MODULE 1
MINI PROJECT

14/07/2025
VISHNU N
GitHub

Problem Scenario: Create Use case specification for the given requirement. Use the following DDL and DML to create Video Management database.

List of Tables:
Table #1: CUSTOMER_MASTER
Table #2: MOVIES_MASTER
Table #3: CUSTOMER_ISSUE_DETAILS
Table #4: LIBRARY_CARD_MASTER
Table #5: CUSTOMER_CARD_DETAILS

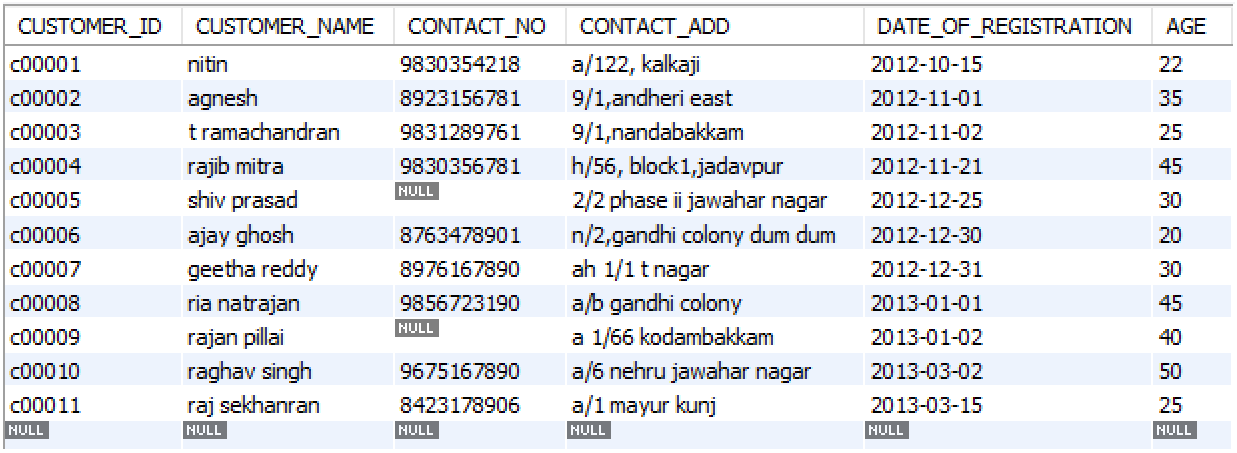
Table creation and values insertion:

```
create database vmd;
use vmd;

-- CUSTOMER_MASTER Table
CREATE TABLE CUSTOMER_MASTER (
   CUSTOMER_ID VARCHAR(10),
   CUSTOMER_NAME VARCHAR(30) NOT NULL,
   CONTACT_NO BIGINT(20),
   CONTACT_ADD VARCHAR(100),
   DATE_OF_REGISTRATION DATE NOT NULL,
   AGE VARCHAR(15) NOT NULL,
   CONSTRAINT MT_CTS1 PRIMARY KEY (CUSTOMER_ID)
);

INSERT INTO CUSTOMER_MASTER (CUSTOMER_ID, CUSTOMER_NAME, CONTACT_NO,
CONTACT_ADD, DATE_OF_REGISTRATION, AGE) VALUES
('c00001','nitin','9830354218','a/122, kalkaji','2012-10-15','22'),
('c00002','agnesh','8923156781','9/1,andheri east','2012-11-01','35'),
('c00003','t ramachandran','9831289761','9/1,nandabakkam','2012-11-02','25'),
('c00004','rajib mitra','9830356781','h/56, block1,jadavpur','2012-11-21','45'),
('c00005','shiv prasad',NULL,'2/2 phase ii jawahar nagar','2012-12-25','30'),
('c00006','ajay ghosh','8763478901','n/2,gandhi colony dum dum','2012-12-30','20'),
('c00007','geetha reddy','8976167890','ah 1/1 t nagar','2012-12-31','30'),
('c00008','ria natrajan','9856723190','a/b gandhi colony','2013-01-01','45'),
('c00009','rajan pillai',NULL,'a 1/66 kodambakkam','2013-01-02','40'),
('c00010','raghav singh','9675167890','a/6 nehru jawahar nagar','2013-03-02','50'),
('c00011','raj sekhanran','8423178906','a/1 mayur kunj','2013-03-15','25');
```

```
select * from CUSTOMER_MASTER;
```



