Abhay Dnyanoba Chavan
PRN : 250240520001

1.Create MySQL user by your name and create your own default database

Create user & databases & granting permission privilege.

step 1. Login with "root" user.

```
cmd> mysql -u root -p
Password: cdac
```

step 2. Create a new user.

```
mysql> CREATE USER dac1@localhost IDENTIFIED BY 'cdac';
```

step 3. Create a new database/schema.

```
mysql> CREATE DATABASE classwork;
mysql> SHOW DATABASES;
```

step 4. Give all permissions to the new user on the new database.

```
mysql> GRANT ALL PRIVILEGES ON classwork.* TO dac1@localhost;
mysql> FLUSH PRIVILEGES;
```

step 5. mysql> EXIT

step 1. Login with new user and password on MySQL CLI.

```
cmd> mysql -u dac1 -p
Password: cdac
```

2. Solve all Exercises and Assignments with the above user

3. MySQL - SQL - Exercises

SQL_Exercise1

4. MySQL - SQL - Assignments

SQL_Asgn1 - SQL_Asgn3

5. www.oracle.com

Create Oracle Account

Signup for MySQL, Java, and Oracle Newsletters

MySQL Magazine, Oracle Magazine, and Java Magazine

6.MySQL PDFs:-

Intro to MySQL, MySQL Datatypes

7. Download MySQL, Java, and Oracle Documentation

https://dev.mysql.com/doc/refman/9.0/en/

https://docs.oracle.com

8. '1995-01-15' or '95-01-15'

Why is 1970 the cut-off year?

Assignment –1

Creating sample tables and inserting values.

Create the following tables with the given structures and insert sample data as specified: -

A) SALESPEOPLE

Snum	int(4)
Sname	varchar(10)
City	varchar(10)
Comm	float(3,2)

```
 create table SALESPEOPLE(
     Snum int(4),
     Sname varchar(10),
     City varchar(10),
     Comm float(3,2)
     );
```

B) CUSTOMERS

Cnum	int(4)
Cname	varchar(10)
City	varchar(10)
Rating	int(4)
Snum	int(4)

```
create table customer(
    -> Cnum int(4),
    -> Cname varchar(10),
    -> City varchar(10),
    -> Rating int(4),
    -> Snum int(4));
```

C) ORDERS

Onum	int(4)
Amt	float(7,2)
Odate	date
Cnum	int(4)
Snum	int(4)

```
create table orders (
    Onum int(4),
    Amt float(7,2),
Odate date,
   Cnum int(4),
   Snum int(4));
```

SALES PEOPLE

SNUM	SNAME	CITY	COMM
1001	Peel	London	.12
1002	Serres	San Jose	.13
1004	Motika	London	.11
1007	Rifkin	Barcelona	.15
1003	Axelrod	New York	.10

```
 insert into salespeople values
    -> (1002,'serres','San Jose','0.13'),
    -> (1004,'Motika','London','0.11'),
    -> (1007,'Rifkin','Barcelone','0.15'),
    -> (1003,'Axelrod','New York','0.10');
```

CUSTOMERS

CNUM	CNAME	CITY	RATING	SNUM
2001	Hoffman	London	100	1001
2002	Giovanni	Rome	200	1003
2003	Liu	San Jose	200	1002
2004	Grass	Berlin	300	1002
2006	Clemens	London	100	1001
2008	Cisneros	San Jose	300	1007
2007	Pereira	Rome	100	1004

```
insert into customer values
   (2001,'Hoffman','London',100,1001),
     (2002,'Giovanni','Rome',200,1003),
    (2003,'Liv','San Jose',200,1002),
    (2004,'Grass','Berlin',300,1002),
    (2006,'Clemens','London',100,1001),
    (2008,'Cisneros','Sen Jose',300,1007),
    (2007,'Pereira','Rome',100,1004);
```

ORDERS

ONUM	AMT	ODATE	CNUM	SNUM
3001	18.69	03-OCT-1990	2008	1007
3003	767.19	03- OCT -1990	2001	1001
3002	1900.10	03- OCT -1990	2007	1004
3005	5160.45	03- OCT -1990	2003	1002
3006	1098.16	03- OCT -1990	2008	1007
3009	1713.23	04- OCT -1990	2002	1003
3007	75.75	04- OCT -1990	2004	1002

