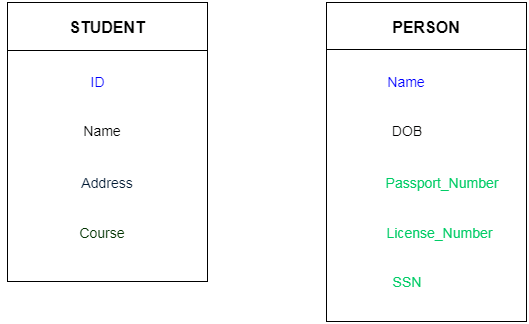
**Keys**

* Keys play an important role in the relational database.
* It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.
* Keys are very important part of Relational database model. They are used to establish and identify relationships between tables and also to uniquely identify any record or row of data inside a table.
* A Key can be a single attribute or a group of attributes, where the combination may act as a key.

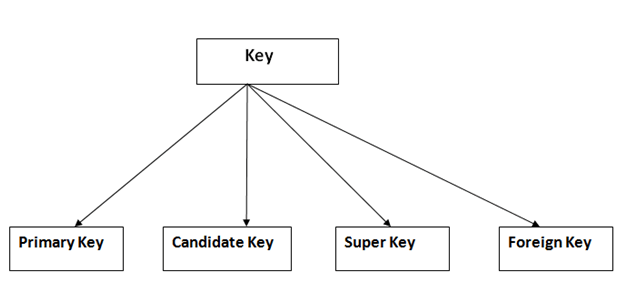
**For example:** In Student table, ID is used as a key because it is unique for each student. In PERSON table, passport\_number, license\_number, SSN are keys since they are unique for each person.



* **Why we need a Key?**

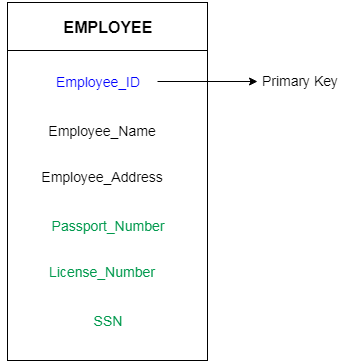
Here are some reasons for using SQL key in the DBMS system.

* Keys help you to 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 you can uniquely identify a table record despite these challenges.
* Allows you to establish a relationship between and identify the relation between tables
* Help you to enforce identity and integrity in the relationship.
* **Types of key:**



**1. 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 EMPLOYEE table, ID can be primary key since it is unique for each employee. In the EMPLOYEE table, we can even select License\_Number and Passport\_Number as primary key since they are also unique.
* For each entity, selection of the primary key is based on requirement and developers.



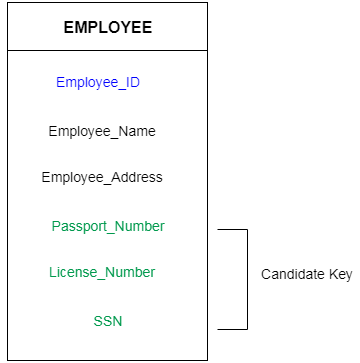
**Rules for defining Primary key:**

* Two rows can't have the same primary key value
* It must for every row to have a primary key value.
* The primary key field cannot be null.
* The value in a primary key column can never be modified or updated if any foreign key refers to that primary key.

**2. Candidate Key**

* A candidate key is an attribute or set of an attribute which can uniquely identify a tuple.
* The remaining attributes except for primary key are considered as a candidate key. The candidate keys are as strong as the primary key.
* Candidate Key is a super key with no repeated attributes.
* The Primary key should be selected from the candidate keys.
* Every table must have at least a single candidate key. A table can have multiple candidate keys but only a single primary key.

**For example:** In the EMPLOYEE table, id is best suited for the primary key. Rest of the attributes like SSN, Passport\_Number, and License\_Number, etc. are considered as a candidate key.

