SQL - Relational Database

A database is a collection of data. A database could be as simple as a text file with a list of names. Or it could be as complex as a large, relational database management system, complete with in-built tools to help you maintain the data.

Methods for Storing Data

1. **Text File**

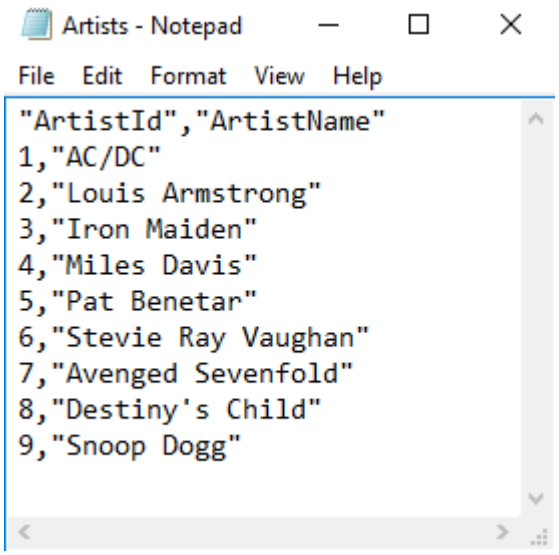
   Imagine we have a text file called Artists.csv, and that the contents look like this screenshot.

   This is a text file. More specifically, it's a comma separated values (CSV) file. The commas separate each field within a row.

   The commas give it structure. It enables us to distinguish the artist ID from the artist name. We could easily add more fields and separate them by more commas.

   Each row represents a different record. In this case, each row represents a different artist. Technically, this is a database.

   It contains data that is structured in a way that's easy to retreive. With a small list like this, a text file may serve our purposes perfectly.

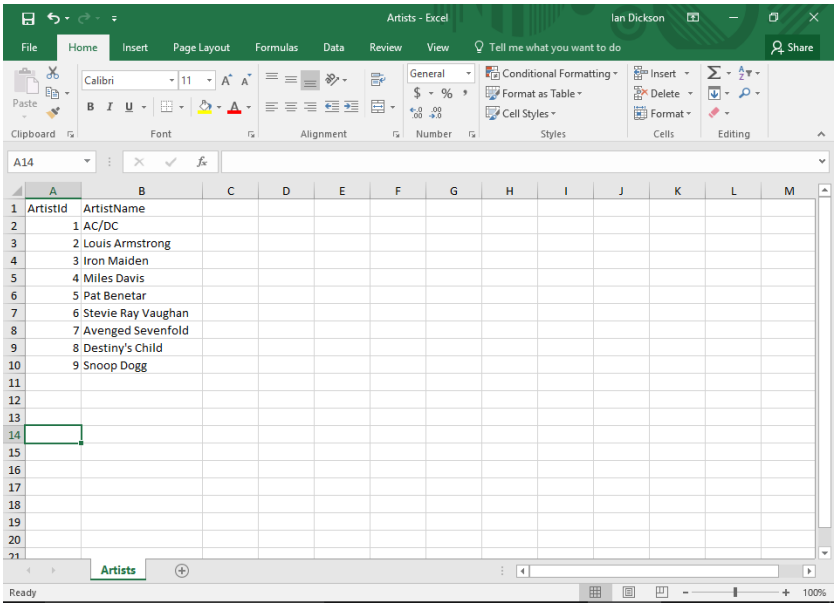


2. **Spreadsheet**

   Another option would be to store the data in a spreadsheet using spreadsheet software (for example, Microsoft Excel). That

way, we could do some extra things with our list (such as format it, or sort by artist name, etc).

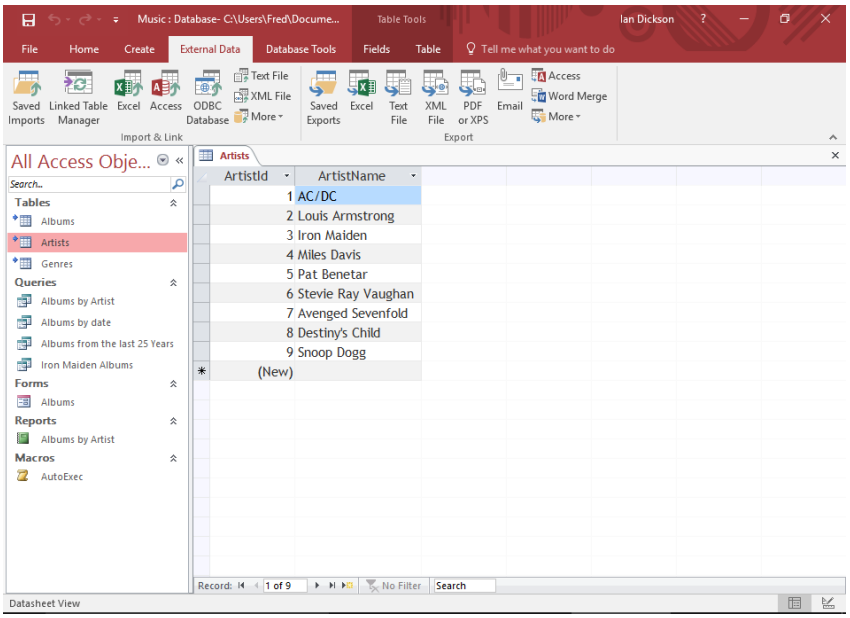
A spreadsheet program like Excel makes these tasks relatively easy to do. Also, programs like Excel organize the data into rows and columns, making your data easier to comprehend.



3. Database Software

A better option would be to store the data in a database table using specialized database software, such as Microsoft Access.

Database management systems like this are purpose built for data storage and retrieval.



A Database Management System (DBMS), is a software program that enables the creation and management of databases. Generally, these databases will be more complex than the text file/spreadsheet example in the previous lesson. In fact, most of today's database systems are referred to as a Relational Database Management System (RDBMS), because of their ability to store related data across multiple tables.

Some of the more popular relational database management systems include:

- Microsoft Access
- Filemaker
- SQL Server
- MySQL
- Oracle
- SQLite

What is a Table?

