**Amrita Vishwa Vidyapeetham**

**Amrita School of Engineering, Coimbatore**

**Department of Computer Science and Engineering**

**2019 – 2021 Odd Semester**

**B.Tech/III Year CSE/V Semester L T P C**

**15CSE302/Database Management Systems 2 0 2 3**

**Lab Exercise: 3 (CO2) [BTL3] Date:10 Aug 2020**

**Rollno:CB.EN.U4CSE18243**

1. **Create the following tables and Populate with records(5 marks)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **Customer** | |  |  | | **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 | | |  |  |  |  | | --- | --- | --- | --- | | **SalesPerson** | | |  | | **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 | | 1005 | Fran | London | .26 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Orders** | |  |  |
| **ONUM** | **AMT** | **ODATE** | **CNUM** | **SNUM** |
| 3001 | 18.69 | 10/03/2019 | 2008 | 1007 |
| 3003 | 767.19 | 11/03/2019 | 2001 | 1001 |
| 3002 | 1900.10 | 10/03/2019 | 2007 | 1004 |
| 3005 | 5160.45 | 10/03/2019 | 2003 | 1002 |
| 3006 | 1098.16 | 12/03/2019 | 2008 | 1007 |
| 3009 | 1713.23 | 10/04/2019 | 2002 | 1003 |
| 3007 | 75.75 | 10/04/2019 | 2002 | 1003 |
| 3008 | 4723 .00 | 10/05/2019 | 2006 | 1001 |
| 3010 | 1309.95 | 10/06/2019 | 2004 | 1002 |
| 3011 | 9891.88 | 10/06/2019 | 2006 | 1001 |

**Answers :**

1. **CREATION AND INSERTION OF SALESPERSON TABLE**
2. **CREATE**

Query :