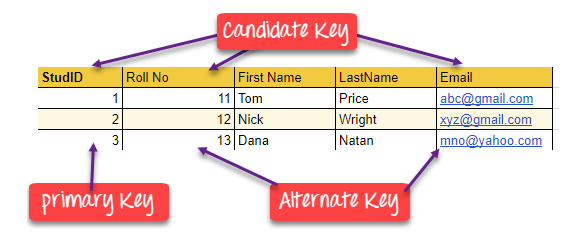


**Properties of Candidate key:**

* It must contain unique values
* Candidate key may have multiple attributes
* Must not contain null values
* It should contain minimum fields to ensure uniqueness
* Uniquely identify each record in a table

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **StudID** | **Roll No** | **First Name** | **LastName** | **Email** |
| 1 | 11 | Tom | Price | [abc@gmail.com](mailto:abc@gmail.com) |
| 2 | 12 | Nick | Wright | [xyz@gmail.com](mailto:xyz@gmail.com) |
| 3 | 13 | Dana | Natan | [mno@yahoo.com](mailto:mno@yahoo.com) |



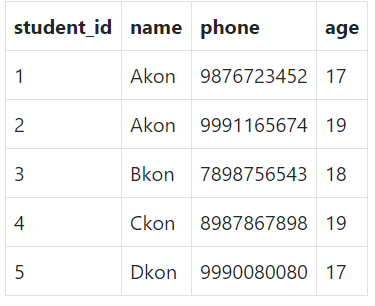
**3. Super Key**

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

**For example:** In the above EMPLOYEE table, for (EMPLOEE\_ID, EMPLOYEE\_NAME) the name of two employees can be the same, but their EMPLYEE\_ID can't be the same. Hence, this combination can also be a key.

The super key would be EMPLOYEE-ID, (EMPLOYEE\_ID, EMPLOYEE-NAME), etc.

**For example:**



In the table defined above super key would include **student\_id, (student\_id, name), phone** etc.

Confused? The first one is pretty simple as student\_id is unique for every row of data; hence it can be used to identity each row uniquely.

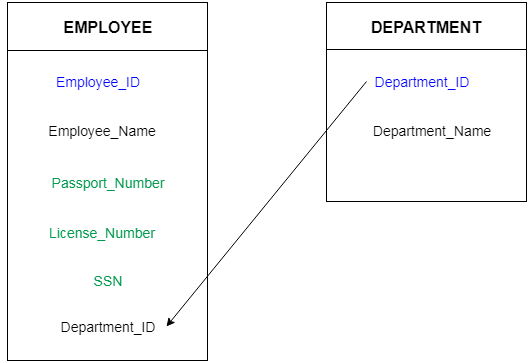
Next comes, (student\_id, name), now name of two students can be same, but their student\_id can't be same hence this combination can also be a key.

Similarly, phone number for every student will be unique, hence again, phone can also be a key.

So, they all are super keys.

**4. Foreign Key**

* Foreign keys are 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, Department\_Id as a new attribute in the EMPLOYEE table.
* Now in the EMPLOYEE table, Department\_Id is the foreign key, and both the tables are related.



|  |  |
| --- | --- |
| **DeptCode** | **DeptName** |
| 001 | Science |
| 002 | English |
| 005 | Computer |

**Example:**

|  |  |  |
| --- | --- | --- |
| **Teacher ID** | **Fname** | **Lname** |
| B002 | David | Warner |
| B017 | Sara | Joseph |
| B009 | Mike | Brunton |

We have two table, teacher and department in a school. However, there is no way to see which teacher work in which department.

In this table, adding the foreign key in Deptcode to the Teacher name, we can create a relationship between the two tables.

|  |  |  |  |
| --- | --- | --- | --- |
| **Teacher ID** | **DeptCode** | **Fname** | **Lname** |
| B002 | 002 | David | Warner |
| B017 | 002 | Sara | Joseph |
| B009 | 001 | Mike | Brunton |

This concept is also known as Referential Integrity.

**5. Alternate Key**

ALTERNATE KEYS is a column or group of columns in a table that uniquely identify every row in that table. A table can have multiple choices for a primary key but only one can be set as the primary key. All the keys which are not primary key are called an Alternate Key.