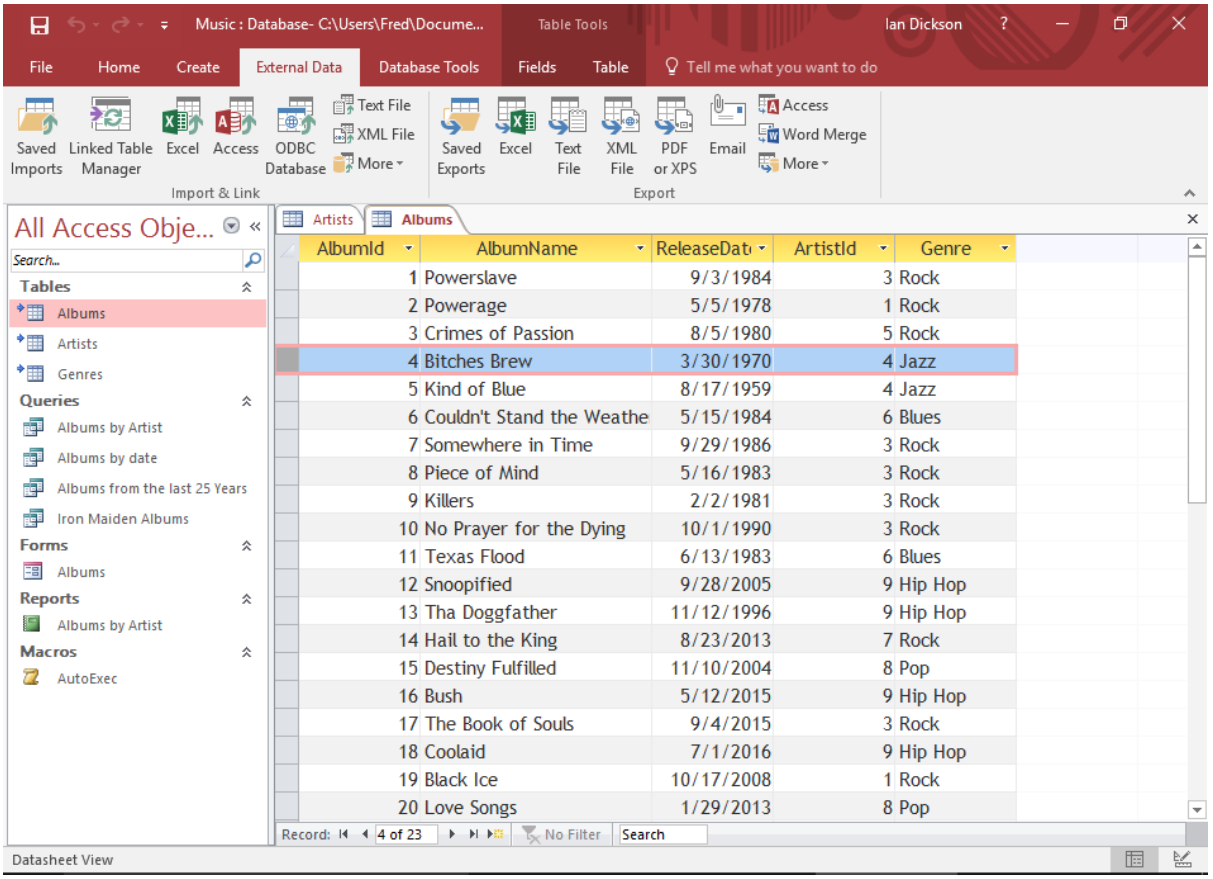
In relational database terms, a table is responsible for storing data in the database. Database tables consist of rows and columns.

Rows

Rows run horizontally. They represent each record. A row is the smallest unit of data that can be inserted into a database.

Rows span multiple columns, and therefore, the definition of a column applies to the cell where the row intersects with that column.

In this example, the fourth row is selected/highlighted.

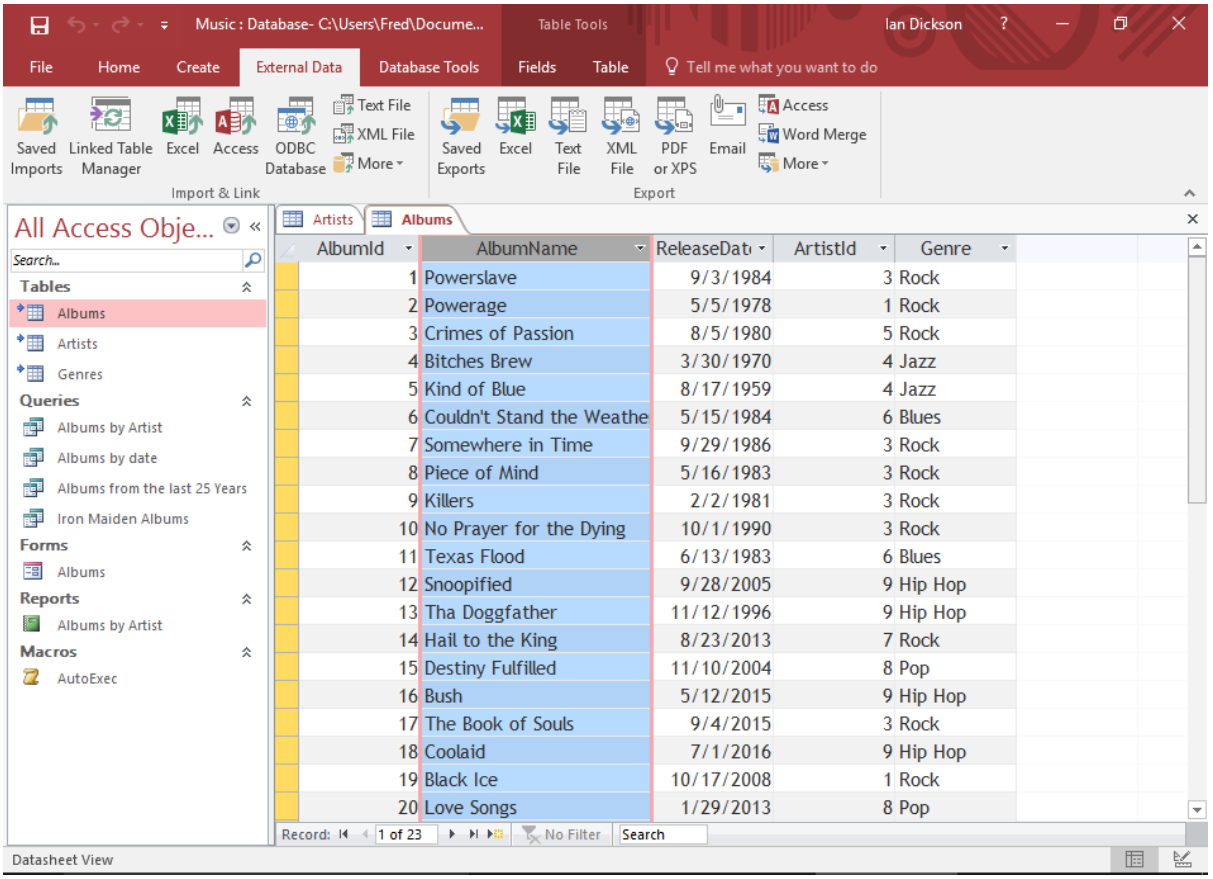


Columns

Columns run vertically. They contain the definition of each field.

