**Netflix Database**

CBD\_2303\_FALL-2019

Database Design

Group no: 1

**OVERVIEW:**

Netflix is a content streaming service provided on internet. It can be accessed via website or the Netflix mobile application. It allows its users to watch different types of movies, tv shows and series.

There are three types of memberships:

* Basic
* Standard
* Premium

Basic plan contains one screen with HD resolution.

Standard plan contains two screens with FHD resolution.

Premium plan contains three screens with UHD resolution.

The Netflix database can be used in following ways:

* To retrieve the data of customer, their subscriptions and payment details.
* To search content available according to certain conditions.
* To retrieve the details of movies, tv shows and series.
* To offer discounts according to the payment cycle of customer.

**DIFFICULTIES FACED:**

* To find the number of tables for the database.
* Designing the ER diagrams and establishing the cardinality between them.
* Designing the queries according to the requirements.
* While designing the number of columns in the tables.

**LESSONS LEARNED:**

* How to create ER diagrams using software such as lucid charts.
* How to apply joins on multiple tables.
* How to fetch data by applying conditions.
* Use of different functions to retrieve data.
* Use of normalization.

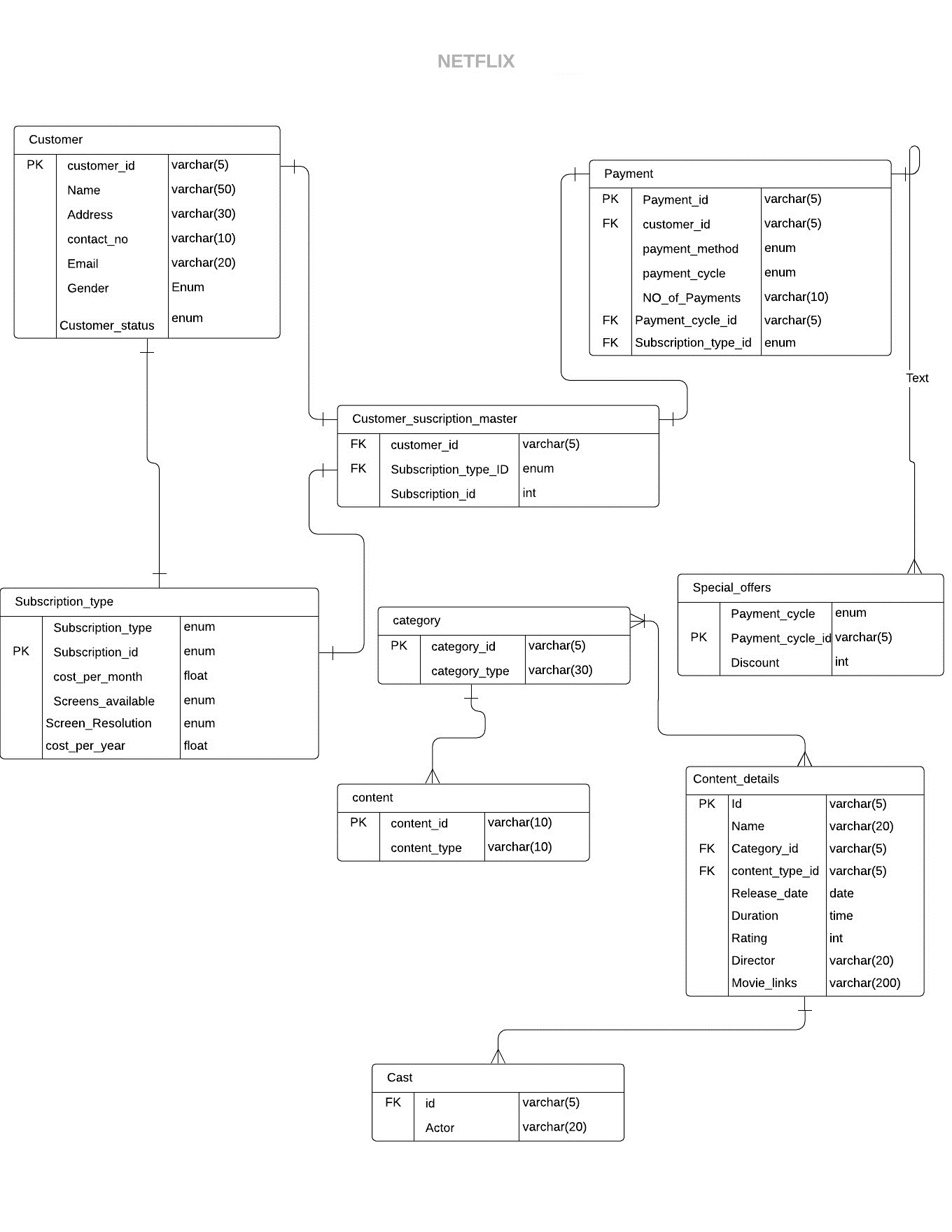
**TOOLS USED**

|  |  |
| --- | --- |
| Lucidchart | ER diagrams |
| Mockaroo | Inserting data |
| XAMPP | Launching Mysql |
| MS word | Documentation |
| MS paint | screenshots |

**ENTITY RELATIONSHIP DIAGRAM**

Entity Relationship Diagram is a type of structural diagram for use in database design, also known as ERD, ER Diagram or ER model. An ERD contains various symbols and connectors that display two important information: the major entities within the scope of the system, and the interrelationships between these entities.