CREATE TABLE SalesPerson(

snum number(4) NOT NULL,

sname varchar2(20),

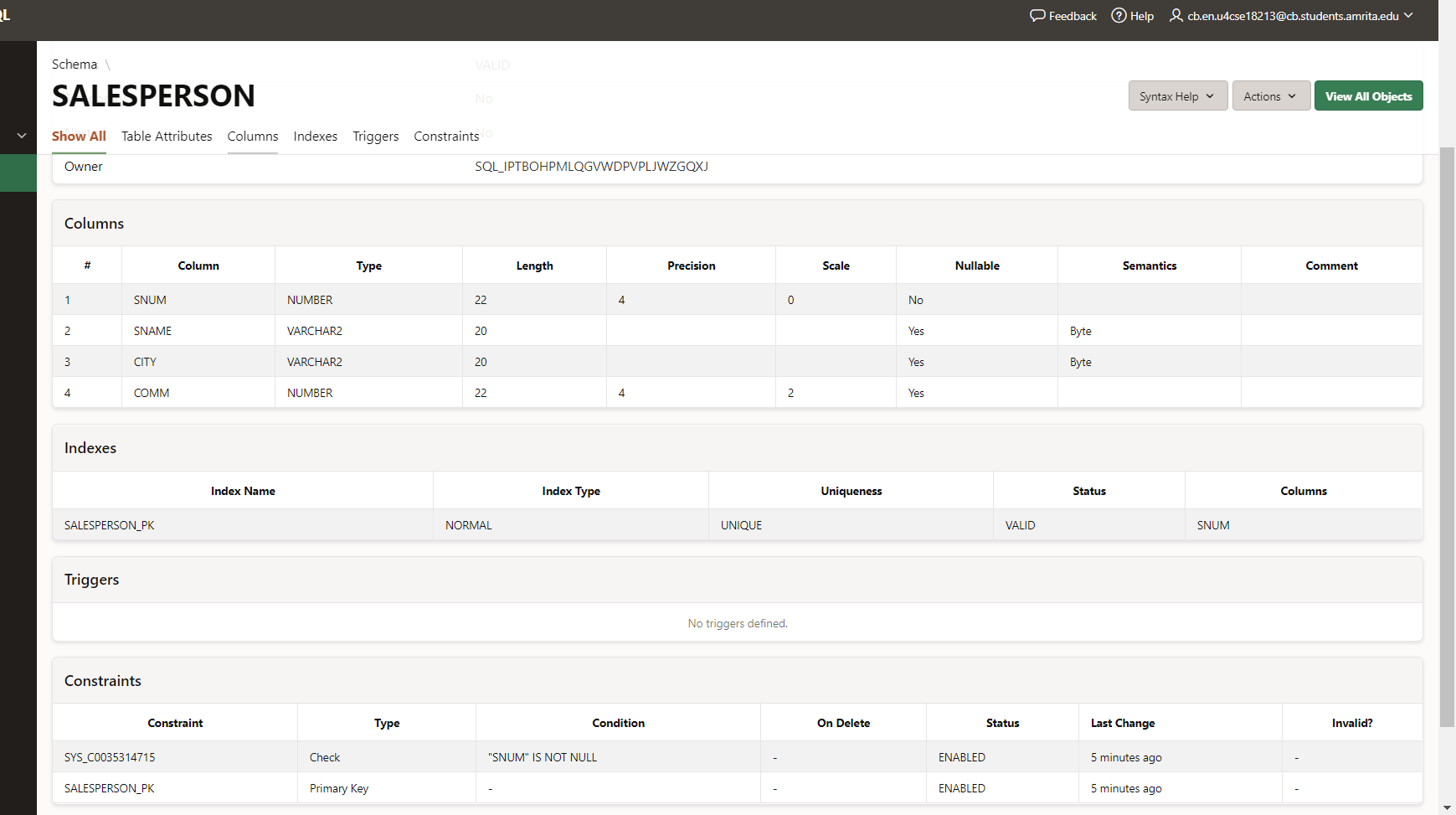
city varchar2(20),

comm numeric(4,2),

CONSTRAINT SalesPerson\_pk PRIMARY KEY(snum)

);

Schema



1. **INSERTION :**

INSERT INTO SalesPerson VALUES (1001,'Peel','London',0.12);

INSERT INTO SalesPerson VALUES (1002,'Serres','San Jose',0.13);

INSERT INTO SalesPerson VALUES (1004,'Motika','London',0.11);

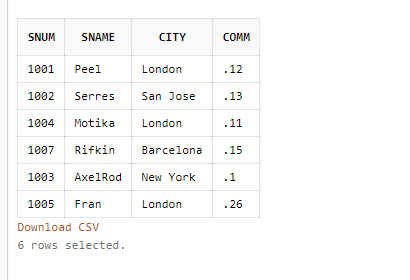
INSERT INTO SalesPerson VALUES (1007,'Rifkin','Barcelona',0.15);

INSERT INTO SalesPerson VALUES (1003,'AxelRod','New York',0.10);

INSERT INTO SalesPerson VALUES (1005,'Fran','London',0.26);

1. **CREATED TABLE :**

SELECT \* FROM SalesPerson;



1. **CREATION AND INSERTION OF CUSTOMER TABLLE**
2. **CREATION**

Query :

