**MUSIC LIBRARY DATABASE**

#To create database

CREATE DATABASE MUSIC\_LIBERARY;

#To select database

USE Music\_Liberary;

#To create table in database

CREATE TABLE Music

(

M\_Id int PRIMARY KEY,

M\_Name VARCHAR(20) not null,

Singer VARCHAR(20) not null,

R\_Date date not null

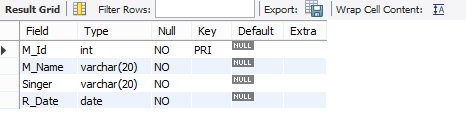
);

#To describe structure of table

DESC Music;

#To retrieve data from table

SELECT \* FROM Music;



#To add data in table

INSERT Music VALUES(1,'Sanam re','Arijit Singh','2020-02-10'),

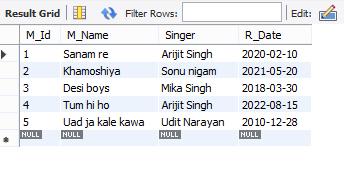
(2,'Khamoshiya','Sonu nigam','2021-05-20'),

(3,'Desi boys','Mika Singh','2018-03-30'),

(4,'Tum hi ho','Arijit Singh','2022-08-15'),

(5,'Uad ja kale kawa','Udit Narayan','2010-12-28');

SELECT \* FROM Music;



#To add new column to the table

ALTER TABLE Music

ADD COLUMN Movie\_name VARCHAR(20) not null;

SELECT \* FROM Music;

#To turn off safe mode

SET SQL\_SAFE\_UPDATES=0;

#To update values for new column

UPDATE Music

SET Movie\_name ='Sanam re'

WHERE M\_Id=1;

UPDATE Music

SET Movie\_name ='Sholey'

WHERE M\_Id=2;

UPDATE Music

SET Movie\_name ='Agnipath'

WHERE M\_Id=3;

UPDATE Music

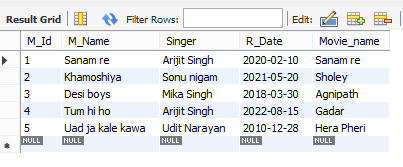
SET Movie\_name ='Gadar'

WHERE M\_Id=4;

UPDATE Music

SET Movie\_name ='Hera Pheri'

WHERE M\_Id=5;



SELECT \* FROM Music;

ALTER TABLE Music

ADD COLUMN Singer\_payment int not null;

SET SQL\_SAFE\_UPDATES=0;

UPDATE Music

SET Singer\_payment =100000

WHERE M\_Id=1;

UPDATE Music

SET Singer\_payment =120000

WHERE M\_Id=2;

UPDATE Music

SET Singer\_payment =590000

WHERE M\_Id=3;

UPDATE Music

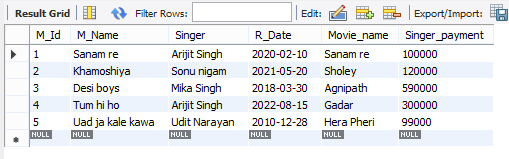
SET Singer\_payment =300000

WHERE M\_Id=4;

UPDATE Music

SET Singer\_payment =99000

WHERE M\_Id=5;

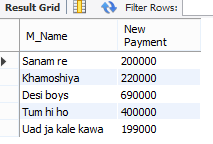


SELECT \* FROM Music;

#Arithmetic operators

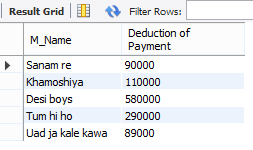
#ADDITION

SELECT M\_Name, Singer\_Payment+100000 AS 'New Payment' from Music;



#SUBTRACTION

SELECT M\_Name, Singer\_Payment-10000 AS 'Deduction of Payment' from Music;

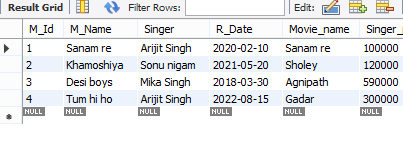


#RELATIONAL OPERATORS

#GREATER THAN OR EQUAL TO

SELECT \* FROM music

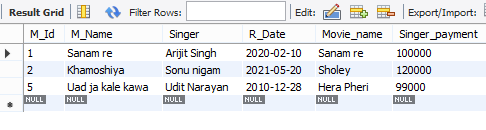
where Singer\_Payment>=100000;



#LESSER THAN OR EQUAL TO

SELECT \* FROM music

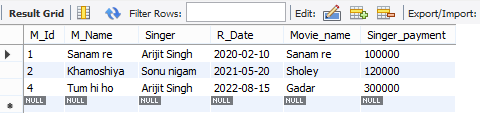
where Singer\_Payment<=200000;



#BETWEEN AND

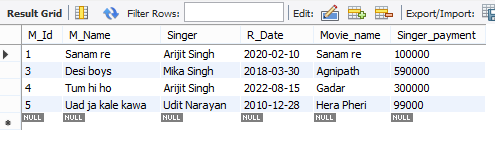
SELECT \* FROM music

where Singer\_Payment BETWEEN 100000 AND 300000;