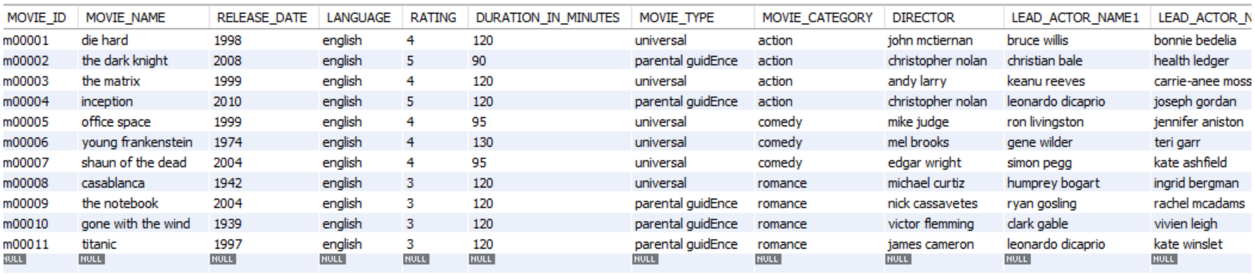
CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO	CONTACT_ADD	DATE_OF_REGISTRATION	AGE
c00001	nitin	9830354218	a/122, kalkaji	2012-10-15	22
c00002	agnesh	8923156781	9/1,andheri east	2012-11-01	35
c00003	t ramachandran	9831289761	9/1,nandabakkam	2012-11-02	25
c00004	rajib mitra	9830356781	h/56, block1,jadavpur	2012-11-21	45
c00005	shiv prasad	NULL	2/2 phase ii jawahar nagar	2012-12-25	30
c00006	ajay ghosh	8763478901	n/2,gandhi colony dum dum	2012-12-30	20
c00007	geetha reddy	8976167890	ah 1/1 t nagar	2012-12-31	30
c00008	ria natrajan	9856723190	a/b gandhi colony	2013-01-01	45
c00009	rajan pillai	NULL	a 1/66 kodambakkam	2013-01-02	40
c00010	raghav singh	9675167890	a/6 nehru jawahar nagar	2013-03-02	50
c00011	raj sekhanran	8423178906	a/1 mayur kunj	2013-03-15	25
NULL	NULL	NULL	NULL	NULL	NULL

-- MOVIES_MASTER Table

```
CREATE TABLE MOVIES_MASTER (
    MOVIE_ID VARCHAR(10),
    MOVIE_NAME VARCHAR(80) NOT NULL,
    RELEASE_DATE VARCHAR(30) NOT NULL,
    LANGUAGE VARCHAR(30),
    RATING INT(2),
    DURATION_IN_MINUTES VARCHAR(10) NOT NULL,
    MOVIE_TYPE VARCHAR(100),
    MOVIE_CATEGORY VARCHAR(40) NOT NULL,
    DIRECTOR VARCHAR(60) NOT NULL,
    LEAD_ACTOR_NAME1 VARCHAR(50) NOT NULL,
    LEAD_ACTOR_NAME2 VARCHAR(60) NOT NULL,
    RENTAL_COST BIGINT(10),
    CONSTRAINT MT_CTS4 PRIMARY KEY (MOVIE_ID)
);
```

```
INSERT INTO MOVIES_MASTER (MOVIE_ID, MOVIE_NAME, RELEASE_DATE, LANGUAGE,
RATING, DURATION_IN_MINUTES, MOVIE_TYPE, MOVIE_CATEGORY, DIRECTOR,
LEAD_ACTOR_NAME1, LEAD_ACTOR_NAME2, RENTAL_COST) VALUES
('m00001','die hard','1998','english',4,'120','universal','action','john mctiernan','bruce
willis','bonnie bedelia',100),
('m00002','the dark knight','2008','english',5,'90','parental guidEnce','action','christopher
nolan','christian bale','health ledger',100),
('m00003','the matrix','1999','english',4,'120','universal','action','andy larry','keanu
reeves','carrie-anee moss',100),
('m00004','inception','2010','english',5,'120','parental guidEnce','action','christopher
nolan','leonardo dicaprio','joseph gordan',100),
('m00005','office space','1999','english',4,'95','universal','comedy','mike judge','ron
livingston','jennifer aniston',100),
('m00006','young frankenstein','1974','english',4,'130','universal','comedy','mel brooks','gene
wilder','teri garr',100),
('m00007','shaun of the dead','2004','english',4,'95','universal','comedy','edgar wright','simon
pegg','kate ashfield',100),
('m00008','casablanca','1942','english',3,'120','universal','romance','michael curtiz','humprey
bogart','ingrid bergman',1000),
('m00009','the notebook','2004','english',3,'120','parental guidEnce','romance','nick
cassavetes','ryan gosling','rachel mcadams',100),
('m00010','gone with the wind','1939','english',3,'120','parental guidEnce','romance','victor
flemming','clark gable','vivien leigh',100),
('m00011','titanic','1997','english',3,'120','parental guidEnce','romance','james cameron','leonardo
dicaprio','kate winslet',100);

select * from MOVIES_MASTER;
```