3008	**4723.00**	**05- OCT -1990**	**2006**	**1001**
3010	**1309.95**	**06- OCT -1990**	**2004**	**1002**
3011	**9891.88**	**06- OCT -1990**	**2006**	**1001**

```
 insert into orders values
     (3003,767.19,'1990-10-03',2001,1001),
    (3002,1900.10,'1990-10-03',2007,1004),
 (3005,5160.45,'1990-10-03',2003,1002),
(3006,1098.16,'1990-10-03',2008,1007),
  (3009,1713.23,'1990-10-04',2002,1003),
  (3007,75.75,'1990-10-04',2004,1002),
 (3008,4723.00,'1990-10-05',2006,1001),
   (3010,1309.95,'1990-10-06',2004,1002),
    (3011,9891.88,'1990-10-06',2006,1001);
```

Assignment –2

Introducing Relational Databases.

1) Which field of the Customers table is the primary key?

ANS.. CNUM

2) What is the 4th column of the Customers table?

Ans..**RATING**

3) What is another word for row? For column?

row -----**line**, **record**, or **entry** depending on the context (e.g., in databases or tables).

column -----**field**, **attribute**, again depending on how you're using the term.

4) Why isn't it possible to see the first five rows of a table?

ANS…

1.it is not possibal if insert only four row in a teable.

2.It is not possibal is write wring command

3.not possibal if give the limitation of only 4 Rows

Assignment –3

Overview of SQL.

1) Does ANSI recognize the data type DATE?

Ans….Yes ,ANSI(American National Standered Insititute) dose recognize the Date And timr In my Sql.

It is Standered date type use to store calender dates (YYYY-MM-DD).

2) Which subdivision of SQL is used to insert values in tables?

Ans…DML(Data Manipulation language) comman DML command;

Inser delete And update….

Assignment –4

Retrieving Information from Tables.

1) Write a select command that produces the order number, amount, and date for all rows in the Orders table.

Ans.. select Onum, Amt,'date' from orders;

2) Write a query that produces all rows from the Customers table for which the salesperson's number is 1001.

```
Ans..select * from customer
        where Snum = 1001;
```

3) Write a query that displays the Salespeople table with the columns in the following order: city, sname, snum, comm.

Ans…select city, sname,snum,comm from salespeople;

4) Write a select command that produces the rating followed by the name of each customer in San Jose.

```
Ans…SELECT Rating, Cname from  customer
     where city = 'San Jose';
```

5) Write a query that will produce the snum values of all salespeople (suppress the

duplicates) with orders in the Orders table

.

Ans…select distinct snum from orders;

Assignment –5 Relational and Logical Operators.

1) Write a query that will give you all orders for more than Rs. 1,000.

Ans…select * from orders

where Amt >1000;

2) Write a query that will give you the names and cities of all salespeople in London with a commission above .10.

Ans……….

select sname,city from salespeople

-> where city = 'London' and Comm > '0.10';

3) Write a query on the Customers table whose output will exclude all customers with a rating <= 100, unless they are located in Rome.

Ans………

```
 select * from  customer
     where Rating <='100' and city = 'Rome';
```

4) What will be the output from the following query?

```
Select * from Orders
where (amt < 1000 OR
NOT (odate = ‘1990-10-03’
AND cnum > 2003));
```

```
| Onum | Amt     | Odate      | Cnum | Snum |
+------+---------+------------+------+------+
| 3001 |   18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 |  767.19 | 1990-10-03 | 2001 | 1001 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 |   75.75 | 1990-10-04 | 2004 | 1002 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+------+---------+------------+------+------+
```

5) What will be the output of the following query?

```
Select * from Orders
where NOT ((odate = ‘1990-10-03’ OR snum
>1006) AND amt >= 1500);
```

```
+------+---------+------------+------+------+
| Onum | Amt     | Odate      | Cnum | Snum |
+------+---------+------------+------+------+
| 3001 |   18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 |  767.19 | 1990-10-03 | 2001 | 1001 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 |   75.75 | 1990-10-04 | 2004 | 1002 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+------+---------+------------+------+------+
```

6) What is a simpler way to write this query?

