# SQL Learning Notes

**Database:** Database is a structured collection of data that is stored, managed, and accessed electronically. It is designed to efficiently store, retrieve, update, and manage data in a way that ensures integrity, consistency, and security**.**

Database management system

Your APP

(Amazon)

Database

**Database Management System:** A **Database Management System (DBMS)** is **software that interacts with databases** to perform tasks such as **storing, retrieving, updating, and managing data** efficiently while ensuring **data integrity, security, and concurrency control**.

**MySQL, PostgreSQL, SQLite, etc., are not databases themselves, but rather Database Management Systems (DBMS)** that provide the tools and functionalities to interact with, manage, and manipulate data stored in databases.

## Creating Databases and Tables

* SHOW DATABASES;
* It will show the databases which is available in the system.
* CREATE DATABASE yethishwar\_rao;
* It will create a database.
* DROP DATABASE yethishwar\_rao;
* It will delete the database permanently.

NOTE: using snake\_case is a good practice naming databases, tables, column\_names etc…

* USE database\_name;
* Switches the active database to the specified one.
* Just use the above query to switch to the working database and work on it.
* SELECT DATABASE();
* Displays the currently selected database.

Inside a Database there will be Tables this is the area where we actually focus on and manipulate the things. A table consists of rows and columns, almost all SQL flavoured languages deals with tables, other than MangoDB and some more.

### Data Types

We have many numerical, character and date time data types, among all some are frequently used like VARCHAR(n), CHAR(n), INT, FLOAT, BIGINT etc…

To create tables use the below query

* CREATE TABLE methodist\_collage

( students VARCHAR(100),

roll\_no INT,

grade CHAR(10)

);

#To see the structure of the table

* DESCRIBE collages\_in\_hyderabad;

-- OR --

* SHOW COLUMNS FROM gurunanak;

-- OR --

* SHOW TABLES;

-- To delete a table permanamtly

* DROP TABLE gurunanak;

-- Insert values into the Table

* INSERT INTO malla\_reddy(students, roll\_no, grade)
* VALUES('Siddhartha\_rao', 90, 'S++');
* INSERT INTO malla\_reddy(students, roll\_no, grade)
* VALUES('Sai Kiran Rao',89,'S++');

We can also interchange the values like this👇… the order does not matter but make sure to tally the column names and values.

* INSERT INTO malla\_reddy(roll\_no, grade, students)
* VALUES(89,'S++','Sai Kiran Rao');

-- Insert multiple rows at a time with single insert command

* INSERT INTO malla\_reddy(students,roll\_no)
* VALUES('Kiara',33),
* ('preethi',24),
* ('Sushil',12),
* ('Rushi',22)
* ;

-- To see the values in a table

* SELECT \* FROM malla\_reddy;

-- By specifying NOT NULL while creating a table... The table dooesn't support any null values

* CREATE TABLE population(

state VARCHAR(20) NOT NULL,

population INT NOT NULL);

If we try to insert values like this, 👇 it will throw error

* INSERT INTO population(state)
* VALUES('TELANGANA')

ERROR: ERROR 1364 (HY000): Field 'age' doesn't have a default value

So to overcome this error we need to specify a default value which is suitable for that data type like below…

The DEFAULT value **only applies when a column is entirely omitted** from the INSERT statement, **not when NULL is explicitly inserted**. 👇

* create table people2(
* -> name varchar(20) DEFAULT 'No value here',
* -> age int default 0);

If we insert null values also it just show the default value instead of ‘NULL’ when we select the table.

If you want MySQL to **always** replace NULL with the default value, you **must** use NOT NULL in the table👇

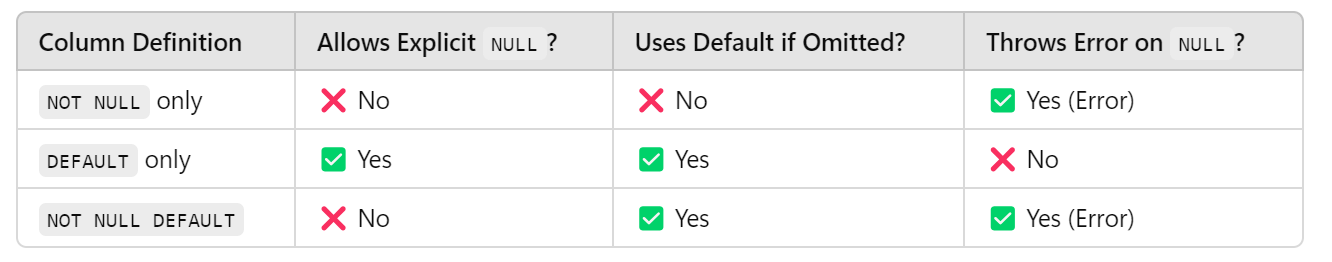
* create table people2(
* -> name varchar(20) NOT NULL DEFAULT 'No value here',
* -> age int NOT NULL default 0);

If you explicitly inserts a null value like this👇 it will take NULL value unless you mention NO NULL wile creating

* insert into people2(name, age)
* -> values(NULL,NULL);

-- If we want double codes in the middle of the name just mention back slash like thisit will ignore the next symbol and proceeds with no ERROR.👇

* INSERT INTO people2(name)
* VALUES('Yethishwar\'s Family');



### #Primary Key

A **Primary Key** is a unique identifier for each record in a database table. It ensures that no two rows have the same value in the primary key column and that the value cannot be **NULL**. This helps maintain data integrity and allows efficient data retrieval.

This query ensures no null values allowed and two different rows will contain unique values.

* create table unique\_people(
* aadhar bigint not null primary key,
* name varchar(100),
* age int);

You can also specify it at the end like this

* PRIMARY KEY (aadhar)

If you try to insert identical values like this it will throw an error like below

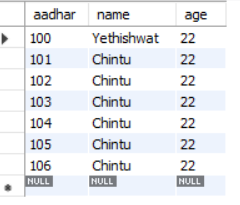
* insert into unique\_people(aadhar,name,age)
* values(1,'Yethishwar',21),
* (1,'Yethishwar',21);

ERROR 1062 (23000): Duplicate entry '1' for key 'unique\_people.PRIMARY'

### #AUTO\_INCREMENT

he **AUTO\_INCREMENT** constraint is used in SQL (specifically in MySQL and some other databases) to automatically generate a unique, incrementing value for a **Primary Key** column.

- If we you want the primary key values to be auto incremented please specify auto\_increment liek this

* CREATE TABLE people4(
* aadhar BIGINT auto\_increment,
* name VARCHAR(100),
* age INT,
* PRIMARY KEY(aadhar)
* );
* INSERT INTO people4(aadhar,name,age)
* VALUES(100,'Yethishwat',22);
* select \* from people4;
* INSERT INTO people4(name,age)
* VALUES('Chintu', 22),
* ('Chintu', 22),
* ('Chintu', 22),
* ('Chintu', 22),
* ('Chintu', 22),
* ('Chintu', 22);

MySQL picks the next available number after the highest existing value in the column, not the lowest missing number.

### FINAL ASSIGNMENT

* create table employee(
* id int primary key auto\_increment,
* last\_name varchar(100) not null,
* first\_name varchar(100) not null,
* middle\_name varchar(100),
* age int not null,
* current\_status varchar(100) not null default 'Employed');

## CRUD(Create, Read, Update, Delete) BASICS