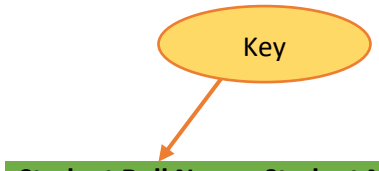
Keys

A key refers to an attribute/a set of attribute that help us identify a row (or tuple) unique, in a table (or relation). Keys play an important role in the relational database. It 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 need keys in DBMS?

A key is used 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. Keys in DBMS ensure that you can uniquely identify a table record despite these challenges.
Allows you to establish a relationship between and identify the relation between tables.
It helps you to enforce identity and integrity in the relationship.

Example:

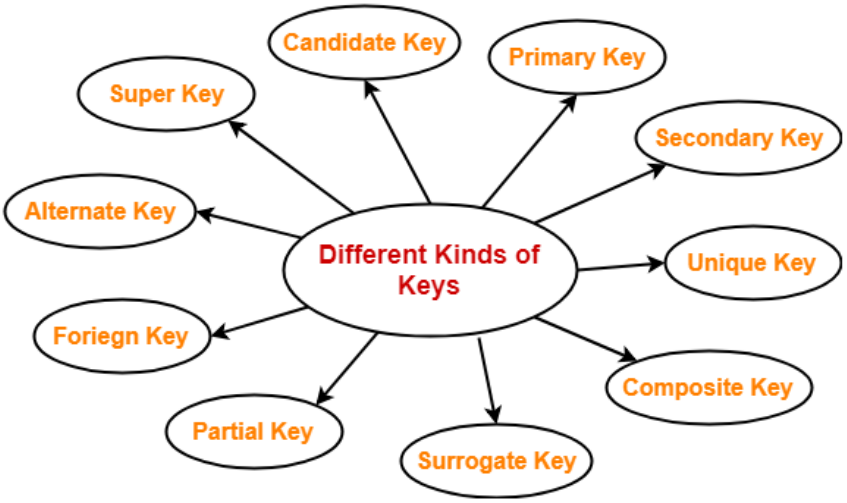


Student Roll No.	Student Name	Student Age
1	Abhilasha	21
2	Priya Raj	22
3	Priya Patel	21
4	Tanya	23
5	Rishika	24
6	Vaibhavi	23

In the above example we can observe, that only Student roll nos. can help us uniquely identify each Student. As, two students have the same name (duplicate) hence that attribute cannot be used as a key. In addition, many students have same age, hence is not a unique attribute either, and hence cannot be used as key.

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.

Type of keys in DBMA (Database Management System):



1. Super key:

- A super key is a set of attributes that can identify each tuple uniquely in the given relation.
- A super key is not restricted to have any specific number of attributes.
- Thus, a super key may consist of any number of attributes.
- A super key is a superset of candidate key. (A candidate key is a super key but visa-versa is not true)

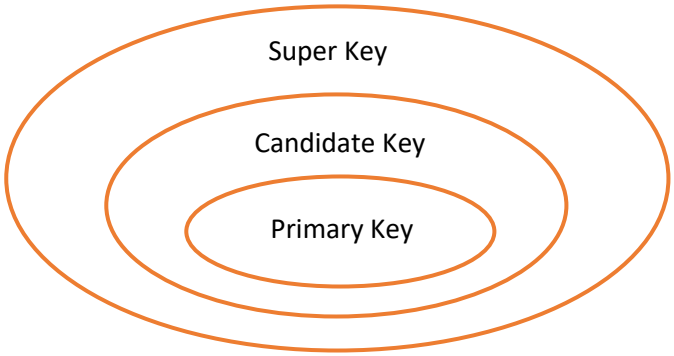
Example:

Employee_Table

Emp_ID	Name	Aadhar_No	Email_ID	Dept_ID
1	Aashna	470532579876	aa@gmail.com	12
2	Naman	543895411234	na@gmail.com	14
3	Naman	909878541984	Na_1@gmail.com	14
4	Tanuja	863650091208	ta@gmail.com	13

Super keys from the above relation:

a. Emp_ID
b. Aadhar_No
c. Email_ID
d. Emp_ID, Aadhar_No
e. Aadhar_No, Email_ID
f. Emp_ID, Email_ID
g. Emp_ID, Aadhar_No, Email_ID
h. Emp_ID, Name
i. Emp_ID, Aadhar_No, Email_ID, Name, Dept_ID
j. etc.



2. Candidate Key:

- A candidate key is an attribute or set of attributes that can uniquely identify a tuple. (And Removing any attribute from the candidate key fails in identifying each tuple uniquely)
- A candidate key is a minimal super key, or a super key with no redundant attributes. (It is called a minimal super key because we select a candidate key from a set of super key such that selected candidate key is the minimum attribute required to uniquely identify the table)
- It is defined as distinct set of attributes from which primary key can be selected.
- The value of candidate key must always be unique and can never be NULL.

Example:

Student_Table

Roll_No	Name	Phone_No	Subject_No
1	Alisha	9999999999	10
2	Sarah	8888888888	20
3	Shubham	7777777777	10

Candidate keys from the **Student_Table** are **Roll_No** or **Phone_No.**
Candidate keys from the **Employee_Table** are **Employe_ID**, **Aadhar_No** or **Email_ID**.

3. Primary Key:

- Primary key is one of the candidate key chosen by the database to uniquely identify the tuple in the relation.
- The value of primary key can never be null.
- The value of primary key must always be unique (not duplicate).
- It can identify only one tuple (a record) at a time.
- No updation I possible in the values of primary key i.e. they cannot be changed.
- The value of primary key must be assigned when inserting a record.
- A relation is allowed to have only one primary key.