Select snum, sname, city, comm From Salespeople

where (comm > .12 OR comm <.14);

```
SELECT snum, sname, city, comm
    FROM Salespeople
    WHERE comm > 0.12 OR comm < 0.14;
```

Assignment –6

Using Special Operators in Conditions.

1) Write two different queries that would produce all orders taken on October 3rd or 4 th, 1990.

A)…. select * from orders

where Odate ='1990-10-03' or Odate = '1990-10-04';

B)….. select * from orders

where Odate in('1990-10-03','1990-10-04');

2) Write a query that selects all of the customers serviced by C

(Hint: the snum field relates the two tables to one another).

```
Select * from customer
    Where snum in (select snum from SALESPEOPLE
    Where sname in('peel','Motika'));
+------+---------+--------+--------+------+
| Cnum | Cname   | City   | Rating | Snum |
+------+---------+--------+--------+------+
| 2001 | Hoffman | London |    100 | 1001 |
| 2006 | Clemens | London |    100 | 1001 |
| 2007 | Pereira | Rome   |    100 | 1004 |
+------+---------+--------+--------+------
```

3) Write a query that will produce all the customers whose names begin with a letter

from ‘A’ to ‘G’.

```
select * from  customer
 where cname like 'A%' or cname like 'B%' or cname like 'C%' or cname like 'D%' or
cname like 'E%' or cname like 'F%' or cname like 'G%';
+------+----------+----------+--------+------+
| Cnum | Cname    | City     | Rating | Snum |
+------+----------+----------+--------+------+
| 2002 | Giovanni | Rome     |    200 | 1003 |
| 2004 | Grass    | Berlin   |    300 | 1002 |
| 2006 | Clemens  | London   |    100 | 1001 |
| 2008 | Cisneros | San Jose |    300 | 1007 |
+------+----------+----------+--------+------+
```

3) Write a query that selects all customers whose names begin with the letter ‘C’

```
.  select * from  customer
    -> where cname like 'C%';
+------+----------+----------+--------+------+
| Cnum | Cname    | City     | Rating | Snum |
+------+----------+----------+--------+------+
| 2006 | Clemens  | London   |    100 | 1001 |
| 2008 | Cisneros | San Jose |    300 | 1007 |
+------+----------+----------+--------+------+
```

5) Write a query that selects all orders except those with zeroes or NULLs in the amt field.

select * from orders

where amt is not null and amt !=0;

Assignment –7

Summarizing Data with Aggregate Functions.

1) Write a query that counts all orders for October 3.

select count(*) from orders where Odate = '1990-10-03' ;

2) Write a query that counts the number of different non-NULL city values in the Customers table.

select count(*) from customer where city is not null ;

3)Write a query that selects each customer’s smallest order.

select cnum, min(amt) from orders group by cnum;

4) Write a query that selects the first customer, in alphabetical order, whose name begins with G.

select min(Cname) from customer where cname like 'G%';

4) Write a query that selects the highest rating in each city.

select city, max(rating) from customer group by city;

6) Write a query that counts the number of salespeople registering orders for each day. (If a salesperson has more than one order on a given day, he or she should be counted only once.)

select Odate,count(distinct Snum) from orders group by Odate;.

ASSIGNMENT -8

FORMATTING QUERY OUTPUT.

1) Assume each salesperson has a 12% commission. Write a query on the orders table that will produce the order number, the salesperson number, and the amount of the salesperson's commission for that order.

```
select Onum,Snum,amt *.12 from orders;
+------+------+----------+
| Onum | Snum | amt *.12 |
+------+------+----------+
| 3001 | 1007 |     2.24 |
| 3003 | 1001 |    92.06 |
| 3002 | 1004 |   228.01 |
| 3005 | 1002 |   619.25 |
| 3006 | 1007 |   131.78 |
| 3009 | 1003 |   205.59 |
| 3007 | 1002 |     9.09 |
| 3008 | 1001 |   566.76 |
| 3010 | 1002 |   157.19 |
| 3011 | 1001 |  1187.03 |
+------+------+----------+
```

2) Write a query on the Customers table that will find the highest rating in each city.

