**Keys in Relational Model**

In the context of a relational database, keys are one of the basic requirements of a relational database model.

* Keys are fundamental components that ensure data integrity, uniqueness and efficient access. It is widely used to identify the tuples(rows) uniquely in the table.
* We also use keys to set up relations amongst various columns and tables of a relational database.
* Keys in DBMS

**Why do we require Keys in a DBMS?**

Keys are important in a Database Management System (DBMS) for several reasons:

* Uniqueness: Keys ensure that each record in a table is unique and can be identified distinctly.
* Data Integrity: Keys prevent data duplication and maintain the consistency of the data.
* Efficient Data Retrieval: By defining relationships between tables, keys enable faster querying and better data organization. Without keys, it would be extremely difficult to manage large datasets and queries would become inefficient and prone to errors.

**Types of Database Keys**

**1. Super Key**

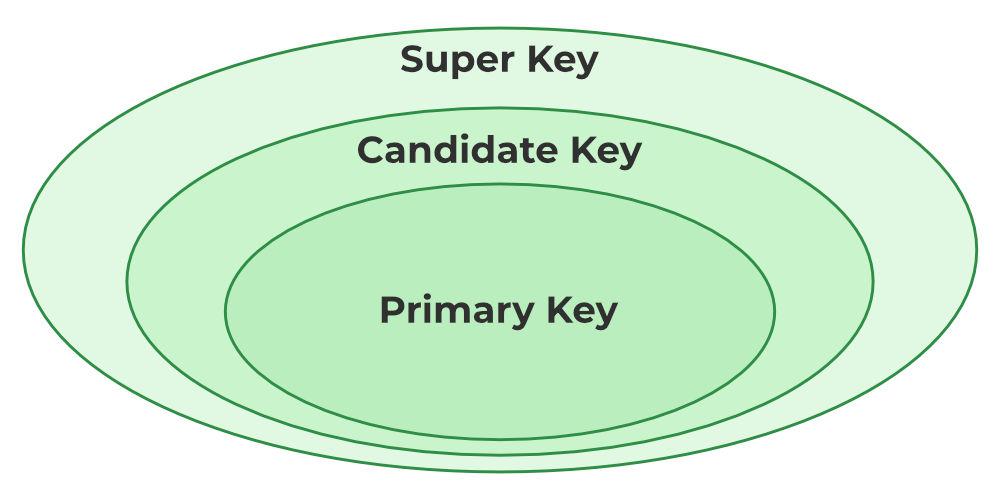
The set of one or more attributes (columns) that can uniquely identify a tuple (record) is known as [Super Key](https://www.geeksforgeeks.org/dbms/super-key-in-dbms/). It may include extra attributes that aren't important for uniqueness but still uniquely identify the row. For Example, STUD\_NO, (STUD\_NO, STUD\_NAME), etc.

* A super key is a group of single or multiple keys that uniquely identifies rows in a table. It supports NULL values in rows.
* A super key can contain extra attributes that aren’t necessary for uniqueness.
* For example, if the "STUD\_NO" column can uniquely identify a student, adding "SNAME" to it will still form a valid super key, though it's unnecessary.

**Example:** Consider the STUDENT table

| **STUD\_NO** | **SNAME** | **ADDRESS** | **PHONE** |
| --- | --- | --- | --- |
| 1 | Shyam | Delhi | 123456789 |
| 2 | Rakesh | Kolkata | 223365796 |
| 3 | Suraj | Delhi | 175468965 |

A super key could be a combination of STUD\_NO and PHONE, as this combination uniquely identifies a student.

Relation between Primary Key, Candidate Key, and Super Key

**2. Candidate Key**

The minimal set of attributes that can uniquely identify a tuple is known as a [candidate key](https://www.geeksforgeeks.org/dbms/candidate-key-in-dbms/). For Example, STUD\_NO in STUDENT relation.

* A candidate key is a minimal super key, meaning it can uniquely identify a record but contains no extra attributes.
* It is a super key with no repeated data is called a candidate key.
* The minimal set of attributes that can uniquely identify a record.
* A candidate key must contain unique values, ensuring that no two rows have the same value in the candidate key’s columns.
* Every table must have at least a single candidate key.
* A table can have multiple candidate keys but only one primary key.