Example:

Primary key from the **Employee_Table** is **Employe_ID**.

4. Alternate Key:

- The candidate keys apart from the one selected, as the primary key are known as the alternate keys.
- The alternate key may or may not exist. If there is only one candidate key in a relation, it does not have an alternate key.

Example:

In the Employee_Table:-

- **Employe_ID** is the selected primary key. Hence, rest of the candidate keys that are **Aadhar_No** and **Email_ID** are considered as the Alternate Keys.

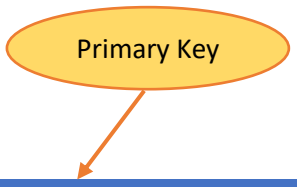
5. Foreign Key:

- Foreign key is a key to link two tables together.
- It can be an attribute (or set of attributes) in one table that refers to the primary key in another table (referential table). Hence, it helps to maintain data and referential integrity.
- Foreign key references the primary key of the table.
- Foreign key may have a name other than that of a primary key.
- Foreign key can take the null values.
- Foreign key may not be unique. (As in the example many people may work in the same department and hence have the same dept_ID)
- Referenced relation may also be called as the master table or primary table whereas the referencing table may also be called as the foreign table.

Example:

Employee_Table (Referencing relation)

Emp_ID	Name	Aadhar_No	Email_ID	Dept_ID
1	Aashna	470532579876	aa@gmail.com	12
2	Naman	543895411234	na@gmail.com	14
3	Naman	909878541984	Na_1@gmail.com	14
4	Tanuja	863650091208	ta@gmail.com	13



Department_Table (Referenced relation)

Dept_ID	Dept_Name
12	Marketing
14	HR
13	Sales

Therefore, here the **Employee_Table** is the referencing relation and the **Dept_ID** is the **foreign key** in that table, while the **Department_Table** is the referenced table, to which the Employee_Table is referencing where the **Dept_ID** is the **Primary key**.

6. Composite Key:

- It is a set of two or more attributes that help identify each tuple in a table uniquely. Or whenever a primary key consists of more than one attribute, it is known as a composite key.
- The attributes in the set may not be unique when considered separately but when taken together, they will ensure uniqueness.
- It is also known as Concatenated Key.

Example:

Emp_Salary_Table

Emp_ID	Name	Salary_Month	Salary
1	Arunita	January	1.2k
2	Karuna	January	2.0k
3	Rakesh	January	1.5k
1	Arunita	February	1.2k
2	Karuna	February	2.0k
3	Rakesh	February	1.5k

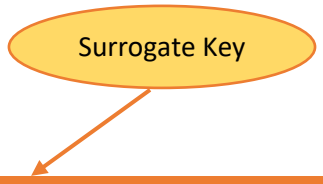
In the **Emp_Salary_table**, none of the attributes are unique, hence 2 attributes are to be taken to uniquely identify each tuple in the table. So, **Emp_ID and Salary_Month** can together for the composite key.

7. Surrogate Key:

- If a relation has no attribute to which can be used to identify the data uniquely (no natural primary key) then we create an attribute for this purpose, which is called as the surrogate key.
  Therefore, a surrogate key is an artificial key which aims to uniquely identify each record in a relation.
- They do not lend any meaning to the data in the table. Hence, it is a fact-less key.
- Surrogate key in DBMS is generally an integer in a sequential manner.
- Also, a surrogate key is added when two tables are merged (which might have their own primary key) but when merged the primary key might lose its properties to be called as a primary key.

Example:

Item_Table



ID	Item_ID	Item_Name	Colour_ID	Colour_desc
1	M_01	Samsung G4	BK	Black
2	M_01	Samsung G4	YL	Yellow
3	M_02	Redmi 9 prime	YL	Yellow
4	M_02	Redmi 9 prime	BK	Black

In the **Item_Table** there is no natural primary key, hence a new column is created as the surrogate key named **ID**, which now works, as a primary key for the relation.

8. Partial Key:

- A partial key is a key using which all the records of the table cannot be identified uniquely.
- However, a bunch of related tuples can be identified from the table using the partial key.

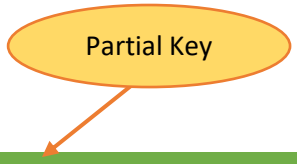
Example:

Let us consider the **Employee_Table** and a **Dependent_Table**:

Employee_Table

Emp_ID	Name	Aadhar_No	Email_ID	Dept_ID
1	Aashna	470532579876	aa@gmail.com	12
2	Naman	543895411234	na@gmail.com	14
3	Naman	909878541984	Na_1@gmail.com	14
4	Tanuja	863650091208	ta@gmail.com	13

Dependent_Table



Emp_ID	Dependent_Name	Relation
1	Sarika	Mother
1	Manoj	Father
2	Tanuja	Spouse
2	Sandhya	Daughter
2	Karuna	Mother

In the **Dependent_Table**, the **Emp_ID** is the **partial key** because one emp_ID gives a bunch of tuples rather than uniquely identifying every record.

9. Unique Key:

- It uniquely identifies a tuple in a relation; hence, it is unique for all the records of the table.
- Once assigned, its value cannot be changed. (It is non-updatable)
- It may have one Null value.

Example:

- Voter ID can be an example of Unique key
- It is unique to all citizens. If it gets lost a duplicate Voter ID is issues but the Voter ID number remains the same, hence it is non-updatable.
- Few citizens may not have got their voter ID, hence its value is Null for them.
- Another example can be Aadhar No.