Put the output in this form:

For the city (city), the highest rating is : (rating).

```
select concat('for the city ',city , ',the highest rating is : ', max(rating))from customer group by city ;
+---------------------------------------------------------------------+
| concat('for the city ',city , ',the highest rating is : ', max(rating)) |
+---------------------------------------------------------------------+
| for the city London,the highest rating is : 100                     |
| for the city Rome,the highest rating is : 200                       |
| for the city San Jose,the highest rating is : 300              |
| for the city Berlin,the highest rating is : 300               |
+-------------------------------------------------------------------+
```

3) Write a query that lists customers in descending order of rating. Output the rating field first, followed by the customer's name and number.

select rating ,cname ,cnum from customer order by rating desc;

4) Write a query that totals the orders for each day and places the results in descending order.

select odate, sum(amt) from orders group by odate order by sum(amt) desc ;

Assignment – 9

Querying Multiple Tables at Once.

1) Write a query that lists each order number followed by the name of the customer who made the order.

```
select orders.onum ,customer.cname from orders,customer
   where orders.cnum = customer.cnum;
```

2) Write a query that gives the names of both the salesperson and the customer for

each order along with the order number.

select salespeople.Sname,customer.cname,orders.onum from orders,customer,salespeople where orders.snum = customer.snum and orders.snum = salespeople.snum;

3) Write a query that produces all customers serviced by salespeople with a commission above 12%. Output the customer's name, the salesperson's name, and the salesperson's rate of commission.

select customer.cname,salespeople.sname from customer,salespeople where customer.snum=salespeople.snum and comm >0.12 ;

4) Write a query that calculates the amount of the salesperson's commission on each order by a customer with a rating above 100.

select salespeople.sname, sum(orders.amt*salespeople.comm) from orders,customer,salespeople where orders.snum = customer.snum and orders.snum =salespeople.snum and customer.rating >100 group by sname;

Assignment – 10

Joining a Table to Itself.

1) Write a query that produces all pairs of salespeople who are living in the same city. Exclude combinations of salespeople with themselves as well as duplicate rows with the order reversed.

```
select a.sname,b.sname,a.city from salespeople b,salespeople a where a.city =
b.city and a.snum < b.snum ;
```

2) Write a query that produces the names and cities of all customers with the same rating as Hoffman.

```
select cname, city from customer  where rating =(select rating from customer
where cname = 'Hoffman') ;
```

Assignment – 11

Subqueries.

1) Write a query that uses a subquery to obtain all orders for the customer named Cisneros. Assume you do not know his customer number (cnum).

```
SELECT *
FROM orders
WHERE cnum = (SELECT cnum FROM customers WHERE cname = 'Cisneros');
```

2) Write a query that produces the names and ratings of all customers who have above-average orders.

```
SELECT cname, rating
FROM customers
WHERE cnum IN (
   SELECT cnum
   FROM orders
   GROUP BY cnum
   HAVING AVG(amount) > (SELECT AVG(amount) FROM orders)
);
```

3) Write a query that selects the total amount in orders for each salesperson for whom this total is greater than the amount of the largest order in the table.

```
SELECT snum, SUM(amount) AS total_sales
FROM orders
GROUP BY snum
HAVING SUM(amount) > (SELECT MAX(amount) FROM orders);
```

Assignment – 12

Using the operators IN, ANY, and ALL.

1) Write a query that selects all customers whose ratings are equal to or greater than ANY of Serres'.

```
SELECT *
FROM customers
WHERE rating >= ANY (SELECT rating FROM customers WHERE cname = 'Serres');
```

2) Write a query using ANY or ALL that will find all salespeople who have no customers located in their city.

```
SELECT *
FROM salespeople
```

WHERE snum <> ALL (SELECT snum FROM customers WHERE city = salespeople.city);

3) Write a query that selects all orders for amounts greater than any for the customers in London.

```
SELECT *
FROM orders
WHERE amount > ANY (SELECT amount FROM orders WHERE cnum IN
   (SELECT cnum FROM customers WHERE city = 'London')
);
```

5) Write the above query using MIN or MAX.

```
SELECT *
FROM orders
WHERE amount > (SELECT MIN(amount) FROM orders WHERE cnum IN
   (SELECT cnum FROM customers WHERE city = 'London')
);
```

Assignment – 13

Using the UNION clause.