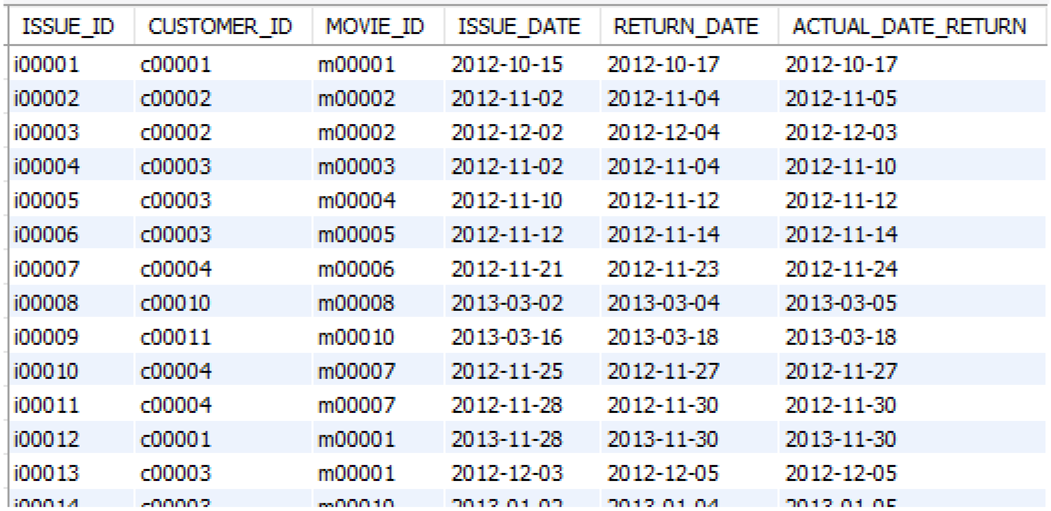
MOVIE_ID	MOVIE_NAME	RELEASE_DATE	LANGUAGE	RATING	DURATION_IN_MINUTES	MOVIE_TYPE	MOVIE_CATEGORY	DIRECTOR	LEAD_ACTOR_NAME1	LEAD_ACTOR_N
m00001	die hard	1998	english	4	120	universal	action	john mctiernan	bruce willis	bonnie bedelia
m00002	the dark knight	2008	english	5	90	parental guidEnce	action	christopher nolan	christian bale	health ledger
m00003	the matrix	1999	english	4	120	universal	action	andy larry	keanu reeves	carrie-anee moss
m00004	inception	2010	english	5	120	parental guidEnce	action	christopher nolan	leonardo dicaprio	joseph gordan
m00005	office space	1999	english	4	95	universal	comedy	mike judge	ron livingston	jennifer aniston
m00006	young frankenstein	1974	english	4	130	universal	comedy	mel brooks	gene wilder	teri garr
m00007	shaun of the dead	2004	english	4	95	universal	comedy	edgar wright	simon pegg	kate ashfield
m00008	casablanca	1942	english	3	120	universal	romance	michael curtiz	humprey bogart	ingrid bergman
m00009	the notebook	2004	english	3	120	parental guidEnce	romance	nick cassavetes	ryan gosling	rachel mcadams
m00010	gone with the wind	1939	english	3	120	parental guidEnce	romance	victor flemming	clark gable	vivien leigh
m00011	titanic	1997	english	3	120	parental guidEnce	romance	james cameron	leonardo dicaprio	kate winslet
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- CUSTOMER_ISSUE_DETAILE Table

```
CREATE TABLE CUSTOMER_ISSUE_DETAILS (
   ISSUE_ID VARCHAR(10) NOT NULL,
   CUSTOMER_ID VARCHAR(10) NOT NULL,
   MOVIE_ID VARCHAR(10),
   ISSUE_DATE DATE NOT NULL,
   RETURN_DATE DATE NOT NULL,
   ACTUAL_DATE_RETURN DATE NOT NULL,
   CONSTRAINT MT_CTS5 PRIMARY KEY (ISSUE_ID),
   CONSTRAINT MT_MEM FOREIGN KEY (CUSTOMER_ID) REFERENCES
CUSTOMER_MASTER(CUSTOMER_ID),
   CONSTRAINT MT_MEM1 FOREIGN KEY (MOVIE_ID) REFERENCES
MOVIES_MASTER(MOVIE_ID)
);

INSERT INTO CUSTOMER_ISSUE_DETAILS (ISSUE_ID, CUSTOMER_ID, MOVIE_ID,
ISSUE_DATE, RETURN_DATE, ACTUAL_DATE_RETURN) VALUES
('i00001','c00001','m00001','2012-10-15','2012-10-17','2012-10-17'),
('i00002','c00002','m00002','2012-11-02','2012-11-04','2012-11-05'),
('i00003','c00002','m00002','2012-12-02','2012-12-04','2012-12-03'),
('i00004','c00003','m00003','2012-11-02','2012-11-04','2012-11-10'),
('i00005','c00003','m00004','2012-11-10','2012-11-12','2012-11-12'),
('i00006','c00003','m00005','2012-11-12','2012-11-14','2012-11-14'),
('i00007','c00004','m00006','2012-11-21','2012-11-23','2012-11-24'),
('i00008','c00010','m00008','2013-03-02','2013-03-04','2013-03-05'),
('i00009','c00011','m00010','2013-03-16','2013-03-18','2013-03-18'),
('i00010','c00004','m00007','2012-11-25','2012-11-27','2012-11-27'),
('i00011','c00004','m00007','2012-11-28','2012-11-30','2012-11-30'),
('i00012','c00001','m00001','2013-11-28','2013-11-30','2013-11-30'),
('i00013','c00003','m00001','2012-12-03','2012-12-05','2012-12-05'),
('i00014','c00003','m00010','2013-01-02','2013-01-04','2013-01-05'),
('i00015','c00003','m00011','2013-02-03','2013-02-05','2013-02-06'),
('i00016','c00003','m00011','2013-03-05','2013-03-07','2013-03-07'),
('i00017','c00003','m00008','2013-04-15','2013-04-17','2013-04-17'),
('i00018','c00002','m00010','2015-01-15','2015-01-17','2015-01-17'),
('i00019','c00004','m00001','2012-11-15','2012-11-17','2012-11-17');
```

```
select * from CUSTOMER_ISSUE_DETAILS;
```