You give each column a name, so that it is describes the data that is stored. Examples of column names could include FirstName, LastName, ProductId, Price, etc

In this example, the second column (called AlbumName) is selected/highlighted.



Many Tables

Most relational databases contain many tables.

For example, a database might have a Customers table, a Products table, an Orders table, and many more. In fact, it's not uncommon for a corporate database to contain hundreds of tables.

What is a Key in a Relational Database?

Within the confines of a Relational Database, a Key is defined as a column or a group of columns (attributes) leveraged to uniquely locate records in a table of a Relational Database.

Relational Database Keys have various forms of constraint to conform to such as columns, which cannot hold duplicate values or null values.

The various MySQL Keys are:

- Primary Key
- Unique Key
- Candidate Key
- Foreign Key
- Super Key

What is a MySQL Primary Key?

According to the rules of a Relational Database, each table should have only one primary key. Apart from this, Primary Keys aren't allowed to entertain duplicates (ND), null (NN), or no change (NC) values.

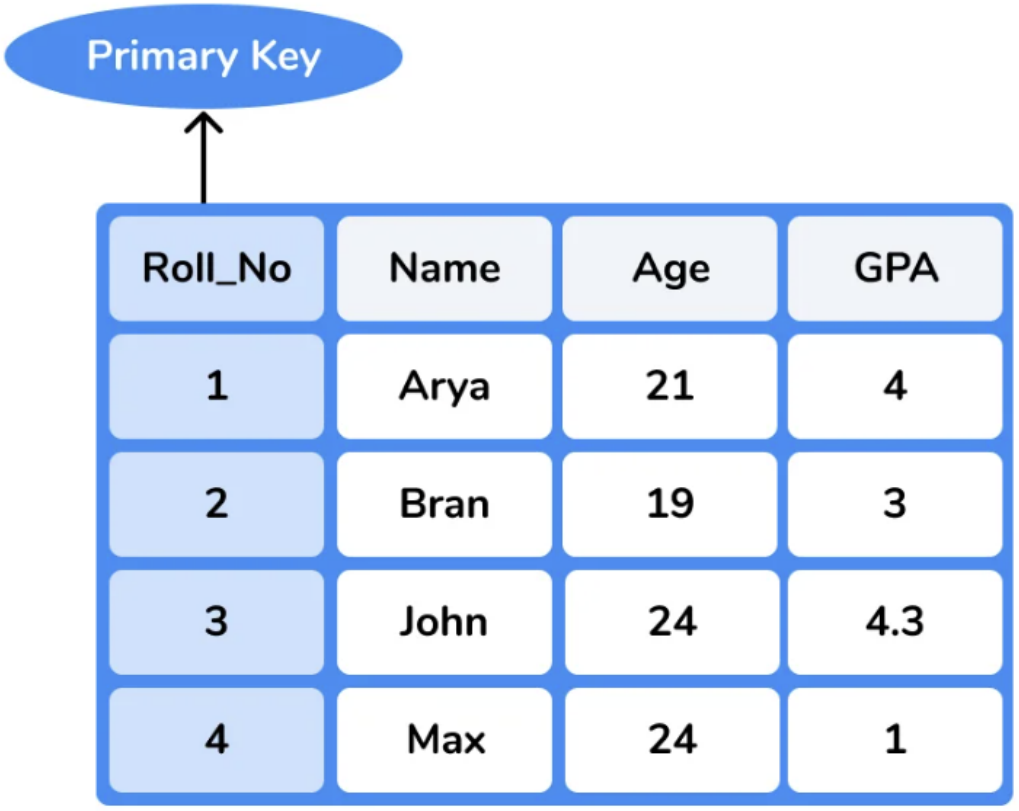
If you try to update or insert NULL values to the Primary Key, it will throw an error. Apart from this, if the Primary Key contains multiple columns, the combination of values within these columns should be unique for it to be valid.

Since MySQL works swiftly with integers, the data type of the Primary MySQL Key column should be an integer.

For instance, INT or BIGINT. It is suggested that you make sure that the value ranges for the Primary Key are enough for storing all the possible rows a table can have.

In MySQL, this column generally has the AUTO_INCREMENT attribute that automatically generates a sequential integer whenever you supply a new row into the table.

Primary Key



Using Create Table to Define a Primary MySQL Key

```
CREATE TABLE table_name(
    primary_key_column1 datatype,
    primary_key_column2 datatype,
    ...,
    PRIMARY KEY(column_list)
);


CREATE TABLE roles(
   role_id INT AUTO_INCREMENT,
   role_name VARCHAR(50),
   PRIMARY KEY(role_id)
);
```