1) Create a union of two queries that shows the names, cities, and ratings of all customers. Those with rating of 200 or greater will also have the words "High Rating", while the others will have the words "Low Rating".

```
select cname,city,rating,'High Rating' as ratint from customer where rating >= 200 union
select cname,city,rating,'low Rating' as Rating from customer where rating < 200;
```

2) Write a command that produces the name and number of each salesperson and each customer with more than one current order. Put the results in alphabetical order.

```
SELECT cname AS name, cnum AS number
   FROM customer
    WHERE cnum IN (SELECT cnum FROM orders GROUP BY cnum HAVING COUNT(*) > 1)
    UNION
   SELECT sname AS name, snum AS number
    FROM salespeople
   ORDER BY name
```

3) Form a union of three queries. Have the first select the snums of all salespeople in San Jose; the second, the cnums of all customers in San Jose; and the third the onums of all orders on October 3. Retain duplicates between the last two queries but eliminate any redundancies between either of them and the first.
(Note: in the sample tables as given, there would be no such redundancy. This is besides the point.)

```
select snum from salespeople where city = 'San Jose' union (select cnum from customer where city = 'San Jose' union select onum from orders where odate =' 1990-10-03');
```

Assignment – 14

Entering, Deleting, and Changing Field Values.

1) Write a command that puts the following values, in their given order, into the salespeople table: city – San Jose, name – Blanco, comm – NULL, cnum – 1100.

insert into salespeople(city,sname,comm,snum) values ('san jose ' ,'Blanco',null,'1100');

2) Write a command that removes all orders from customer Clemens from the Orders table.

Delete from orders where cnum =(select cnum from customer where cname =' Clemens ')

2) Write a command that increases the rating of all customers in Rome by 100.

update customer set rating = rating +100 where city = 'Rome';

3) Salesperson Serres has left the company. Assign her customers to Motika.

UPDATE customers   SET snum = (SELECT snum FROM salespeople WHERE sname = 'Motika') WHERE snum = (SELECT snum FROM salespeople WHERE sname = 'Serres');

Assignment – 15

Using Subqueries with DML Commands.

1) Assume there is a table called Multicust, with all of the same column definitions as Salespeople. Write a command that inserts all salespeople with more than one customer into this table.

Insert into Multicust(snum,sname,city,comm) select snum,sname,city,comm

From Salespeople where snum in (select snum from customer group by snum having count(*) >1);

2) Write a command that deletes all customers with no current orders.

Delete from customer where cnum not in(select distinct cnum from orders);

3) Write a command that increases by twenty percent the commissions of all salespeople with total orders above Rs. 3,000.

Update salespeople set comm = comm *1.2 where snum in(select snum from orders group by snum having sum(amt)>3000);

Assignment – 16

Creating Tables and Indexes.

1) Write a command that will enable a user to pull orders grouped by date out of the Orders table quickly.

create index index_name on orders(odate);

select odate,count(*) from orders group by odate order by odate;

2) If the Orders table has already been created, how can you force the onum field to be unique (assume all current values are unique)?

Create unique index idx_unique_index on Orders(onum);

3) Create an index that would permit each salesperson to retrieve his or her orders grouped by date quickly.

Create index idx_salesperson_orders on orders(snum,odate);

Select snum, odate from orders group by snum,odate order by snum,odate;

4) Let us assume that each salesperson is to have only one customer of a given rating, and that this is currently the case. Enter a command that enforces it.

alter table customer add constraint unique (snum,rating);

update customer set rating = 101 where snum = 1001 and rating = 100;

Assignment – 17

Constraining the Values of your data.

