**Types of keys in DBMS**

**Note:** Guys I have been getting comments that there are no examples of keys here. If you click on the hyperlink provided below in green colour, you would see the complete separate tutorial of each key with examples.

[Primary Key](https://beginnersbook.com/2015/04/primary-key-in-dbms/) – A primary is a column or set of columns in a table that uniquely identifies tuples (rows) in that table.

[Super Key](https://beginnersbook.com/2015/04/super-key-in-dbms/) – A super key is a set of one of more columns (attributes) to uniquely identify rows in a table.

[Candidate Key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) – A super key with no redundant attribute is known as candidate key

[Alternate Key](https://beginnersbook.com/2015/04/alternate-key-in-dbms/) – Out of all candidate keys, only one gets selected as primary key, remaining keys are known as alternate or secondary keys.

[Composite Key](https://beginnersbook.com/2015/04/composite-key-in-dbms/) – A key that consists of more than one attribute to uniquely identify rows (also known as records & tuples) in a table is called composite key.

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

# Primary key in DBMS

**Definition**: A **primary key** is a minimal set of attributes (columns) in a table that uniquely identifies tuples (rows) in that table.

## Primary Key Example in DBMS

Lets take an example to understand the concept of primary key. In the following table, there are three attributes: Stu\_ID, Stu\_Name & Stu\_Age. Out of these three attributes, one attribute or a set of more than one attributes can be a primary key.

Attribute Stu\_Name alone cannot be a primary key as more than one students can have same name.

Attribute Stu\_Age alone cannot be a primary key as more than one students can have same age.

Attribute Stu\_Id alone is a primary key as each student has a unique id that can identify the student record in the table.

**Note:** In some cases an attribute alone cannot uniquely identify a record in a table, in that case we try to find a set of attributes that can uniquely identify a row in table. We will see the example of it after this example.

**Table Name: STUDENT**

|  |  |  |
| --- | --- | --- |
| Stu\_Id | Stu\_Name | Stu\_Age |
| 101 | Steve | 23 |
| 102 | John | 24 |
| 103 | Robert | 28 |
| 104 | Steve | 29 |
| 105 | Carl | 29 |

## Points to Note regarding Primary Key

* We denote usually denote it by underlining the attribute name (column name).
* The value of primary key should be unique for each row of the table. The column(s) that makes the key cannot contain duplicate values.
* The attribute(s) that is marked as primary key is not allowed to have null values.
* Primary keys are not necessarily to be a single attribute (column). It can be a set of more than one attributes (columns). For example {Stu\_Id, Stu\_Name} collectively can identify the tuple in the above table, but we do not choose it as primary key because Stu\_Id alone is enough to uniquely identifies rows in a table and we always go for minimal set. Having that said, we should choose more than one columns as primary key only when there is no single column that can uniquely identify the tuple in table.

## Another example of primary key – More than one attributes

Consider this table ORDER, this table keeps the daily record of the purchases made by the customer. This table has three attributes: Customer\_ID, Product\_ID & Order\_Quantity.

Customer\_ID alone cannot be a primary key as a single customer can place more than one order thus more than one rows of same Customer\_ID value. As we see in the following example that customer id 1011 has placed two orders with product if 9023 and 9111.

Product\_ID alone cannot be a primary key as more than one customers can place a order for the same product thus more than one rows with same product id. In the following table, customer id 1011 & 1122 placed an order for the same product (product id 9023).

Order\_Quantity alone cannot be a primary key as more more than one customers can place the order for the same quantity.

Since none of the attributes alone were able to become a primary key, lets try to make a set of attributes that plays the role of it.

{Customer\_ID, Product\_ID} together can identify the rows uniquely in the table so this set is the primary key for this table.

**Table Name: ORDER**

|  |  |  |
| --- | --- | --- |
| Customer\_ID | Product\_ID | Order\_Quantity |
| 1011 | 9023 | 10 |
| 1122 | 9023 | 15 |
| 1099 | 9031 | 20 |
| 1177 | 9031 | 18 |
| 1011 | 9111 | 50 |

**Note:** While choosing a set of attributes for a primary key, we always choose the minimal set that has minimum number of attributes. For example, if there are two sets that can identify row in table, the set that has minimum number of attributes should be chosen as primary key.

