



## Table, key and foreign key definitions, update behaviors

A key refers to an attribute/a set of attributes that help us identify a row (or tuple) uniquely in a table (or relation). A key is also used when we want to establish relationships between the different columns and tables of a relational database. The individual values present in a key are commonly referred to as key values.

### Why do we require Keys in DBMS?

We use a key for defining various types of integrity constraints in a database. A table, on the other hand, represents a collection of the records of various events for any relation. Now, there might be thousands of these records, and some of these might even be duplicated.

Thus, we need a way in which one can identify all of these records uniquely and separately, i.e., without any duplicates. This hassle is removed with the help of keys.

For example, let us consider a database of all the students who are studying in a college. What attribute of all the students, according to you, will identify each of these people uniquely? We can refer to these students by their names, departments, sections, and year. Similarly, we can also mention only the university roll number and fetch all the other details based on that roll number.

The keys in DBMS can be a combination of multiple attributes (or columns), or they can be just one single attribute. The primary motive of the keys is to provide every record with a unique identity of its own.

### What are the different types of Keys in DBMS?

Keys are of seven broad types in DBMS:

1. Candidate Key
2. Primary Key
3. Foreign Key
4. Super Key
5. Alternate Key
6. Composite Key
7. Unique Key

#### 1. Primary Key

The primary key refers to a column or a set of columns of a table that helps us identify all the records uniquely present in that table. A table can consist of just one primary key. Also, this primary key cannot consist of the same values reappearing/repeating for any of its rows. All the values of a primary key have to be different, and there should be no repetitions.

The PK (PRIMARY KEY) constraint that we put on a column/set of columns won't allow these to have a null value or a duplicate. Any table can consist of only a single primary key constraint. A foreign key (explained below) that refers to it can never change the values present in the primary key.



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2. Super Key

A super key refers to the set of all those keys that help us uniquely identify all the rows present in a table. It means that all of these columns present in a table that can identify the columns of that table uniquely act as the super keys.

A super key is a candidate key's superset (candidate key has been explained below). We need to pick the primary key of any table from the super key's set so as to make it the table's identity attribute.

3. Candidate Key

The candidate keys refer to those attributes that identify rows uniquely in a table. In a table, we select the primary key from a candidate key. Thus, a candidate key has similar properties as that of the primary keys that we have explained above. In a table, there can be multiple candidate keys.

4. Alternate Key

As we have stated above, any table can consist of multiple choices for the primary key. But, it can only choose one. Thus, all those keys that did not become a primary key are known as alternate keys.

5. Foreign Key

We use a foreign key to establish relationships between two available tables. The foreign key would require every value present in a column/set of columns to match the referential table's primary key. A foreign key helps us to maintain data as well as referential integrity.

6. Composite Key

The composite key refers to a set of multiple attributes that help us uniquely identify every tuple present in a table. The attributes present in a set may not be unique whenever we consider them separately. Thus, when we take them all together, it will ensure total uniqueness.

7. Unique Key

A unique key refers to a column/a set of columns that identify every record uniquely in a table. All the values in this key would have to be unique. Remember that a unique key is different from a primary key. It is because it is only capable of having one null value. A primary key, on the other hand, cannot have a null value.

***Practice Questions on Types of Keys in DBMS***

1. Look at the relation scheme  $A = \{F, E, H, G, J, I, L, K, N, M\}$  and a set of some functional dependencies  $\{\{F, E\} \rightarrow \{H\}, \{E\} \rightarrow \{J, I\}, \{F, G\} \rightarrow \{L, K\}, L \rightarrow \{N\}, K \rightarrow \{M\}$  on A. What would be the key for A here?

A.  $\{F, E\}$

B.  $\{F, E, G\}$



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

C. {F, E, G, L, K}

D. {F}

**Answer: B.** {F, E, G}

**Explanation:** We check the attribute that is the closure of all the provided options. A set whose closure would provide us with the entire relation A will be the accurate answer.

A:  $\{F, E\}^+ = \{EFGIJ\} \neq A$

B:  $\{F, E, G\}^+ = \{EFGHIJKLMN\} = A$

C:  $\{F, E, G, L, K\}^+ = \{EFGHIJKLMN\} = A$

D:  $\{F\}^+ = \{F\} \neq A$

Both of these options, B & C, would provide us with the entire relation scheme. But we would choose the minimal option out of these to be the accurate answer because the candidate key must be the minimal super key.

2. A relation X consists of eight attributes BADCFEHG. The fields of X consist of just atomic values.  $E = \{DG \rightarrow H, B \rightarrow AD, A \rightarrow DEG, F \rightarrow B, E \rightarrow FH\}$  refers to a set of FDs (functional dependencies) so that  $E^+$  is exactly the set of functional dependencies that hold for X. Here, the relation X consists of how many candidate keys?

A. 6

B. 5

C. 4

D. 3

**Answer: C.** 4

**Explanation:** Let us take the LHS of every FD given here in this question and then find the attribute closures.

$DG^+ = H$

$B^+ = BADFEHG$

$A^+ = BADFEHG$

$F^+ = BADFEHG$

$E^+ = BADFEHG$



SNS COLLEGE OF TECHNOLOGY, COIMBATORE –35  
(An Autonomous Institution)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Thus, we can see that the closures of B, A, F, E have the entire relationship, but not for the attribute C.  
Thus, there are a total of four candidate keys, namely BC, AC, FC and EC.