Example:

In this table, StudID, Roll No, Email is qualified to become a primary key. But since StudID is the primary key, Roll No, Email becomes the alternative key.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **StudID** | **Roll No** | **First Name** | **LastName** | **Email** |
| 1 | 11 | Tom | Price | [abc@gmail.com](mailto:abc@gmail.com) |
| 2 | 12 | Nick | Wright | [xyz@gmail.com](mailto:xyz@gmail.com) |
| 3 | 13 | Dana | Natan | [mno@yahoo.com](mailto:mno@yahoo.com) |

**6. Compound Key**

**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. However, when combined with the other column or columns the combination of composite keys become unique. The purpose of the compound key in database is to uniquely identify each record in the table.

**Example:**

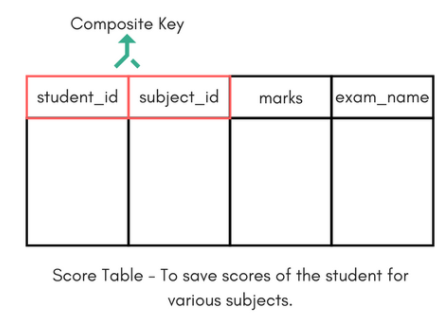
|  |  |  |  |
| --- | --- | --- | --- |
| **OrderNo** | **PorductID** | **Product Name** | **Quantity** |
| B005 | JAP102459 | Mouse | 5 |
| B005 | DKT321573 | USB | 10 |
| B005 | OMG446789 | LCD Monitor | 20 |
| B004 | DKT321573 | USB | 15 |
| B002 | OMG446789 | Laser Printer | 3 |

In this example, OrderNo and ProductID can't be a primary key as it does not uniquely identify a record. However, a compound key of Order ID and Product ID could be used as it uniquely identified each record.

**7. Composite Key**

**COMPOSITE KEY** is a combination of two or more columns that uniquely identify rows in a table. The combination of columns guarantees uniqueness, though individually uniqueness is not guaranteed. Hence, they are combined to uniquely identify records in a table.

The difference between compound and the composite key is that any part of the compound key can be a foreign key, but the composite key may or maybe not a part of the foreign key.



In the above picture we have a Score table which stores the marks scored by a student in a particular subject.

In this table student\_id and subject\_id together will form the primary key, hence it is a composite key.

**8. Surrogate Key**

**SURROGATE KEYS** is an artificial key which aims to uniquely identify each record. This kind of partial key in DBMS is unique because it is created when you don't have any natural primary key. They do not lend any meaning to the data in the table. Surrogate key is usually an integer. A surrogate key is a value generated right before the record is inserted into a table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fname** | **Lastname** | **Start Time** | **End Time** |  |
| Anne | Smith | 09:00 | 18:00 |  |
| Jack | Francis | 08:00 | 17:00 |  |
| Anna | McLean | 11:00 | 20:00 |  |
| Shown | Willam | 14:00 | 23:00 |  |

Above, given example, shown shift timings of the different employee. In this example, a surrogate key is needed to uniquely identify each employee.

**Features of the surrogate key:**

* It is automatically generated by the system.
* It holds anonymous integer.
* It contains unique value for all records of the table.
* The value can never be modified by the user or application.
* Surrogate key is called the factless key as it is added just for our ease of identification of unique values and contains no relevant fact (or information) that is useful for the table.

Surrogate keys in SQL are allowed when

* No property has the parameter of the primary key.
* In the table when the primary key is too big or complicated.

