Cardinality

In SQL (Structured Query Language), the term cardinality refers to the uniqueness of data values contained in a particular column (attribute) of a database table. The lower the cardinality, the more duplicated elements in a column. Thus, a column with the lowest possible cardinality would have the same value for every row. However cardinality conjointly sometimes refers to the relationships between tables. Cardinality between tables is often one-to-one, many-to-one or many-to-many.

When dealing with columnar value sets, there are three types of cardinality: **high-cardinality, normal-cardinality, and low-cardinality.**

High-cardinality refers to columns with values that are very uncommon or unique. High-cardinality column values are typically identification numbers, email addresses, or user names. An example of a data table column with high-cardinality would be a USERS table with a column named USER_ID. This column would contain unique values of 1-n. Each time a new user is created in the USERS table, a new number would be created in the USER_ID column to identify them uniquely. Since the values held in the USER_ID column are unique, this column's cardinality type would be referred to as high-cardinality.

Normal-cardinality refers to columns with values that are somewhat uncommon. Normal-cardinality column values are typically names, street addresses, or vehicle types. An example of a data table column with normal-cardinality would be a CUSTOMER table with a column named LAST_NAME, containing the last names of customers. While some people have common last names, such as Smith, others have uncommon last names. Therefore, an examination of all of the values held in the LAST_NAME column would show "clumps" of names in some places (e.g. a lot of Smiths) surrounded on both sides by a long series of unique values. Since there is a variety of possible values held in this column, its cardinality type would be referred to as normal-cardinality.

Low-cardinality refers to columns with few unique values. Low-cardinality column values are typically status flags, Boolean values, or major classifications such as gender. An example of a data table column with low-cardinality would be a CUSTOMER table with a column named NEW_CUSTOMER. This column would contain only two distinct values: Y or N, denoting whether the customer was new or not. Since there are only two possible values held in this column, its cardinality type would be referred to as low-cardinality.

What are Keys?

A DBMS key is an attribute or set of an attribute which helps you to identify a row (tuple) in a relation (table). They allow you to find the relation between two tables. Keys help you uniquely identify a row in a table by a combination of one or more columns in that table.

Roll Number	Student Name	Father Name
11	Muhammad Ali	Farooq Ali
22	Imran Khan	Rehman Khan
33	Feroz Ahmed	Saleem Ahmed

In the above-given example, Roll Number is a primary key because it uniquely identifies a Student record. In this table, no other students can have the same roll number in class.

Why we need a Key?

Here, are reasons for using Keys in the DBMS system.

- Keys help you to identify any row of data in a table. In a real-world application, a table could contain thousands of records. Moreover, the records could be duplicated. Keys 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
- Help you to enforce identity and integrity in the relationship.

Various Keys in Database Management System

DBMS has following seven types of Keys each have their different functionality:

- Super Key
- Primary Key
- Candidate Key
- Alternate Key
- Foreign Key
- Compound Key
- Composite Key

Emp-NIC	Emp ID	Emp Name
9812-34509899-2	AB05	Saleem
9876-51234522-1	AB06	Feroz
1999-37890988-9	AB07	Jamal

In the above-given example, Emp-NIC and Emp Id are superkeys.

What is a Primary Key?

A column or group of columns in a table which helps us to uniquely identify every row in that table is called a primary key. This DBMS can't be a duplicate. The same value can't appear more than once in the table.

Rules for defining Primary key:

- Two rows can't have the same primary key value
- It must for every row to have a primary key value.
- The primary key field cannot be null.
- The value in a primary key column can never be modified or updated if any foreign key refers to that primary key.

Example:

In the following example, StudID is a Primary Key.

StudID	Roll No	First Name	Last Name	Email
1	11	Asif	Khan	abc@email.com
2	12	Nadeem	Ahmed	xyz@email.com
3	13	Basit	Farooqi	aaa@email.com

What is the Alternate key?

All the keys which are not primary key are called an alternate key. It is a candidate key which is currently not the primary key. However, a table may have single or multiple choices for the primary key.

Example: In this table. StudID, Roll No, Email are qualified to become a primary key. But since StudID is the the following example, StudID is a Primary Key.

StudID	Roll No	First Name	Last Name	Email
1	11	Asif	Khan	abc@email.com
2	12	Nadeem	Ahmed	xyz@email.com
3	13	Basit	Farooqi	aaa@email.com

What is a Candidate Key?