SELECT \* FROM Music

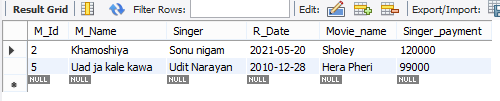
WHERE Singer in('Arijit Singh', 'Mika Singh', 'Udit Narayan');



#LIKE OPERATORS

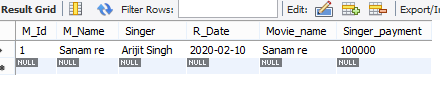
SELECT \* FROM Music

WHERE M\_Name LIKE '%A';



SELECT \* FROM Music

WHERE M\_Name LIKE '\_\_\_A%';

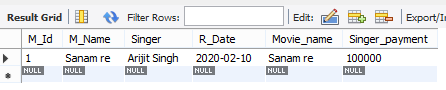


#LOGICAL OPERATORS

#AND

SELECT \* FROM Music

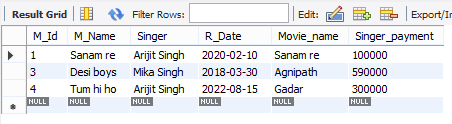
WHERE Singer='Arijit Singh' And M\_name='Sanam Re';



#OR

SELECT \* FROM Music

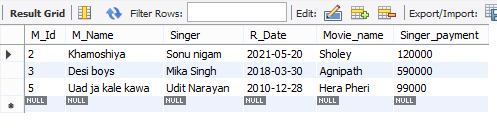
WHERE Singer='Arijit Singh' OR M\_name='Desi boys';



#NOT

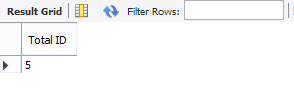
SELECT \* FROM Music

WHERE NOT Singer='Arijit Singh';

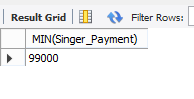


#AGGREGATE FUNCTIONS

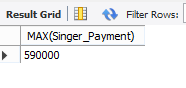
SELECT COUNT(M\_Id) AS 'Total ID' FROM Music ;



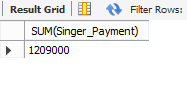
SELECT MIN(Singer\_Payment) FROM Music ;



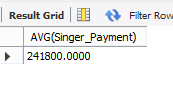
SELECT MAX(Singer\_Payment) FROM Music;



SELECT SUM(Singer\_Payment) FROM Music ;

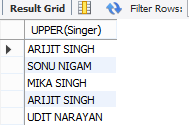


SELECT AVG(Singer\_Payment) FROM Music;

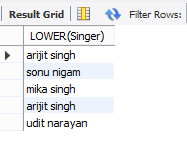


#SINGLE ROW FUNCTION

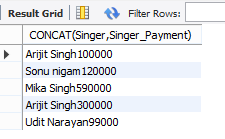
SELECT UPPER(Singer) FROM Music;



SELECT LOWER(Singer) FROM Music;

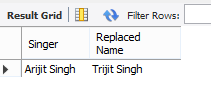


SELECT CONCAT(Singer,Singer\_Payment) FROM Music;



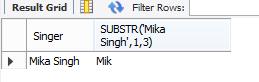
SELECT Singer, REPLACE('Arijit Singh','A','T') AS 'Replaced Name' FROM Music

WHERE M\_Id=1;



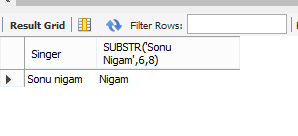
SELECT Singer, SUBSTR('Mika Singh',1,3) FROM Music

WHERE M\_Id=3;

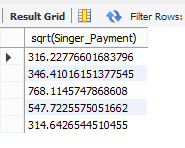


SELECT Singer, SUBSTR('Sonu Nigam',6,8) FROM Music

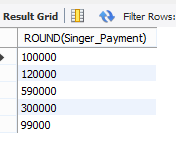
WHERE M\_Id=2;



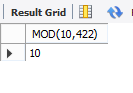
SELECT sqrt(Singer\_Payment) from Music;



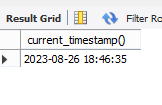
SELECT ROUND(Singer\_Payment) from Music;



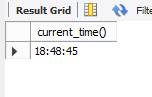
SELECT MOD(10,422);



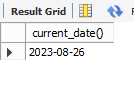
SELECT current\_timestamp();



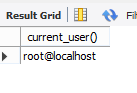
SELECT current\_time();



select current\_date();



SELECT current\_user();



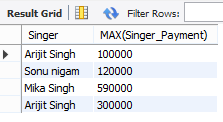
#GROUP BY CLAUASE

SELECT Singer, MAX(Singer\_Payment)

FROM Music

GROUP BY M\_Id

Having MAX(Singer\_Payment)>=100000;



