

# **Keys in DBMS**

## **Introduction:**

- In database management systems (DBMS), keys are crucial for identifying and establishing relationships between data entities within tables. They ensure data integrity, enforce constraints, and facilitate efficient querying. Keys are defined based on their ability to uniquely identify records (tuples) within a table. Different types of keys include super keys, candidate keys, primary keys, alternate keys, and foreign keys, each serving specific roles in database design and management. Keys play an important role in the relational database.

## **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:**

- Candidate Key
- Primary Key
- Foreign Key
- Super Key
- Alternate Key

### 1. Primary Key:

#### ○ What is the Primary Key in DBMS?

- 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.

#### ○ Importance of using Primary Key in DBMS:

- Using a primary key in DBMS design is of utmost importance. Ensuring the integrity, efficiency & effectiveness of the database system is crucial, and it plays a significant role. One key significance of using a primary key is data uniqueness. It helps to identify records uniquely and prevent data redundancy.
- Another crucial aspect is efficient data retrieval and manipulation. The database management system can quickly locate and retrieve specific records based on their unique identifiers with a primary key. Furthermore, a primary key is a foundation for establishing relationships between tables in a relational database.

#### ○ How Primary Key in DBMS Works?

- A primary key is a distinct label assigned to every entry in a table within a database. It's used to uniquely identify each row or record to easily retrieve, update, or delete. Primary keys must have unique values and cannot contain null values.
- The primary key in DBMS, such as MySQL and Oracle, is typically an auto-incrementing integer. This means the database automatically assigns a new number every time you create a new record, ensuring each row has its unique identifier.

### ○ Characteristics of Primary Key in DBMS

Here are the key characteristics of primary key in DBMS:

- **Uniqueness:** A primary key must ensure the uniqueness of each value within a table. No two records could have the same value for the primary key attribute.
- **Non-nullability:** A primary key must have non-null values. This means that the primary key attribute for each record cannot be empty or null. This constraint ensures that the primary key serves as a reliable identifier for every record in the table.
- **Stability:** The values of a primary key should ideally remain stable over time. It means that once a value is assigned as a primary key for a record, it should only change if necessary. This stability helps maintain consistency and reliability in data relationships and references.
- **Minimal:** A primary key, consisting of the smallest set of attributes necessary to identify each record uniquely, should be minimal. This helps optimize storage space and improve query performance by reducing the index size or data structure associated with the primary key.
- **Easily accessible:** The primary key should be easily accessible, allowing efficient data retrieval and manipulation. It is a key reference point for locating specific records within the table or establishing relationships with other tables. By providing quick and direct access, it enhances the performance and usability of the database system.

### ○ Examples of Primary Key in DBMS

#### ○ Example -1:

- Consider a table called “Students” that stores information about students in a school. Let’s assume we have the following attributes: StudentID, Name, Age, and Grade. In this example, we will designate the “StudentID” as the primary key.

StudentID	Name	Age	Grade
001	Kavya Patel	15	10
002	Dev Joshi	17	12

- In the table above, each row represents a student record. The “StudentID” column serves as the primary key, uniquely identifying each student in the table. No two students can have the same “StudentID” value.
- For example, the first row shows that “Kavya Patel” has a StudentID of “001”. This ID is unique to him and allows us to specifically identify and retrieve his information from the “Students” table.
- The primary key in DBMS plays a crucial role in maintaining data integrity, ensuring that each student’s record is unique and identifiable within the table. It

allows for efficient retrieval, manipulation, and referencing of data in the database management system.

- **Example -2:**

- **Customers**

CustomerID	Name	Email	PhoneNo
001	Kavya Patel	kav12@gmail.com	9382949493
002	Dev Joshi	dev34@gmail.com	8493848494

- **Primary Key:** CustomerID

## 2. Candidate Key

- **What Is A Candidate Key?**

- A candidate key in a database management system (DBMS) is a unique identifier for a record within a table that can be chosen as the primary key. It possesses the essential characteristics required for a primary key: uniqueness and minimal redundancy. While multiple candidate keys may exist within a table, the chosen primary key becomes the definitive means of uniquely identifying each record. Selecting a candidate key is crucial for database normalization and ensures efficient data retrieval and integrity. It acts as a foundation for establishing relationships between tables and maintaining the overall structure and reliability of the database.

- **How A DBMS Candidate Key Is Different From A Primary Key?**

- **Uniqueness:**

- **Candidate Key:** A candidate key is a set of one or more attributes that can uniquely identify a record in a table. Multiple candidate keys may exist within a table, and each candidate key must ensure uniqueness.
- **Primary Key:** The primary key is a specific candidate key chosen as the main means of uniquely identifying records within a table. Unlike candidate keys, a table can have only one primary key, and it ensures uniqueness and non-null values.

- **Selection:**

- **Candidate Key:** Database designers identify all potential candidate keys based on their uniqueness and minimal redundancy. These are options for serving as primary keys.
- **Primary Key:** The primary key is the chosen candidate key from the pool of potential keys. Designers select the primary key based on various factors, including simplicity and efficiency.