1) Create the Orders table so that all onum values as well as all combinations of cnum and snum are different from one another, and so that NULL values are excluded from the date field.

Create table orders3 (

Onum int primary key,

Cnum int ,

Snum int ,

Amt decimal(10,2),

Odatee date not null,

Foreign key (cnum) references customer(cnum),

Foreign key (snum) references selespeople(snum),

Constraint unique_cnum_snum unique (Cnum, Snum)

);

2) Create the Salespeople table so that the default commission is 10% with no NULLS permitted, snum is the primary key, and all names fall alphabetically between A and M, inclusive (assume all names will be uppercase).

Create table salespeople2 (snum int primary key,Sname varchar(50) not null check (sname between 'A' and 'M'), city varchar(50), comm decimal(3,2) not null default 0.10 check (comm between 0.10 and 1.00));

3) Create the Orders table, making sure that the onum is greater than the cnum, and the cnum is greater that the snum. Allow no NULLS in any of these three fields.

Create table orders4 (onum int not null , cnum int not null ,snum int not null,amt decimal(10,2), odate date not null,check(onum>cnum), check (cnum>snum));

Assignment – 18

Maintaining the Integrity of your Data.

1) Create a table called Cityorders. This will contain the same onum, amt and snum fields as the Orders table, and the same cnum and city fields as the Customers table, so that each customer’s order will be entered into this table along with his or her city. Onum will be the primary key of Cityorders. All of the fields in Cityorders will be constrained to match the Customers and Orders tables. Assume

the parent keys in these tables already have the proper constraints.

```
Create table Cityorders(
Onum int primary key,
Amt decimal(10,2),
Snum int,
Cnum int,
City varchar(50)
);

INSERT INTO Cityorders (Onum, Amt, Snum, Cnum, City)
SELECT orders.Onum, orders.Amt, orders.Snum, customers.Cnum, customers.City
FROM Orders,customers
Where  orders.Cnum = customers.Cnum;
```

2) Redefine the Orders table as follows:- add a new column called *prev*, which will identify, for each order, the onum of the previous order for that current customer. Implement this with a foreign key referring to the Orders table itself. The foreign key should refer as well to the cnum of the customer, providing a definite enforced link between the current order and the one referenced.

```
Alter table orders
Add column prev int,
Add constraint fk_prev_orders foreign key (prev) references orders (onum);
Update orders o1 set prev = (
Set prev = (
Select max(o2.onum)  from orders o2
```

Where o2.cnum = o1.cnum and o2.onum<o1.onum

);

Assignment – 19

Views.

1) Create a view that shows all of the customers who have the highest ratings

Create view v1 as select * from custromer where rating = (select Max(rating)from customer);

select * from v1;

2) Create a view that shows the number of salespeople in each city.

Create view v2 as select count(city),city from salespeople group by city ;

select * from v2;

3) Create a view that shows the average and total orders for each salesperson after

his or her name. Assume all names are unique.

create view v3 as select sname ,avg(amt),sum(amt) from orders,salespeople where orders.Snum = salespeople .snum group by sname;

select * from v3;

4) Create a view that shows each salesperson with multiple customers.

Create view v4 as select salespeople.sname,salespeople.snum,count(customer.cnum) from customer,salespeople where customer.snum =salespeople.snum group by salespeople.snum,salespeople.sname having count(cnum)>1;

select * from v4;

Assignment – 20

Changing Values through Views.

1) Which of these views are updateable (will allow DML operations)?

#1 Create View Dailyorders

as Select Distinct cnum, snum, onum, odate from Orders;

```
select * from Dailyorders;
+------+------+------+------------+
| cnum | snum | onum | odate      |
+------+------+------+------------+
| 2008 | 1007 | 3001 | 1990-10-03 |
| 2001 | 1001 | 3003 | 1990-10-03 |
| 2007 | 1004 | 3002 | 1990-10-03 |
| 2003 | 1002 | 3005 | 1990-10-03 |
| 2008 | 1007 | 3006 | 1990-10-03 |
| 2002 | 1003 | 3009 | 1990-10-04 |
| 2004 | 1002 | 3007 | 1990-10-04 |
| 2006 | 1001 | 3008 | 1990-10-05 |
| 2004 | 1002 | 3010 | 1990-10-06 |
| 2006 | 1001 | 3011 | 1990-10-06 |
+------+------+------+------------+
```

