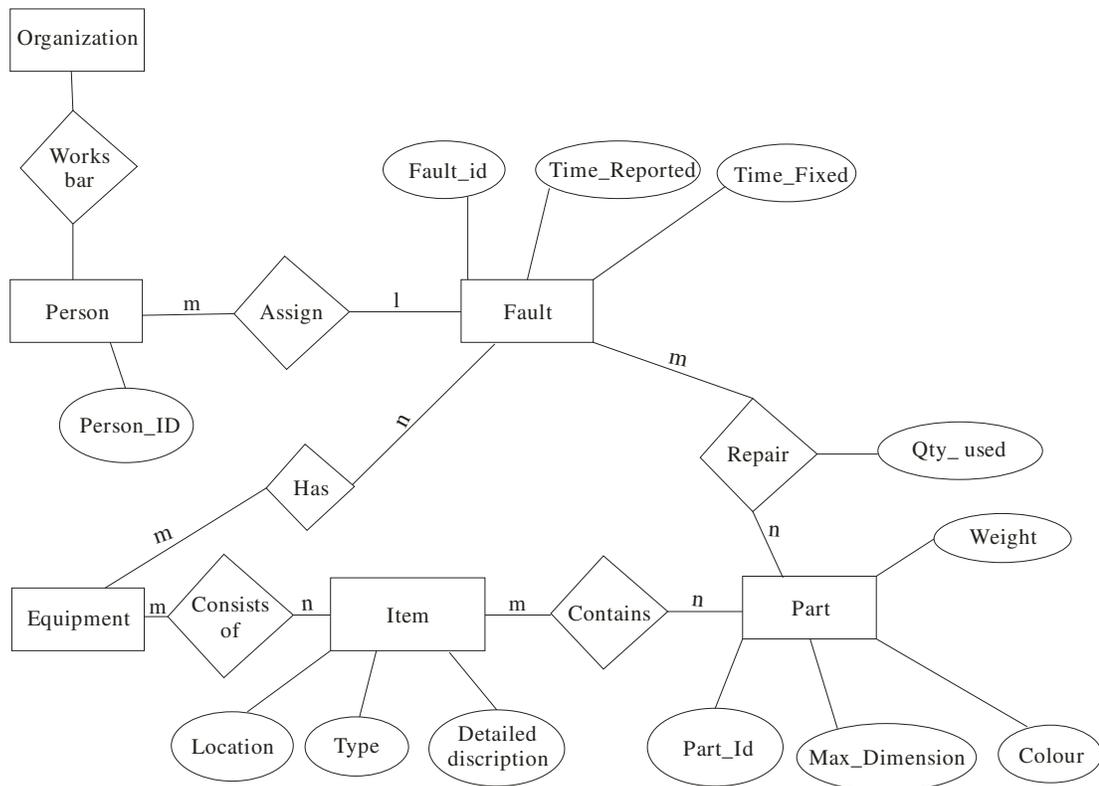
Q.164   Draw an E_R Model for the following:
An organization uses number of items of a equipment to produce goods. Each item is at one LOCATION, of one TYPE and has a DETAILED_DISCRIPTION. Faults on the equipment are identified by a unique FAULT_ID and are reported at a TIME_REPORTED. Any number of persons may be assigned to a fault and work on the fault until it is fixed. The TIME_FIXED is recorded as is the TIME_SPENT by each person on a fault. Any number of parts may be used to repair a fault. The QTY_USED of each parts is recorded against the fault. Each part is identified by a PART_ID and has a given weight and MAX_DIMENSION and can have any number of colors.

Ans:

Q.165   Explain the concept of generalization and aggregation in E_R diagrams. Give one example for each one of them.