**Example:** For the STUDENT table below, STUD\_NO can be a candidate key, as it uniquely identifies each record.

| **STUD\_NO** | **SNAME** | **ADDRESS** | **PHONE** |
| --- | --- | --- | --- |
| 1 | Shyam | Delhi | 123456789 |
| 2 | Rakesh | Kolkata | 223365796 |
| 3 | Suraj | Delhi | 175468965 |

**Table:**STUDENT\_COURSE

| **STUD\_NO** | **TEACHER\_NO** | **COURSE\_NO** |
| --- | --- | --- |
| 1 | 001 | C001 |
| 2 | 056 | C005 |

A composite candidate key example: {STUD\_NO, COURSE\_NO} can be a candidate key for a STUDENT\_COURSE table.

**3. Primary Key**

There can be more than one candidate key in relation out of which one can be chosen as the primary key. For Example, STUD\_NO, as well as STUD\_PHONE, are candidate keys for relation STUDENT but STUD\_NO can be chosen as the [primary key](https://www.geeksforgeeks.org/dbms/primary-key-in-dbms/)(only one out of many candidate keys).

* A primary key is a unique key, meaning it can uniquely identify each record (tuple) in a table.
* It must have unique values and cannot contain any duplicate values.
* A primary key cannot be NULL, as it needs to provide a valid, unique identifier for every record.
* A primary key does not have to consist of a single column. In some cases, a composite primary key (made of multiple columns) can be used to uniquely identify records in a table.
* Databases typically store rows ordered in memory according to primary key for fast access of records using primary key.

**Example:**

*STUDENT table -> Student(STUD\_NO, SNAME, ADDRESS, PHONE) , STUD\_NO is a primary key*

**Table:** STUDENT

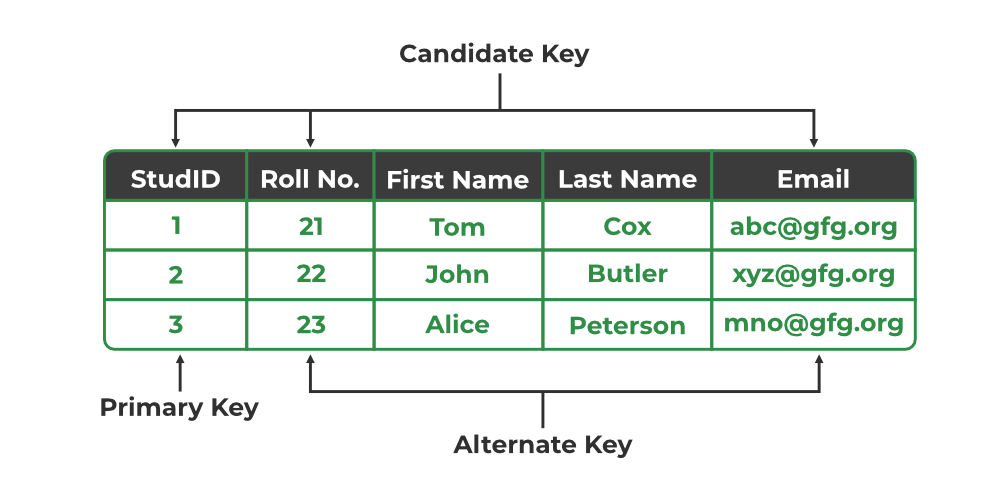
| **STUD\_NO** | **SNAME** | **ADDRESS** | **PHONE** |
| --- | --- | --- | --- |
| 1 | Shyam | Delhi | 123456789 |
| 2 | Rakesh | Kolkata | 223365796 |
| 3 | Suraj | Delhi | 175468965 |

**4. Alternate Key**

An [alternate key](https://www.geeksforgeeks.org/dbms/alternate-key-in-dbms/) is any candidate key in a table that is not chosen as the primary key. In other words, all the keys that are not selected as the primary key are considered alternate keys.