- **Constraints:**

- **Candidate Key:** Candidate keys need to adhere to the basic criteria of uniqueness and minimal redundancy but don't have specific constraints imposed by the database system.
- **Primary Key:** The primary key is subject to constraints, such as ensuring uniqueness and non-null values. It acts as a cornerstone for maintaining data integrity and enforcing relationships.

- **Multiplicity:**

- **Candidate Key:** A table can have multiple candidate keys, and each candidate key uniquely identifies records.
- **Primary Key:** A table can have only one primary key, which serves as the primary means of identification for establishing relationships with other tables.

- **Example of Candidate Key in DBMS**

Let us understand the concept of candidate key in DBMS with the help of the following Student table.

Student\_Rollno, Student\_Name, Student\_ID\_no, and Student\_email. These are the attributes of the student\_detail table, in which some are unique and some are not. here Student\_Name is not unique because the name of the student can be the same as we observe Student\_Rollno 01 and 04 have the same name (Vikas) so this column is not unique.

Student_Rollno	Student_Name	Student_ID_no	Student_email
01	Vikas	1233329	v@gmail.com
02	Mohan	1234534	m@gmail.com
03	Lalit	1234328	l@gmail.com
04	Vikas	1239875	vk@gmail.com

**Candidate Keys:**

Student\_Rollno

Student\_ID\_no

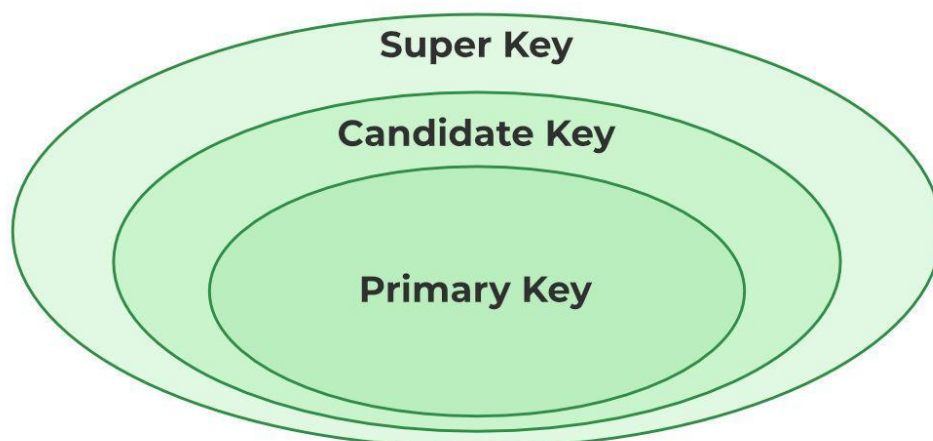
Student\_email

**3. Super Key****○ What Is A Super Key In DBMS?****Definition:**

- A super key is a set of one or more columns (attributes) in a table that, when combined, can uniquely identify each row in that table.
- Super keys make ensuring that the values of the attributes that they contain are unique across all rows in a table. While every table has at least one super key, it may have multiple super keys.
- Adding zero or more attributes to the candidate key generates the super key.
- Super Key values may also be NULL.

**Key Points:**

- **Uniqueness:**
  - A super key ensures that each row in the table can be uniquely identified based on the values in the columns it includes.
- **Flexibility:**
  - It can include more columns than necessary to uniquely identify rows, allowing for different combinations of columns.

**Relation between Primary Key, Candidate Key, and Super Key:**

○ **Examples of Super Key**

- Let's consider an EMPLOYEE\_DETAIL table example where we have the following attribute:

**Emp\_SSN:** The SSN number is stored in this field.

**Emp\_Id:** An attribute that stores the value of the employee identification number.

**Emp\_name:** An attribute that stores the name of the employee holding the specified employee id.

**Emp\_email:** An attribute that stores the email id of the specified employees.

The EMPLOYEE\_DETAIL table is given below that will help you understand better:

Emp_SSN	Emp_Id	Emp_name	Emp_email
11051	01	John	john@email.com
11052	02	Merry	merry@email.com
19801	03	Riddle	riddle@email.com
41201	04	Cary	cary@email.com

So, from the above table, we conclude the following set of the super keys.

**Set of super keys obtained**

{ Emp\_SSN }

{ Emp\_Id }

{ Emp\_email }

{ Emp\_SSN, Emp\_Id }

{ Emp\_Id, Emp\_name }

{ Emp\_SSN, Emp\_Id, Emp\_email }

{ Emp\_SSN, Emp\_name, Emp\_Id }

- These all are the set of super keys which, together or combining with other prime attributes, can identify a table uniquely.
- Just like, if we set Super key on Emp\_SSN, it will be able to identify all other tuples of the table very easily. Similarly, if we set the Super key on {Emp\_Id, Emp\_name}, we can easily get the value or details of the other remaining attributes of the employee. So, in this way, we can create and search out the super keys from a table.
- Now, the question is how to identify the Primary key from a super key because the super key is the superset. So, the answer is that a Primary key is picked up from the super key set only by selecting a single attribute that can provide a unique identification to the whole table. Hence, it is the responsibility of the DBA to choose the most appropriate and suitable primary key out of the candidate keys or super key.

