

# Types of Keys in DBMS

In DBMS, there are seven types of keys available. Each type has its significance. These are namely:

- Candidate Key
- Primary Key
- Foreign Key
- Super Key
- Alternate Key
- Composite Key
- Unique Key

The primary key maintains the integrity constraints of the relationships. The referential integrity constraints use a foreign key. We will now discuss and understand each key thoroughly.

## What is Key in DBMS?

A key is an integral part of any relation/ table in DBMS. Every relation or table in a relational database should have a key defined for itself. Specifically, there should be a primary key for each relation. As per the [Gate Syllabus for CSE](#), there are different types of keys in DBMS, which will be discussed in the other sections of this article.

### Key in DBMS Definition

A [key in DBMS](#) is defined as an attribute or a set of attributes uniquely identifying a tuple (row) in the relation. It ensures that no two tuples in a relation are the same. Moreover, it helps in maintaining or establishing relationships among different relations.

### Candidate Key in DBMS

A candidate key refers to the minimal attributes uniquely identifying a tuple in a relation. One of the candidate keys is selected as a primary key for a relation.

A relation can have one or more candidate keys. One key is selected from the many candidate keys to be the relationship's primary key.

### Primary Key in DBMS

The primary key is one of the candidate keys of the relational schema whose field values must not be null. To understand the primary key better, we can see the [difference between primary key and unique key](#). The following are the primary key constraints:

- It must be one of the candidate's keys.
- The field values are not allowed to be null.
- At most, one primary key is allowed.

## Alternate Key in DBMS

An alternate key is the candidate key of the relationship, except for the primary key. The alternate key constraints are:

- It must be candidate key.
- Field values allowed null values.
- Many alternate keys are allowed for a relational schema.

**Note:** Alternate Key allows unique + null values.

## Super Key in DBMS

Collecting all the keys that enable us to identify each row in a table specifically is referred to as a super key. This indicates that all of the columns of a table that may uniquely identify its columns serve as the super keys. It may not be the minimal set of attributes.

A super key is the superset of a candidate key. To make the primary Key the table's identity attribute, we must choose it from the list of super keys. Every candidate key is also a super key, but vice versa is not true.

## Foreign Key in DBMS

A foreign key is a set of attributes that references the primary Key or the alternate Key of the same or some other relation. It is also called a referential key. The foreign key is very different from the primary key and the [difference between the foreign key and the primary](#) can be checked here. The foreign key constraints are as follows:

- Used to relate data between tables or referenced relation and referencing relation.
- It allows null values.
- Referencing a relation record whose foreign key value is null is out of referential integrity constraints.

## Composite Key in DBMS

The term "composite key" refers to a set of characteristics that enable us to identify each tuple in a table in a particular way. If we look at the attributes in a set separately, they could not all be unique. Consequently, when we combine them all, it will guarantee complete uniqueness. Composite keys are essential for the MSQ-based questions in the [GATE question paper](#). It is the candidate key with at least one attribute key (can be two or more).

## Unique Key in DBMS

A column or group of columns known as a unique key uniquely identifies each record in a table. In this key, each value must be distinct. Because a unique key can only have one null value while a primary key cannot, a unique key differs from a primary key.

## Types of Keys in DBMS with Example

Types of keys in DBMS help us reduce redundancy and remove database anomalies. Let us see an example for each type to better understand the keys in DBMS for the [GATE exam](#).

### Example of Simple Candidate Key in DBMS

Consider the relational schema

Student (Sid, Sname, DOB, gender, course, mobile\_number)

The examples of simple candidate keys that each set can uniquely identify each student in the Student table are as given below:

- (Sid)
- (mobile\_number)
- (DOB)
- (Sname)

### Example of Super Key in DBMS

Consider the relational schema

Student (Sid, Sname, DOB, gender, course, mobile\_number)

The examples of super keys in DBMS that each set can uniquely identify each student in the Student table are given below:

- (Sid, Sname, DOB, gender, course, mobile\_number)
- (Sid, course, mobile\_number)
- (Sid, Sname, DOB)
- (Sid, Sname, mobile\_number)

### Example of Compound Candidate Key in DBMS

Consider the relational schema

Student (Sid, Sname, DOB, gender, course, mobile\_number)

The examples of compound candidate keys that each set can uniquely identify each student in the Student table are as given below:

- (Sid, Sname)
- (Sid, mobile\_number)
- (Sid, DOB)
- (Sname, mobile\_number)

### Example of Overlapping Candidate Key in DBMS

Candidate keys in DBMS: (Sid, Sname), (Sname, SMobile). Here Sname is common to both keys. So, these candidate keys are called overlapping candidate keys.

### Example of Candidate Key in DBMS

Consider the relational schema

Student (Sid, Sname, DOB, gender, course, mobile\_number)

The examples of candidate keys in DBMS that each set can uniquely identify each student in the Student table are given below:

- (Sid, DOB)
- (Sname, mobile\_number)

### Applications of Types of Keys in DBMS

The various types of keys in DBMS help in achieving normalization. We have discussed all the types of keys in DBMS: candidate key, super key, foreign key, primary key, alternate key, composite key, and unique key. Following are the main applications of using types of keys in DBMS:

- In relational DBMS, keys play an important role.
- Keys help in determining a row uniquely in relation.