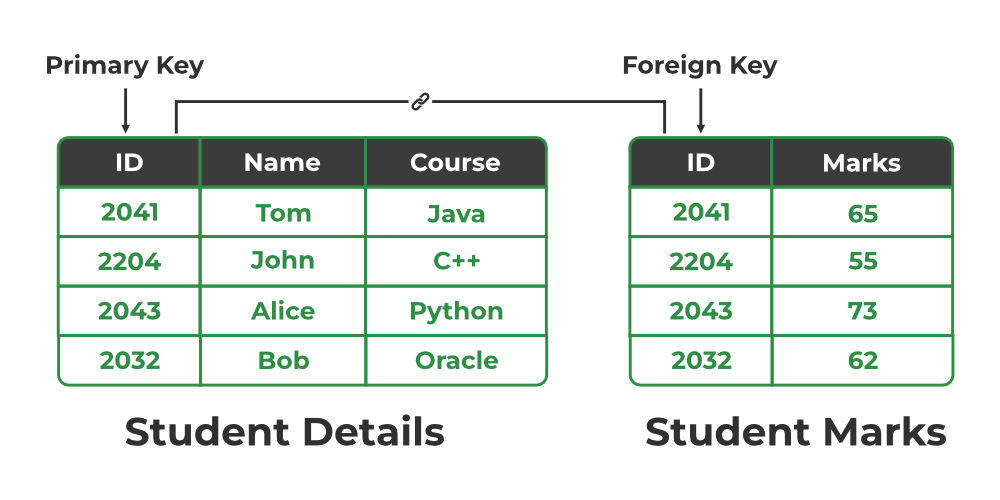
* An alternate key is also referred to as a secondary key because it can uniquely identify records in a table, just like the primary key.
* An alternate key can consist of one or more columns (fields) that can uniquely identify a record, but it is not the primary key

**Example:** In the STUDENT table, both STUD\_NO and PHONE are candidate keys. If STUD\_NO is chosen as the primary key, then PHONE would be considered an alternate key.

Primary Key, Candidate Key, and Alternate Key

**5. Foreign Key**

A [foreign key](https://www.geeksforgeeks.org/sql/foreign-key-constraint-in-sql/) is an attribute in one table that refers to the primary key in another table. The table that contains the foreign key is called the referencing table and the table that is referenced is called the referenced table.

Relation between Primary Key and Foreign Key

* A foreign key in one table points to the primary key in another table, establishing a relationship between them.
* It helps connect two or more tables, enabling you to create relationships between them. This is important for maintaining data integrity and preventing data redundancy.
* They act as a cross-reference between the tables.

**Example:** Consider the STUDENT\_COURSE table

| **STUD\_NO** | **TEACHER\_NO** | **COURSE\_NO** |
| --- | --- | --- |
| 1 | 005 | C001 |
| 2 | 056 | C005 |

**Explanation:**

* Here, STUD\_NO in the STUDENT\_COURSE table is a foreign key that references the STUD\_NO primary key in the STUDENT table.
* Unlike the Primary Key of any given relation, Foreign Key can be NULL as well as may contain duplicate tuples i.e. it need not follow uniqueness constraint. For Example, STUD\_NO in the STUDENT\_COURSE relation is not unique.
* It has been repeated for the first and third tuples. However, the STUD\_NO in STUDENT relation is a primary key and it needs to be always unique and it cannot be null.

**6. Composite Key**

Sometimes, a table might not have a single column/attribute that uniquely identifies all the records of a table. To uniquely identify rows of a table, a combination of two or more columns/attributes can be used. It still can give duplicate values in rare cases. So, we need to find the optimal set of attributes that can uniquely identify rows in a table.

* It acts as a primary key if there is no primary key in a table
* Two or more attributes are used together to make a[composite key](https://www.geeksforgeeks.org/dbms/composite-key-in-database/).
* Different combinations of attributes may give different accuracy in terms of identifying the rows uniquely.

**Example:**In the STUDENT\_COURSE table, {STUD\_NO, COURSE\_NO} can form a composite key to uniquely identify each record.