CREATE TABLE Customer(

cnum number(4) NOT NULL,

cname varchar2(20),

city varchar2(20),

rating number(3),

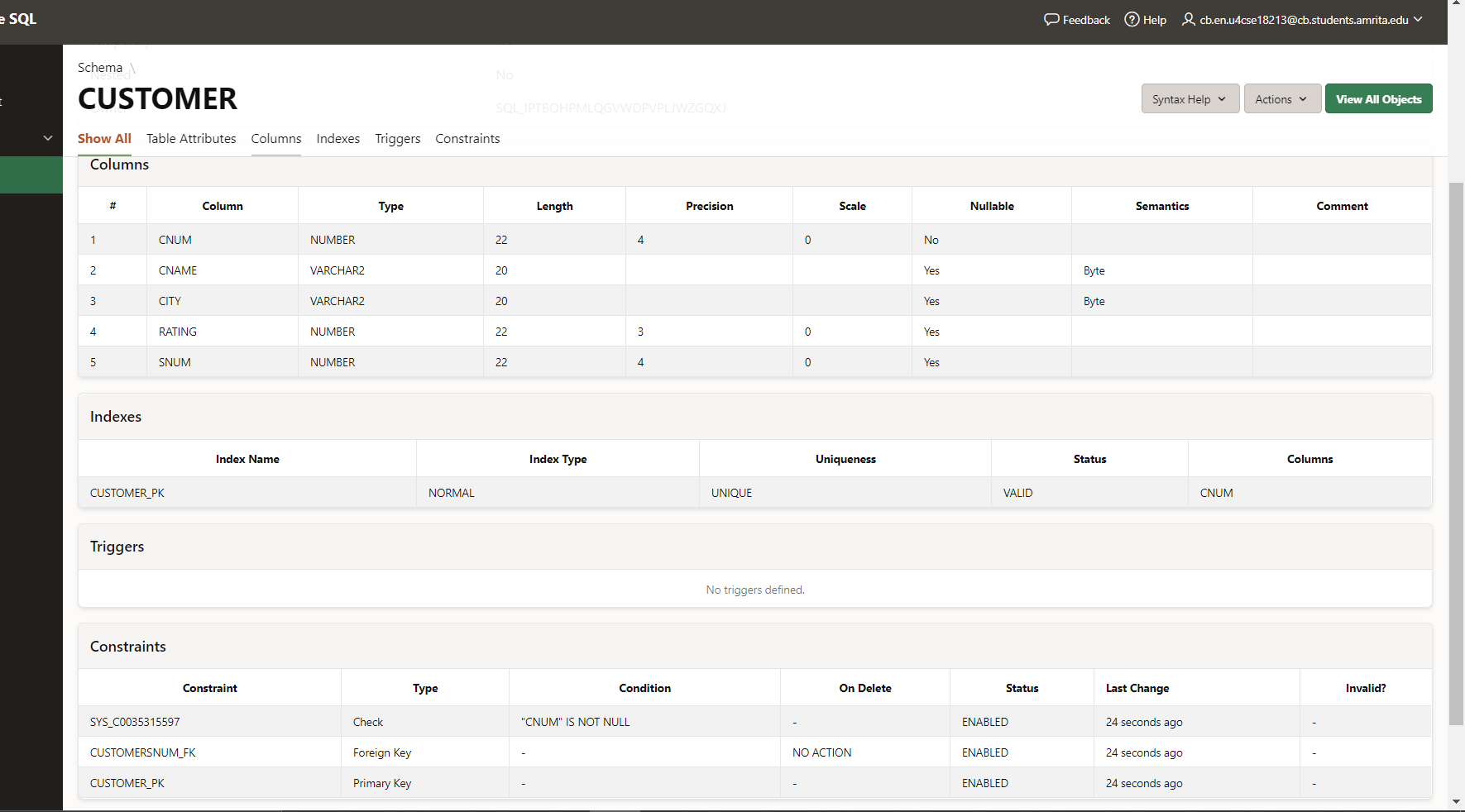
snum number(4),

CONSTRAINT customer\_pk PRIMARY KEY(cnum),

CONSTRAINT customersnum\_fk FOREIGN KEY(snum) REFERENCES SalesPerson(snum)

);

Schema :

****

1. **INSERTION :**

INSERT INTO customer VALUES (2001,'Hoffman','London',100,1001);

INSERT INTO customer VALUES (2002,'Giovanni','Rome',200,1003);

INSERT INTO customer VALUES (2003,'Liu','San Jose',200,1002);

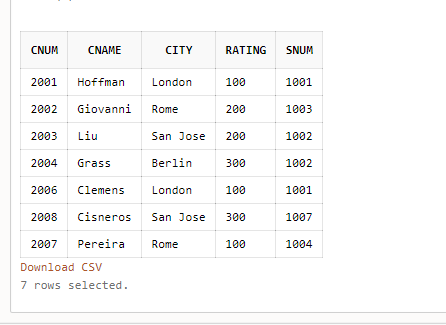
INSERT INTO customer VALUES (2004,'Grass','Berlin',300,1002);

INSERT INTO customer VALUES (2006,'Clemens','London',100,1001);

INSERT INTO customer VALUES (2008,'Cisneros','San Jose',300,1007);

INSERT INTO customer VALUES (2007,'Pereira','Rome',100,1004);

1. **CREATED TABLE :**



1. **CREATION AND INSERTION OF ORDERS :**
2. **CREATION :**

Query :