#SUB QUERY

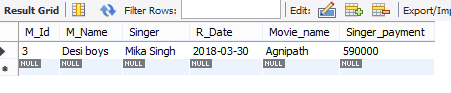
select \* from Music

WHERE Singer\_Payment=

(

select max(Singer\_Payment) from Music

);



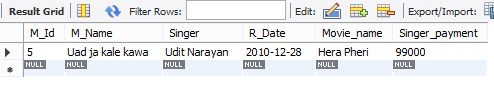
select \* from Music

WHERE Singer\_Payment=

(

select MIN(Singer\_Payment) from Music

);

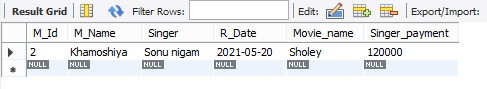


select \* from Music

ORDER BY Singer\_Payment desc

limit 1

offset 2;



#To create table

CREATE TABLE Music\_review

(

S\_Id int PRIMARY KEY ,

M\_lOCATION VARCHAR(20) NOT NULL,

Views int NOT NULL,

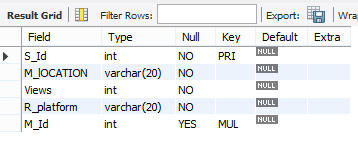
R\_platform VARCHAR(20) NOT NULL,

M\_Id int,

Foreign key (M\_Id) references Music(M\_Id)

);

DESC Music\_review;



INSERT Music\_review VALUES(1,'Shimla',2000000,'Youtube',1),

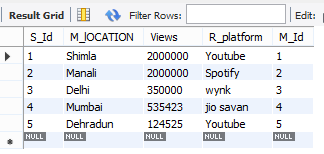
(2,'Manali',2000000,'Spotify',2),

(3,'Delhi',350000,'wynk',3),

(4,'Mumbai',535423,'jio savan',4),

(5,'Dehradun',124525,'Youtube',5);

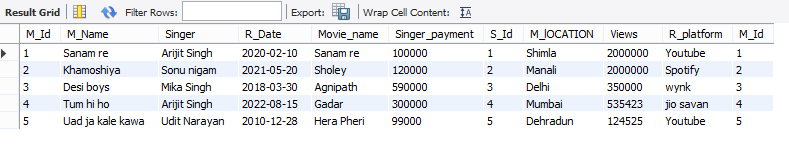
SELECT \* FROM Music\_review;



#JOINS

SELECT \* FROM Music,Music\_review

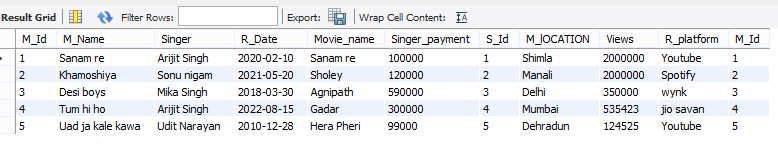
where Music.M\_Id=Music\_review.M\_Id;



SELECT \* FROM Music

JOIN Music\_review

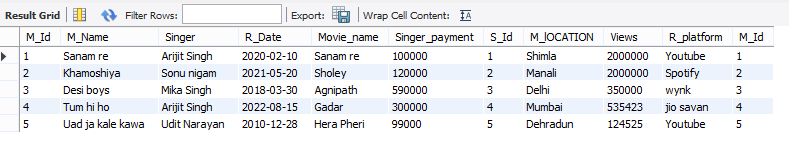
ON Music.M\_Id=Music\_review.M\_Id;



#INNER JOIN

SELECT \* FROM music INNER JOIN Music\_review

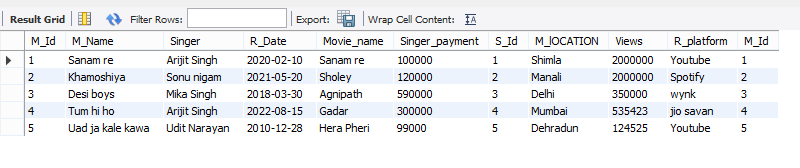
ON Music.M\_Id=Music\_review.M\_Id;



#OUTER JOIN

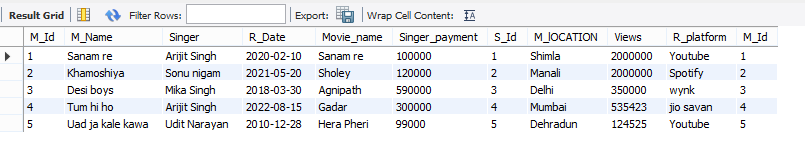
SELECT \* FROM Music LEFT OUTER JOIN Music\_review

ON Music.M\_Id = Music\_review.M\_Id;



SELECT \* FROM Music RIGHT OUTER JOIN Music\_review

ON Music.M\_Id = Music\_review.M\_Id;



**DONE BY**

Tanmay

Sowjanya bhat

Ajay singh

Devi dayal