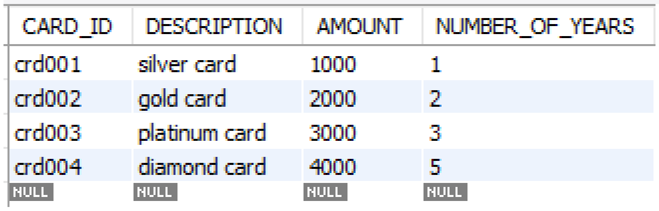
ISSUE_ID	CUSTOMER_ID	MOVIE_ID	ISSUE_DATE	RETURN_DATE	ACTUAL_DATE_RETURN
i00001	c00001	m00001	2012-10-15	2012-10-17	2012-10-17
i00002	c00002	m00002	2012-11-02	2012-11-04	2012-11-05
i00003	c00002	m00002	2012-12-02	2012-12-04	2012-12-03
i00004	c00003	m00003	2012-11-02	2012-11-04	2012-11-10
i00005	c00003	m00004	2012-11-10	2012-11-12	2012-11-12
i00006	c00003	m00005	2012-11-12	2012-11-14	2012-11-14
i00007	c00004	m00006	2012-11-21	2012-11-23	2012-11-24
i00008	c00010	m00008	2013-03-02	2013-03-04	2013-03-05
i00009	c00011	m00010	2013-03-16	2013-03-18	2013-03-18
i00010	c00004	m00007	2012-11-25	2012-11-27	2012-11-27
i00011	c00004	m00007	2012-11-28	2012-11-30	2012-11-30
i00012	c00001	m00001	2013-11-28	2013-11-30	2013-11-30
i00013	c00003	m00001	2012-12-03	2012-12-05	2012-12-05
i00014	c00003	m00010	2013-01-02	2013-01-04	2013-01-05

