KEYS

- It is an **attribute or set of attributes** which helps you to **identify a row in a table**.
- They allow you to find the relation between two tables.
- Keys help you **uniquely identify a row in a table** by a combination of one or more columns in that table.
- Key is also helpful for finding unique record from the table.

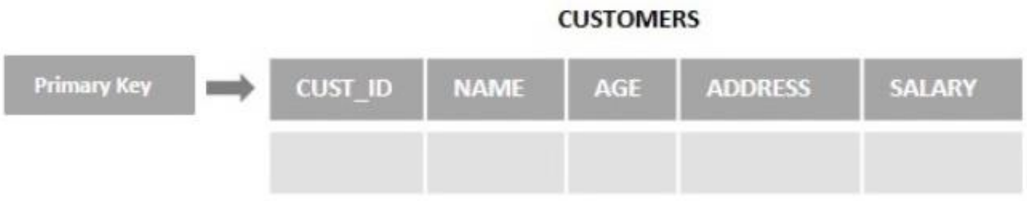
Types of Keys

- Primary Key
- Candidate Key
- Alternate Key
- Super Key
- Foreign Key
- Composite Key

Primary Key

- The Primary Key is an attribute or set of attributes in a table that uniquely identify every row in that table.
- The value of the primary key should be NOT NULL.
- A single table can have only one primary key.
- Every primary key is the candidate key.

Example :


Primary Key

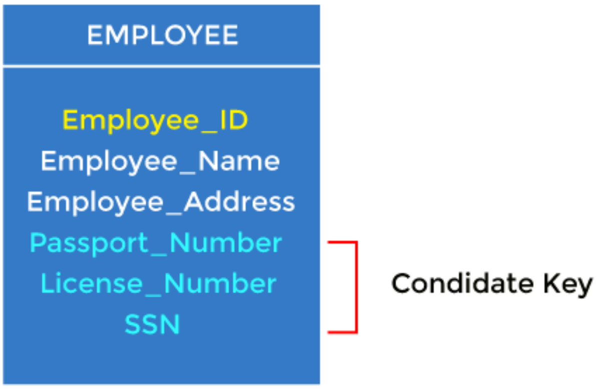
CUSTOMERS

CUST_ID	NAME	AGE	ADDRESS	SALARY

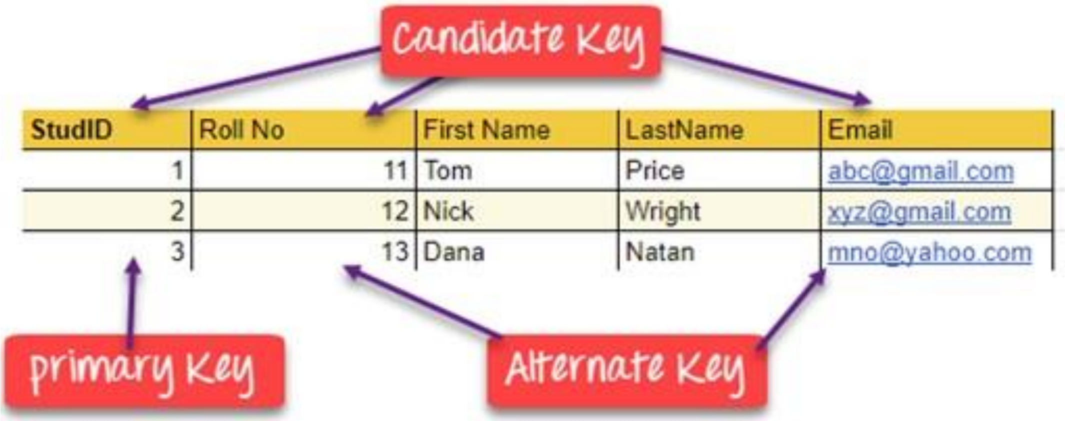
Candidate Key

- A candidate key is an **attribute or set of attributes** that can uniquely identify a tuple.
- A single table can have **multiple candidate keys but only a single primary key** and **Unique values** must be present in all columns. The **Primary key should be selected from the candidate keys**.
- Except for the primary key, the remaining attributes are considered a candidate key.
- The candidate keys are as strong as the primary key.
- Candidate Key is a super key with no repeated attributes.
- **Every table must have at least a single candidate key.**
- Candidate key Must not contain null values.

Example 1 : SSN, PAN No, Account No, Passport No. in the EMPLOYEE table, id is best suited for the primary key. The rest of the attributes, like SSN, Passport_Number, License_Number, etc., are considered a candidate key.