#### **4. Foreign Key**

##### **○ What is a Foreign Key?**

- A foreign key is different from a super key, candidate key or primary key because a foreign key is the one that is used to link two tables together or create connectivity between the two.
- A foreign key is the one that is used to link two tables together via the primary key. It means the columns of one table points to the primary key attribute of the other table. It further means that if any attribute is set as a primary key attribute will work in another table as a foreign key attribute.
- The use of a foreign key is simply to link the attributes of two tables together with the help of a primary key attribute. it is used for creating and maintaining the relationship between the two relations.

##### **○ Examples of Foreign Key**

###### **Example -1:**

- Consider two tables: **Orders** and **Customers**.

**Customers Table:**

<b>CustomerID</b>	<b>Name</b>	<b>Email</b>	<b>Phone</b>
1	Kavita Joshi	kav23@gmail.com	9574948439
2	Manav Patel	man43@gmail.com	6849403853



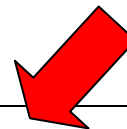
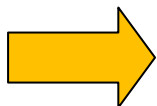
**Orders Table:**

OrderID	CustomerID	OrderDate	TotalAmount
101	1	2024-06-1	200.00
102	2	2024-06-16	150.00
103	1	2024-07-01	300.00

- CustomerID is a foreign key that references CustomerID in the Customers table.
- This relationship ensures that every CustomerID in the Orders table must exist in the Customers table.

**Example -2: Employees and Departments****Employees Table:**

Employee ID	FirstName	LastName	DepartmentID	Position	Salary
1	Nikita	Dave	101	Manager	70000.00
2	Manav	Patel	102	Analyst	30000.00

**Foreign Key****Departments Table:****Primary key**

DepartmentID	DepartmentName
101	IOS
102	Android

**In this scenario:**

- The Employees table has a column DepartmentID which is a foreign key referencing the DepartmentID column in the Departments table.
- This relationship indicates that each employee belongs to a specific department defined in the Departments table.

## 5. Alternate Key

### ○ What is an Alternate Key?

- An alternate is a secondary candidate key that is capable of identifying a row uniquely.
- However, such a key is not used as a primary key because, as we have discussed in our previous section that out of all the generated candidate keys, only one key is selected as the primary key.
- And the other remaining keys are known as Alternate Keys or Secondary Keys.
- An alternate key is none other than a candidate key, so the use/role of an alternate key is the same. It means an alternate key is also used to identify those columns in a table that can uniquely identify all the records of the table.

### ○ Example of Alternate Key

- Let's consider an EMPLOYEE\_DETAIL table example where we have the following attribute:

**Emp\_SSN:** The SSN number is stored in this field.

**Emp\_Id:** An attribute that stores the value of the employee identification number.

**Emp\_name:** An attribute that stores the name of the employee holding the specified employee id.

**Emp\_email:** An attribute that stores the email id of the specified employees.

Emp_SSN	Emp_Id	Emp_name	Emp_email
11051	01	John	john@email.com
11052	02	Merry	merry@email.com
19801	03	Riddle	riddle@email.com
41201	04	Cary	cary@email.com

## **Exercise: Employee and Department Tables**

### **Employee Table**

<b>Column Name</b>	<b>Data Type</b>
emp_id	INT
first_name	VARCHAR(50)
last_name	VARCHAR(50)
email	VARCHAR(100)
phone_number	VARCHAR(20)
hire_date	DATE
salary	INT
dept_id	INT

### **Department Table**

<b>Column Name</b>	<b>Data Type</b>
dept_id	INT
dept_name	VARCHAR(50)
location	VARCHAR(100)

### **Tasks:**

**1. Define the Primary Keys:**

- Identify the primary keys for both the employee and department tables.

**2. Define the Foreign Key:**

- Identify the foreign key in the employee table and specify the referenced table and column.

**3. Define Candidate Keys:**

- Identify all candidate keys for the employee table.
- Identify all candidate keys for the department table.

**4. Define Super Keys:**

- Identify all super keys for the employee table.
- Identify all super keys for the department table.

### Solution:

1. **Primary Keys:**

- **employee table:** emp\_id
- **department table:** dept\_id

2. **Foreign Key:**

- **employee table:** dept\_id references departments(dept\_id)

3. **Candidate Keys:**

- **employee table:** emp\_id, email
- **department table:** dept\_id,

4. **Super Keys:**

- **employee table:** {emp\_id}, {email}, {emp\_id, email}, {emp\_id, first\_name}, {emp\_id, last\_name}, {emp\_id, phone\_number}, {emp\_id, hire\_date}, {emp\_id, salary}, {emp\_id, dept\_id, ...}
- **department table:** {dept\_id}, {dept\_name}, {dept\_id, dept\_name}, {dept\_id, location}, {dept\_id, dept\_name, location, ...},