-- LIBRARY_CARD_MASTER

```
CREATE TABLE LIBRARY_CARD_MASTER (
   CARD_ID VARCHAR(10),
   DESCRIPTION VARCHAR(30) NOT NULL,
   AMOUNT BIGINT(50),
   NUMBER_OF_YEARS BIGINT(10) NOT NULL,
   CONSTRAINT MT_CTS2 PRIMARY KEY (CARD_ID)
);

INSERT INTO LIBRARY_CARD_MASTER (CARD_ID, DESCRIPTION, AMOUNT,
NUMBER_OF_YEARS) VALUES
('crd001','silver card',1000,1),
('crd002','gold card',2000,2),
('crd003','platinum card',3000,3),
('crd004','diamond card',4000,5);

select * from LIBRARY_CARD_MASTER;
```

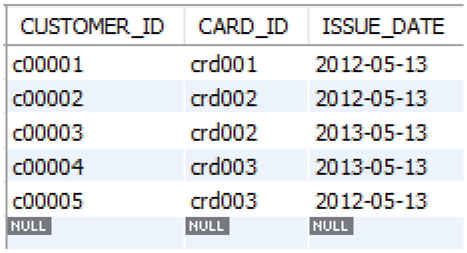


CARD_ID	DESCRIPTION	AMOUNT	NUMBER_OF_YEARS
crd001	silver card	1000	1
crd002	gold card	2000	2
crd003	platinum card	3000	3
crd004	diamond card	4000	5
NULL	NULL	NULL	NULL

```
-- CUSTOMER_CARD_DETAILS table
CREATE TABLE CUSTOMER_CARD_DETAILS (
   CUSTOMER_ID VARCHAR(10),
   CARD_ID VARCHAR(10),
   ISSUE_DATE DATE NOT NULL,
   CONSTRAINT MT_CTS3 PRIMARY KEY (CUSTOMER_ID),
   CONSTRAINT MT_CTS41 FOREIGN KEY (CUSTOMER_ID) REFERENCES
CUSTOMER_MASTER(CUSTOMER_ID),
   CONSTRAINT MT_CTS42 FOREIGN KEY (CARD_ID) REFERENCES
LIBRARY_CARD_MASTER(CARD_ID)
);

INSERT INTO CUSTOMER_CARD_DETAILS (CUSTOMER_ID, CARD_ID, ISSUE_DATE)
VALUES
('c00001','crd001','2012-05-13'),
('c00002','crd002','2012-05-13'),
('c00003','crd002','2013-05-13'),
('c00004','crd003','2013-05-13'),
('c00005','crd003','2012-05-13');

select * from CUSTOMER_CARD_DETAILS;
```

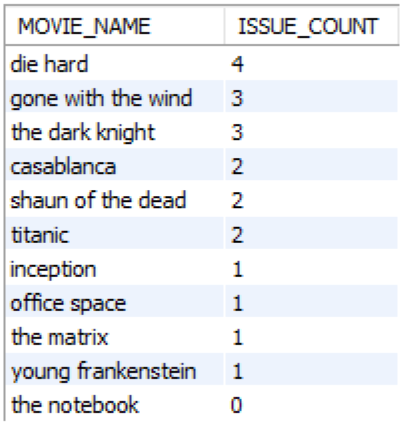


CUSTOMER_ID	CARD_ID	ISSUE_DATE
c00001	crd001	2012-05-13
c00002	crd002	2012-05-13
c00003	crd002	2013-05-13
c00004	crd003	2013-05-13
c00005	crd003	2012-05-13
NULL	NULL	NULL

Queries:

1. Write a query to display movie names and number of times that movie is issued to customers. In case movies are never issued to customers display number of times as 0. Display the details in sorted order based on number of times (in descending order) and then by movie name (in ascending order). The Alias name for the number of movies issued is ISSUE_COUNT.

