



Database Keys

- **What are keys in DBMS?**

In DBMS, keys are attributes (columns) or sets of attributes that help to identify a row (tuple) in a table (relation). Keys help uniquely identify a row in a table by a combination of one or more columns in that table. They help retrieve data from the table and also establish a relation between tables.

Example:-

Employee ID	Employee Name	Department
1	Amayra	HR
2	Deepa	Data Analyst
3	Harshita	Software

In the above example, Employee ID is a primary key because it uniquely identifies an employee record. In this table, no other employee can have the same employee ID.

- **Why do we need Keys?**

In a nutshell we choose a Relational Database Management System (RDBMS) over a regular file system because we want a database that's organised and doesn't have unnecessary repetition. We achieve this organisation through a method called normalisation. This technique helps us tidy up the database by removing unnecessary duplication and sorting out issues like data deletion or insertion.

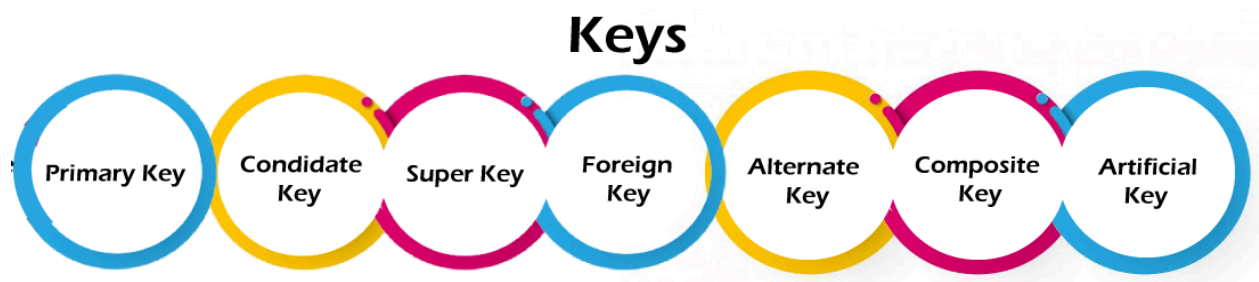
Normalisation is used for mainly two purposes:

- Eliminating redundant(useless) data.
- Making sure that the way data is connected makes logical sense.

We achieve this normalisation by using "keys".

- Keys help you to identify any row of data in a table, even when dealing with a large number of records in real-world applications.
- It allows you to establish a relationship and identify the relation between tables.
- Ensures the identity and integrity of relationships.

- **Types Of Keys:**



Mainly, there are 7 types of keys in DBMS. They are as follows:

1. Super Key
2. Candidate key
3. Primary Key
4. Alternate Key
5. Composite Key
6. Surrogate Key
7. Foreign Key
8. compound key

1. Super Key:

A super key is a set of attributes (columns) that can uniquely identify records within a table. A candidate key is a minimal super key. In addition to the attributes of the candidate key, a super key can include extra attributes (columns).

Example :-

Roll No.	Name	Branch	Email
1	Neha	CSE	neha@gmail.com
2	Deepika	EEE	deepika@gmail.com
3	Harsh	ME	harsh@gmail.com

Here, Email and Roll No. are candidate keys as they can alone uniquely identify records.

Email

Roll No + Name

Roll No + Branch

Roll No + Email

Name + Branch + Email

Roll No + Branch + Name

Roll No + Branch + Name + Email

Each of these combinations can uniquely identify a student in the table. When you put all these combinations together, it's called the **super key set**.

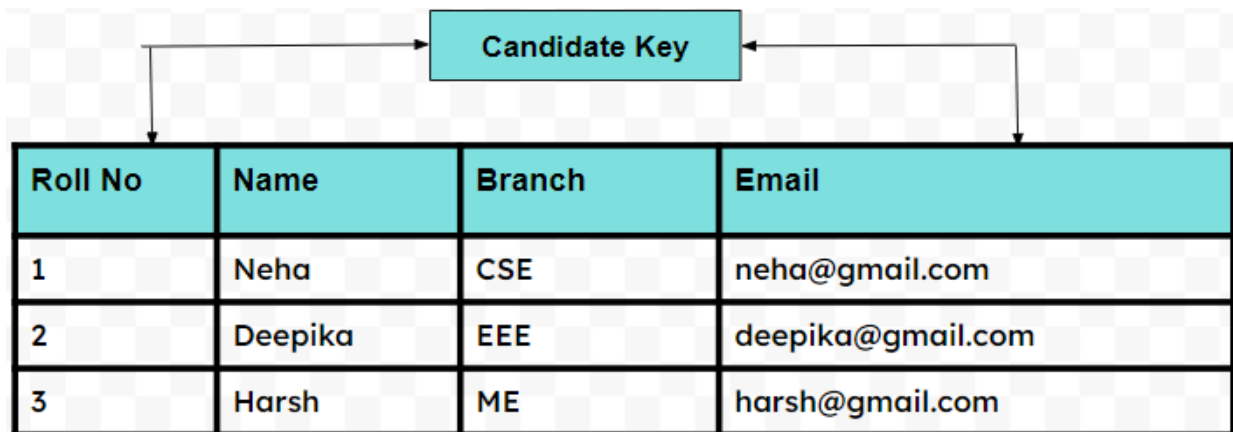
Why is "Name + Branch" not a Super Key?

