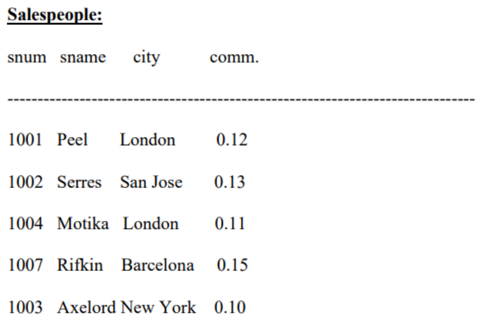
PRACTICAL-9

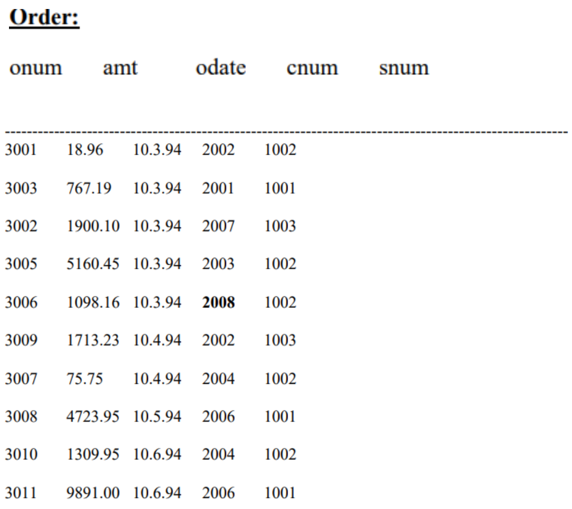
AIM- To study and execute various JOIN commands to perform data retrieval and manipulation from Salespeople, Customer, and Order tables based on specific requirements.

DESCRIPTION:-



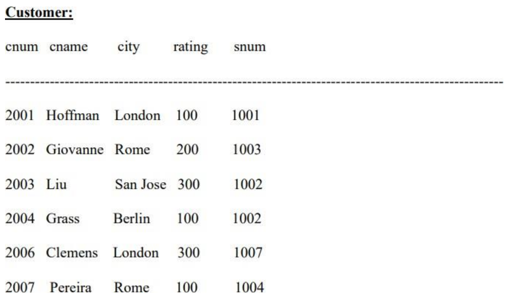
Salespeople:

snum	sname	city	comm.
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1004	Motika	London	0.11
1007	Rifkin	Barcelona	0.15
1003	Axelord	New York	0.10



Order:

onum	amt	odate	cnum	snum
3001	18.96	10.3.94	2002	1002
3003	767.19	10.3.94	2001	1001
3002	1900.10	10.3.94	2007	1003
3005	5160.45	10.3.94	2003	1002
3006	1098.16	10.3.94	**2008**	1002
3009	1713.23	10.4.94	2002	1003
3007	75.75	10.4.94	2004	1002
3008	4723.95	10.5.94	2006	1001
3010	1309.95	10.6.94	2004	1002
3011	9891.00	10.6.94	2006	1001



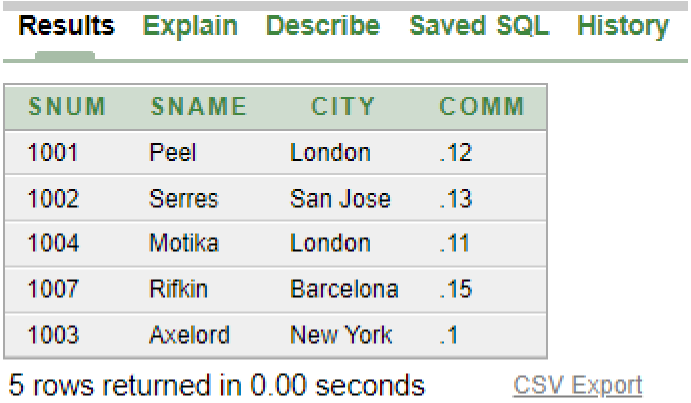
Customer:

cnum	cname	city	rating	snum
2001	Hoffman	London	100	1001
2002	Giovanne	Rome	200	1003
2003	Liu	San Jose	300	1002
2004	Grass	Berlin	100	1002
2006	Clemens	London	300	1007
2007	Pereira	Rome	100	1004

