Types of SQL Keys

We have following types of keys in SQL which are used to fetch records from tables and to make relationship among tables or views.

1. Super Key

Super key is a set of one or more than one keys that can be used to identify a record uniquely in a table.**Example :**Primary key, Unique key, Alternate key are subset of Super Keys.

1. Candidate Key

A Candidate Key is a set of one or more fields/columns that can identify a record uniquely in a table. There can be multiple Candidate Keys in one table. Each Candidate Key can work as Primary Key.

**Example:** In below diagram ID, RollNo and EnrollNo are Candidate Keys since all these three fields can be work as Primary Key.

1. Primary Key

Primary key is a set of one or more fields/columns of a table that uniquely identify a record in database table. It can not accept null, duplicate values. Only one Candidate Key can be Primary Key.

1. Alternate key

A Alternate key is a key that can be work as a primary key. Basically it is a candidate key that currently is not primary key.

**Example:** In below diagram RollNo and EnrollNo becomes Alternate Keys when we define ID as Primary Key.

1. Composite/Compound Key

Composite Key is a combination of more than one fields/columns of a table. It can be a Candidate key, Primary key.

1. Unique Key

Uniquekey is a set of one or more fields/columns of a table that uniquely identify a record in database table. It is like Primary key but it can accept only one null value and it can not have duplicate values. For more help refer the article[Difference between primary key and unique key](http://www.dotnet-tricks.com/Tutorial/sqlserver/V2bS260912-Difference-between-Primary-Key-and-Unique-Key.html).

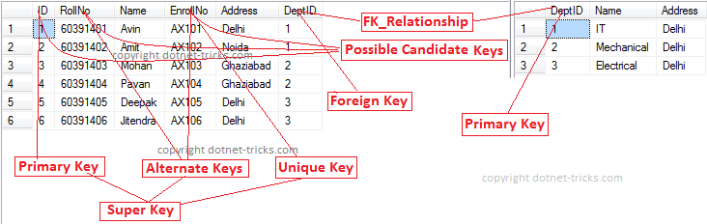
1. Foreign Key

Foreign Key is a field in database table that is Primary key in another table. It can accept multiple null, duplicate values. For more help refer the article [Difference between primary key and foreign key](http://www.dotnet-tricks.com/Tutorial/sqlserver/TENc260912-Difference-between-Primary-Key-and-Foreign-Key.html).

**Example :** We can have a DeptID column in the Employee table which is pointing to DeptID column in a department table where it a primary key.

**Defined Keys -**

1. **CREATE TABLE Department**
2. **(**
3. **DeptID int PRIMARY KEY,**
4. **Name varchar (50) NOT NULL,**
5. **Address varchar (200) NOT NULL, )**
6. **CREATE TABLE Student**
7. **(**
8. **ID int PRIMARY KEY,**
9. **RollNo varchar(10) NOT NULL,**
10. **Name varchar(50) NOT NULL,**
11. **EnrollNo varchar(50) UNIQUE,**
12. **Address varchar(200) NOT NULL,**
13. **DeptID int FOREIGN KEY REFERENCES Department(DeptID)**
14. **)**



In SQL, the definition of a superkey is a set of columns in a table for which there are no two rows that will share the same combination of values. So, the superkey is unique for each and every row in the table. A superkey can also be just a single column.

Example of a superkey

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.

|  |
| --- |
|  |

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.

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.

Super Keys : Super key stands for superset of a key.  
A Super Key is a set of one or more attributes that are taken collectively and can identify all other attributes uniquely.  
   
  
Candidate Keys  
Candidate Keys are super keys for which no proper subset is a super key. In other words candidate keys are minimal super keys.  
  
Primary Key:  
It is a candidate key that is chosen by the database designer to identify entities with in an entity set. Primary key is the minimal super keys. In the ER diagram primary key is represented by underlining the primary key attribute. Ideally a primary key is composed of only a single attribute. But it is possible to have a primary key composed of more than one attribute.  
   
Composite Key  
Composite key consists of more than one attributes.  
   
Example: Consider a Relation or Table R1. Let A,B,C,D,E are the attributes of this relation.  
   
R(A,B,C,D,E)  
A→BCDE This means the attribute 'A' uniquely determines the other attributes B,C,D,E.  
BC→ADE This means the attributes 'BC' jointly determines all the other attributes A,D,E in the relation.  
   
Primary Key :A  
Candidate Keys :A, BC  
Super Keys : A,BC,ABC,AD  
   
ABC,AD are not Candidate Keys since both are not minimal super keys.