**Keys**

Table of Contents

[Keys 2](#_Toc115660467)

[Why we need a Key? 2](#_Toc115660468)

[Types of Keys 2](#_Toc115660469)

[Primary Key 3](#_Toc115660470)

[Candidate key 4](#_Toc115660471)

[Super Key 4](#_Toc115660472)

[Foreign key 4](#_Toc115660473)

# Keys

* Keys play an important role in a relational database.
* They are used to uniquely identify any record or row of data from the table.
* They are also used to establish and identify relationships between tables.

**For example:**

**Example 1:** In Student table, \_id is used as a key because it is **unique** for each student.

**Example 2:** In Person table, passport\_number, license\_number, aadhar\_no are keys since they are **unique** for each person.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Student | | \*\_id | | name | | address | | course | | fees | | |  | | --- | | Person | | \*\_id | | name | | \*passport\_number | | \*licence\_number | | \*aadhar\_no | |

## Why we need a Key?

Here are some reasons for using SQL keys in the DBMS system.

1. Keys identify any row of data in a table.

In a real-world application, a table could contain thousands of records. Moreover, the records could be duplicated. Keys ensure that a record can be uniquely identified in a table despite these challenges.

1. Keys allows us to establish a relationship between and identify the relation between tables
2. Keys enforce identity and integrity in the relationship.

## Types of Keys

There are mainly eight different types of Keys in DBMS and each key has its different functionality:

1. **Primary Key:** is a column or group of columns in a table that uniquely identify every row.
2. **Super Key:** A super key is a group of single or multiple keys which identifies rows in a table.
3. **Candidate Key:** is a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes.
4. **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.
5. **Composite Key:** An artificial key which aims to uniquely identify each record is called a surrogate key. These kinds of keys are unique because they are created when you don't have any natural primary key.
6. **Alternate Key:** is a column or group of columns in a table that uniquely identify every row.
7. **Compound Key:** has two or more attributes that allow you to uniquely recognize a specific record. It is possible that each column may not be unique by itself within the database.
8. **Surrogate Key:** An artificial key which aims to uniquely identify each record is called a surrogate key. These kinds of key are unique because they are created when you don't have any natural primary key.

# Primary Key

* It is the first key which is used to identify one and only one instance of an entity uniquely. An entity can contain multiple keys as we saw in PERSON table. The key which is most suitable from those lists become a primary key.
* In the Person table, \_id can be primary key since it is unique for each person. However we can also select license\_number, passport\_number or aadhar\_no as primary key since they are also unique.
* For each entity, selection of the primary key is based on requirement and choice of developers.
* A table can have only 1 primary key

|  |
| --- |
| Person |
| \*\_id |
| fname |
| lname |
| \*passport\_number |
| \*licence\_number |
| \*aadhar\_no |

# Candidate key

* A candidate key is an attribute or set of an attribute which can uniquely identify a record.
* The remaining attributes except for primary key are considered as a candidate key.
* The candidate keys are as **strong** as the primary key.

**For example:** In the Person table, **\_id** is best suited for the primary key. Rest of the attributes like **passport\_number, license\_number and aadhar\_no** are considered as a candidate key.

# Super Key

* Super key is a set of an attribute which can uniquely identify a tuple.
* Super key is a superset of candidate keys.

**For example:** In Person table, for (\_id, name) the name of two persons can be the same, but their \_id can't be the same. Hence, this **combination** can also be a key.

The super key would be \_id, (\_id, name), etc.

Other super keys:

1. \_id
2. \_id, fname
3. \_id, lname
4. \_id, fname, lname
5. \_id, passport\_no
6. passport\_no
7. passport\_no, fname
8. passport\_no, lname
9. passport\_no, fname, lname
10. driving\_licence
11. aadhar\_no

# Foreign key

* Foreign key is the column of the table which is used to **point to** the primary key of **another table**.
* In a company, every employee works in a specific department, and employee and department are two different entities. So we can't store the information of the department in the employee table. That's why we **link** these **two tables** through the primary key of one table.
* We add the primary key of the Department table, \_id as a new attribute in the Employee table.
* Now in the Employee table, **dept\_id** is the **foreign key**, and both the tables are **related**.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Employee | | \*\_id | | Name | | passport\_number | | aadhar\_no | | dept\_id | | |  | | --- | | Department | | \*\_id | | name | |