QUERY:

```
CREATE TABLE Salespeople (
   snum NUMBER(4) PRIMARY KEY,
   sname VARCHAR2(20) NOT NULL,
   city VARCHAR2(15),
   comm NUMBER(5,2));

INSERT INTO Salespeople VALUES (1001, 'Peel', 'London', 0.12);
INSERT INTO Salespeople VALUES (1002, 'Serres', 'San Jose', 0.13);
INSERT INTO Salespeople VALUES (1004, 'Motika', 'London', 0.11);
INSERT INTO Salespeople VALUES (1007, 'Rifkin', 'Barcelona', 0.15);
INSERT INTO Salespeople VALUES (1003, 'Axelord', 'New York', 0.10);
```

Output:



SNUM	SNAME	CITY	COMM
1001	Peel	London	.12
1002	Serres	San Jose	.13
1004	Motika	London	.11
1007	Rifkin	Barcelona	.15
1003	Axelord	New York	.1

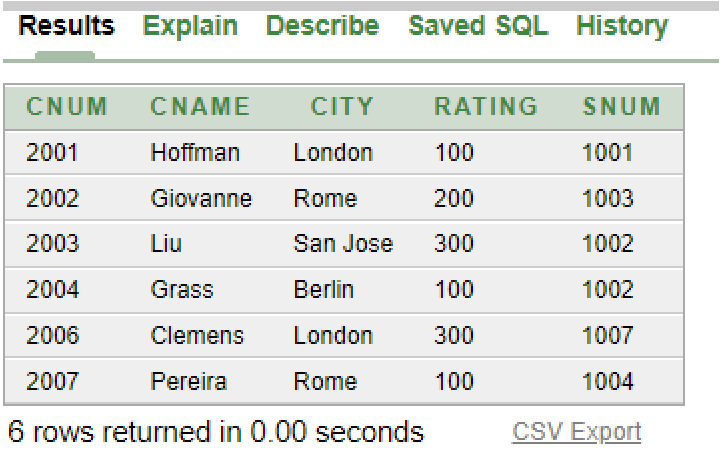
5 rows returned in 0.00 seconds

QUERY:

```
CREATE TABLE Customer (
   cnum NUMBER(4) PRIMARY KEY,
   cname VARCHAR2(20) NOT NULL,
   city VARCHAR2(20),
   rating NUMBER(3),
   snum NUMBER(4),
   FOREIGN KEY (snum) REFERENCES Salespeople(snum));
```

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

INSERT INTO Customer VALUES (2002, 'Giovanne', 'Rome', 200, 1003);

INSERT INTO Customer VALUES (2003, 'Liu', 'San Jose', 300, 1002);

INSERT INTO Customer VALUES (2004, 'Grass', 'Berlin', 100, 1002);

INSERT INTO Customer VALUES (2006, 'Clemens', 'London', 300, 1007);

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

Output:



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

6 rows returned in 0.00 seconds

QUERY:

```
CREATE TABLE "Order" (
  onum NUMBER(4) PRIMARY KEY,
  amt NUMBER(7,2),
  odate DATE,
  cnum NUMBER(4),
  snum NUMBER(4),
  FOREIGN KEY (cnum) REFERENCES Customer(cnum),
  FOREIGN KEY (snum) REFERENCES Salespeople(snum));
```

```
INSERT INTO "Order" VALUES (3001, 18.96, TO_DATE('10.3.94', 'DD.MM.YY'), 2002, 1002);

INSERT INTO "Order" VALUES (3003, 767.19, TO_DATE('10.3.94', 'DD.MM.YY'), 2001, 1001);

INSERT INTO "Order" VALUES (3002, 1900.10, TO_DATE('10.3.94', 'DD.MM.YY'), 2007, 1003);

INSERT INTO "Order" VALUES (3005, 5160.45, TO_DATE('10.3.94', 'DD.MM.YY'), 2003, 1002);

INSERT INTO "Order" VALUES (3006, 1098.16, TO_DATE('10.3.94', 'DD.MM.YY'), 2008, 1002);
```

INSERT INTO "Order" VALUES (3009, 1713.23, TO_DATE('10.4.94', 'DD.MM.YY'), 2002, 1003);