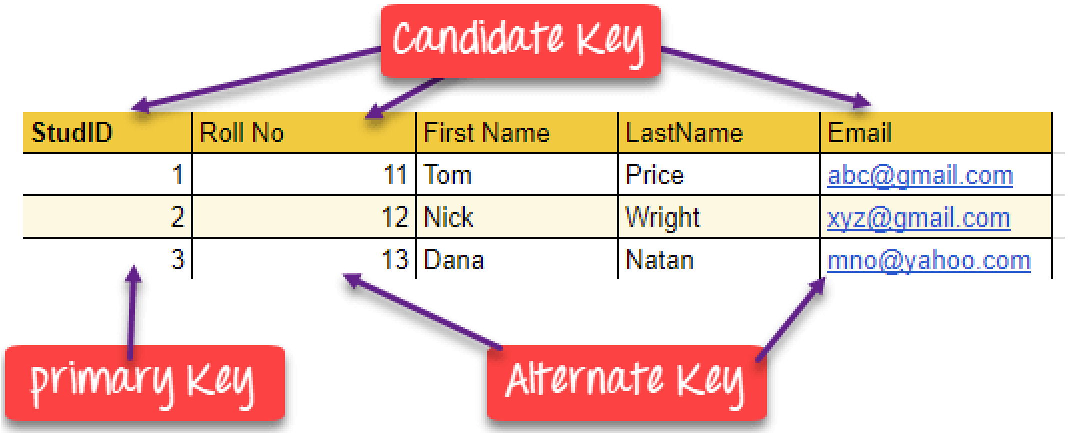
A super key with no repeated attribute is called candidate key. The Primary key should be selected from the candidate keys. Every table must have at least a single candidate key.

Properties of Candidate key:

- It must contain unique values
- Candidate key may have multiple attributes
- Must not contain null values
- It should contain minimum fields to ensure uniqueness
- Uniquely identify each record in a table

Example: In the given table Stud ID, Roll No, and email are candidate keys which help us to uniquely identify the student record in the table. The following example, StudID is a Primary Key.

StudID	Roll No	First Name	Last Name	Email
1	11	Asif	Khan	abc@email.com
2	12	Nadeem	Ahmed	xyz@email.com
3	13	Basit	Farooqi	aaa@email.com



StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

Candidate Key
primary Key
Alternate Key

Foreign key

A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It acts as a cross-reference between tables because it references the primary key of another table, thereby establishing a link between them.

The majority of tables in a relational database system adhere to the foreign key concept. In complex databases and data warehouses, data in a domain must be added across multiple tables, thus maintaining a relationship between them. The concept of referential integrity is derived from foreign key theory. Foreign keys and their implementation are more complex than primary keys.

For any column acting as a foreign key, a corresponding value should exist in the link table. Special care must be taken while inserting data and removing data from the foreign key column, as a careless deletion or insertion might destroy the relationship between the two tables.

For instance, if there are two tables, customer and order, a relationship can be created between them by introducing a foreign key into the order table that refers to the customer ID in the customer table. The customer ID column exists in both customer and order tables. The customer ID in the order table becomes the foreign key, referring to the primary key in the customer table. To insert an entry into the order table, the foreign key constraint must be satisfied. An attempt to enter a customer ID that is not present in the customer table fails, thus maintaining the table's referential integrity.

Department Table

DeptCode	DeptName
001	CIT
002	SE
005	Electronics

Teacher Table

TeacherID	Name	Subject
B01	Asif Khan	Maths
B02	Muhammad Ali	English
B03	Adnana Ahmed	Urdu

In this example, we have two table, teacher and department in an Institute. However, there is no way to see which search work in which department.

In this table, adding the foreign key in Deptcode to the Teacher name, we can create a relationship between the two tables.

Department Table

DeptCode	DeptName
001	CIT
002	SE
005	Electronics

Teacher Table

TeacherID	Name	Subject	DeptCode
B01	Asif Khan	Maths	001
B02	Muhammad Ali	English	002
B03	Adnana Ahmed	Urdu	005

The rule that a value of a foreign key must appear as a value of some specific table is called a referential constraint (Referential integrity constraint is a concerned with foreign key)

Primary Key

- Helps you to uniquely identify a record in the table.
- Primary Key never accept null values.
- Primary key is a clustered index and data in the DBMS table are physically organized in the sequence of the clustered index.
- You can have the single Primary key in a table.

Foreign Key

- It is a field in the table that is the primary key of another table.
- A foreign key may accept multiple null values.
- A foreign key cannot automatically create an index, clustered or non-clustered. However, you can manually create an index on the foreign key.
- You can have multiple foreign keys in a table.

Compound key

Compound key has many fields which allow you to uniquely recognize a specific record. It is possible that each column may be not unique by itself within the database. However, when combined with the other column or columns the combination of composite keys become unique.

OrderNo	PorductID	Product Name	Quantity
B005	JAP102459	Mouse	5
B005	DKT321573	USB	10
B005	OMG446789	LCD Monitor	20
B004	DKT321573	USB	15
B002	OMG446789	Laser Printer	3

In this example, OrderNo and ProductID cannot be a primary key as it does not uniquely identify a record. However, a compound key of Order ID and Product ID could be used as it uniquely identified each record.

Composite Key.

A key which has multiple attributes to uniquely identify rows in a table is called a composite key. The difference between compound and the composite key is that any part of the compound key can be a foreign key, but the composite key may or maybe not a part of the foreign key.

How to create Primary Key or Super Key in the table.