Assume you need to maintain track of good data in a table. The data includes information on each well's geographic location and depth, as indicated in the Well table below:

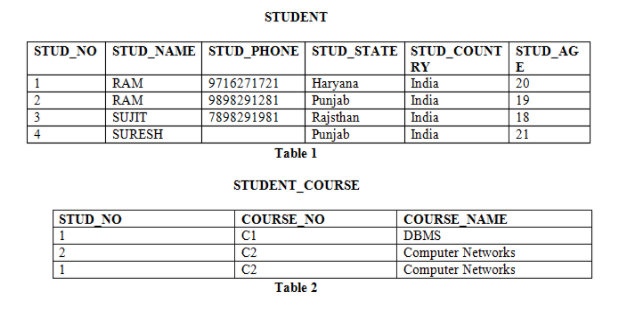
| **Longitude** | **Latitude** | **Depth** |
| --- | --- | --- |
| 220 | 140 | 5.6 |
| 220 | 160 | 5.6 |
| 220 | 170 | 7.5 |
| 340 | 170 | 8.2 |
| 340 | 510 | 9.4 |

The table above shows two wells with the same longitude, latitude, or depth. As a result, you can't select the primary key from one of these three columns because they don't uniquely identify the row.

You'll need to create a SURROGATE key column, which can be a unique auto-number column. Here's an example of an auto-number SURROGATE field called WellId in the table:

| **WellId** | **Longitude** | **Latitude** | **Depth** |
| --- | --- | --- | --- |
| 1 | 220 | 140 | 5.6 |
| 2 | 220 | 160 | 5.6 |
| 3 | 220 | 170 | 7.5 |
| 4 | 340 | 170 | 8.2 |
| 5 | 340 | 510 | 9.4 |

* **Example to understand Concept of Keys**



**Candidate Key:** The minimal set of attributes which can uniquely identify a tuple is known as candidate key. For Example, STUD\_NO in STUDENT relation.

* The value of Candidate Key is unique and non-null for every tuple.
* There can be more than one candidate key in a relation. For Example, STUD\_NO is candidate key for relation STUDENT.
* The candidate key can be simple (having only one attribute) or composite as well. For Example, {STUD\_NO, COURSE\_NO} is a composite candidate key for relation STUDENT\_COURSE.
* No of candidate keys in a Relation are **nC(floor(n/2)),** for example if a Relation have 5 attribute i.e. R (A, B, C, D, E) then total no of candidate keys are 5C(floor(5/2))=10.

**Super Key:**The set of attributes which can uniquely identify a tuple is known as Super Key. For Example, STUD\_NO, (STUD\_NO, STUD\_NAME) etc.

* Adding zero or more attributes to candidate key generates super key.
* A candidate key is a super key but vice versa is not true.

**Primary Key:** There can be more than one candidate key in relation out of which one can be chosen as the primary key. For Example, STUD\_NO, as well as STUD\_PHONE both, are candidate keys for relation STUDENT but STUD\_NO can be chosen as the primary key (only one out of many candidate keys).

**Foreign Key:** If an attribute can only take the values which are present as values of some other attribute, it will be a foreign key to the attribute to which it refers. The relation which is being referenced is called referenced relation and the corresponding attribute is called referenced attribute and the relation which refers to the referenced relation is called referencing relation and the corresponding attribute is called referencing attribute. The referenced attribute of the referenced relation should be the primary key for it.

For Example, STUD\_NO in STUDENT\_COURSE is a foreign key to STUD\_NO in STUDENT relation.

It may be worth noting that unlike, Primary Key of any given relation, Foreign Key can be NULL as well as may contain duplicate tuples i.e., it need not follow uniqueness constraint.

For Example, STUD\_NO in STUDENT\_COURSE relation is not unique. It has been repeated for the first and third tuple. However, the STUD\_NO in STUDENT relation is a primary key and it needs to be always unique and it cannot be null.

Difference Between Primary key & Foreign key

|  |  |
| --- | --- |
| **Primary Key** | **Foreign Key** |
| Helps you to uniquely identify a record in the table. | It is a field in the table that is the primary key of another table. |
| Primary Key never accept null values. | A foreign key may accept multiple null values. |
| Primary key is a clustered index and data in the DBMS table are physically organized in the sequence of the clustered index. | A foreign key cannot automatically create an index, clustered or non-clustered. However, you can manually create an index on the foreign key. |
| You can have the single Primary key in a table. | You can have multiple foreign keys in a table. |