Using Alter Table to Define a Primary MySQL Key

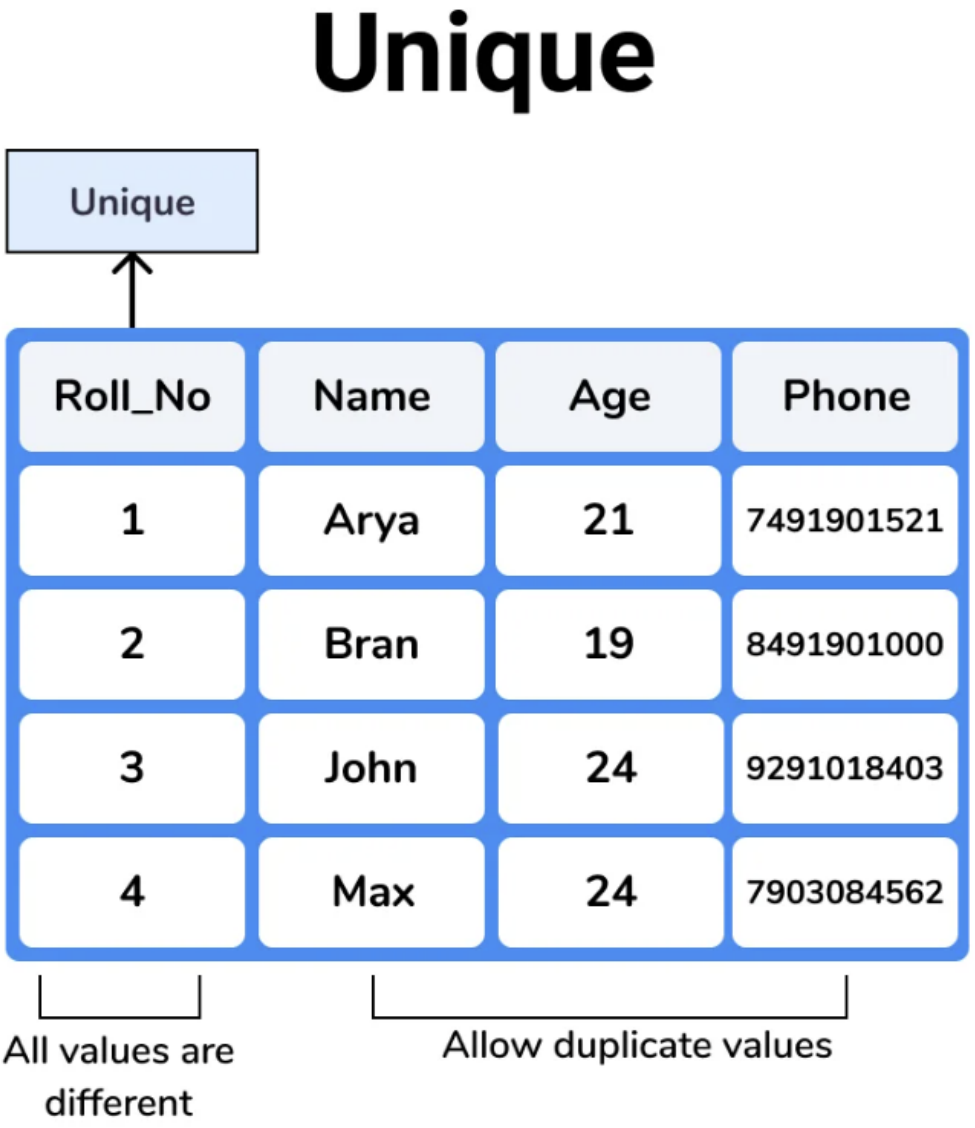
```
ALTER TABLE table_name
ADD PRIMARY KEY(column_list);

ALTER TABLE pkdemos
ADD PRIMARY KEY(id);
```

What is a MySQL Unique Key?

A group of one or more table fields/columns that uniquely identify a record in a database table is known as a unique key in MySQL Keys.

It's similar to a primary key in that it can only accept one null value and cannot have duplicate values.

Both the unique key and the primary key ensure that a column or set of columns is unique.



Using Create Table to Define a Unique MySQL Key

```
CREATE TABLE <table_name>
(
Column_name1 datatype(),
Column_name2 datatype(),…
Column_namen datatype(),
UNIQUE (column_name1, column_name2)
);

CREATE TABLE VATSA(
    ID INT AUTO_INCREMENT PRIMARY KEY,
    Company_name varchar(100) UNIQUE,
    Address varchar(250) UNIQUE
);
```

Using Alter Table to Define a Unique MySQL Key

```
ALTER TABLE <table_name>
ADD UNIQUE (column_name);

ALTER TABLE friend
ADD CONSTRAINT uq_col
UNIQUE (friend_name);
```

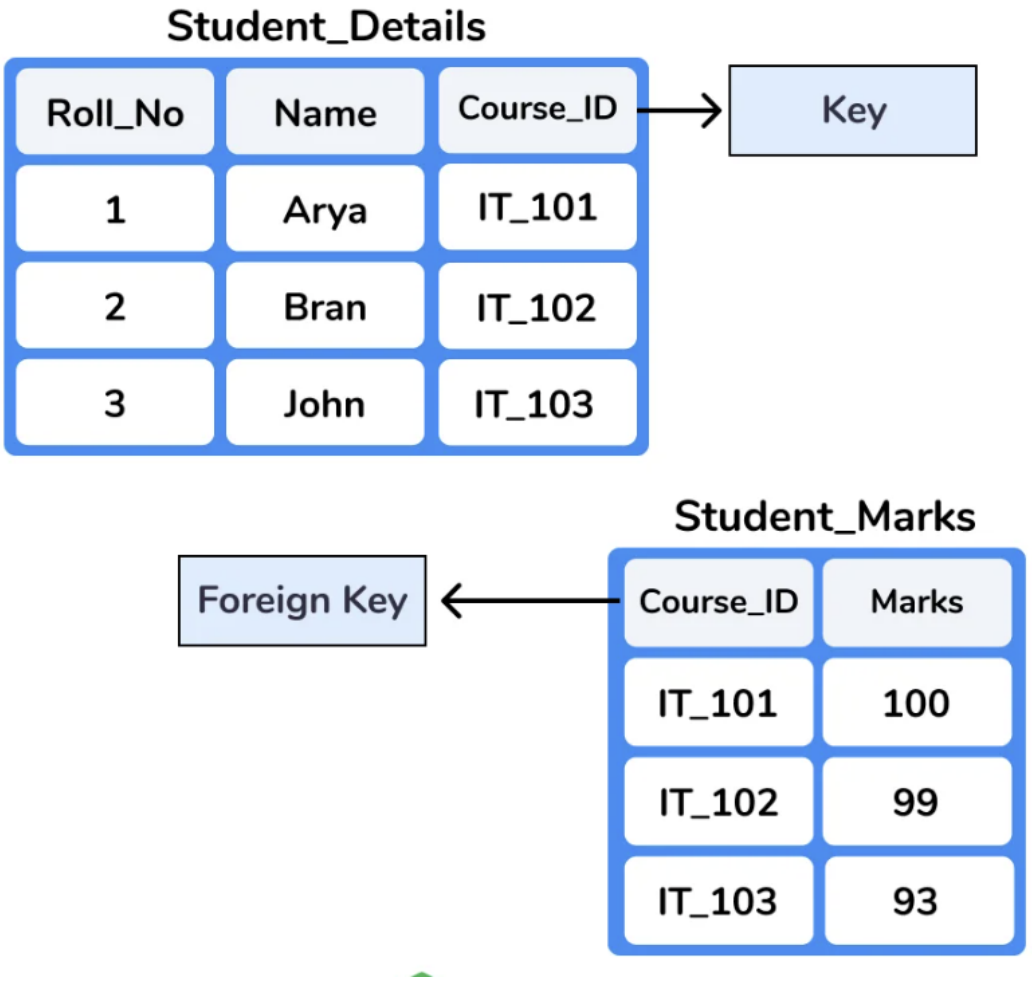
What is a MySQL Foreign Key?

The Foreign Key in MySQL Keys allows MySQL to ensure referential integrity by placing constraints on data in the related tables.

In MySQL, you can have more than one Foreign Key for a table where each Foreign Key references a Primary Key of various parent tables.

However, once a Foreign Key constraint is in place, the Foreign Key columns from the child table need to have the corresponding row in the parent Key columns of the parent table.