CREATE TABLE orders(

onum number(4) NOT NULL,

amt numeric(8,2),

odate varchar2(10),

cnum number(4),

snum number(4),

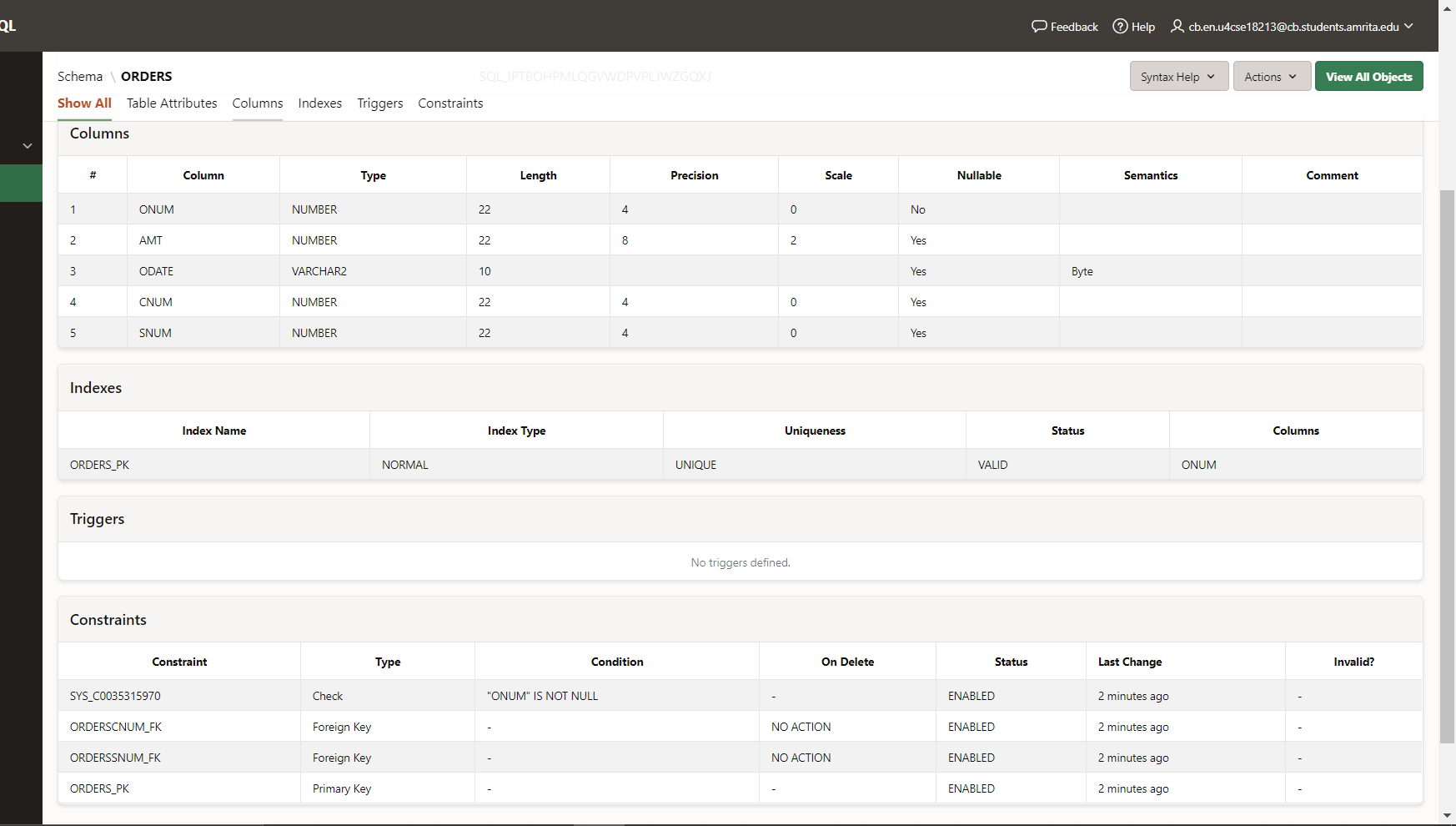
CONSTRAINT orders\_pk PRIMARY KEY(onum),