## How to define primary key in RDBMS?

In the above example, we already had a table with data and we were trying to understand the purpose and meaning of primary key, however you should know that generally we define the primary key during table creation. We can define the primary key later as well but that rarely happens in the real world scenario.

Lets say we want to create the table that we have discussed above with the customer id and product id set working as primary key. We can do that in SQL like this:

Create table ORDER

(

Customer\_ID int not null,

Product\_ID int not null,

Order\_Quantity int not null,

Primary key (Customer\_ID, Product\_ID)

)

Suppose we didn’t define the primary key while creating table then we can define it later like this:

ALTER TABLE ORDER

ADD CONSTRAINT PK\_Order PRIMARY KEY (Customer\_ID, Product\_ID);

**Another way:**  
When we have only one attribute as primary key, like we see in the first example of STUDENT table. we can define the key like this as well:

Create table STUDENT

(

Stu\_Id int primary key,

Stu\_Name varchar(255) not null,

Stu\_Age int not null

)

# Super key in DBMS

**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. Often [DBMS beginners](https://beginnersbook.com/2015/04/dbms-tutorial/) get confused between super key and [candidate key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/), so we will also discuss candidate key and its relation with super key in this article.

## 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.

* {Emp\_SSN}
* {Emp\_Number}
* {Emp\_SSN, Emp\_Number}
* {Emp\_SSN, Emp\_Name}
* {Emp\_SSN, Emp\_Number, Emp\_Name}
* {Emp\_Number, 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.

# Candidate Key in DBMS

**Definition of Candidate Key in DBMS**: A [super key](https://beginnersbook.com/2015/04/super-key-in-dbms/) with no redundant attribute is known as candidate key. Candidate keys are selected from the set of super keys, the only thing we take care while selecting candidate key is that the candidate key should not have any redundant attributes. That’s the reason they are also termed as minimal super key.

## Candidate Key Example

Lets take an example of table “Employee”. This table has three attributes: Emp\_Id, Emp\_Number & Emp\_Name. Here Emp\_Id & Emp\_Number will be having unique values and Emp\_Name can have duplicate values as more than one employees can have same name.

Emp\_Id Emp\_Number Emp\_Name

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

E01 2264 Steve

E22 2278 Ajeet

E23 2288 Chaitanya

E45 2290 Robert

How many super keys the above table can have?  
1. {Emp\_Id}  
2. {Emp\_Number}  
3. {Emp\_Id, Emp\_Number}  
4. {Emp\_Id, Emp\_Name}  
5. {Emp\_Id, Emp\_Number, Emp\_Name}  
6. {Emp\_Number, Emp\_Name}

Lets select the candidate keys from the above set of super keys.

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.

The **candidate keys** we have selected are:  
{Emp\_Id}  
{Emp\_Number}

**Note**: A [primary key](https://beginnersbook.com/2015/04/primary-key-in-dbms/) is selected from the set of candidate keys. That means we can either have Emp\_Id or Emp\_Number as primary key. The decision is made by DBA (Database administrator)

# Foreign key in DBMS

**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.

**For example**:  
In the below example the Stu\_Id column in Course\_enrollment table is a foreign key as it points to the primary key of the Student table.

Course\_enrollment table:

|  |  |
| --- | --- |
| Course\_Id | Stu\_Id |
| C01 | 101 |
| C02 | 102 |
| C03 | 101 |
| C05 | 102 |
| C06 | 103 |
| C07 | 102 |

Student table:

|  |  |  |
| --- | --- | --- |
| Stu\_Id | Stu\_Name | Stu\_Age |
| 101 | Chaitanya | 22 |
| 102 | Arya | 26 |
| 103 | Bran | 25 |
| 104 | Jon | 21 |

**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.

# Composite key in DBMS

**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.

# Alternate key in DBMS

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.