

# Database Keys

## Table of Contents

- [Keys](#)
- [Primary Key](#)
- [Candidate Key](#)
- [Foreign key](#)
- [Compound key](#)
- [Composite key](#)
- [Surrogate Key](#)

## Keys

Any attribute in the table which uniquely identifies each record in the table is called key. It can be a single attribute or a combination of attributes. For example, in STUDENT table, STUDENT\_ID is a key, since it is unique for each student. In PERSON table, his passport number, driving license number, phone number, SSN, email address is keys since they are unique for each person.

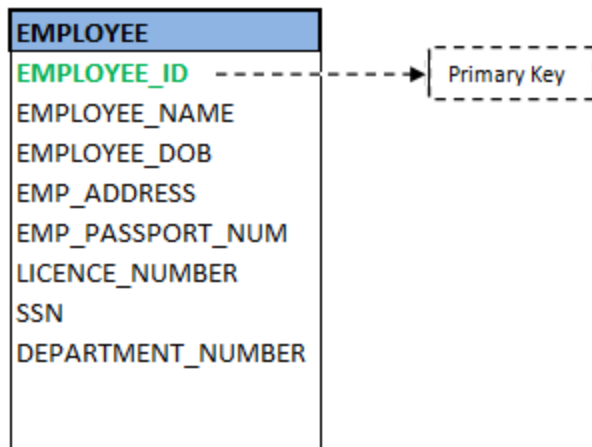
STUDENT
STUDENT_ID
STUDENT_NAME
ADDRESS
DOB
COURSE

PERSON
EMPLOYEE_NAME
EMPLOYEE_DOB
EMP_ADDRESS
EMP_PASSPORT_NUM
LICENCE_NUMBER
SSN

## Primary Key

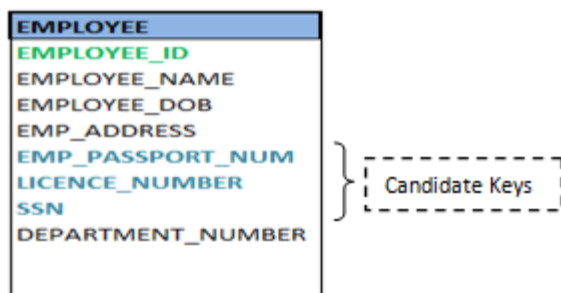
It is the first and foremost key which is used to uniquely identify a record. It can be a single attribute or a combination of attributes. For an entity, there could be multiple keys as we saw in PERSON table. Most suitable key from those lists becomes a primary key. In the Person table above, we can select SSN as primary key, since it is unique for each person. We can even select Passport Number or license number as primary key as they are also unique for a person. However, selection of primary key for each entity is based on requirement and developer.

For a student, STUDENT\_ID is a primary key and for an employee EMPLOYEE\_ID is a primary key.



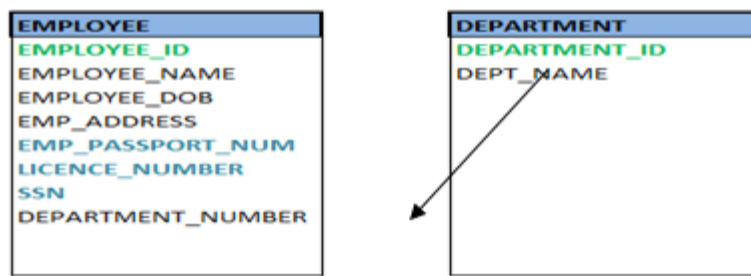
## Candidate Key

As we discussed above, an employee is identified by his ID in his office. Apart from his ID, does he have any other unique keys, so that he can be identified from others? Yes, he has passport number, PAN number, SSN number (if applicable), driving license number, email address etc. These also identify specific person uniquely. But we can choose any one of these unique attributes as primary key in the table. Rest of the attributes, which hold as strong as primary key are considered as Candidate key/secondary key. In our example of employee table, **EMPLOYEE\_ID** is best suited for primary key as it's from his own employer. Rest of the attributes like passport number, SSN, license Number etc are considered as candidate key.



## Foreign key

In a company there would be different departments – Accounting, Human Resource (HR), development, Quality, etc. An employee, who works for that company, works in specific department. But we know that employee and department are two different entities. So we cannot store his department information in employee table. Instead what we do is we link these two [tables](#) by means of primary key of one of the table i.e.; in this case, we pick the primary key of department table – **DEPARTMENT\_ID** and add it as a new attribute/column in the Employee table. Now **DEPARTMENT\_ID** is a foreign key for Employee table, and both the tables are related!



**Note:** – Names of the attribute in both the tables can be different. It's all when we really create the table via script matters!

## Compound key

A key in a table is formed by combining more than one attributes/columns of the same table. These columns of the table **can or cannot be keys** in the table. The compound key acts as a primary key only when all the columns in the compound keys are together, individually those columns are not keys. In other words, unique record from the table is fetched only if we combine more than one column. If we use them individually, we will not get any unique record.

In the example of M: N relationship – ‘Student enrolls for a course’, STUDENT\_ID and COURSE\_ID, when combined together gives the particular course to which he is enrolled for. Only STUDENT\_ID or COURSE\_ID alone does not inform correct data.

In the table above, STUDENT\_ID, 100 alone gives us multiple courses. To know about particular course we need both STUDENT\_ID and COURSE\_ID. In this case, both the IDs are primary keys from their table, but in STUDENT\_COURSE table, they form primary key when they are combined together. Hence they are compound key.

STUDENT_COURSE		
STUDENT_ID	COURSE_ID	No_of_Hours
100	401	20
100	402	15
101	401	25
103	403	10

## Composite key

Composite key is similar to compound key, but the columns which are part of composite keys are **always keys** in that table.

## **Surrogate Key**

Surrogate key is a kind of primary key, but it is not defined by the designer. It is a system generated random number, which uniquely identifies the entity in the system and not available for the user.