SHOW TABLES AND DATABASE

* show databases;
* show tables;
* See the main command which helped you make this table(Helps you knowing the constraints etc)
  + SHOW CREATE TABLE person;
* Get info about indexes
  + SHOW INDEX FROM person;

USE DATBASE

* use vit\_adv\_java\_19mid0023;

SELECT COMMANDS

* SELECT \* from table\_name
* NULL
  + SELECT \* FROM customer\_info WHERE salary is null;
  + SELECT \* FROM customer\_info WHERE salary is not null;

DROP

* DROP TABLE table\_name;
* DROP DATABASE <database\_name>;
* ALTER TABLE table\_name

DROP COLUMN column\_name;

DELETE ALL THE ROWS/COLUMN OF A TABLE

* DELETE FROM table\_name;

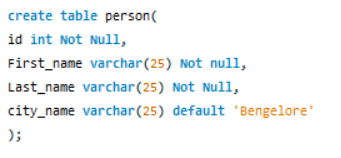
COMMENTS

* # Single line comment
* /\*Multiline comment\*/

CREATE DATABASE / TABLEr

* CREATE DATABASE DB\_NAME;
* CREATE TABLE table\_name ( column1 datatype constraint, column2 datatype constraint, ... columnN datatype constraint);

INSERT/ DELETING ROWS IN A TABLE

* insert into customer\_info(first\_name, last\_name ,salary) values ('John’, ‘Daniel’ ,5000), ('Krish','Naik',60000),('Darius’,'Bengali’,70000),('Chandan’, ‘Kumar',40000),('Ankit', 'Sharma’, NULL);
* DELETE FROM customer\_info where last\_name is null;
* DEFAULT VALUE
  + 

META DATA OF A TABLE

* DESCRIBE TABLE\_NAME;
* SELECT Field, Type FROM information\_schema.columns WHERE table\_name = 'customer\_info';

CONSTRAINTS

* create table customer\_info{id integer auto\_increment,first\_name varchar(25),last\_name varchar(25),salary integer,primary key(id)}
* Know your constraints
  + SHOW CREATE TABLE person;
  + SELECT \*

FROM INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS

WHERE TABLE\_SCHEMA = 'person'

AND TABLE\_NAME = 'customers';

* Unique based on two things
  + Alter table person add constraint uc\_person unique(age, first\_name);
* Modifying constraints
  + alter table student modify age int not null;
* Primary key
  + create table personl(

id int not null,

first\_name varchar(25) not null,

last\_name varchar(25),

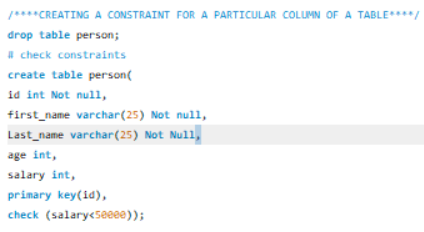
age int,

constraint pk\_person primary key(id,last\_name));

* + Alter table person1

add primary key(id)

* Dropping primary key
  + ALTER TABLE table\_name DROP PRIMARY KEY;
* Dropping a constraint
  + Alter table person drop index uc\_person;



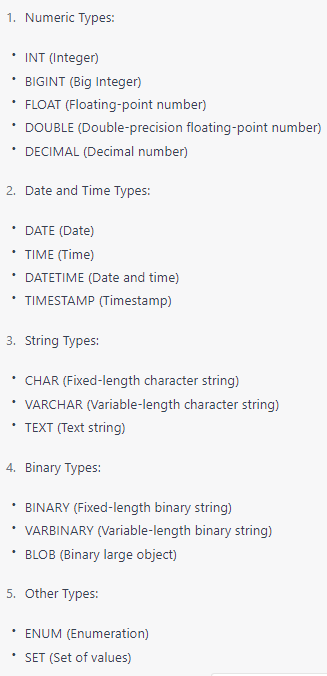
ALTER

* Auto\_increment(Setting value for auto increment)
  + ALTER TABLE customer\_info AUTO\_INCREMENT = 1;
* Adding/ Dropping columns
  + ALTER TABLE my\_table ADD COLUMN column1 INT, ADD COLUMN column2 VARCHAR(50), ADD COLUMN column3 DATE;
  + ALTER TABLE my\_table DROP COLUMN column1, DROP COLUMN column2, DROP COLUMN column3;
* Modify multiple datatypes
  + ALTER TABLE my\_table MODIFY COLUMN column1 VARCHAR(50), MODIFY COLUMN column2 INT, MODIFY COLUMN column3 DATE;
* Modifying constraints
  + alter table student modify age int not null;
* Adding unique after the table is created
  + Alter table person ADD unique(first\_name);
* Unique based on two things
  + Alter table person add constraint uc\_person unique(age, first\_name);
  + Here the constraint name is uc\_person
* Dropping a constraint
  + Alter table person drop index uc\_person;

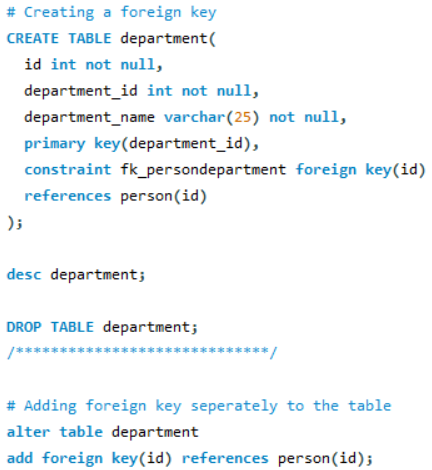
Update

* UPDATE customer\_info set salary = 50000 where id = 4;