A super key is not a combination of any elements (columns); a super key has at least one candidate key. So, it's not a super key because, to be a super key, it needs the candidate key (Roll No or Email). Super keys can have extra attributes along with the candidate key.

2. Candidate Key

A candidate key is a set of attributes (columns) in a table that can uniquely identify each record without any redundancy. In other words, It's a minimal super key.

Example:-



Roll No is a candidate key (CK) because it's minimal and sufficient to uniquely identify each record.

Email is also a candidate key (CK) for the same reason.

On the other hand, Roll No + Name is not a candidate key (not CK) because Roll No alone is sufficient to uniquely identify a record; the inclusion of Name is redundant or unnecessary.

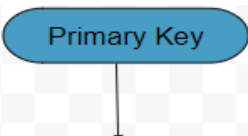
The same logic applies to all combinations like Roll No + Name, Roll No + Branch, Email + Name, etc.

So, from all the potential super keys, those that are sufficient on their own (like Roll No and Email) to uniquely identify each record are termed as candidate keys.

3. Primary Key

The primary key is the heart of a database table. A primary key is a unique identifier for each row in a database table, making sure there are no repeated values. There can only be one primary key in a table, and it cannot contain null values.

Example:



Roll No	Name	Branch	Email
1	Neha	CSE	neha@gmail.com
2	Deepika	EEE	deepika@gmail.com
3	Harsh	ME	harsh@gmail.com

Mandatory Criteria for Primary Key:

- Cannot be NULL
- Must be unique (no duplicates)

In this table, both Roll No and Email meet these criteria, so both could be primary keys.

but, according to the definition, a table can have only one primary key.

To decide between Roll No and Email, we consider additional "good to have" criteria:

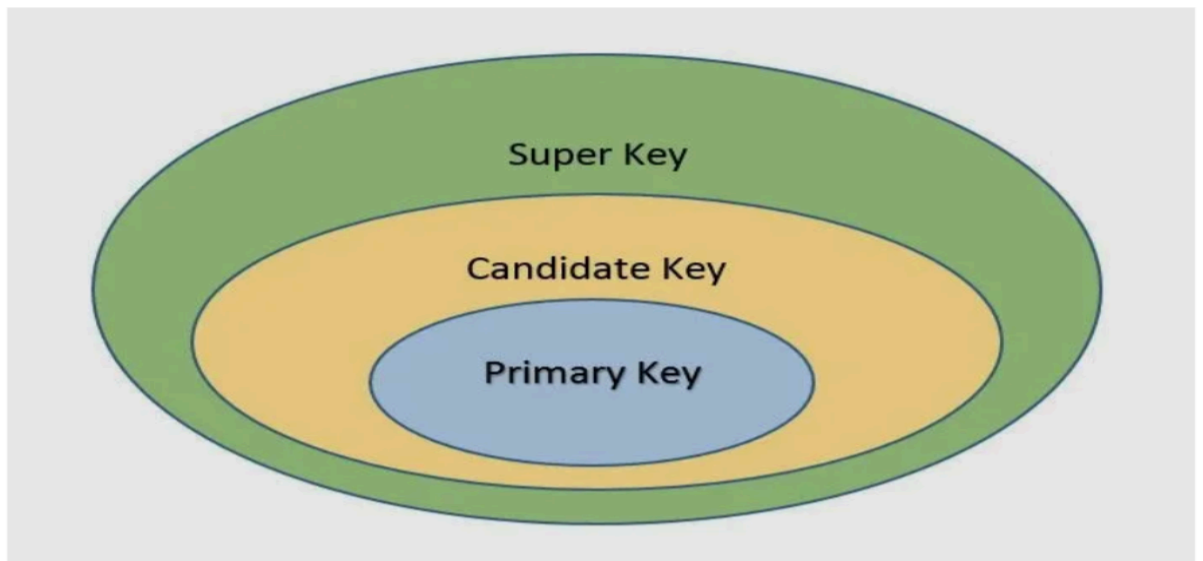
Good to Have Criteria:

Numerical: Roll No is often numeric or alphanumeric, while Email is alphabetic. As the primary key is commonly numeric, Roll No aligns better with this criterion.