Let us have a look at the entity-relationship diagram of Netflix Database and how the different entities relate to each other:



**TABLES**

CUSTOMER TABLE

|  |  |  |
| --- | --- | --- |
| FIELD NAME | DATA TYPE | CONSTRAINT |
| CUSTOMER\_ID | VARCHAR | PRIMARY KEY |
| NAME | VARCHAR |  |
| ADDRESS | VARCHAR |  |
| CONTACT\_NO | VARCHAR |  |
| EMAIL | VARCHAR |  |
| GENDER | ENUM |  |
| CUSTOMER\_STATUS | ENUM |  |

SUBSCRIPTION\_TYPE TABLE

|  |  |  |
| --- | --- | --- |
| FIELD NAME | DATA TYPE | CONSTRAINT |
| SUBSCRIPTION\_TYPE | ENUM |  |
| SUBSCRIPTION\_ID | ENUM | PRIMARY KEY |
| COST\_PER\_MONTH | FLOAT |  |
| SCREENS\_AVAILABLE | ENUM |  |
| SCREEN\_RESOLUTION | ENUM |  |
| COST\_PER\_YEAR | FLAOT |  |

CUSTOMER\_SUBSCRIPTION\_MASTER TABLE

|  |  |  |
| --- | --- | --- |
| FIELD NAME | DATA TYPE | CONSTRAINT |
| CUSTOMER\_ID | VARCHAR | FOREIGN KEY |
| SUBSCRIPTION\_TYPE\_ID | ENUM | FOREIGN KEY |
| SUBSCRIPTION\_ID | INT |  |

CATEGORY TABLE

|  |  |  |
| --- | --- | --- |
| FIELD NAME | DATA TYPE | CONSTRAINT |
| CATEGORY\_ID | VARCHAR | PRIMARY KEY |
| CATEGORY\_TYPE | VARCHAR |  |

CONTENT TABLE

|  |  |  |
| --- | --- | --- |
| FIELD NAME | DATA TYPE | CONSTRAINT |
| CONTENT\_ID | VARCHAR | PRIMARY KEY |
| CONTENT\_TYPE | VARCHAR |  |

CONTENT\_DETAILS TABLE

|  |  |  |
| --- | --- | --- |
| FIELD NAME | DATA TYPE | CONSTRAINT |
| ID | VARCHAR | PRIMARY KEY |
| NAME | VARCHAR |  |
| CATEGORY\_ID | VARCHAR | FOREIGN KEY |
| CONTENT\_TYPE\_ID | VARCHAR | FOREIGN KEY |
| RELEASE\_DATE | DATE |  |
| DURATION | TIME |  |
| RATING | INT |  |
| DIRECTOR | VARCHAR |  |

PAYMENT TABLE

|  |  |  |
| --- | --- | --- |
| FIELD NAME | DATA TYPE | CONSTRAINT |
| PAYMENT\_ID | VARCHAR | PRIMARY KEY |
| CUSTOMER\_ID | VARCHAR | FOREIGN KEY |
| PAYMENT\_METHOD | ENUM |  |
| PAYMENT\_CYCLE | ENUM |  |
| NO\_OF\_PAYMENTS | VARCHAR |  |
| PAYMENT\_CYCLE\_ID | VARCHAR | FOREIGN KEY |
| SUBSCRIPTION\_TYPE\_ID | ENUM | FOREIGN KEY |

