What is Normalization?

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalisation in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

Database Normal Forms

Here is a list of Normal Forms

1NF (First Normal Form)

2NF (Second Normal Form)

3NF (Third Normal Form)

BCNF (Boyce-Codd Normal Form)

4NF (Fourth Normal Form)

5NF (Fifth Normal Form)

6NF (Sixth Normal Form)

The Theory of Data Normalization in MySQL server is still being developed further. For example, there are discussions even on 6th Normal Form. However, in most practical applications, normalization achieves its best in 3rd Normal Form.

Database Normalization With Examples

Database Normalization Example can be easily understood with the help of a case study. Assume, a video library maintains a database of movies rented out. Without any normalization in database, all information is stored in one table as shown below. Let's understand Normalization database with tables example:

Full Names	Physical Address	Movies rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

Database Normalization With Example

Here you see Movies Rented column has multiple values. Now let's move into 1st Normal Forms:

1NF (First Normal Form) Rules

Each table cell should contain a single value.

Each record needs to be unique.

The above table in 1NF-

1NF Example

Full Names	Physical Address	Movies rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

What is a KEY in SQL?

A KEY in SQL is a value used to identify records in a table uniquely. An SQL KEY is a single column or combination of multiple columns used to uniquely identify rows or tuples in the table. SQL Key is used to identify duplicate information, and it also helps establish a relationship between multiple tables in the database.

Note: Columns in a table that are NOT used to identify a record uniquely are called non-key columns.

What is a Primary Key?

Primary key in Database

A primary is a single column value used to identify a database record uniquely.

<u>It has following attributes</u>

A primary key cannot be NULL

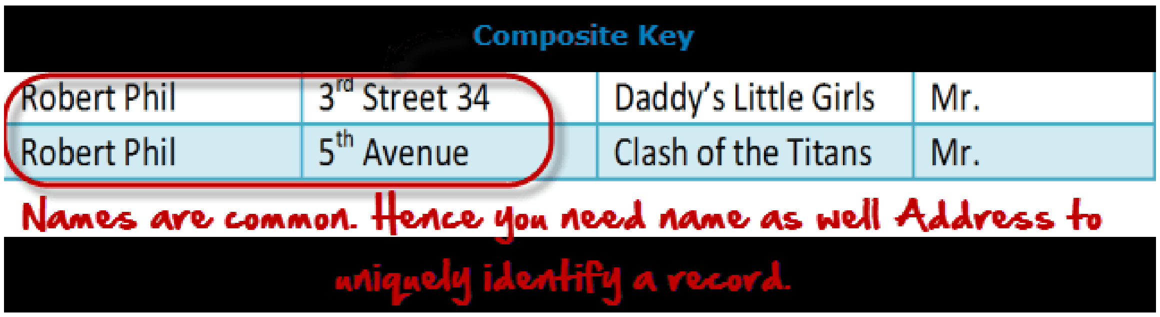
A primary key value must be unique

The primary key values should rarely be changed

The primary key must be given a value when a new record is inserted.

What is Composite Key?

A composite key is a primary key composed of multiple columns used to identify a record uniquely.In our database, we have two people with the same name Robert Phil, but they live in different places.



Composite Key

Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

Names are common. Hence you need name as well Address to uniquely identify a record.

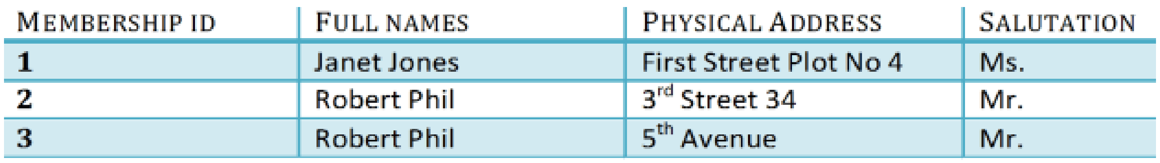
Hence, we require both Full Name and Address to identify a record uniquely. That is a composite key.

2NF (Second Normal Form) Rules

Rule 1- Be in 1NF

Rule 2- Single Column Primary Key

It is clear that we can't move forward to make our simple database in 2nd Normalization form unless we partition the table above.



MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Membership ID	Movies rented
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

We have divided our 1NF table into two tables viz. Table 1 and Table2. Table 1 contains member information. Table 2 contains information on movies rented.

We have introduced a new column called Membership_id which is the primary key for table 1. Records can be uniquely identified in Table 1 using membership id.