Syntax

CREATE TABLE <TableName> (ColumName1 type, columName2 type, … , ***PRIMARY KEY*** (CulumnName));

Example

```
mysql>CREATE TABLE Students (
   Roll int NOT NULL,
   Name varchar(20) NOT NULL,
   FatherName varchar(25) NULL,
   Gender text(13) NULL,
   Technology varchar(20) NULL,
    PRIMARY KEY (Roll)
     );
```

How to add Primary Key in existing table.

```
Mysql>ALTER TABLE Students ADD PRIMARY KEY (Roll);
```

How to create Alternate or Candidate Key in the table.

Syntax

```
CREATE TABLE <TableName> (ColumName1 type UNIQUE, columName2 type, … , PRIMARY KEY
(CulumnName) );
```

Example

```
mysql>CREATE TABLE Students (
    Roll int NOT NULL,
    Name varchar(20) NOT NULL,
    ContactNo varchar(15) NOT NULL UNIQUE,
    Gender text(13) NOT NULL,
     NICNo varchar(18) NOT NULL UNIQUE,
    Technology varchar(20) NULL,
     PRIMARY KEY (Roll)
      );
```

How to add Primary Key in existing table.

```
Mysql> ALTER TABLE Students ADD PRIMARY KEY (Roll) ;
```

How to Create FOREIGN KEY in MySQL.

In this example that the "Roll" column in the "Students" table. The "Roll" column in the "Students" table is the PRIMARY KEY in the "Students" table. The "Roll" column in the "Marksheet" table is a FOREIGN KEY in the "Marksheet" table.

```
Mysql>CREATE TABLE Marksheet (
            Id  int AUTO_INCREMENT,
            Roll int NOT NULL,
            Month Date,
            Maths int,
            English int,
            Urdu int,
            PRIMARY KEY (Id),
            FOREIGN KEY (Roll) REFERENCES Students(Roll)
);
```

How to add foreign key in existing table.

```
Mysql> ALTER TABLE Marksheet ADD FOREIGN KEY (Roll) REFERENCES Students(Roll);
```

How to remove foreign key in existing table.

```
Mysql>ALTER TABLE Marksheet DROP FOREIGN KEY Roll;
```

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables. The FOREIGN KEY constraint also prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the table it points to.

How to Create Composite KEY in MySQL

Syntax

```
CREATE TABLE SAMPLE_TABLE
(COL1 integer,
 COL2 varchar(30),
 COL3 varchar(50),
PRIMARY KEY (COL1, COL2));
```

Example

```
mysql>CREATE TABLE Students (
    Roll int NOT NULL,
    Name varchar(20) NOT NULL,
    ContactNo varchar(15) NOT NULL,
    Gender text(13) NOT NULL,
     NICNo varchar(18) NOT NULL,
    Technology varchar(20) NULL,
     PRIMARY KEY (Roll,NICNo)
      );
```

Exercise

Theory Question

1) What is cardinality and its types?
2) How many type of relationship with in the tables in RDBMS?
3) Write list of keys in the tables (Concept of RDBMS).
4) How you can differentiate between the Composite and compound key?
5) What difference between the primary and foreign key.

Practical Question

1) Create table of students with their fields Roll number, Name, Father Name Contact, Gender and Technology with apply primary key on Roll number field.
2) Remove primary key to Roll Number Field in the table Students.
3) Add foreign key of Roll number field here Parent table Student and child table Marksheet in the existing tables.

Objective and MCQ's

1) Mapping cardinalities are useful in describing
   a) Unary relationships
   b) Binary relationships
   c) Composite relationships
   d) Simple relationships
2) Structure of primary key for relationship set depends on the
   a) Entity sets
   b) Roles
   c) Mapping cardinalities
   d) Mapping expression
3) The rule that a value of a foreign key must appear as a value of some specific is called a
   a) Referential constaint
   b) Index
   c) Integrity constraint
   d) Functional dependency
4) If the closure of an attribute set is the entire relation then the attribute set is a ________.
   a) Candidate key
   b) Super key
   c) Primary key
   d) No a key
5) A table can have only one __________.
   a) Secondary key
   b) Alternate key
   c) Primary key
   d) Unique key
6) In the relational modes, cardinality is term as:
   a) Number of tuples
   b) Number of attributes
   c) Number of table
   d) Number of constraints.
7) An entity set that does not have sufficient attributes to form a primary key is a ___
   a) Strong entity set

b) Weak entity set
c) Simple entity set
d) primary entity set

8) In case of entity integrity, the primary key may be _____
a) Not null
b) Null
c) Both null or not null
d) Nothing

9) Key to represt relationship between tables is called _______
a) Primary key
b) Secondary key
c) Foreign key
d) Not of these

10) A primary key is combined with a foreign key creates _____
a) Many to many relationship between the tables connect them.
b) Network model between the tables that connect them.
c) Parent-child relationship between the tables that connect them.
d) Nothing.