SPECIAL\_OFFERS TABLE

|  |  |  |
| --- | --- | --- |
| FIELD NAME | DATA TYPE | CONSTRAINT |
| PAYMENT\_CYCLE\_ID | VARCHAR | PRIMARY KEY |
| PAYMENT\_CYCLE | ENUM |  |
| DISCOUNT | INT |  |

CAST TABLE

|  |  |  |
| --- | --- | --- |
| FIELD NAME | DATA TYPE | CONSTRAINT |
| ID | VARCHAR | FOREIGN KEY |
| ACTOR | VARCHAR |  |

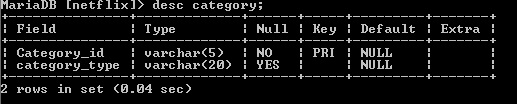
**Queries**

1. **Create table queries.**

create table category

(category\_id varchar(5) primary key,

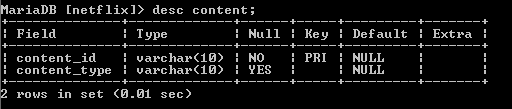
category\_type varchar(20));



create table content

(content\_id varchar(10) primary key,

content\_type varchar(10));



CREATE TABLE content\_detail

(id varchar(5) Primary Key, name varchar(20) ,

category\_id varchar(5) ,

content\_type\_id varchar(5) ,

release\_date date ,

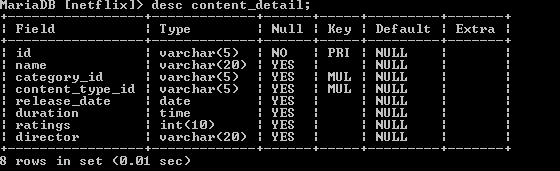
duration time ,

ratings int ,

director varchar(20) ,

FOREIGN KEY (category\_id) REFERENCES category(Category\_id),

FOREIGN KEY (content\_type\_id) REFERENCES content(content\_id));



CREATE TABLE customer

(customer\_id varchar(5) primary key,

Name varchar(50) ,

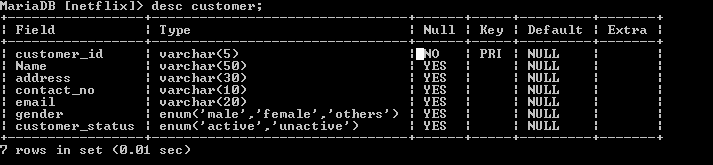
address varchar(30) ,

contact\_no varchar(10) ,

email varchar(20) ,

gender enum('male','female','others') ,

customer\_status enum('active','unactive'));



CREATE TABLE customer\_subscription\_master

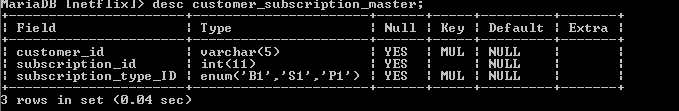
( customer\_id varchar(5) ,

subscription\_id int(11) ,

subscription\_type\_ID enum('B1','S1','P1') ,

FOREIGN KEY (customer\_id) REFERENCES customer (customer\_id),

FOREIGN KEY (subscription\_type\_ID) REFERENCES subscription\_type (subscription\_id));



CREATE TABLE payment

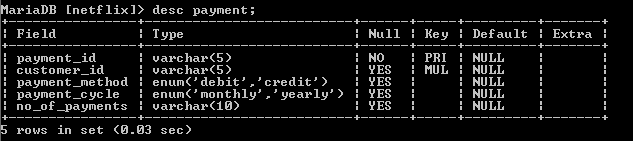
( payment\_id varchar(5) Primary key, customer\_id varchar(5) ,

payment\_method enum('debit','credit') ,

payment\_cycle enum('monthly','yearly') ,

no\_of\_payments varchar(10) ,

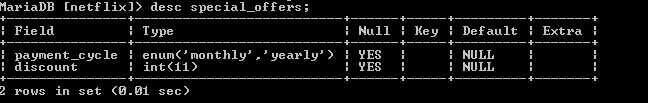
FOREIGN KEY (customer\_id) REFERENCES customer (customer\_id));



CREATE TABLE special\_offers

(payment\_cycle enum('monthly','yearly') ,

discount int );