#2 Create View Custotals

```
as Select cname, sum(amt) Sum_Amt from Orders, Customer
where Orders.cnum=Customer.cnum
Group by cname;
select * from Custotals;
+----------+----------+
| cname    | Sum_Amt  |
+----------+----------+
| Cisneros |  1116.85 |
| Hoffman  |   767.19 |
| Pereira  |  1900.10 |
| Liv      |  5160.45 |
| Giovanni |  1713.23 |
| Grass    |  1385.70 |
| Clemens  | 14614.88 |
+----------+----------+
```

#3 Create view Thirdorders

```
as Select * from Dailyorders where
odate='1990-10-03';

select * from Thirdorders;
+------+------+------+------------+
| cnum | snum | onum | odate      |
+------+------+------+------------+
| 2008 | 1007 | 3001 | 1990-10-03 |
| 2001 | 1001 | 3003 | 1990-10-03 |
| 2007 | 1004 | 3002 | 1990-10-03 |
| 2003 | 1002 | 3005 | 1990-10-03 |
| 2008 | 1007 | 3006 | 1990-10-03 |
+------+------+------+------------+
```

#4 Create view Nullcities

```
Create view Nullcities
as Select snum, sname, city
from Salespeople
where city is NULL
OR sname BETWEEN 'A' and 'MZ';
```

```
select * from Nullcities;
+------+---------+----------+
| snum | sname   | city     |
+------+---------+----------+
| 1004 | Motika  | London   |
| 1003 | Axelrod | new      |
| 1100 | Blanco  | san jose |
+------+---------+----------+
```

2) Create a view of the Salespeople table called Commissions. This view will include only the snum and comm fields. Through this view, someone could enter or change commissions, but only to values between .10 and .20.

```
CREATE VIEW Commissions AS
SELECT snum, comm
FROM Salespeople;
```

3) Some SQL implementations have a built-in constant representing the current date, sometimes called "CURDATE" or "SYSDATE". The word SYSDATE can therefore be used in a SQL statement, and be replaced by the current date when the

value is accessed by commands such as Select or Insert. We will use a view of the Orders table called Entryorders to insert rows into the Orders table. Create the Orders table, so that SYSDATE is automatically inserted for odate if no value is given. Then create the Entryorders view so that no values can be given.

```
CREATE TABLE Orders1 ( Onum INT PRIMARY KEY,   Amt DECIMAL(10,2), Odate TIMESTAMP DEFAULT CURRENT_TIMESTAMP, Cnum INT,Snum INT );
CREATE VIEW Entryorders AS
SELECT Onum, Amt, Cnum, Snum
FROM Orders;

INSERT INTO Entryorders (Onum, Amt, Cnum, Snum)
VALUES (3012, 2500.00, 2005, 1003);
Select * from Entryorders;
```

Assignment - 21

Grant and Revoke.

1) Give Amit the right to change the ratings of the customers.

```
GRANT UPDATE (rating) ON Customers TO 'Amit'@'your_host';
```

2) Give Manoj the right to give other users the right to query the Orders table.

```
GRANT SELECT ON Orders TO 'Manoj'@'localhost';
```

3) Take the INSERT privilege on Salespeople away from Ajita.

If user nahi hai to user bano ………………….

REVOKE INSERT ON Salespeople FROM 'Ajita'@'localhost';

4) Grant Abhijeet the right to insert or update the Customers table while keeping her possible rating values in the range of 100 to 500.

GRANT INSERT, UPDATE ON Customers TO 'Abhijeet'@'localhost';

5) Allow Vikram to query the Customers table, but restrict his access to those customers whose rating is the lowest.

```
CREATE VIEW LowestRatedCustomers AS
SELECT * FROM Customers
WHERE rating = (SELECT MIN(rating) FROM Customers);
GRANT SELECT ON LowestRatedCustomers TO 'Vikram'@'localhost';
```