Small: Roll No is generally smaller in size than Email, making it more efficient for indexing and storage.

Consistent: Roll No is likely to remain constant over time, while Email may change.

Considering these criteria, Email fails in smallness and constancy. Roll No emerges as the more suitable choice for the primary key, meeting both the mandatory and "good to have" conditions. Therefore, Roll No is selected as the primary key for this table.



4. Alternate Key

An alternate key is a candidate key that is not used as the primary key.

Example:

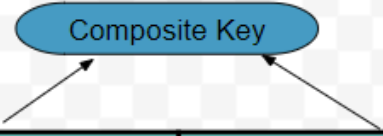
Primary Key		Alternate Key	
Roll No	Name	Branch	Email
1	Neha	CSE	neha@gmail.com
2	Deepika	EEE	deepika@gmail.com
3	Harsh	ME	harsh@gmail.com

In our case "Roll No" and "Email " are both candidate keys and "Roll No" is chosen as the primary key, then "Email" becomes the alternate key.

5. Composite Key

A composite key is a primary key that is made up of two or more attributes. Composite keys are used when a single attribute is not sufficient to uniquely identify a tuple in a table.

Example:



Student_ID	Course_ID	Date	Payment
101	Python	2022-3-15	Online
102	SQL	2022-3-16	Offline
103	Power BI	2022-3-16	Offline
101	Python	2022-3-17	Online
104	SQL	2022-4-12	Online

student_id: Can't be a primary key as one student can enrol in multiple courses.

course_id: Can't be a primary key as one course can have multiple students.

date and payment: Can have duplicates.

In this table, a single attribute like "Student_ID" or "Course_id" is not sufficient to uniquely identify a tuple. Therefore, a composite key, in this case, consists of the combination of "Student_ID" and "Course_id." This combination ensures uniqueness and is used as the composite key to identify each record uniquely in the table.

6. Surrogate Key:

A surrogate key is a unique identifier generated by the database management system (DBMS), not derived from the actual data. It is often used as the primary key in database tables.

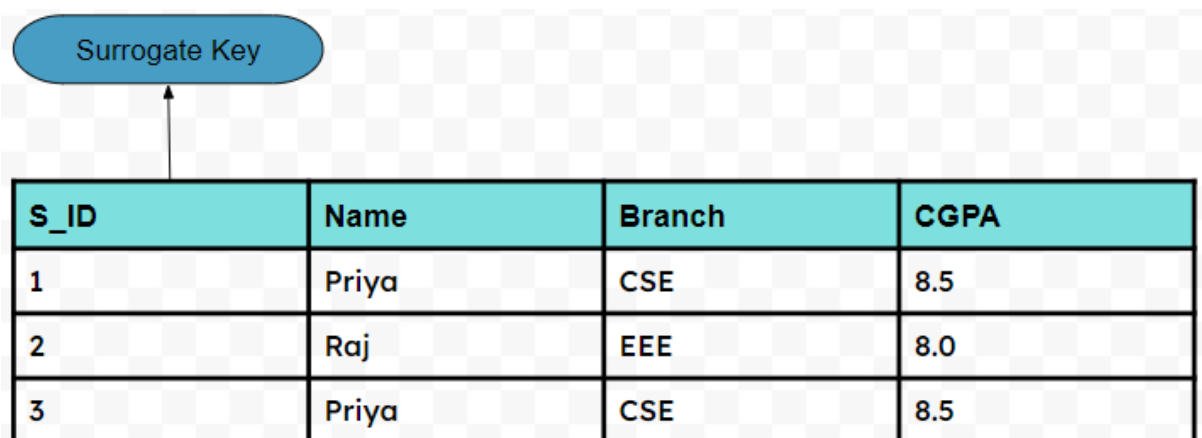
Example:

Name	Branch	CGPA
Priya	CSE	8.5
Raj	EEE	8.0
Priya	CSE	8.5

In this table, which column can become the primary key?