CONSTRAINT orderscnum\_fk FOREIGN KEY(cnum) REFERENCES customer(cnum),

CONSTRAINT orderssnum\_fk FOREIGN KEY(snum) REFERENCES SalesPerson(snum)

);

Schema :



1. **INSERTION :**

INSERT INTO orders VALUES (3001,18.69,'10/03/2019',2008,1007);

INSERT INTO orders VALUES (3003,767.19,'11/03/2019',2001,1001);

INSERT INTO orders VALUES (3002,1900.10,'10/03/2019',2007,1004);

INSERT INTO orders VALUES (3005,5160.45,'10/03/2019',2003,1002);

INSERT INTO orders VALUES (3006,1098.16,'12/03/2019',2008,1007);

INSERT INTO orders VALUES (3009,1713.23,'10/04/2019',2002,1003);

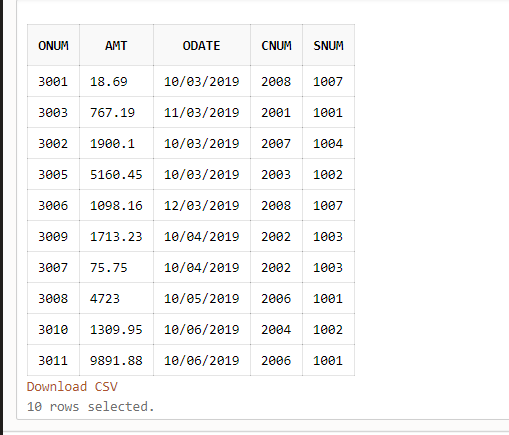
INSERT INTO orders VALUES (3007,75.75,'10/04/2019',2002,1003);

INSERT INTO orders VALUES (3008,4723.00,'10/05/2019',2006,1001);

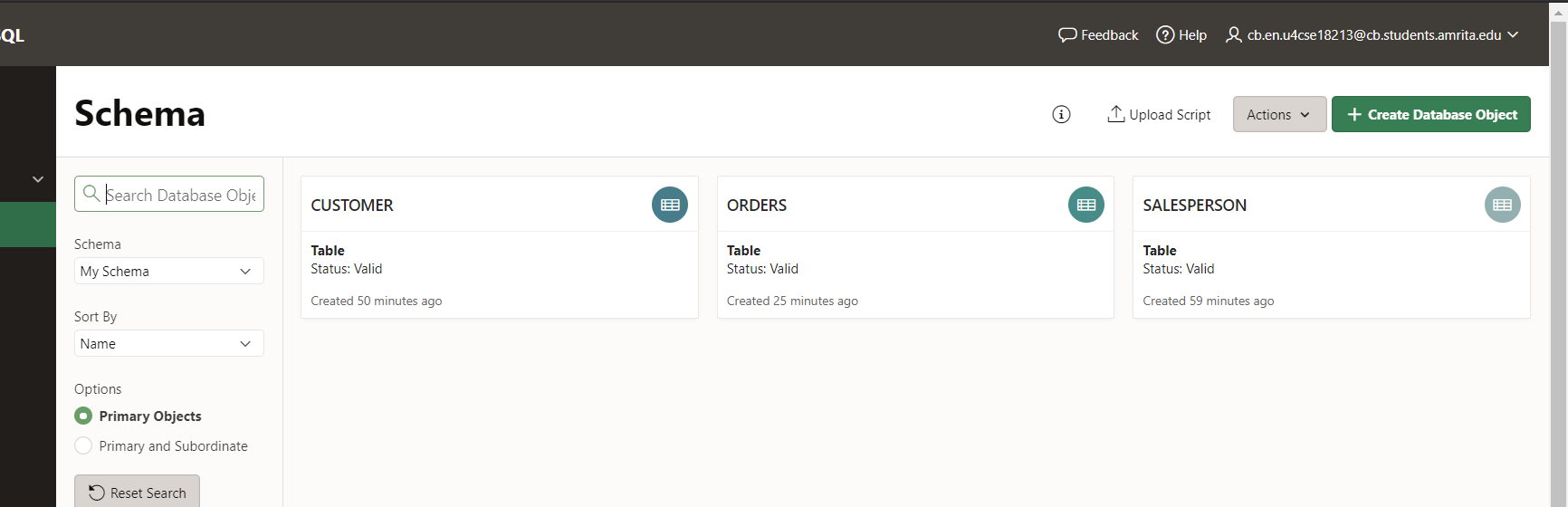
INSERT INTO orders VALUES (3010,1309.95,'10/06/2019',2004,1002);

