Data manipulation language - It is used to change the values in the table

1. Insert

-- the below inserting operation is called implicitng inserting because we filled the values for every column.

insert into student_details values ('Pavan', 390, 'kakinada'),

('Vinay', 391, 'Chennai'),

('Venkat', 392, 'Bangalore'),

('Sai', 393, 'mumbai'),

('Jai', 394, 'hyderbad');

-- If we want to insert values in a specific column then it is called as explicit inserting

Insert into student_details(Student_name,student_id) values('Peter', 395);

2. Update - it is used to modify any exisitng value in the table

/* In the initial time of running update statement sql will throught safety mode error because if we try to update the data

the old data will be changed in order to remaind us it will through an error. For this scenario we need to run the below statement first*/

Update student_details set address = 'Vizag' where student_name = 'Peter';

set SQL_safe_updates = 0;

insert into student_details(address) values('guntur');

select * from student_details;

update student_details set student_name = 'Raju', student_id = 397, address = 'Tirupati' where address = 'guntur';

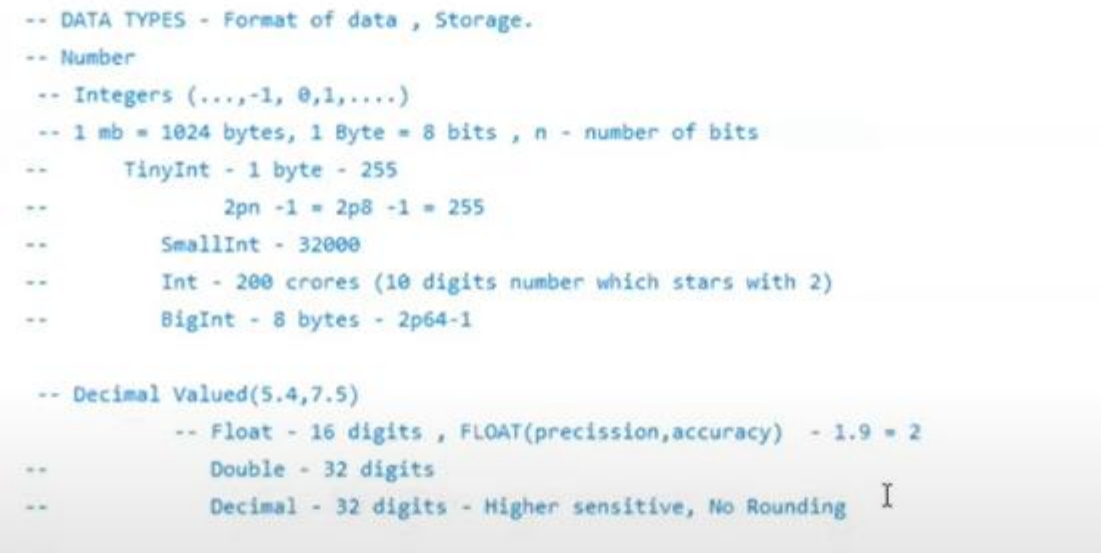
-- to turn on the safety mode we need to run the below statement

set SQL_safe_updates = 1;

3. Delete - to delete the records in the table either single or multiple or all records

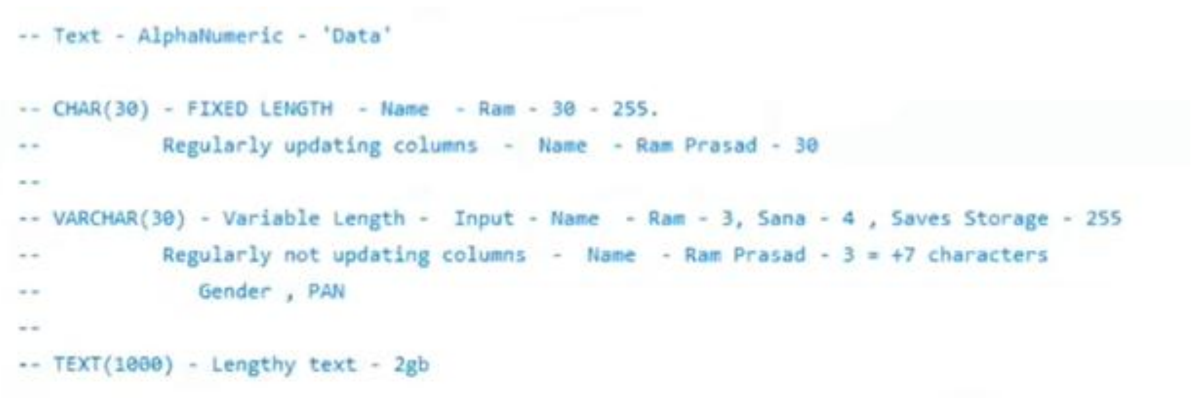
delete from student_details where student_id = 394;