Foreign Key



How to add a Foreign MySQL Key?

```
ALTER TABLE zoo
ADD FOREIGN KEY
(FK_species) REFERENCES species (id);
```

How to Delete/Drop a Foreign MySQL Key?

```
ALTER TABLE table_name
DROP FOREIGN KEY constraint_name;
```

How to Disable Foreign MySQL Key Checks?

```
// disable
SET foreign_key_checks = 0;
// enable
```

```
SET foreign_key_checks = 1;
```

What is a MySQL Candidate Key?

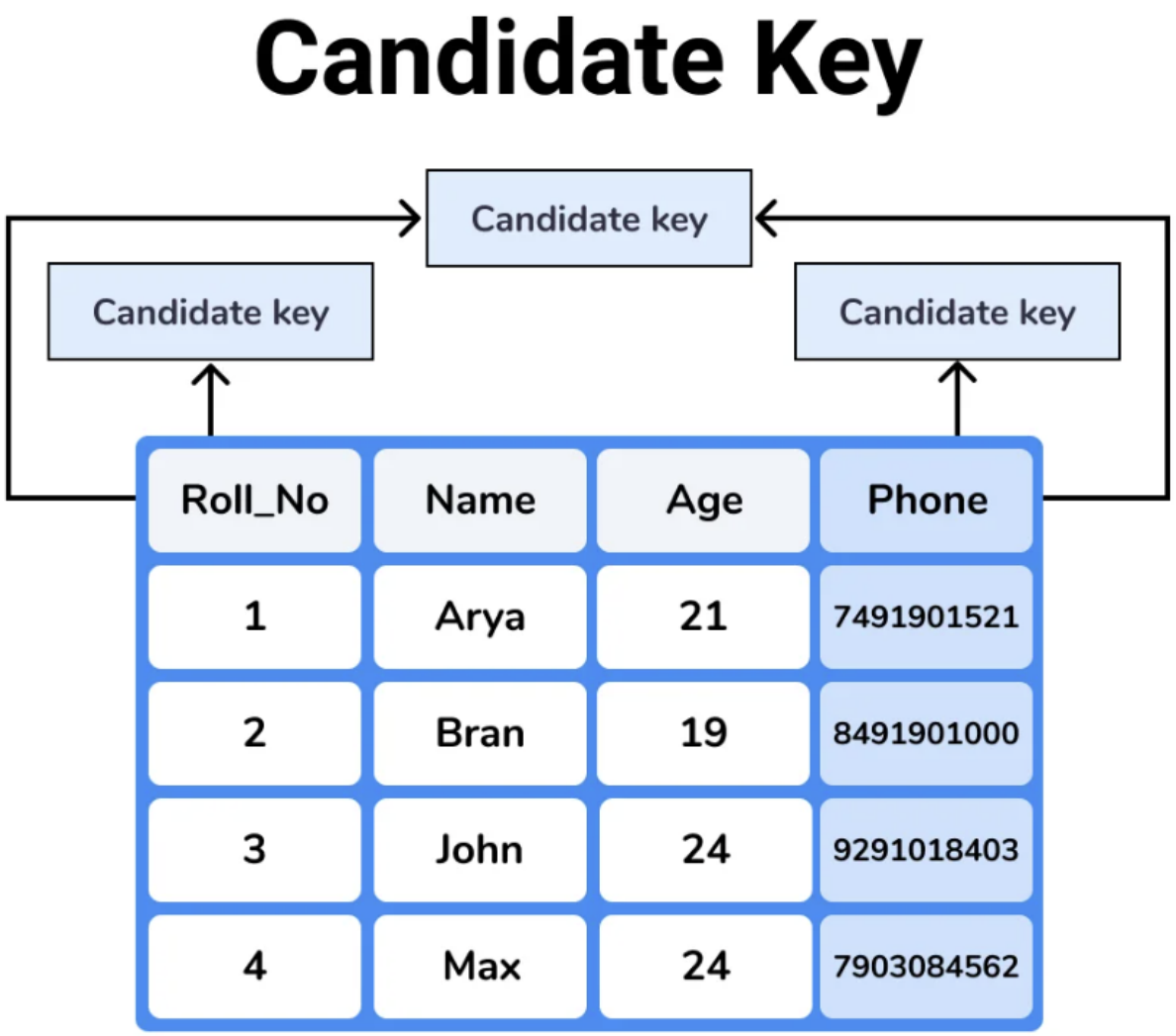
A characteristic or group of attributes that can uniquely identify a tuple is known as a candidate key in MySQL Keys.

The remaining properties, with the exception of the primary key, are considered candidate keys. Candidates have the same strength as the primary key.

Take a look at the “Employee” table as an example. Emp_Id, Emp_Number, and Emp_Name are the three attributes in this table. Emp_Id and Emp_Number will have unique values, however, Emp_Name may contain duplicate values because multiple employees may have the same name.

The candidate keys here are {Emp Id} and {Emp Number}.

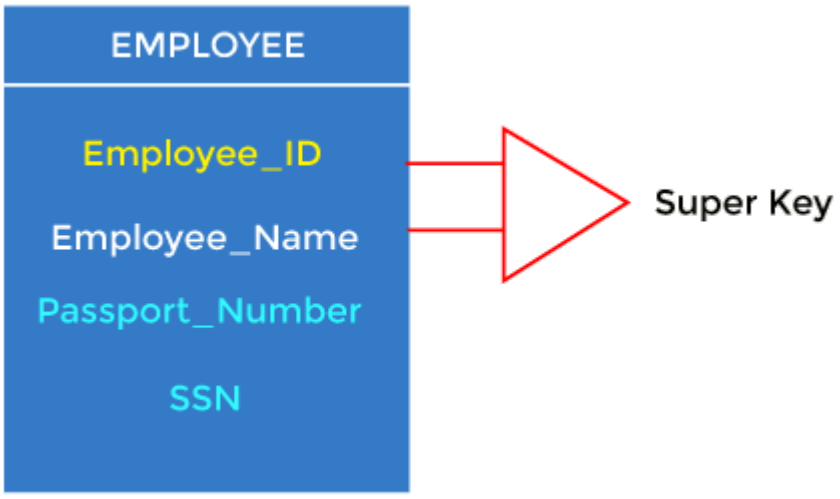
```
ALTER TABLE MyTable ADD UNIQUE KEY `my_unique_key` (`Emp_Id`, `Emp_Number`);
```