Example 2 : Stud ID, Roll No, and email are candidate keys which help us to uniquely identify the student record in the table.


Candidate Key
primary Key
Alternate Key

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

Alternate key

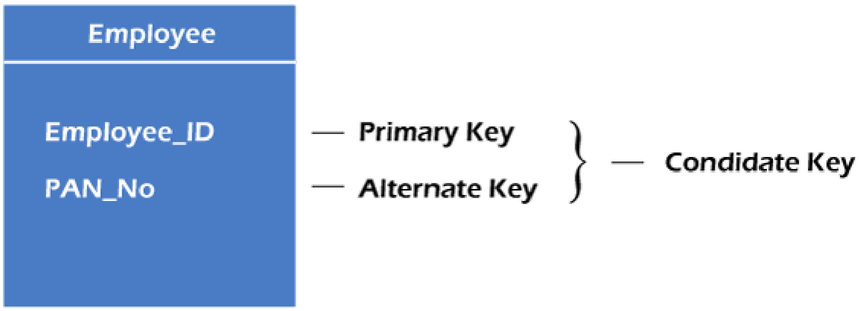
There may be one or more attributes or a combination of attributes that uniquely identify each tuple in a relation. These attributes or combinations of the attributes are called the **candidate keys**.

One key is chosen as the primary key from these candidate keys, and the remaining candidate key, if it exists, is termed the Alternate key.

Total no. of alternate key = Total no. of candidate keys - primary key

The alternate key may or may not exist. If there is only one candidate key in a relation, it does not have an alternate key.

Example: Employee relation has two attributes, Employee_Id and PAN_No, that act as candidate keys. In this relation, Employee_Id is chosen as the primary key, so the other candidate key, PAN_No, acts as the Alternate key.

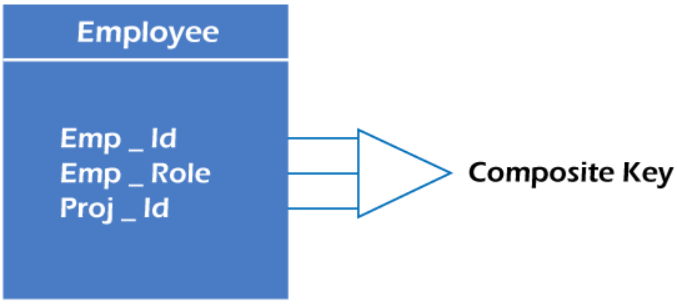


Composite Key

A composite key is a **combination of multiple columns** that uniquely identify a tuple in a table. Sometimes single attribute fails to uniquely identify the tuples in a table. However, when taken all together, they ensure uniqueness.

Whenever a primary key consists of more than one attribute, it is known as a composite key. This key is also known as Concatenated Key.

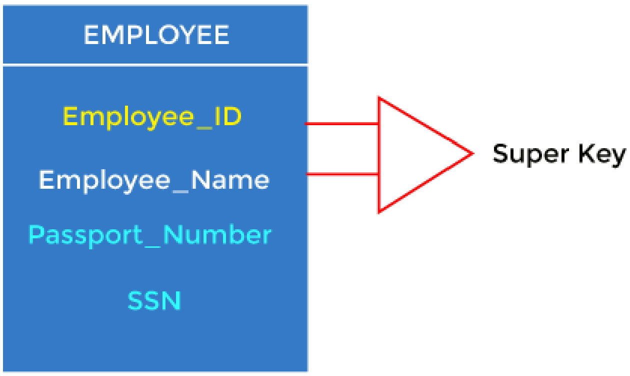
Example:



Super Key

- Super Key is a single attribute or combination of attributes that can be used **to uniquely identify a row** in a table.
- A single table can have multiple super keys.
- A **candidate key and primary key** can be a super key
- A super key is a superset of a candidate key.

Example 1 : In the Employee table, the combination (Employee_ID, Employee_Name) is a Super Key. The name of two employees can be the same, but their Employee_ID can't be the same.