CREATE TABLE subscription\_type

( subscription\_type enum('Basic','Standard','Premium') ,

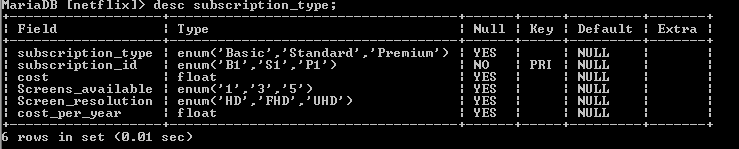
subscription\_id enum('B1','S1','P1') primary key ,

cost float ,

Screens\_available enum('1','3','5') ,

Screen\_resolution enum('HD','FHD','UHD') ,

cost\_per\_year float );

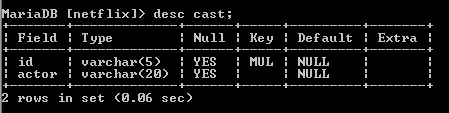


create table cast

(id varchar(5) ,

actor varchar(20),

foreign key(id) references content\_detail (id));



1. **Insert Queries**

insert into cast values ('1', 'Rock');



insert into customer

(customer\_id, name, address, contact\_no, email, gender, customer\_status)

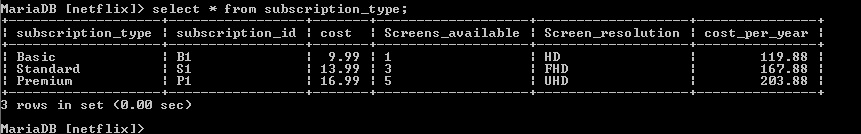
values (1, 'Antin', '07 Lake View Court', '911-887-0711', 'hpendrigh0@hugedomains.com', 'Male', 'Active');



insert into Subscription\_type

(subscription\_type, subscription\_id, cost, Screens\_available, Screen\_resolution)

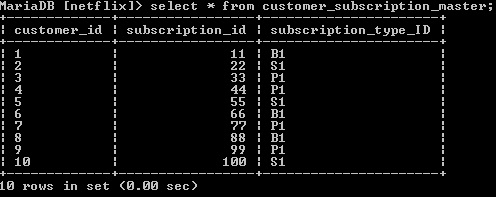
values ('Basic', 'B1', '9.99', '1', 'HD');



insert into customer\_subscription\_master

(customer\_id, subscription\_id, subscription\_type\_ID)

values (1, 11, 'B1');



insert into category

(category\_id, category\_type)

values ('M1', 'Movies');



insert into content

(content\_id, content\_type)

values ('H1', 'Horror');



insert into payment

(payment\_id, customer\_id, payment\_method, payment\_cycle, no\_of\_payments)

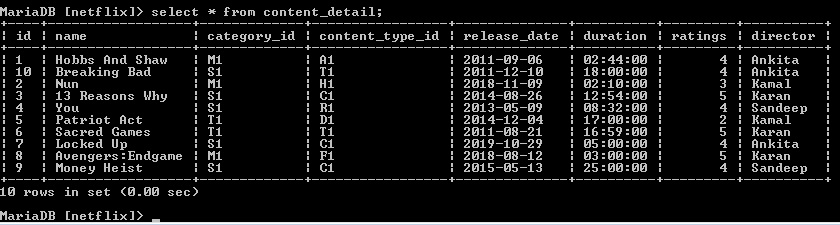
values ('p1', 1, 'debit','monthly', 2);



insert into content\_detail

(id, name, category\_id, Content\_type\_id, Director , release\_date, ratings,duration)

values (1, 'Hobbs And Shaw', 'M1', 'A1', 'Ankita', '2011-09-06', 4,'2:44:00');

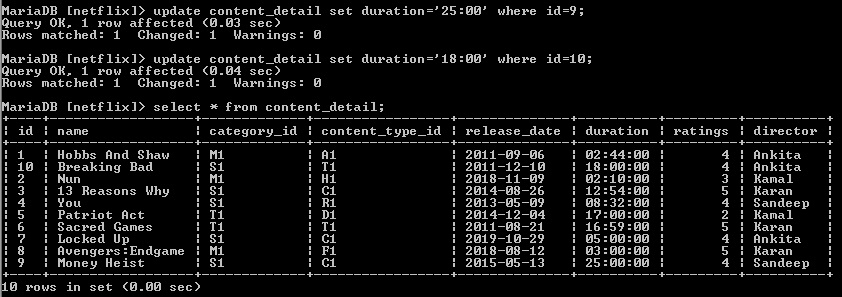


1. **Alter and update queries**

update content\_detail

set duration='18:00'