Database - Foreign Key

In Table 2, Membership_ID is the Foreign Key

Membership ID	Movies rented
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

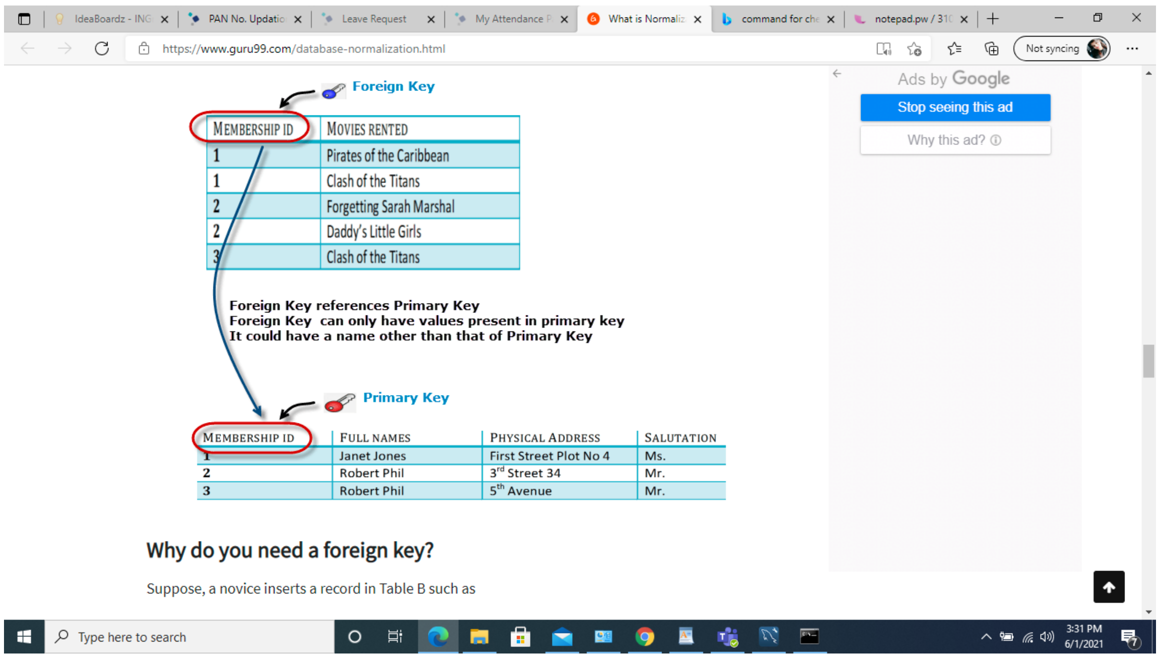
Foreign Key references the primary key of another Table! It helps connect your Tables

A foreign key can have a different name from its primary key

It ensures rows in one table have corresponding rows in another

Unlike the Primary key, they do not have to be unique. Most often they aren't

Foreign keys can be null even though primary keys can not



Membership ID	Movies Rented
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

Foreign Key references Primary Key
Foreign Key can only have values present in primary key
It could have a name other than that of Primary Key

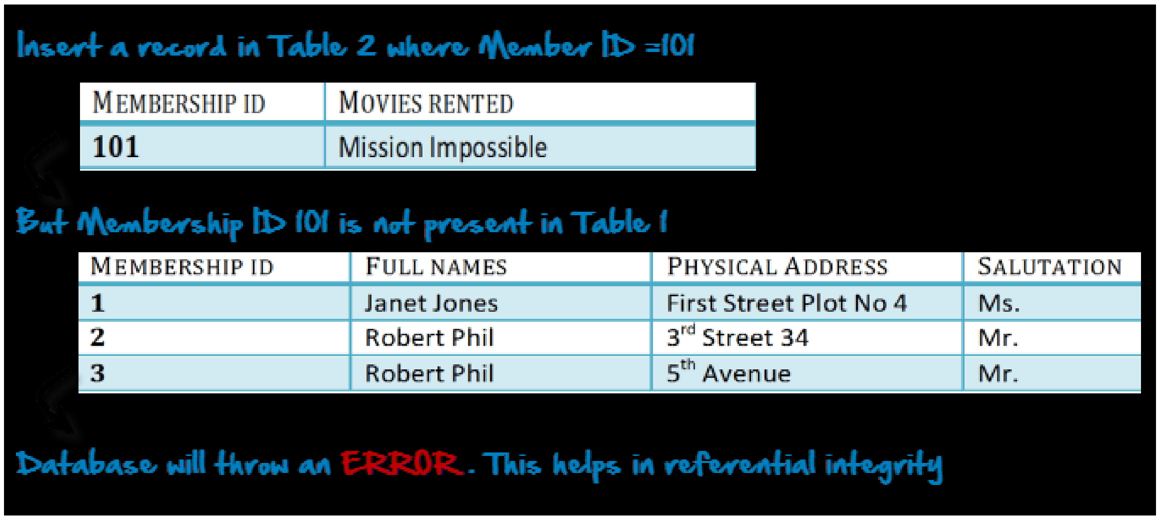
Membership ID	Full Names	Physical Address	Salutation
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Why do you need a foreign key?

Suppose, a novice inserts a record in Table B such as

Why do you need a foreign key?