DATATYPES



Foreign key



# CREATING A INDEX

create index index\_city\_first\_name

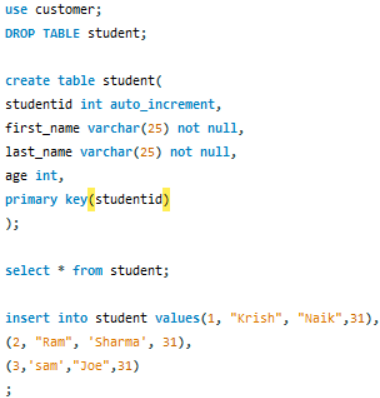
on person(first\_name,city\_name);

# Dropping Index

alter table student

drop index index\_age;

VIEW



Graphical user interface, text, application

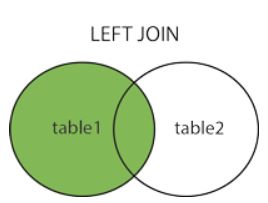
Description automatically generated

Text

Description automatically generated

JOIN

* CARTESIAN PRODUCT
  + select student.studentid, first\_name,last\_name,age,department\_name from student inner join department ;
* INNER JOIN
  + Diagram, venn diagram

    Description automatically generated
* FULL OUTER JOIN
  + SELECT column\_name(s)  
    FROM table1  
    FULL OUTER JOIN table2ON table1.column\_name = table2.column\_nameWHERE condition;
* LEFT JOIN
  + 
* RIGHT JOIN
  + Diagram, venn diagram

    Description automatically generated

W3 Schools

MySQL

* relational database management system (RDBMS).
* open-source.
* both small and large applications
* very fast, reliable, scalable
* cross-platform
* PHP is server side
* SQL is used to insert, search, update, and delete database records.

RDBMS

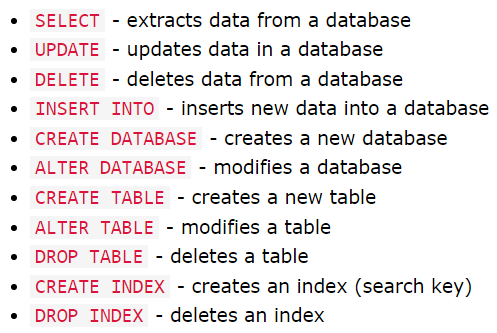
* maintain a relational database.
* Defines database relationships in the form of tables.
* The tables are related to each other - based on data common to each.
* Basis for all modern database systems such as MySQL, Microsoft SQL Server, Oracle, and Microsoft Access.
* uses [SQL queries](https://www.w3schools.com/sql/default.asp) to access the data in the database.

What is a Database Table?

* collection of related data entries
* consists of columns and rows
  + Column specific information about every record in the table.
  + Record (or row) is each individual entry

SEMICOLON

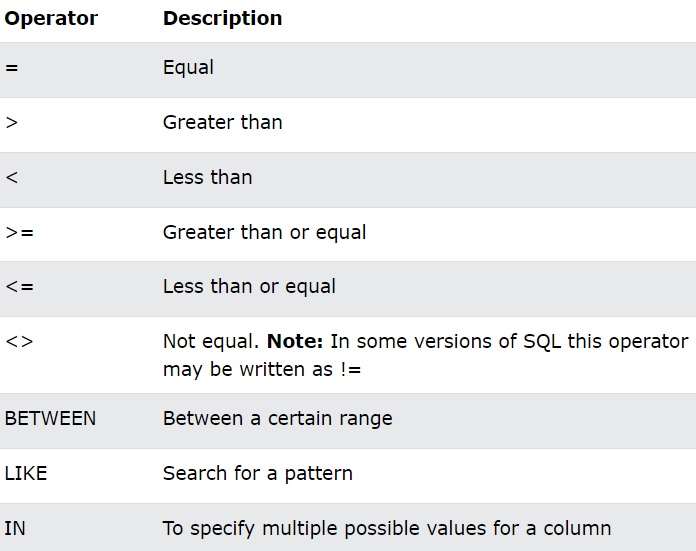
* Semicolon is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.



SELECT STATEMENT

* select data from a database.
* data returned is stored in a result table, called the result-set.
  + Select \* from table
  + Select column1, column2 From Table
* Select DISTINCT
  + return only distinct (different) values
  + column often contains many duplicate values
    - SELECT DISTINCT column1, column2, ...  
      FROM table\_name;
* number of different (distinct)
  + SELECT COUNT(DISTINCT COUNTRY) From Customers;

WHERE STATEMENT

* used to filter records
* 
* A picture containing graphical user interface

  Description automatically generatedA picture containing polygon

  Description automatically generated

## AND, OR and NOT Operators

## filter records based on more than one condition

## AND

## if all the conditions separated by AND are TRUE.

## OR

## if any of the conditions separated by OR is TRUE

## NOT

## displays a record if the condition(s) is NOT TRUE

## Chart Description automatically generated with low confidence

## ALIAS

* SELECT c.\* FROM CITY c WHERE ID = 1661;

BASIC STRUCTURE