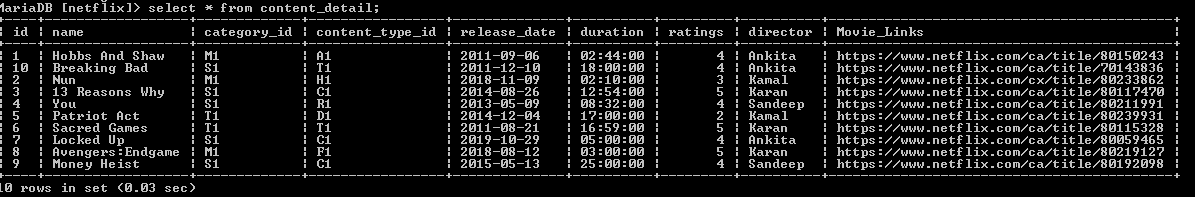
where id=10;



update content\_detail

set movie\_links='https://www.netflix.com/ca/title/80150243'

where id=1;

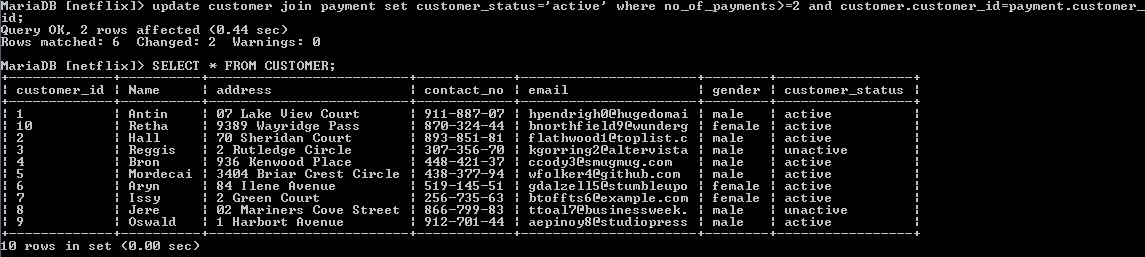


update customer join payment

set customer\_status='unactive'

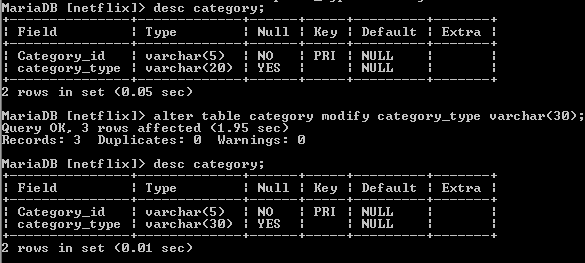
where no\_of\_payments<2

and customer.customer\_id=payment.customer\_id;



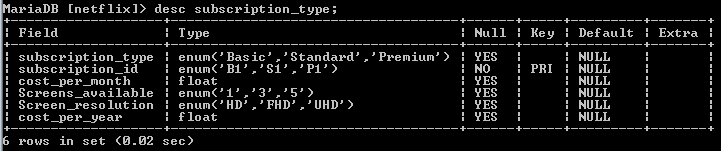
alter table category

modify category\_type varchar(30);



alter table subscription\_type

change cost cost\_per\_month float;



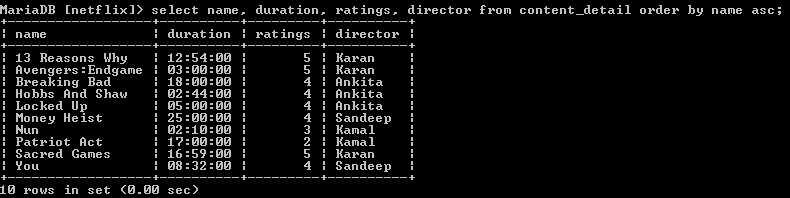
1. **Functions and clauses:**

Order by clause

select name, duration, ratings, director

from content\_detail

order by name asc;

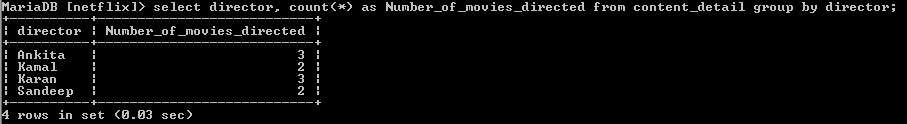


Group by clause

select director, count(\*) as Number\_of\_movies\_directed

from content\_detail

group by director;