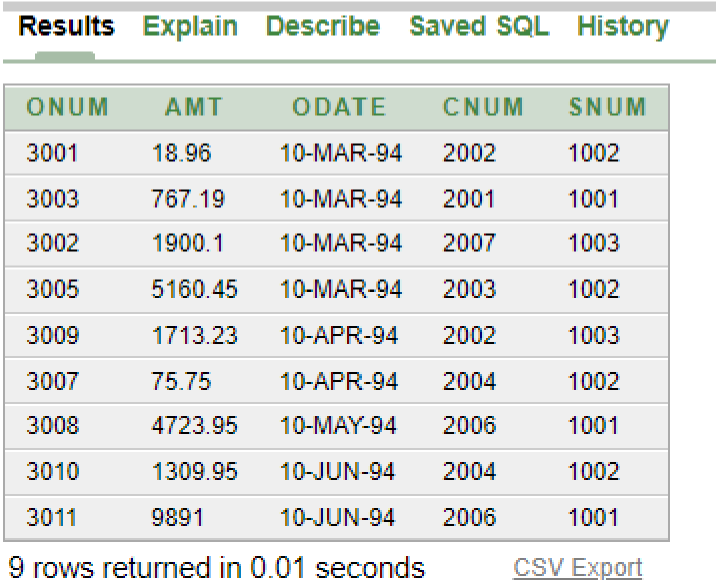
INSERT INTO "Order" VALUES (3007, 75.75, TO_DATE('10.4.94', 'DD.MM.YY'), 2004, 1002);

INSERT INTO "Order" VALUES (3008, 4723.95, TO_DATE('10.5.94', 'DD.MM.YY'), 2006, 1001);

INSERT INTO "Order" VALUES (3010, 1309.95, TO_DATE('10.6.94', 'DD.MM.YY'), 2004, 1002);

INSERT INTO "Order" VALUES (3011, 9891.00, TO_DATE('10.6.94', 'DD.MM.YY'), 2006, 1001);

OUTPUT:



ONUM	AMT	ODATE	CNUM	SNUM
3001	18.96	10-MAR-94	2002	1002
3003	767.19	10-MAR-94	2001	1001
3002	1900.1	10-MAR-94	2007	1003
3005	5160.45	10-MAR-94	2003	1002
3009	1713.23	10-APR-94	2002	1003
3007	75.75	10-APR-94	2004	1002
3008	4723.95	10-MAY-94	2006	1001
3010	1309.95	10-JUN-94	2004	1002
3011	9891	10-JUN-94	2006	1001

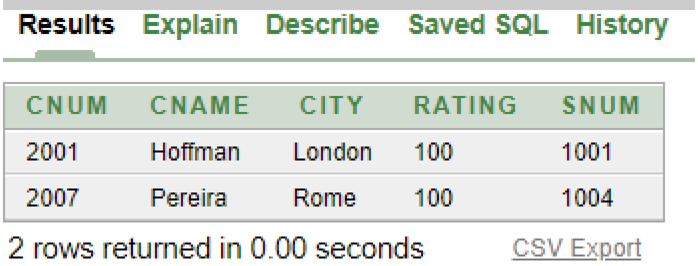
9 rows returned in 0.01 seconds

1. All customers serviced by Peel or Motika

QUERY:

Select Distinct c.* FROM Customer c JOIN Salespeople s ON c.snum = s.snum WHERE s.sname IN ('Peel', 'Motika');

OUTPUT:



CNUM	CNAME	CITY	RATING	SNUM
2001	Hoffman	London	100	1001
2007	Pereira	Rome	100	1004