Suppose, a novice inserts a record in Table B such as



Insert a record in Table 2 where Member ID =101

Membership ID	Movies Rented
101	Mission Impossible

But Membership ID 101 is not present in Table 1

Membership ID	Full Names	Physical Address	Salutation
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Database will throw an ERROR. This helps in referential integrity

You will only be able to insert values into your foreign key that exist in the unique key in the parent table. This helps in referential integrity.

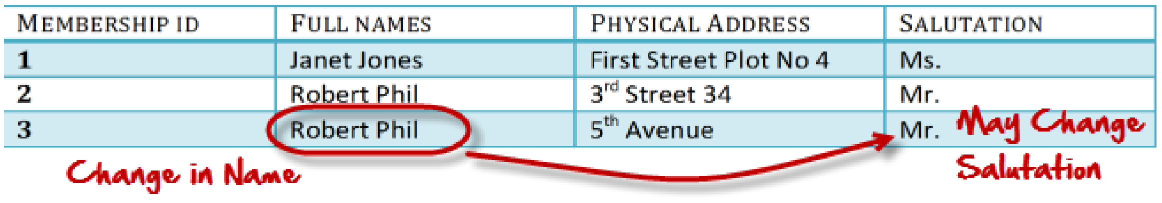
The above problem can be overcome by declaring membership id from Table2 as foreign key of membership id from Table1

Now, if somebody tries to insert a value in the membership id field that does not exist in the parent table, an error will be shown!

What are transitive functional dependencies?

A transitive functional dependency is when changing a non-key column, might cause any of the other non-key columns to change

Consider the table 1. Changing the non-key column Full Name may change Salutation**.**



MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Change in Name
May Change Salutation

3NF (Third Normal Form) Rules

Rule 1- Be in 2NF

Rule 2- Has no transitive functional dependencies

To move our 2NF table into 3NF, we again need to again divide our table.

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION ID
1	Janet Jones	First Street Plot No 4	2
2	Robert Phil	3rd Street 34	1
3	Robert Phil	5th Avenue	1

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

SALUTATION ID	SALUTATION
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

DDL Full Form

DDL stands for Data Defination Language.

These commands are used to change the structure of a database and database objects.For example, DDL commands can be used to add, remove, or modify tables with in a database.

The DDL commands are:

CREATE

ALTER

DROP

TRUNCATE

RENAME

1. CREATE :

This command is used to create table in the relational database .

This can be done by specifying the names and datatypes of various columns.

Syntax:

```
CREATE TABLE TABLE_NAME

(

        column_name1 datatype1,

        column_name2 datatype2,

        column_name3 datatype3,

        column_name4 datatype4

);
```

The column_name in create table command will tell the name of the column and corresponding datatype will specify the datatype of that column.Here in this table the three column_names namely – Student_id is of type int ,Name is of type varchar and Marks is of type int.

for example:

```
CREATE TABLE

Employee

(Student_id INT,

Name VARCHAR(100),

Marks INT);
```

Student_id

Name

Marks

2. ALTER :

Alter command is used for altering the table in many forms like:

Add a column

Rename existing column

Drop a column

Modify the size of the column or change datatype of the column

ADD using ALTER –

Syntax to add column :

```
ALTER TABLE table_name ADD(
        column_name datatype);
```

The above command will add a new column to the table.And the resulting table will have one more column like this:

```
ALTER TABLE Student
ADD
(Address    VARCHAR(200));
```

Here this command will add a new column “Address” in the table Student of datatype varchar(200);

Student_id

Name

Marks

Address

RENAME using ALTER –

Syntax to rename column :

ALTER TABLE

table_name

RENAME

old_column_name TO new_column_name;

The above command will rename the existing column to new column.

ALTER TABLE

Employee

RENAME

Marks TO Age;

The command above will change the column_name from Marks to Age;

Student_id

Name

Age

Address

DROP using ALTER –

Syntax to Drop a column :

ALTER TABLE

table_name

DROp

(column_name);

The above command will delete the existing column.

For example:

ALTER TABLE Employee

DROP

(Age);

Here the column_name =”Age”, has been deleted by this command;

Student_id

Name

Address

MODIFY using ALTER –

Syntax to Modify a column :

ALTER TABLE

Employee MODIFY

(column_name datatype);

The above command will modify the existing column .

For example:

ALTER TABLE

student

MODIFY

(name varchar(300));

The above command will modify the column_name “Name” by changing the size of that column.

Student_id

Name

Address

3. TRUNCATE :

This command removes all the records from a table. But this command will not destroy the table’s structure.

Syntax :

TRUNCATE TABLE table_name

This will delete all the records from the table.For example the below command will remove all the records from table student.

Example:

TRUNCATE TABLE Student;

4. DROP :

This command completely removes the table from the database along with the destruction of the table structure.

Syntax –

DROP TABLE table_name

This will delete all the records as well as the structure of the table.

This is the main difference between TRUNCATE AND DROP.-TRUNCATE only removes the records whereas DROP completely destroys the table.

Example:

DROP TABLE Student;

This command will remove the table records as well as destroys the schema too.

This is all about the DDL commands.