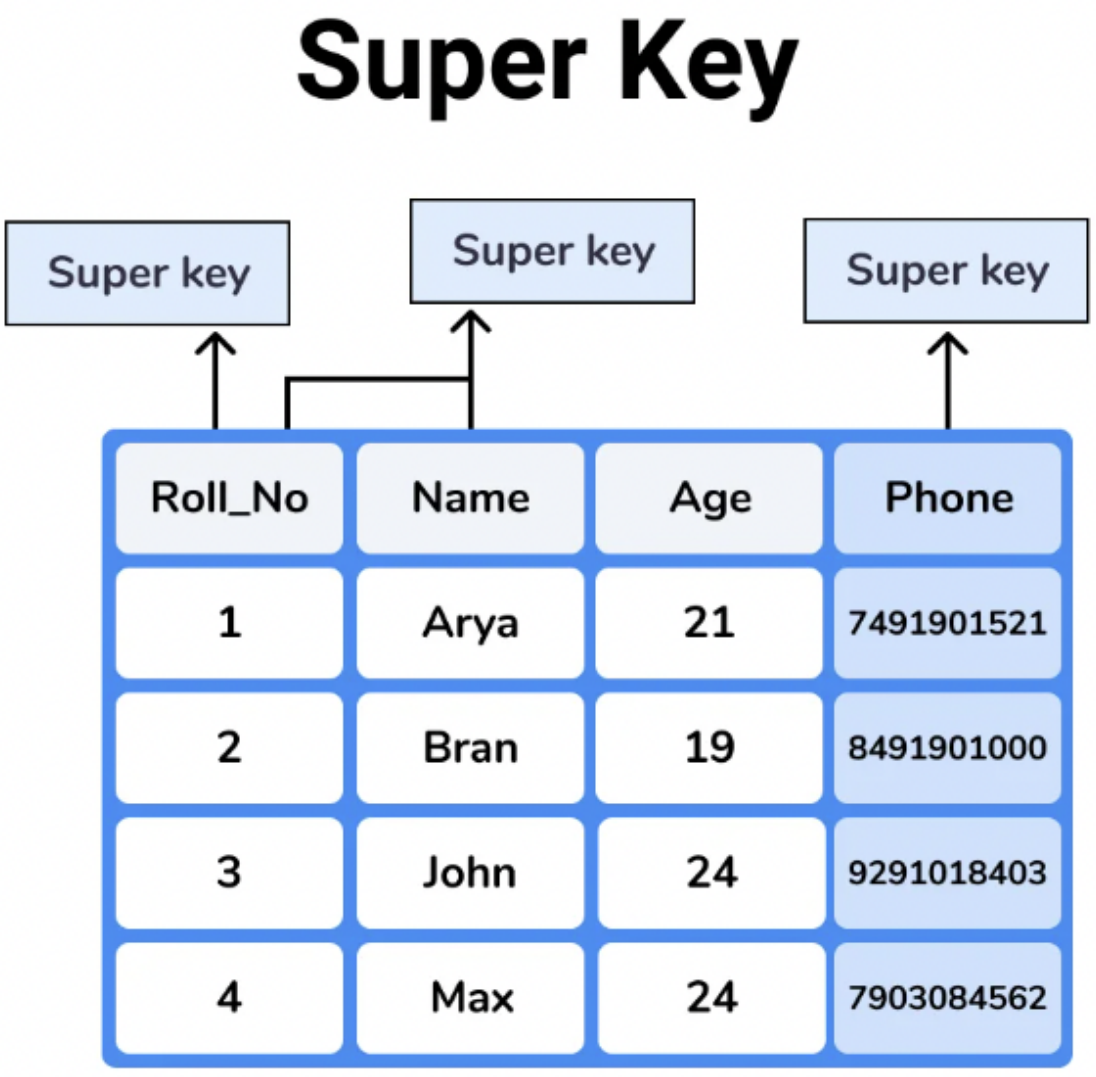


What is MySQL Super Key?

In a Relational Database, a Super Key is defined as a set of attributes that can identify every tuple of a relation. Since Super Key values are unique, tuples that have the same Super Key value should also have the same non-key attribute values.

For example, (EMPLOYEE ID, EMPLOYEE NAME) in the following EMPLOYEE database, two employees' names can be the same, but their EMPLYEE ID cannot. As a result, this combination could be a crucial.





How candidate key is different from super key?

Answer is simple – Candidate keys are selected from the set of super keys, the only thing we take care while selecting candidate key is: It should not have any redundant attribute. That's the reason they are also termed as minimal super key.

Let's take an example to understand this:

Table: Employee

Emp_SSN	Emp_Number	Emp_Name
123456789	226	Steve
999999321	227	Ajeet
888997212	228	Chaitanya
777778888	229	Robert

Super keys: The above table has following super keys. All of the following sets of super key are able to uniquely identify a row of the employee table.

{Emp_SSN}

{Emp_Number}

{Emp_SSN, Emp_Number}

{Emp_SSN, Emp_Name}

{Emp_SSN, Emp_Number, Emp_Name}

{Emp_Number, Emp_Name}

Candidate Keys: As I mentioned in the beginning, a candidate key is a minimal super key with no redundant attributes. The following two set of super keys are chosen from the above sets as there are no redundant attributes in these sets.

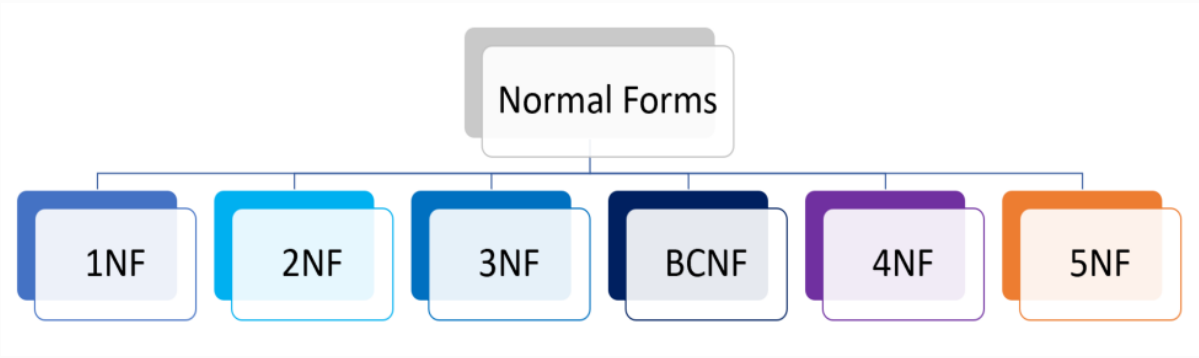
{Emp_SSN}

{Emp_Number}

Only these two sets are candidate keys as all other sets are having redundant attributes that are not necessary for unique identification.

Database Normalization

Normalization or the database normalization is a process to organize the data into database tables. To make a good database design, you have to follow Normalization practices. Without normalization, a database system might be slow, inefficient, and might not produce the expected result. Normalization reduces data redundancy and inconsistent data dependency.