INSERT INTO orders VALUES (3011,9891.88,'10/06/2019',2006,1001);

1. **CREATED TABLE :**



**ALL TABLES CREATED:**

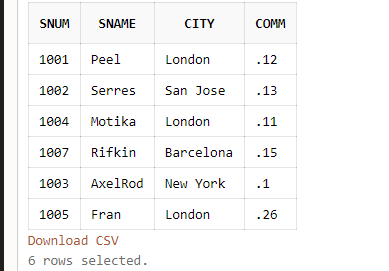


**Queries (30 marks)**

1. List all the columns of the Salespeople table.

Query : SELECT \* from Salesperson

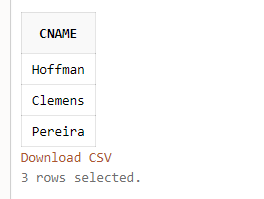
Output :



1. List all customers with a rating of 100.

Query : SELECT cname from customer where rating=100;

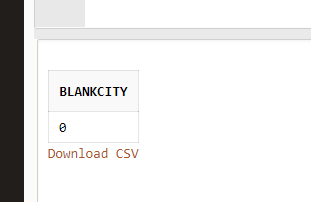
Output :



1. Find all records in the Customer table with NULL values in the city column.

Query : SELECT count(\*) as blankcity from customer where city=NULL;

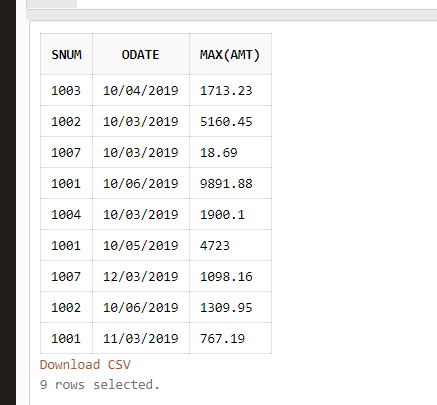
Output :



1. Find the largest order taken by each salesperson on each date.

Query : SELECT snum,odate,MAX(amt) from orders GROUP BY odate,snum;

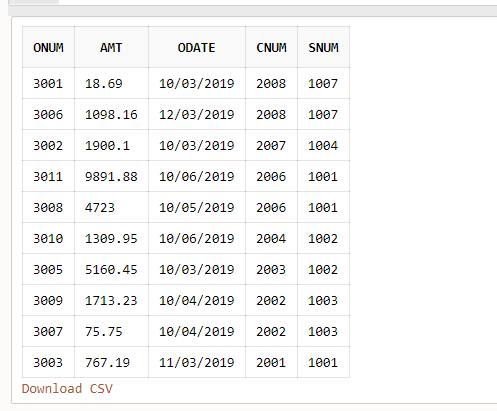
Ouput ;



1. Arrange the Orders table by descending customer number.

Query : SELECT \* from orders order by cnum DESC;

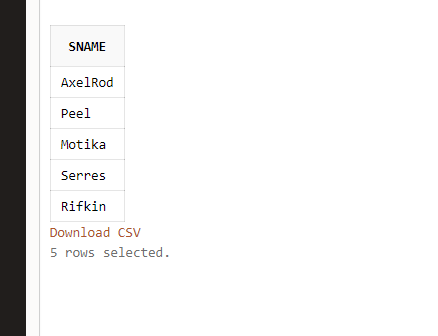
Output :



1. Find which salespeople currently have orders in the Orders table.

Query : SELECT DISTINCT sname from salesperson,orders where salesperson.snum=orders.snum;