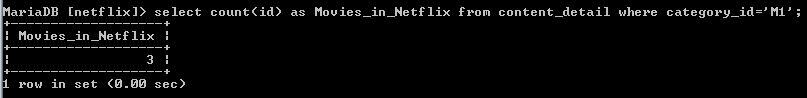


Count()

select count(id) as Movies\_in\_Netflix

from content\_detail

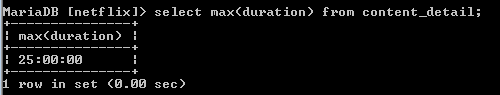
where category\_id='M1';



Max()

select max(duration)

from content\_detail;



Min()

select min(duration)

from content\_detail;

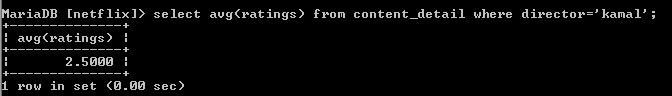


Avg()

select avg(ratings)

from content\_detail

where director='kamal';

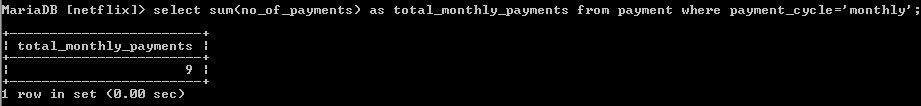


Sum()

select sum(no\_of\_payments) as total\_monthly\_payments

from payment

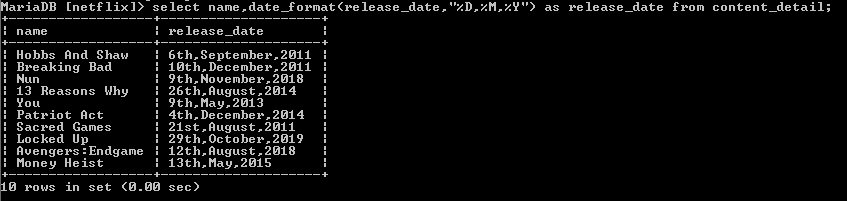
where payment\_cycle='monthly';



Format()

select name,date\_format(release\_date,"%D,%M,%Y") as release\_date

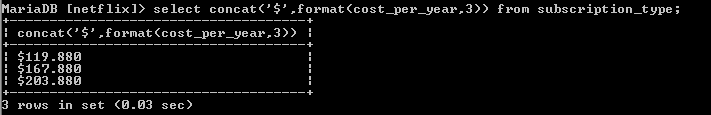
from content\_detail;



Concat()

select concat('$',format(cost\_per\_year,3))

from subscription\_type;



Year()

select name, release\_date

from content\_detail

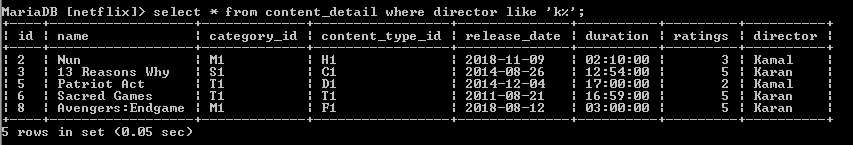
where year(release\_date) between 2014 and 2018;



Like

select \* from content\_detail

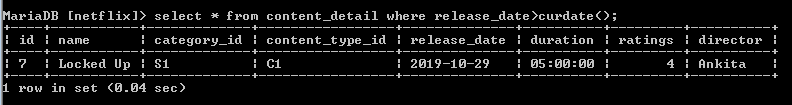
where director like 'k%';



Curdate()

select \* from content\_detail

where release\_date>curdate();



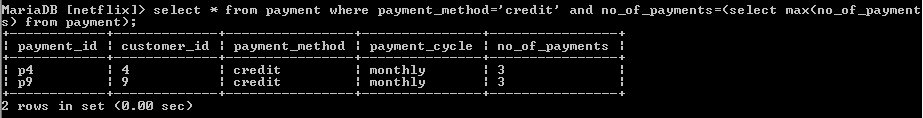
1. **Complex queries**

**Retrieve payment details of customer who made maximum number of payments.**

select \* from payment

where payment\_method='credit'

and no\_of\_payments=(select max(no\_of\_payments) from payment);

