

The key is defined as the column or attribute of the database table. For example if a table has id, name and address as the column names then each one is known as the key for that table. We can also say that the table has 3 keys as id, name and address. **The keys are also used to identify each record in the database table .**

Primary Key:-

- Every database table should have one or more columns designated as the primary key . The value this key holds should be unique for each record in the database. For example, assume we have a table called **Employees** (SSN-social security No) that contains personnel information for every employee in our firm. We need to select an appropriate primary key that would uniquely identify each employee.

Primary Key

- The primary key must contain unique values, must never be null and uniquely identify each record in the table.
- As an example, a student id might be a primary key in a student table, a department code in a table of all departments in an organisation.

Primary Keys



<u>StudentId</u>	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

Unique Key

- The UNIQUE constraint uniquely identifies each record in a database table.
- **Allows Null value.** But only one Null value.
- A table can have more than one UNIQUE Key Column[s]
- **A table can have multiple unique keys**

Differences between Primary Key and Unique Key:

- **Primary Key**

1. A primary key *cannot* allow null (a primary key cannot be defined on columns that allow nulls).
2. Each table can have only one primary key.

- **Unique Key**

1. A unique key can allow null (a unique key can be defined on columns that allow nulls.)
2. Each table can have multiple unique keys.

Secondary Key or Alternate Key

- A table may have one or more choices for the primary key. Collectively these are known as candidate keys. One is selected as the primary key. Those not selected are known as secondary keys or alternative keys.
- Table with **student id , First Name and last name**
- we identified two candidate keys, studentId and **firstName + lastName**. The studentId would be the most appropriate for a primary key leaving the other candidate key as secondary or alternative key. It should be noted for the other key to be candidate keys, we are assuming you will never have a person with the same first and last name combination. As this is unlikely we might consider fistName+lastName to be a suspect candidate key as it would be restrictive of the data you might enter.

Super key

- A Super key is **any combination of fields within a table that uniquely** identifies each record within that table.
- A superkey is a combination of columns that uniquely identifies any row within a relational database management system (RDBMS) table

Example of Super key

- Suppose we have a table that holds all the managers in a company, and that table is called Managers. The table has columns called **ManagerID, Name, Title, and DepartmentID**. Every manager has his/her own ManagerID, so that value is always unique in each and every row

To be continued:-

- This means that if we combine the ManagerID column value for any given row with any other column value, then **we will have a unique set of values**. So, for the combinations of (ManagerID, Name), (ManagerID, Title), (ManagerID, DepartmentID), (ManagerID, Name, DepartmentID), etc – there will be no two rows in the table that share the exact same combination of values, because the ManagerID will always be unique and different for each row. This means that pairing the Manager ID with any other column(s) will ensure that the combination will also be unique across all rows in the table.

Now the Concept of super key

- And that is exactly what defines a superkey – **it's any combination of column(s) for which that combination of values will be unique across all rows in a table.** So, all of those combinations of columns in the Manager table that we gave earlier would be considered to be superkeys. Even the ManagerID column is considered to be a superkey, although a special type of superkey as you can read more about below.

Example of Super key

For example, imagine a table used to store customer master details that contains columns such as:

- Customer name
- Customer ID
- Social security number (SSN)
- Address
- Date of birth
- A certain set of columns may be extracted and guaranteed unique to each customer. Examples of superkeys are as follows:
- Name+SSN+Birthdate
- ID+Name+SSN
- However, this process may be further reduced. It can be assumed that each customer ID is unique to each customer. So, the superkey may be reduced to just one field, customer ID, which is the candidate key.

Candidate key

- **A candidate is a subset of a super key. A candidate key is a single field or the least combination of fields that uniquely identifies each record in the table.** The least combination of fields distinguishes a candidate key from a super key. Every table must have at least one candidate key but at the same time can have several.

Candidate key=Minimal Super Key

- for example if ABC is a super key with three attributes A, B C and if A and B alone are sufficient to determine the rows in the table , then AB will be the candidate key

Super Key and Candidate Key

- Informally, a superkey is a set of attributes within a table whose values can be used to uniquely identify a tuple. A [candidate key](#) is a minimal set of attributes necessary to identify a tuple; this is also called a minimal superkey. For example, given an employee schema consisting of the attributes employeeID, name, job, and departmentID, we could use the employeeID in combination with any or all other attributes of this table to uniquely identify a tuple in the table. Examples of superkeys in this schema would be {employeeID, Name}, {employeeID, Name, job}, and {employeeID, Name, job, departmentID}.

Candidate Key Condition:-

- In order to be eligible for a candidate key it must pass certain criteria:-
- It must contain unique values
- It must not contain null values
- It contains the minimum number of fields to ensure uniqueness
- It must uniquely identify each record in the table

Candidate Keys

StudentId	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

- Candidate Key – A Candidate Key can be any column or a combination of columns that can **qualify as unique key in database**. There can be multiple Candidate Keys in one table. Each Candidate Key can qualify as Primary Key.
- **Candidate key** -> Any column or set of columns that are **candidate to become primary key** are Candidate key. *(There can be **one or more candidate key(s)** in the table,*
- Primary Key – A Primary Key is a column or a combination of columns that **uniquely identify a record**. *There can be **only one Primary key** in the table*
- Only one Candidate Key can be Primary Key.

- An example of Primary key and Candidate key can be ID and SSN number in a Employee table, Since both can identify each employee uniquely they are candidate key and any one can become primary key. Now if you have to choose between them as primary key, I will go ID as primary key because SSN is sensitive information and may not be allow/not safe to use as String in queries as frequently as ID. Second reason of choosing ID over SSN as primary key can be use of ID as primary tracking ID within organization and its frequent use all over the place. Once you choose a primary key, All candidate key are like unique keys.

Foreign Key

- A foreign key is generally a primary key from one table that appears as a field in another where the first table has a relationship to the second. In other words, if we had a table A with a primary key X that linked to a table B where X was a field in B, then X would be a foreign key in B.
- An example might be a **student table** that contains the `course_id` the student is attending. Another table lists the **courses** on offer with `course_id` being the primary key. **The 2 tables are linked through `course_id` and as such `course_id` would be a foreign key in the student table.**

Foreign Key

- A foreign key is a key used to link two tables together. This is sometimes called a referencing key.
- Foreign Key is a column or a combination of columns whose values match a Primary Key in a different table.
- **The relationship between 2 tables matches the Primary Key in one of the tables with a Foreign Key in the second table.**

Foreign Key

- In simpler words, the foreign key is defined in a second table, but it refers to the primary key in the first table. For example, a table called Employee has a primary key called **employee_id**. Another table called Employee Details has a foreign key which references **employee_id** in order to uniquely identify the relationship between both the tables.
- The table containing the foreign key is called the **referencing** or **child table**,

<u>studentId</u>	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042

Foreign Keys

Relationship

Primary Keys

<u>courseId</u>	courseName
A004	Accounts
C002	Computing
P301	History
S042	Short Course