Neither 'Name,' 'Branch,' nor 'CGPA' can be the primary key, as they can be repeated. Even combinations like 'Name + Branch,' 'Branch + CGPA,' 'Name + Branch + CGPA,' etc., cannot serve as the primary key.

When there is no natural primary key in a table, we need to create one artificially to uniquely identify a row. This key is known as the surrogate key or synthetic primary key of the table.

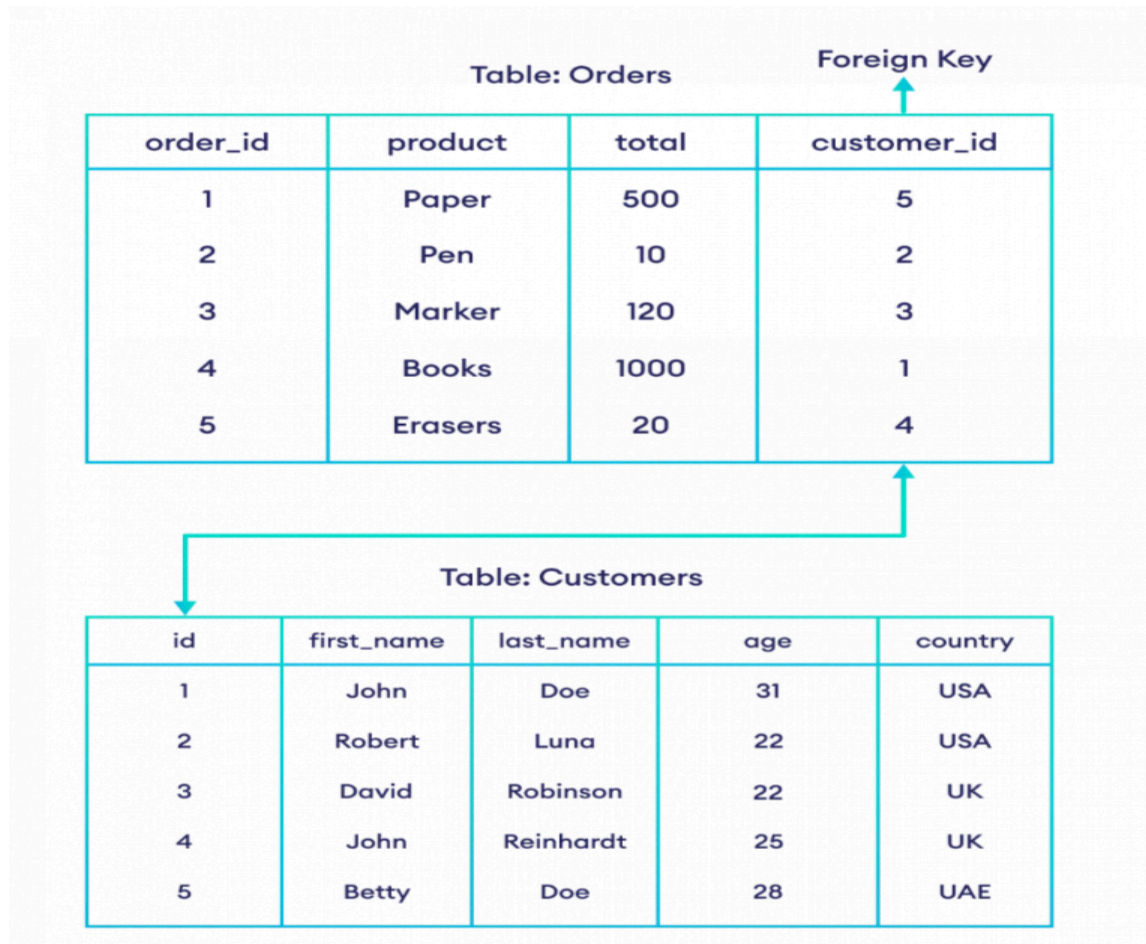


Here, 'S_ID' serves as the surrogate key, artificially created for this purpose.

7. Foreign Key

A foreign key is a primary key from one table that is used to establish a relationship with another table.

Example:



Consider two tables - "Customers" and "Orders." In the "Orders" table, the "customer_id" column is a foreign key that references the primary key "id" in the "Customers" table.

This foreign key connection ensures that each entry in the "Orders" table is associated with a unique customer from the "Customers" table.