To understand normal forms consider the following unnormalized database table. Now we will normalize the data in the below table using normal forms.

Project Code	Project Name	Project Manager	Project Budget	Employee No.	Employee Name	Department No.	Department Name	Hourly Rate
PC010	Reservation System	Mr. Ajay	120500	S100	Mohan	D03	Database	21.00
PC010	Reservation System	Mr. Ajay	120500	S101	Vipul	D02	Testing	16.50
PC010	Reservation System	Mr. Ajay	120500	S102	Riyaz	D01	IT	22.00
PC011	HR System	Mrs. Charu	500500	S103	Pavan	D03	Database	18.50
PC011	HR System	Mrs. Charu	500500	S104	Jitendra	D02	Testing	17.00
PC011	HR System	Mrs. Charu	500500	S315	Pooja	D01	IT	23.50
PC012	Attendance System	Mr. Rajesh	710700	S137	Rahul	D03	Database	21.50
PC012	Attendance System	Mr. Rajesh	710700	S218	Avneesh	D02	Testing	15.50
PC012	Attendance System	Mr. Rajesh	710700	S109	Vikas	D01	IT	20.50

UNF

First Normal Form (1NF)

A database table is said to be in 1NF if it contains no repeating fields/columns. The process of converting the UNF table into 1NF is as follows:

Separate the repeating fields into new database tables along with the key from the unnormalized database table.

The primary key of new database tables may be a composite key

1NF of the above UNF table is as follows:

Primary Key

Project Code	Project Name	Project Manager	Project Budget
PC010	Reservation System	Mr. Ajay	120500
PC011	HR System	Mrs. Charu	500500
PC012	Attendance System	Mr. Rajesh	710700

Composite Key (Unique Key)

Project Code	Employee No.	Employee Name	Department No.	Department Name	Hourly Rate
PC010	S100	Mohan	D03	Database	21.00
PC010	S101	Vipul	D02	Testing	16.50
PC010	S102	Riyaz	D01	IT	22.00
PC011	S103	Pavan	D03	Database	18.50
PC011	S104	Jitendra	D02	Testing	17.00
PC011	S315	Pooja	D01	IT	23.50
PC012	S137	Rahul	D03	Database	21.50
PC012	S218	Avneesh	D02	Testing	15.50
PC012	S109	Vikas	D01	IT	20.50

1NF

Second Normal Form (2NF)

A database table is said to be in 2NF if it is in 1NF and contains only those fields/columns that are functionally dependent(means the value of the field is determined by the value of another field(s)) on the primary key. In 2NF we remove the partial dependencies of any non-key field.

The process of converting the database table into 2NF is as follows:

Remove the partial dependencies(A type of functional dependency where a field is only functionally dependent on the part of the primary key) of any non-key field.

If field B depends on field A and vice versa. Also for a given value of B, we have only one possible value of A and vice versa, Then we put the field B into a new database table where B will be the primary key and also marked as a foreign key in a parent table.

2NF of the above 1NF tables is as follows:

Primary Key

Project Code	Project Name	Project Manager	Project Budget
PC010	Reservation System	Mr. Ajay	120500
PC011	HR System	Mrs. Charu	500500
PC012	Attendance System	Mr. Rajesh	710700

Composite Key

Project Code	Employee No.	Hourly Rate
PC010	S100	21.00
PC010	S101	16.50
PC010	S102	22.00
PC011	S103	18.50
PC011	S104	17.00
PC011	S315	23.50
PC012	S137	21.50
PC012	S218	15.50
PC012	S109	20.50

Primary Key

Employee No.	Employee Name	Department No.	Department Name
S100	Mohan	D03	Database
S101	Vipul	D02	Testing
S102	Riyaz	D01	IT
S103	Pavan	D03	Database
S104	Jitendra	D02	Testing
S315	Pooja	D01	IT
S137	Rahul	D03	Database
S218	Avneesh	D02	Testing
S109	Vikas	D01	IT

2NF

Third Normal Form (3NF)

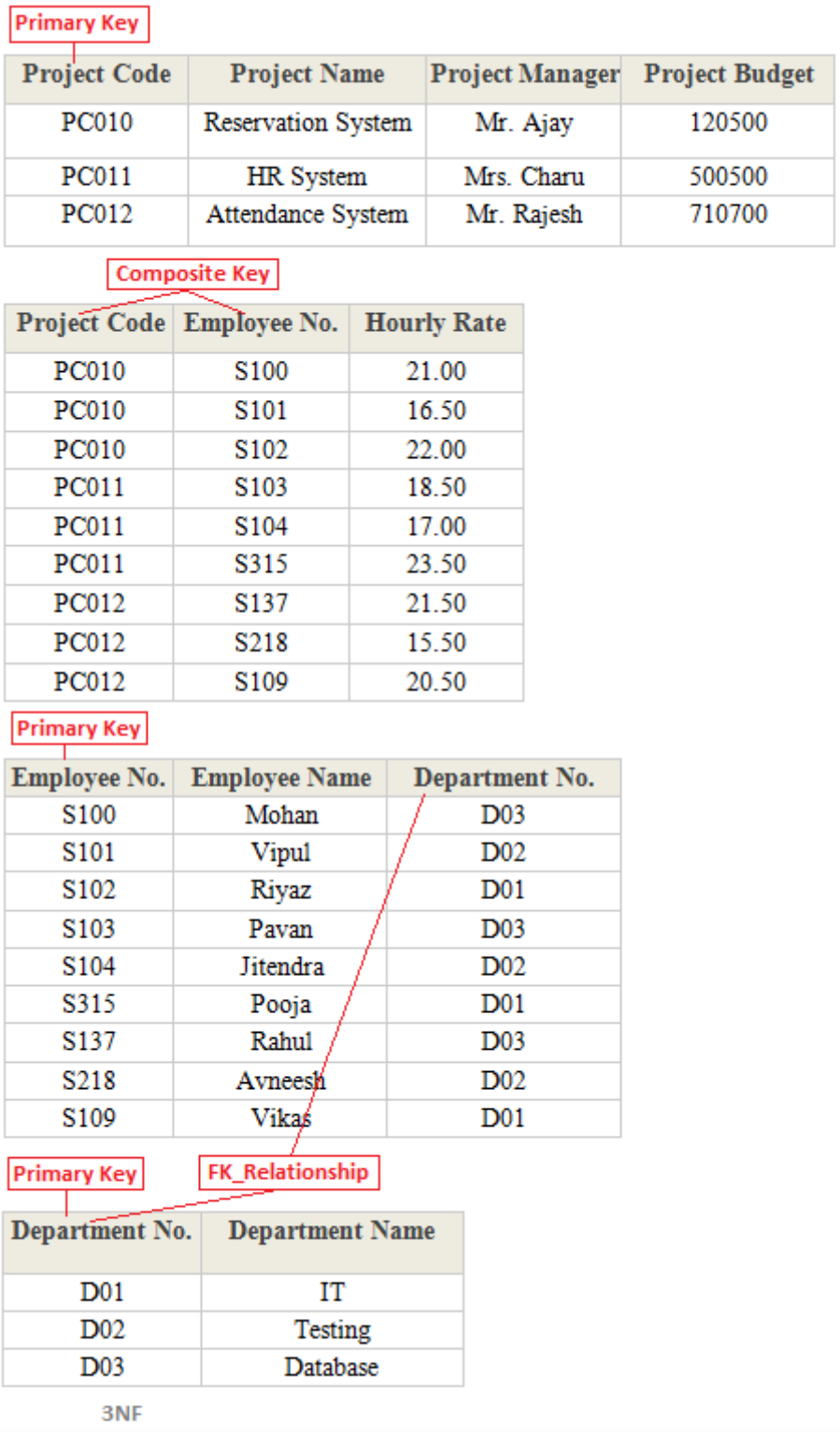
A database table is said to be in 3NF if it is in 2NF and all non-keys fields should be dependent on the primary key We can also say a table to be in 3NF if it is in 2NF and no fields of the table are transitively functionally dependent on the primary key. The process of converting the table into 3NF is as follows:

Remove the transitive dependencies(A type of functional dependency where a field is functionally dependent on the Field that

is not the primary key. Hence its value is determined, indirectly by the primary key)

Make a separate table for transitive dependent Fields.

3NF of the above 2NF tables is as follows:



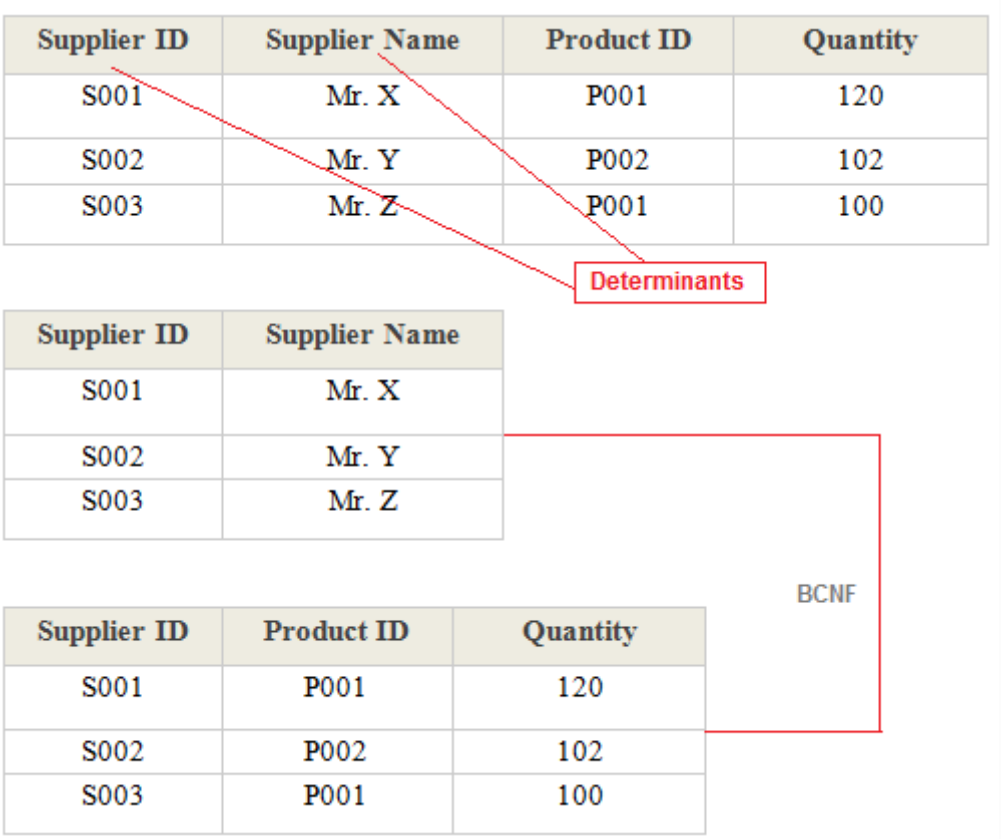
3NF

Boyce Code Normal Form (BCNF)

A database table is said to be in BCNF if it is in 3NF and contains each and every determinant is a candidate key. The process of converting the table into BCNF is as follows:

1. Remove the nontrivial functional dependency.
2. Make a separate table for the determinants.

BCNF of the below table is as follows:



Fourth Normal Form (4NF)

A database table is said to be in 4NF if it is in BCNF and the primary key has a one-to-one relationship to all non-keys fields We can also say a table to be in 4NF if it is in BCNF and contains no multi-valued dependencies. The process of converting the table into 4NF is as follows:

1. Remove the multivalued dependency.

2. Make a separate table for multivalued Fields.

4NF of the below table is as follows:

Employee Name	Skills	Language
Mohan	C Sharp	Hindi
Mohan	Asp.Net	Hindi
Mohan	SQL Server	Hindi
Mohan	C Sharp	English
Mohan	Asp.Net	English
Mohan	SQL Server	English

Employee Name	Skills
Mohan	C Sharp
Mohan	Asp.Net
Mohan	SQL Server

4NF

Employee Name	Language
Mohan	Hindi
Mohan	English

Fifth Normal Form (5NF)

A database table is said to be in 5NF if it is in 4NF and contains no redundant values or We can also say a table to be in 5NF if it is in 4NF and contains no join dependencies. The process of converting the table into 5NF is as follows:

5NF of the below table is as follows:

1. Remove the join dependency.
2. Break the database table into smaller and smaller tables to remove all data redundancy.

Company	Product	Supplier
Godrej	Soap	Mr. Amit
Godrej	Shampoo	Mr. Pavan
Godrej	Shampoo	Mr. Amit
H.Lever	Soap	Mr. Amit
H.Lever	Shampoo	Mr. Pavan
H.Lever	Soap	Mr. Sachin

Company	Product
Godrej	Soap
Godrej	Shampoo
H.Lever	Soap
H.Lever	Shampoo

Company	Supplier
Godrej	Mr. Amit
Godrej	Mr. Pavan
H.Lever	Mr. Amit
H.Lever	Mr. Pavan
H.Lever	Mr. Sachin

Product	Supplier
Soap	Mr. Amit
Shampoo	Mr. Pavan
Shampoo	Mr. Amit
Soap	Mr. Sachin