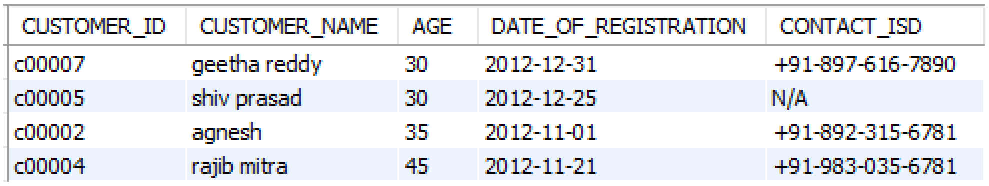
```
select M.MOVIE_NAME, count(CI.ISSUE_ID) as ISSUE_COUNT from MOVIES_MASTER M
left join CUSTOMER_ISSUE_DETAILS CI on M.MOVIE_ID = CI.MOVIE_ID
group by M.MOVIE_ID, M.MOVIE_NAME
order by ISSUE_COUNT desc, M.MOVIE_NAME asc;
```



MOVIE_NAME	ISSUE_COUNT
die hard	4
gone with the wind	3
the dark knight	3
casablanca	2
shaun of the dead	2
titanic	2
inception	1
office space	1
the matrix	1
young frankenstein	1
the notebook	0

2. Write a query to display id, name, age, contact no of customers whose age is greater than 25 and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name. Customers age > 25 and registered in 2012 with formatted contact number

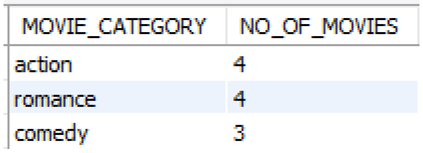
```
select * from CUSTOMER_MASTER;
select CUSTOMER_ID, CUSTOMER_NAME, AGE, DATE_OF_REGISTRATION,
case
        when CONTACT_NO is not null
   then concat('+91-',
                   SUBSTRING(CONTACT_NO, 1, 3), '-',
                   SUBSTRING(CONTACT_NO, 4, 3), '-',
                   SUBSTRING(CONTACT_NO, 7, 4)) -- +91-000-000-0000
        else 'N/A'
end as CONTACT_ISD
from CUSTOMER_MASTER
where AGE > 25 and year(DATE_OF_REGISTRATION) = 2012
order by AGE asc, CUSTOMER_NAME asc;
```



CUSTOMER_ID	CUSTOMER_NAME	AGE	DATE_OF_REGISTRATION	CONTACT_ISD
c00007	geetha reddy	30	2012-12-31	+91-897-616-7890
c00005	shiv prasad	30	2012-12-25	N/A
c00002	agnesh	35	2012-11-01	+91-892-315-6781
c00004	rajib mitra	45	2012-11-21	+91-983-035-6781

3. Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order. Hint: Use NO_OF_MOVIES as alias name for number of movies.

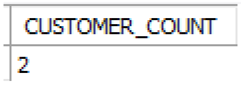
```
select MOVIE_CATEGORY, count(*) AS NO_OF_MOVIES from MOVIES_MASTER
group by MOVIE_CATEGORY
order by NO_OF_MOVIES desc, MOVIE_CATEGORY asc;
```



MOVIE_CATEGORY	NO_OF_MOVIES
action	4
romance	4
comedy	3

4. Write a query to display the number of customers having card with description “Gold card”. Use CUSTOMER_COUNT as alias name for number of customers.