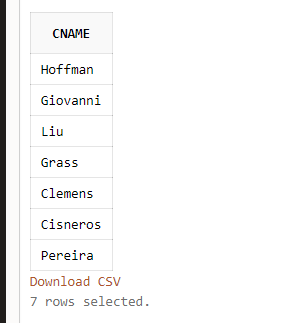
Output :



1. List names of all customers matched with the salespeople serving them.

Query : SELECT cname from customer,salesperson where customer.snum=salesperson.snum;

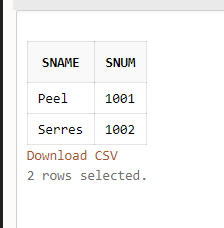
Output :



1. Find the names and numbers of all salespeople who had more than one customer.

Query : SELECT sname,snum from Salesperson WHERE (SELECT COUNT(\*) from customer where salesperson.snum=customer.snum) > 1;

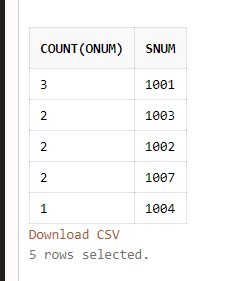
Output :



1. Count the orders of each of the salespeople and output the results in descending order.

Query : SELECT snum,COUNT(onum) from orders GROUP BY snum ORDER BY COUNT(onum) DESC.

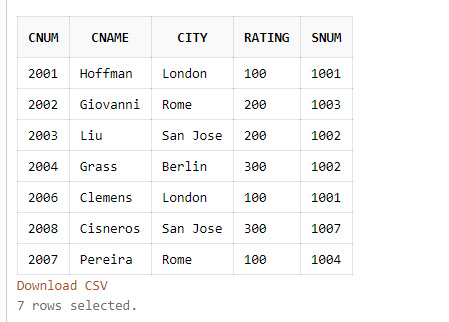
Output :



1. List the Customer table if and only if one or more of the customers in the Customer table are located in San Jose.

Query : SELECT \* from customer where (SELECT COUNT(cnum) from customer where city='San Jose')>1;

Output :



1. Match salespeople to customers according to what city they lived in.

Query : select salesperson.sname,customer.cname from salesperson,customer where salesperson.city=customer.city

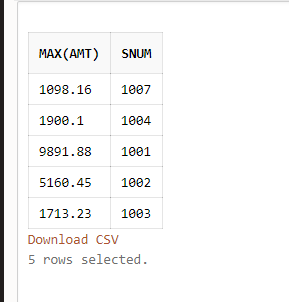
Output :



1. Find the largest order taken by each salesperson.

Query : select max(amt),snum from orders group by snum;

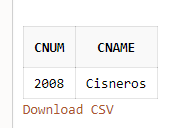
Output :



1. Find customers in San Jose who have a rating above 200.

Query : select cnum,cname from customer where rating>200 and city like 'San Jose'

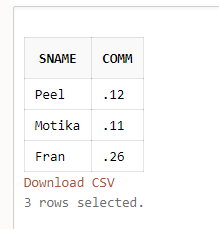
Output :



1. List the names and commissions of all salespeople in London.

Query : select sname,comm from salesperson where city like 'London'

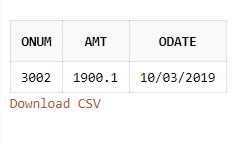
Output :



1. List all the orders of salesperson Motika from the Orders table.

Query : select onum,amt,odate from orders,salesperson where orders.snum=salesperson.snum and sname='Motika'

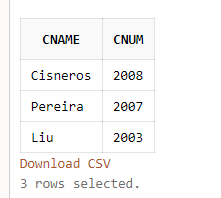
Output :



1. Find all customers with orders on October 3.

Query : select cname from customer,orders where customer.cnum=orders.cnum and odate like '10/03%'

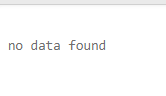
Output :



1. Give the sums of the amounts from the Orders table, grouped by date, eliminating all those dates where the SUM was not at least 2000.00 above the MAX amount.