Data types



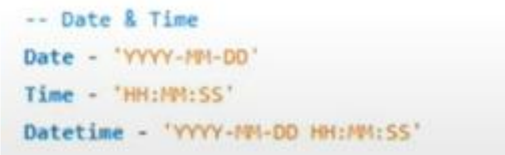
```
-- DATA TYPES - Format of data , Storage.
-- Number
 -- Integers (...,-1, 0,1,....)
 -- 1 mb = 1024 bytes, 1 Byte = 8 bits , n - number of bits
--      TinyInt - 1 byte - 255
--              2pn -1 = 2p8 -1 = 255
--         SmallInt - 32000
--         Int - 200 crores (10 digits number which stars with 2)
--         BigInt - 8 bytes - 2p64-1

 -- Decimal Valued(5.4,7.5)
           -- Float - 16 digits , FLOAT(precission,accuracy)  - 1.9 = 2
--            Double - 32 digits
--            Decimal - 32 digits - Higher sensitive, No Rounding
```



```
-- Text - AlphaNumeric - 'Data'

-- CHAR(30) - FIXED LENGTH  - Name  - Ram - 30 - 255.
--          Regularly updating columns  -  Name  - Ram Prasad - 30
--
-- VARCHAR(30) - Variable Length -  Input - Name  - Ram - 3, Sana - 4 , Saves Storage - 255
--          Regularly not updating columns  -  Name  - Ram Prasad - 3 = +7 characters
--             Gender , PAN
--
-- TEXT(1000) - Lengthy text - 2gb
```



```
-- Date & Time
Date - 'YYYY-MM-DD'
Time - 'HH:MM:SS'
Datetime - 'YYYY-MM-DD HH:MM:SS'
```

Constraints:

Regular constraints:

Null: it is used to represent unknown values and it is not equal to zero, also can be used for all data types

Not Null: if any value that should be mandatory

```
CREATE TABLE products (
    id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL
);
```

Default: The default value will be taken which was given while creating column

```
CREATE TABLE settings (
    id INT PRIMARY KEY,
    theme VARCHAR(20) DEFAULT 'light'
);
```

Check: it is used like a conditional statement. If we want people above age 18 we need to use check constraint.
ex; check (age >=18)

```
CREATE TABLE accounts (
    id INT PRIMARY KEY,
    balance DECIMAL(10, 2) CHECK (balance >= 0)
);
```

Key constrainst:

Unique key: Value should be unique and doesn’t allow duplicates but allows single null values because it also considers a null value as unique.

```
CREATE TABLE employees (
    id INT PRIMARY KEY,
    email VARCHAR(100) UNIQUE
);
```

Pirmary key: Combination on unique and not null. Will not allow null values

```
CREATE TABLE users (
    id INT PRIMARY KEY,
    name VARCHAR(100)
);
```

Foreign key: Ensures the value in a column matches a value in another table's primary key. Maintains referential integrity between related tables.

```
CREATE TABLE orders (
    order_id INT PRIMARY KEY,
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES users(id)
);
```

Auto_increment: it will generate sequence of value generated automatically

```
-- MySQL
CREATE TABLE customers (
   id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(100)
);


-- PostgreSQL
CREATE TABLE customers (
   id SERIAL PRIMARY KEY,
   name VARCHAR(100)
);
```