****

**Retrieve those customers with premium subscription.**

select name, customer.customer\_id, subscription\_type\_ID, subscription\_type.subscription\_type

from customer

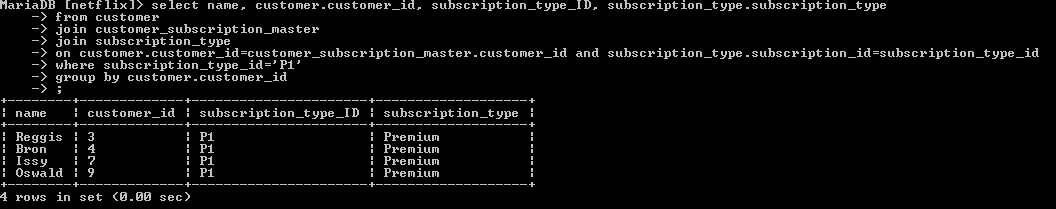
join customer\_subscription\_master

join subscription\_type

on customer.customer\_id=customer\_subscription\_master.customer\_id and subscription\_type.subscription\_id=subscription\_type\_id

where subscription\_type\_id='P1'

group by customer.customer\_id;

****

**Retrieve all subscription details of customer including name and id of customer.**

select customer.customer\_id, name, subscription\_type, cost, screens\_available,screen\_resolution

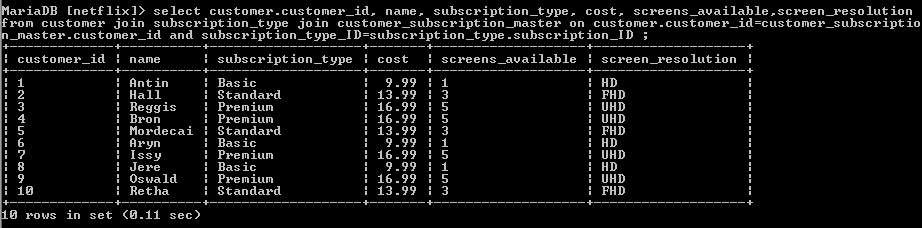
from customer

join subscription\_type join

customer\_subscription\_master

on customer.customer\_id=customer\_subscription\_master.customer\_id

and subscription\_type\_ID=subscription\_type.subscription\_ID ;



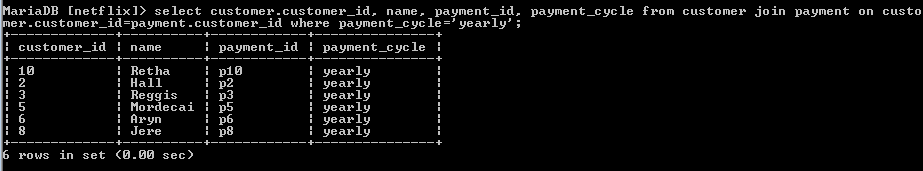
**Retrieve those customers who are paying yearly.**

select customer.customer\_id, name, payment\_id, payment\_cycle

from customer join payment

on customer.customer\_id=payment.customer\_id

where payment\_cycle='yearly';

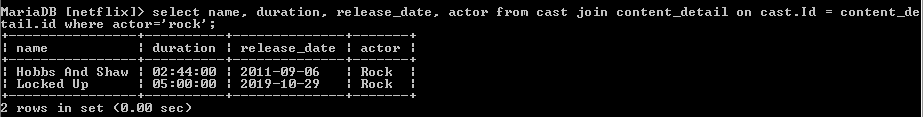


**Retrieve those movies, shows or series whose actor is rock.**

select name, duration, release\_date, actor

from cast join content\_detail on cast.Id = content\_detail.id

where actor='rock';



**Retrieve active female customers who watch horror movie.**

select \* from customer join content

on customer\_status='active' and gender='female'

where content\_id=(select content\_id from content where content\_type='Horror');



**Discount to those customers who are paying yearly with standard subscription.**

select name,customer.customer\_id, subscription\_type.subscription\_type,special\_offers.payment\_cycle\_id,

