SHOW DATABASES;

CREATE DATABASE DB-NAME;

USE DB-NAME;

//CREATE TABLE

CREATE TABLE TB-NAME (

ID INT NOT NULL AUTO INCREMENT,

NAME VARCHAR(20)

);

//INSERT VALUES

INSERT INTO TB-NAME VALUES(2, “GDUGUH”);

//TO SHOW EVERYTHING INSIDE OF THE TABLE

SELECT \* FROM TB-NAME;

SELECT \* FROM DB-NAME.TB-NAME;

SELECT \* FROM TB-NAME;

//INCLUDING MORE CONDITIONS

SELECT \* FROM TB-NAME

WHERE CUSTOMER-ID = 1

ORDER BY FIRST-NAME;

SELECT \* FROM TB-NAME

-- WHERE CUSTOMER-ID = 1 //TO COMMENT THIS

ORDER BY FIRST-NAME;

SELECT FIRST-NAME, LAST-NAME

FROM TB-NAME;

SELECT FIRST-NAME, LAST-NAME, POINTS, POINTS + 10

FROM TB-NAME;

//USING ARITHMETIC OPERATION WITHOUT CHANGING THE VALUE

SELECT

FIRST-NAME,

LAST-NAME ,

POINTS,

(POINTS + 10 ) \* 100 AS DISCOUNT-FACTOR

FROM TB-NAME;

//BY USING AS WE CAN SHOW THE FIELD NAME AS WE WANT

//TO ADD A SPACE IN BETWEEN

SELECT

FIRST-NAME,

LAST-NAME ,

POINTS,

(POINTS + 10 ) \* 100 AS “DISCOUNT FACTOR”

FROM TB-NAME;

//WE CAN REMOVE DUPLICATES USING DISTINCT

SELECT DISTINCT NAME

FROM TB-NAME;

//WHERE

SELECT \*

FROM TB-NAME

WHERE POINTS>3000;

>

>=

<

<=

=

!= / <> // NOT EQUAL OPERATORS

SELECT \*

FROM TB-NAME

WHERE NAME = “BADSHAH”;

SELECT \*

FROM TB-NAME

WHERE BIRTHDATE > “1990-01-01”;

YEAR-MONTH-DAY

//AND

SELECT \*

FROM TB-NAME

WHERE POINTS>3000 AND BIRTHDAY = “2001-09-3”;

//OR

SELECT \*

FROM TB-NAME

WHERE POINTS>3000 OR BIRTHDAY = “2001-09-3”;

//WHERE NOT

SELECT \*

FROM TB-NAME

WHERE NOT (POINTS>3000 AND BIRTHDAY = “2001-09-3”);

//IN

//INSTEAD OF USING MULTIPLE OR OPERATOR WE CAN USE IN OPERATOR

SELECT \*

FROM TB-NAME

WHERE POINTS=3 OR POINTS =4 OR POINTS =5;

SELECT \*

FROM TB-NAME

WHERE POINTS IN (3, 4, 5);

//ORDER DOESN’T MATTER

SELECT \*

FROM TB-NAME

WHERE POINTS NOT IN (3, 4, 5);

//BETWEEN

SELECT \*

FROM TB-NAME

WHERE POINTS BETWEEN 1000 AND 3000;

//LIKE

SELECT \*

FROM TB-NAME

WHERE LAST-NAME LIKE ‘B%’;

//THIS IS LAST-NAME STARTING WITH B AND CAN HAVE ANY MORE LETTERS AFTER THAT

%B%

//FOR THIS… B CAN WE ANYWHERE OF THE STRING

\_Y

//THIS MEANS THE STRING SHOULD BE OF 2 LETTERS…WE DON’T CARE WHAT THE FIRST CHAR IS BUT THE SECOND CHAR HAS TO BE Y

//THE REGEXP OPERATOR

//IT IS USEFUL WHEN SEARCHING FOR STRINGS

//ALLOW US TO SEARCH FOR MORE COMPLEX PATTERNS

SELECT \*

FROM TB-NAME

-- WHERE LAST-NAME LIKE ‘%B%’

WHERE LAST-NAME REGEXP ‘B’ ;

^B

//MEANS LAST-NAME MUST START WITH B

B$

//SHOWS END OF STRING..I.E, STRING MUST END WITH B

B$ | ^S | A

//WE CAN DO LIKE THIS TOO

[GIM]E

GE/IE/ME

E[FMQ]

EF/EM/EQ

[A-H]Q

TAKING ALL FROM A TO H

//IS NULL

SELECT \* FROM TB-NAME

WHERE POINT IS NULL;

//WHERE THE FIELD IS EMPTY

IS NOT NULL

//FOR FIELD NOT EMPTY