Query : select sum(amt),odate from orders group by odate having (sum(amt)-max(amt))>2000

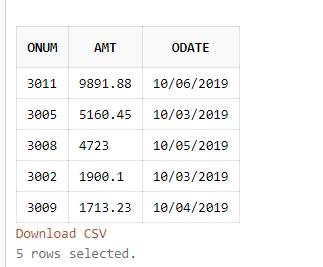
Output :



1. Select all orders that had amounts that were greater than at least one of the orders from October6.

Query : select onum,amt,odate from orders where amt> ANY (select amt from orders where odate like '10/06/%');

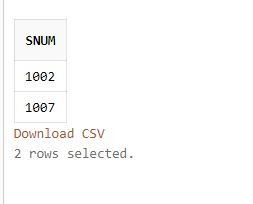
Query :



1. Write a query that uses the EXISTS operator to extract all salespeople who have customers with a rating of 300.

Query : Select a.snum from salesperson a where exists ( select b.snum from customer b where b.rating = 300 and a.snum = b.snum)

Output :



1. Find all pairs of customers having the same rating.

Query : Select a.cname, b.cname,a.rating from customer a, customer b where a.rating = b.rating and a.cnum != b.cnum ;

Output :



1. Find all customers whose CNUM is 1000 above the SNUM of Serres.

Query : Select cnum,cname from customer where cnum - (Select snum from salesperson where sname='Serres')>1000

Output :



1. Give the salespeople’s commissions as percentages instead of decimal numbers.

Query : Select comm\*100 as Comm\_Percentage from salesperson

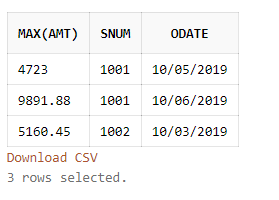
Output :



1. Find the largest order taken by each salesperson on each date, eliminating those MAX orders which are less than $3000.00 in value.

Query : select max(amt),snum,odate from orders group by snum,odate having max(amt)>=3000

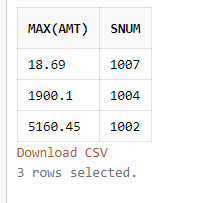
Output :



1. List the largest orders for October 3, for each salesperson.

Query : select max(amt),snum from orders where odate='10/03/2019' group by snum;

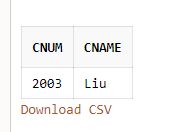
Output :



1. Find all customers located in cities where Serres (SNUM 1002) has customers.

Query : select cnum,cname from customer,salesperson where salesperson.city=customer.city and salesperson.snum=customer.snum and salesperson.sname='Serres'

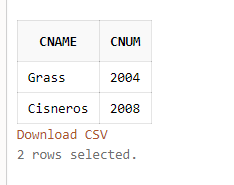
Output :



1. Select all customers with a rating above 200.00.

Query : select cname,cnum from customer where rating>200

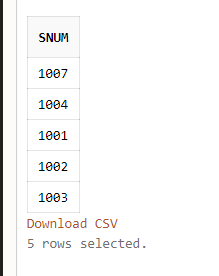
Output :



1. Count the number of salespeople currently listing orders in the Orders table.

Query : select distinct salesperson.snum from salesperson,orders where salesperson.snum=orders.snum

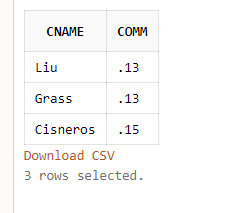
Output :



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

Query : select cname,comm from customer, salesperson where salesperson.snum=customer.snum and salesperson.comm>.12

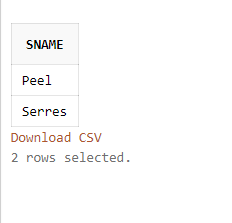
Output :



1. Find salespeople who have multiple customers.

Query : select sname from salesperson where snum in (select snum from customer group by snum having count(\*)>1)

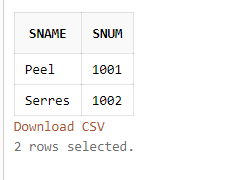
Output :



1. Find salespeople with customers located in their city.

Query : select distinct sname,salesperson.snum from salesperson,customer where salesperson.snum=customer.snum and salesperson.city=customer.city

Output :



1. Find all salespeople whose name starts with ‘P’ and the fourth char ‘e’

Query : select sname from salesperson where sname like 'P\_\_e%'

Output :