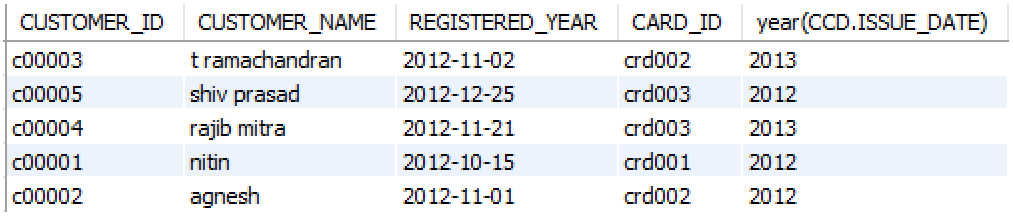
```
select * from library_card_master;
select count(*) AS CUSTOMER_COUNT from CUSTOMER_CARD_DETAILS CCD
join LIBRARY_CARD_MASTER LCM on CCD.CARD_ID = LCM.CARD_ID
where LCM.DESCRIPTION = 'gold card';
```



CUSTOMER_COUNT
2

5. Write a query to display the customer id, customer name, year of registration, library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order. Use REGISTERED_YEAR as alias name for year of registration.

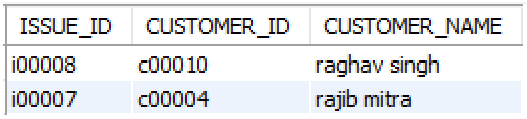
```
select CM.CUSTOMER_ID, CM.CUSTOMER_NAME, CM.DATE_OF_REGISTRATION as
REGISTERED_YEAR, CCD.CARD_ID, year(CCD.ISSUE_DATE)
from CUSTOMER_MASTER CM
join CUSTOMER_CARD_DETAILS CCD on CM.CUSTOMER_ID = CCD.CUSTOMER_ID
order by CM.CUSTOMER_NAME desc;
```



CUSTOMER_ID	CUSTOMER_NAME	REGISTERED_YEAR	CARD_ID	year(CCD.ISSUE_DATE)
c00003	t ramachandran	2012-11-02	crd002	2013
c00005	shiv prasad	2012-12-25	crd003	2012
c00004	rajib mitra	2012-11-21	crd003	2013
c00001	nitin	2012-10-15	crd001	2012
c00002	agnesh	2012-11-01	crd002	2012

6. Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer. Display the records sorted in ascending order based on customer name. fine = actual date > return date

select CID.ISSUE_ID, CM.CUSTOMER_ID, CM.CUSTOMER_NAME
from CUSTOMER_ISSUE_DETAILS CID
join CUSTOMER_MASTER CM on CID.CUSTOMER_ID = CM.CUSTOMER_ID
where CM.CUSTOMER_NAME like 'r%' and CID.ACTUAL_DATE_RETURN > CID.RETURN_DATE
order by CM.CUSTOMER_NAME asc;



ISSUE_ID	CUSTOMER_ID	CUSTOMER_NAME
i00008	c00010	raghav singh
i00007	c00004	rajib mitra

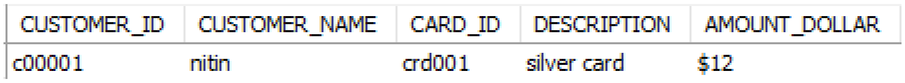
7. Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered. For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details. AMOUNT_DOLLAR = amount/85.8 and round it to zero decimal places and display as $Amount. Example Assume 500 is the amount then dollar value will be $10. Use AMOUNT_DOLLAR as alias name for amount in dollar. Display the records in ascending order based on customer name. customer who took movie on the same day library card issued.

select * from CUSTOMER_MASTER;
select * from CUSTOMER_CARD_DETAILS;
select * from LIBRARY_CARD_MASTER;
select * from CUSTOMER_ISSUE_DETAILS;

insert into CUSTOMER_ISSUE_DETAILS values ('i99999', 'c00001', 'm00002', '2012-05-13', '2012-05-15', '2012-05-15');

select CM.CUSTOMER_ID, CM.CUSTOMER_NAME, CCD.CARD_ID, LCM.DESCRIPTION, concat('$', round(LCM.AMOUNT / 85.8, 0)) as AMOUNT_DOLLAR
from CUSTOMER_MASTER CM
join CUSTOMER_CARD_DETAILS CCD on CM.CUSTOMER_ID = CCD.CUSTOMER_ID