Ans: Generalization: Consider extending the entity set account by classifying accounts as being either savings-account or chequing-account.
Each of these is described by the attributes of account **plus additional attributes.**
(savings has interest-rate and chequing has overdraft-amount.)
We can express the similarities between the entity sets by **generalization.** This is the process of forming containment relationships between a **higher-level** entity set and one or more **lower-level** entity sets.
Aggregation: The E-R model cannot express relationships among relationships.
When would we need such a thing?
Consider a DB with information about employees who work on a particular project and use a number of machines doing that work. We get the E-R diagram shown in Figure 1.
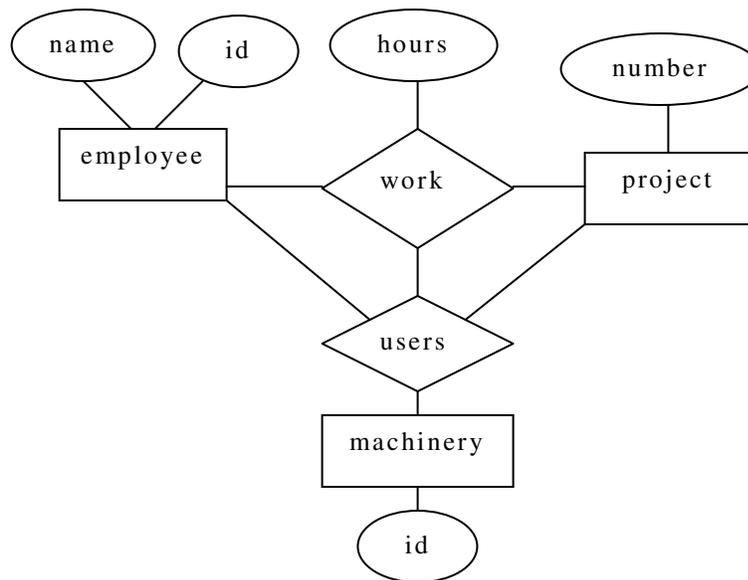

Figure 1: E-R diagram with redundant relationships

Relationship sets **work** and **uses** could be combined into a single set. However, **they** shouldn't be, as this would obscure the logical structure of this scheme.
The solution is to use aggregation.
- An abstraction through which relationships are treated as higher-level entities.
- For our example, we treat the relationship set **work** and the entity sets **employee** and **project** as a higher-level entity set called **work**.
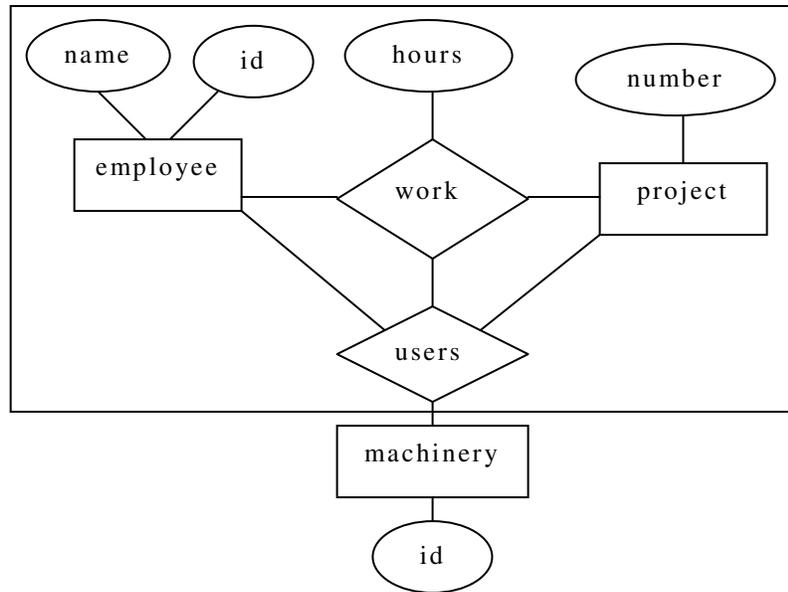- Figure 2 shows the E-R diagram with aggregation.

work

Figure 2: E-R diagram with aggregation