Example 2 :

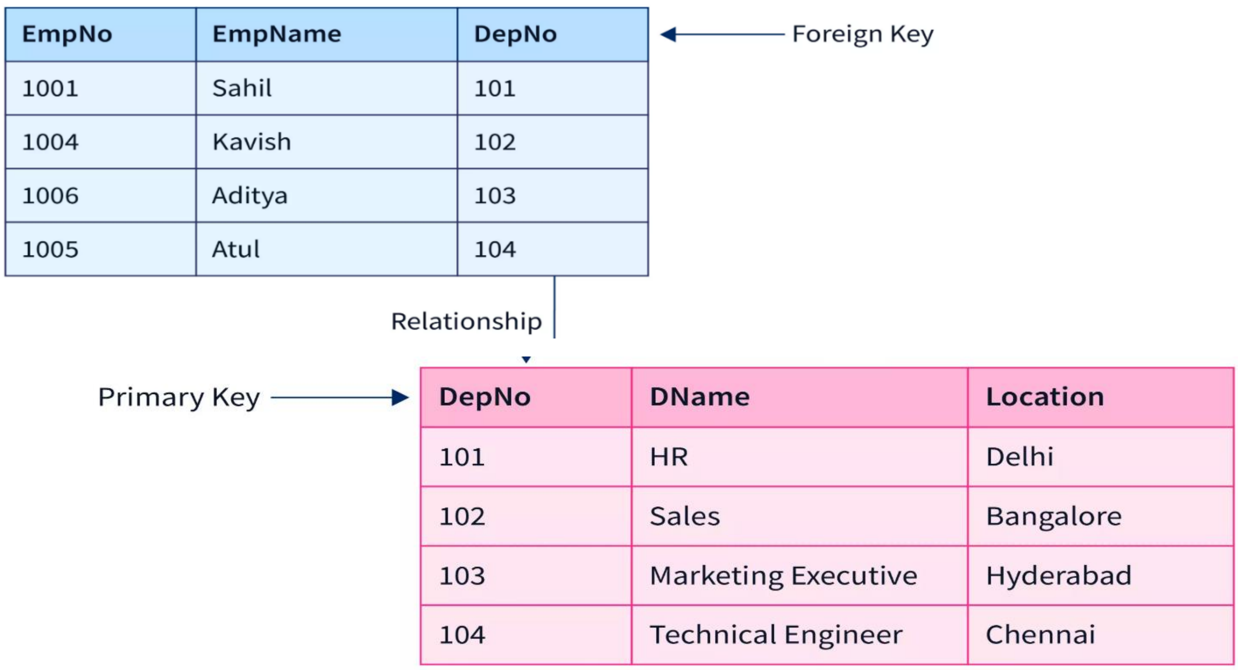
In the above Order Table, we have chosen the Order ID, Customer ID, and Customer Contact ID to uniquely identify tuples.

So, the super key set can be as follows:

- [Order ID]
- [Customer ID]
- [Customer Contact ID]
- [Order ID, Customer ID]
- [Customer Contact ID, Order ID]
- [Customer ID, Customer Contact ID]
- [Order ID, Customer ID, Customer Contact ID]

FOREIGN KEY

- To create a relationship between two tables, we need two keys, one is the primary key, and the other is the foreign key.
- A foreign key is a field or collection of fields in a table whose values are referenced from a primary key in another table.
- The table containing the primary key is known as the **parent table**, whereas the table which contains the foreign key is known as the **child table**.
- A primary key in SQL uniquely identifies records in a table. However, you must have a primary key for using a foreign key in SQL.
- Primary key values in the parent table must **match the values** of the foreign key in another table.
- Foreign Key is a column that creates a relationship between two tables.
- The purpose of Foreign keys is to maintain data integrity and allow navigation between two different instances of an entity.
- The relationship between two tables is known as referential integrity.
- A table may have multiple foreign keys.
- A foreign key can have NULL values.
- You can have duplicate values in foreign keys.
- The parent table records can be deleted if no child table record exists. Parent Table cannot be updated if its child table exists.



EmpNo	EmpName	DepNo
1001	Sahil	101
1004	Kavish	102
1006	Aditya	103
1005	Atul	104

DepNo	DName	Location
101	HR	Delhi
102	Sales	Bangalore
103	Marketing Executive	Hyderabad
104	Technical Engineer	Chennai

Foreign Key
Relationship
Primary Key

Example :

```
CREATE TABLE Department
(DeptNo int PRIMARY KEY,
 DName varchar(266),
 Location varchar(266));


CREATE TABLE Employee
(EmpNo int,
 EmpName varchar(266),
 Salary int,
 DeptNo int,
FOREIGN KEY (DeptNo) REFERENCES Department (DeptNo));
```

DeptNo in the **Employee** table will act as a **foreign key** in this new table. Using the above query, we have linked these two tables based on a common column **DeptNo**.

ADD a foreign key constraint :

```
ALTER TABLE Employee
ADD FOREIGN KEY (DeptNo) REFERENCES Department(DeptNo);
```

DROP a foreign key constraint :

```
ALTER TABLE Employee
DROP FOREIGN KEY DeptNo;
```