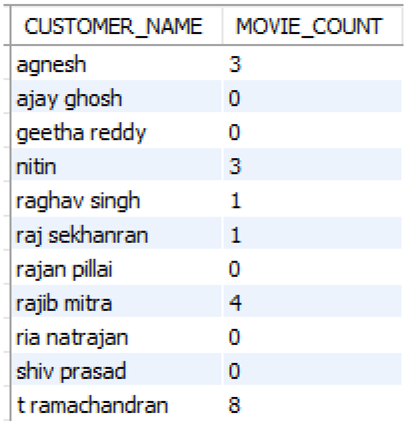
```
join LIBRARY_CARD_MASTER LCM on CCD.CARD_ID = LCM.CARD_ID
join CUSTOMER_ISSUE_DETAILS CID on CM.CUSTOMER_ID = CID.CUSTOMER_ID
where CCD.ISSUE_DATE = CID.ISSUE_DATE
order by CM.CUSTOMER_NAME asc;
```



CUSTOMER_ID	CUSTOMER_NAME	CARD_ID	DESCRIPTION	AMOUNT_DOLLAR
c00001	nitin	crd001	silver card	$12

8. Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0. Use MOVIE_COUNT as alias name for number of movies issued.

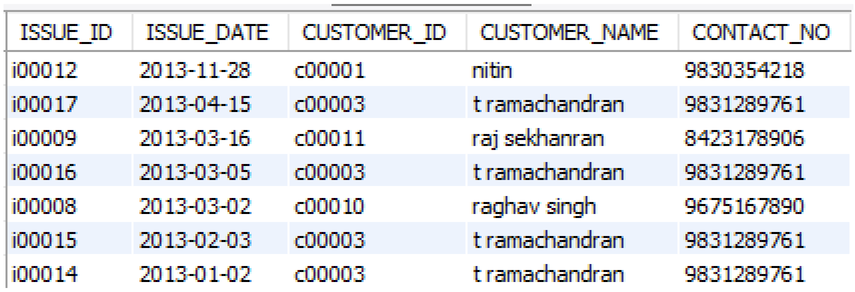
```
select CM.CUSTOMER_NAME, count(CID.ISSUE_ID) as MOVIE_COUNT
from CUSTOMER_MASTER CM
left join CUSTOMER_ISSUE_DETAILS CID on CM.CUSTOMER_ID = CID.CUSTOMER_ID
group by CM.CUSTOMER_ID, CM.CUSTOMER_NAME
order by CM.CUSTOMER_NAME asc;
```



CUSTOMER_NAME	MOVIE_COUNT
agnesh	3
ajay ghosh	0
geetha reddy	0
nitin	3
raghav singh	1
raj sekhanran	1
rajan pillai	0
rajib mitra	4
ria natrajan	0
shiv prasad	0
t ramachandran	8

9. Write a query to display the issue id, issue date, customer id, customer name and contact number for videos that are issued in the year 2013. Display the records in descending order based on issue date of the video.

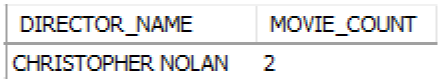
```
select CID.ISSUE_ID, CID.ISSUE_DATE, CM.CUSTOMER_ID, CM.CUSTOMER_NAME,
CM.CONTACT_NO
from CUSTOMER_ISSUE_DETAILS CID
join CUSTOMER_MASTER CM on CID.CUSTOMER_ID = CM.CUSTOMER_ID
where year(CID.ISSUE_DATE) = 2013
order by CID.ISSUE_DATE desc;
```



ISSUE_ID	ISSUE_DATE	CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO
i00012	2013-11-28	c00001	nitin	9830354218
i00017	2013-04-15	c00003	t ramachandran	9831289761
i00009	2013-03-16	c00011	raj sekhanran	8423178906
i00016	2013-03-05	c00003	t ramachandran	9831289761
i00008	2013-03-02	c00010	raghav singh	9675167890
i00015	2013-02-03	c00003	t ramachandran	9831289761
i00014	2013-01-02	c00003	t ramachandran	9831289761

10. Write a query to display the director's name, number of movies directed by the director who directed more than one movie. Display the director name in capital letters. Use DIRECTOR_NAME as alias name for director name column Display the records sorted in ascending order based on director_name.

```
-- select * from MOVIES_MASTER where DIRECTOR = 'christopher nolan';
select upper(DIRECTOR) as DIRECTOR_NAME, count(*) as MOVIE_COUNT
from MOVIES_MASTER
group by DIRECTOR
having count(*) > 1
order by DIRECTOR_NAME asc;
```



DIRECTOR_NAME	MOVIE_COUNT
CHRISTOPHER NOLAN	2