**Definition of Super Key in DBMS**: A super key is a set of one or more attributes (columns), which can uniquely identify a row in a table.

**How candidate key is different from super key?**

Answer is simple – Candidate keys are selected from the set of super keys, the only thing we take care while selecting candidate key is: It should not have any redundant attribute. That’s the reason they are also termed as minimal super key.

Let’s take an example to understand this:  
**Table: Employee**

Emp\_SSN Emp\_Number Emp\_Name

--------- ---------- --------

123456789 226 Steve

999999321 227 Ajeet

888997212 228 Chaitanya

777778888 229 Robert

**Super keys**: The above table has following super keys. All of the following sets of super key are able to uniquely identify a row of the employee table.

1. {Emp\_Id} – No redundant attributes  
   2. {Emp\_Number} – No redundant attributes  
   3. {Emp\_Id, Emp\_Number} – Redundant attribute. Either of those attributes can be a minimal super key as both of these columns have unique values.  
   4. {Emp\_Id, Emp\_Name} – Redundant attribute Emp\_Name.  
   5. {Emp\_Id, Emp\_Number, Emp\_Name} – Redundant attributes. Emp\_Id or Emp\_Number alone are sufficient enough to uniquely identify a row of Employee table.  
   6. {Emp\_Number, Emp\_Name} – Redundant attribute Emp\_Name.

**Candidate Keys**: As I mentioned in the beginning, a candidate key is a minimal super key with no redundant attributes. The following two set of super keys are chosen from the above sets as there are no redundant attributes in these sets.

* {Emp\_SSN}
* {Emp\_Number}

Only these two sets are candidate keys as all other sets are having redundant attributes that are not necessary for unique identification.

**Super key vs Candidate Key**

I have been getting lot of comments regarding the confusion between super key and candidate key. Let me give you a clear explanation.  
1. First you have to understand that all the candidate keys are super keys. This is because the candidate keys are chosen out of the super keys.  
2. How we choose candidate keys from the set of super keys? We look for those keys from which we cannot remove any fields. In the above example, we have not chosen {Emp\_SSN, Emp\_Name} as candidate key because {Emp\_SSN} alone can identify a unique row in the table and Emp\_Name is redundant.

[**Primary key**](https://beginnersbook.com/2015/04/primary-key-in-dbms/):  
A Primary key is selected from a set of candidate keys. This is done by database admin or database designer. We can say that either {Emp\_SSN} or {Emp\_Number} can be chosen as a primary key for the table Employee.

**Foreign keys**

**Definition**: Foreign keys are the columns of a table that points to the [primary key](https://beginnersbook.com/2015/04/primary-key-in-dbms/) of another table. They act as a cross-reference between tables.

**Note**: Practically, the foreign key has nothing to do with the primary key tag of another table, if it points to a unique column (not necessarily a primary key) of another table then too, it would be a foreign key. So, a correct definition of foreign key would be: Foreign keys are the columns of a table that points to the [candidate key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) of another table.

**Definition of Composite key:** A key that has more than one attributes is known as composite key. It is also known as compound key.

**Note:** Any key such as [super key](https://beginnersbook.com/2015/04/super-key-in-dbms/), [primary key](https://beginnersbook.com/2015/04/primary-key-in-dbms/), [candidate key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) etc. can be called composite key if it has more than one attributes.

## Composite key Example

Lets consider a table Sales. This table has four columns (attributes) – cust\_Id, order\_Id, product\_code & product\_count.

**Table – Sales**

cust\_Id order\_Id product\_code product\_count

-------- -------- ------------ -------------

C01 O001 P007 23

C02 O123 P007 19

C02 O123 P230 82

C01 O001 P890 42

None of these columns **alone** can play a role of key in this table.

Column **cust\_Id** alone cannot become a key as a same customer can place multiple orders, thus the same customer can have multiple entires.

Column **order\_Id** alone cannot be a primary key as a same order can contain the order of multiple products, thus same order\_Id can be present multiple times.

Column **product\_code** cannot be a primary key as more than one customers can place order for the same product.

Column **product\_count** alone cannot be a primary key because two orders can be placed for the same product count.

Based on this, it is safe to assume that the key should be having more than one attributes:  
**Key in above table: {cust\_id, product\_code}**

This is a composite key as it is made up of more than one attributes.

**Alternative or Secondary keys**.

As we have seen in the [candidate key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) guide that a table can have multiple candidate keys. Among these candidate keys, only one key gets selected as [primary key](https://beginnersbook.com/2015/04/primary-key-in-dbms/), the remaining keys are known as **alternative or secondary keys**.

## Alternate Key Example

Lets take an example to understand the alternate key concept. Here we have a table Employee, this table has three attributes: Emp\_Id, Emp\_Number & Emp\_Name.

**Table: Employee/strong>**

Emp\_Id Emp\_Number Emp\_Name

------ ---------- --------

E01 2264 Steve

E22 2278 Ajeet

E23 2288 Chaitanya

E45 2290 Robert

There are two candidate keys in the above table:  
{Emp\_Id}  
{Emp\_Number}

DBA (Database administrator) can choose any of the above key as primary key. Lets say Emp\_Id is chosen as primary key.

Since we have selected Emp\_Id as primary key, the remaining key Emp\_Number would be called alternative or secondary key.