Conclusion:

1. Super Key:

Identifies records uniquely. Candidate key is a minimal version.

2. Candidate Key:

Uniquely identifies records with no redundancy.

3. Primary Key

Unique identifier for each row. Only one per table, no null values.

4. Alternate Key

Candidate key not chosen as primary.

5. Composite Key

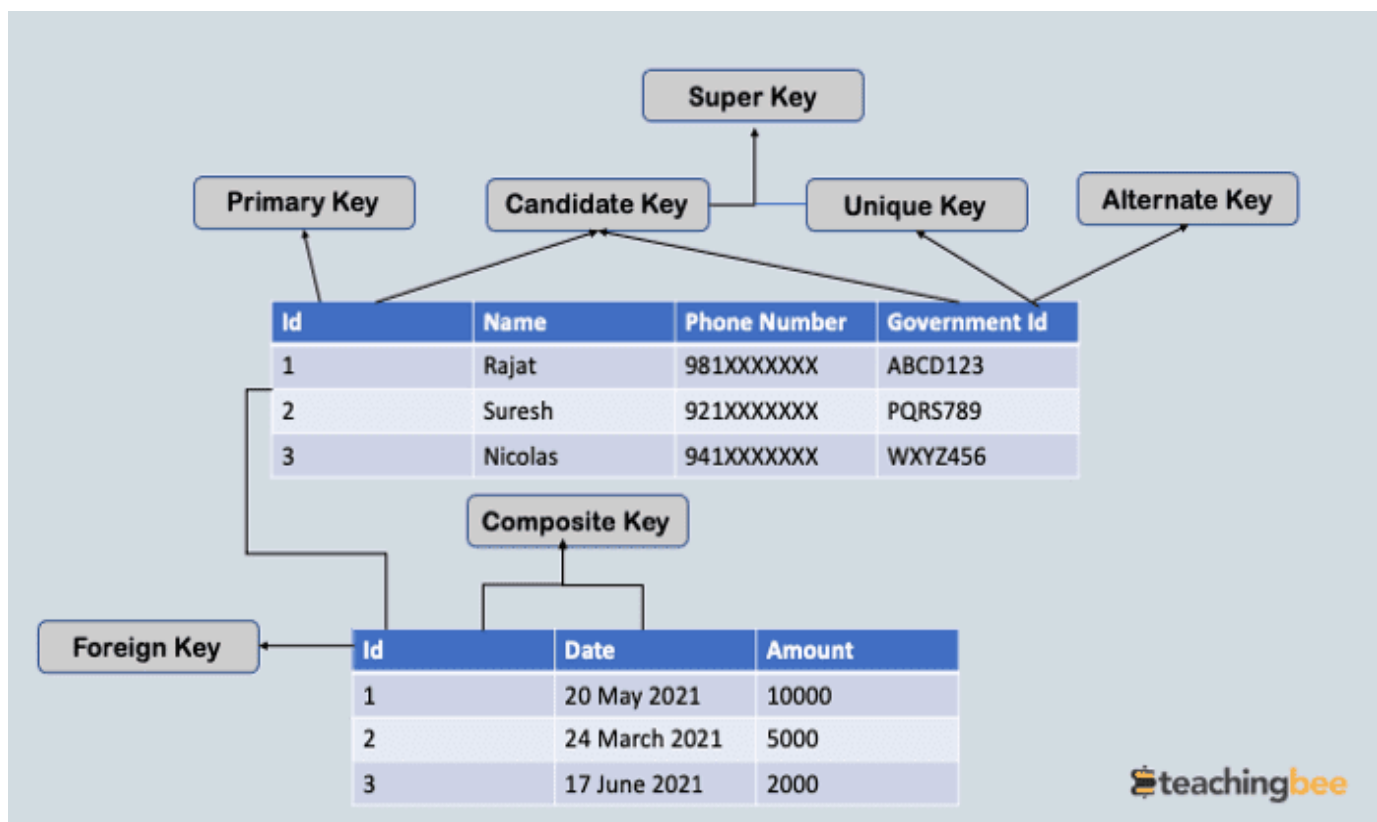
Combines two or more attributes. Used when a single attribute is insufficient.

6. Surrogate Key

Generated by DBMS, not from data. Often used as the primary key.

7. Foreign Key

Links to the primary key in another table. Establishes relationships between tables.



Thank you 🌻

◆ Presented By Deepa