(cost\_per\_year\*special\_offers.discount/100) as yearly\_discount , subscription\_type.cost\_per\_year-(cost\_per\_year\*special\_offers.discount/100) as Amount\_After\_Discount

from customer join customer\_subscription\_master

join subscription\_type

join payment

join special\_offers

on customer.customer\_id=customer\_subscription\_master.customer\_id

and customer\_subscription\_master.subscription\_type\_id=payment.subscription\_type\_id

and customer\_subscription\_master.subscription\_type\_id=subscription\_type.subscrition\_id

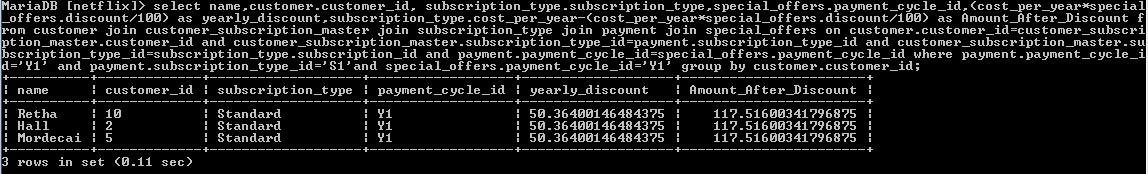
and payment.payment\_cycle\_id=special\_offers.payment\_cycle\_id

where payment.payment\_cycle\_id='Y1'

and payment.subscription\_type\_id='S1'

and special\_offers.payment\_cycle\_id='Y1'

group by customer.customer\_id;



1. **Views**

create view subscriptions\_by\_customers

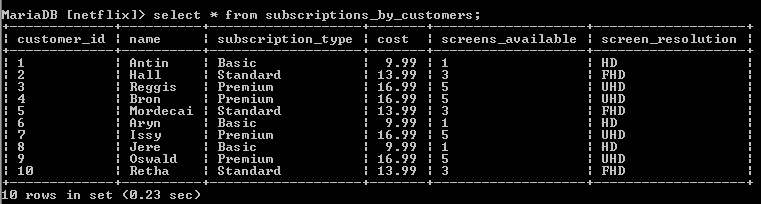
as select customer.customer\_id, name, subscription\_type, cost, screens\_available,screen\_resolution from customer

join subscription\_type join

customer\_subscription\_master

on customer.customer\_id=customer\_subscription\_master.customer\_id

and subscription\_type\_ID=subscription\_type.subscription\_ID ;

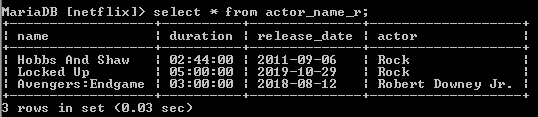


create view actor\_name\_r

as select name, duration, release\_date, actor from cast

join content\_detail on cast.Id = content\_detail.id

where actor like 'R%';



create view discount

as select name,customer.customer\_id, subscription\_type.subscription\_type, special\_offers.payment\_cycle\_id,(cost\_per\_year\*special\_offers.discount/100) as yearly\_discount,

subscription\_type.cost\_per\_year-(cost\_per\_year\*special\_offers.discount/100)

as Amount\_After\_Discount from customer

join customer\_subscription\_master

join subscription\_type

join payment

join special\_offers

on customer.customer\_id=customer\_subscription\_master.customer\_id

and customer\_subscription\_master.subscription\_type\_id=payment.subscription\_type\_id

and customer\_subscription\_master.subscription\_type\_id=subscription\_type.subscription\_id

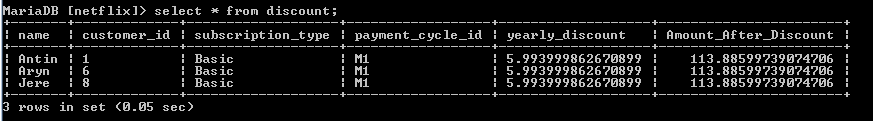
and payment.payment\_cycle\_id=special\_offers.payment\_cycle\_id

where payment.payment\_cycle\_id='M1'

and payment.subscription\_type\_id='B1'

and special\_offers.payment\_cycle\_id='M1'

group by customer.customer\_id;



**References:**

1. Netflix - 6/09/2019- <https://www.netflix.com/ca/>
2. Mockaroo - 9/09/2019- <https://mockaroo.com/>
3. Lucidchart - 8/09/2019- [https://www.lucidchart.com](https://www.lucidchart.com/)
4. Mysql tutorials - 10/09/2019- <http://www.mysqltutorial.org/>
5. W3schools - 10/09/2019- <https://www.w3schools.com/>