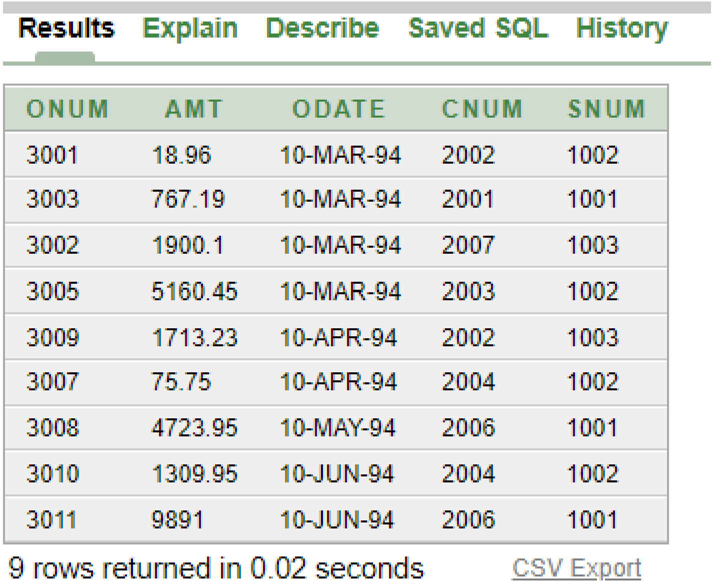
2 rows returned in 0.00 seconds

2. All orders except those with 0 or null value in the amt field

QUERUY:

Select * from "Order" where amt IS NOT NULL AND amt != 0;

OUTPUT:



ONUM	AMT	ODATE	CNUM	SNUM
3001	18.96	10-MAR-94	2002	1002
3003	767.19	10-MAR-94	2001	1001
3002	1900.1	10-MAR-94	2007	1003
3005	5160.45	10-MAR-94	2003	1002
3009	1713.23	10-APR-94	2002	1003
3007	75.75	10-APR-94	2004	1002
3008	4723.95	10-MAY-94	2006	1001
3010	1309.95	10-JUN-94	2004	1002
3011	9891	10-JUN-94	2006	1001

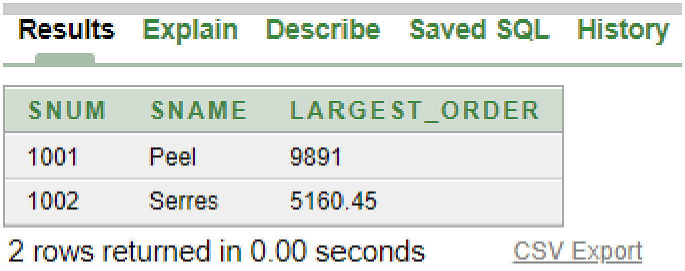
9 rows returned in 0.02 seconds

3. Largest order taken by each sales order value of more than 3000

QUERY:

Select s.snum, s.sname, MAX(o.amt) as largest_order FROM Salespeople s JOIN "Order" o ON s.snum = o.snum WHERE o.amt > 3000 GROUP BY s.snum, s.sname;

OUTPUT:



SNUM	SNAME	LARGEST_ORDER
1001	Peel	9891
1002	Serres	5160.45

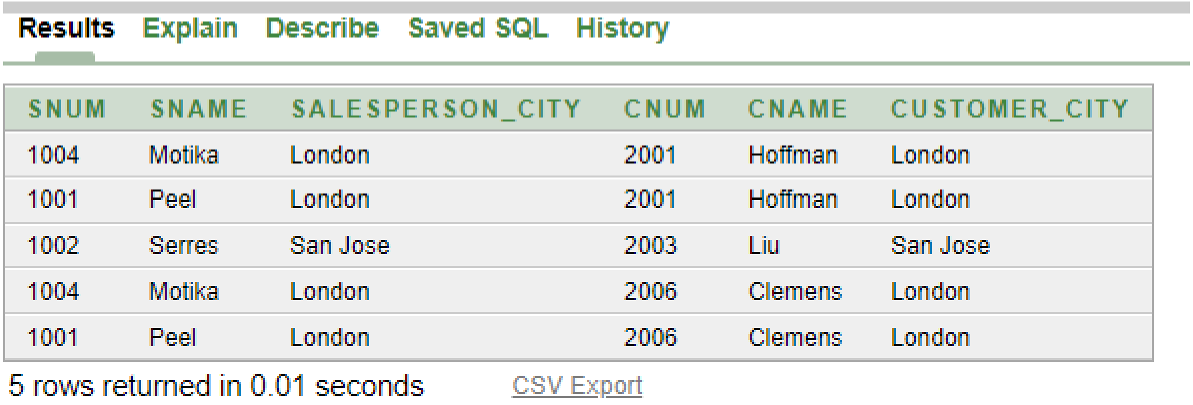
2 rows returned in 0.00 seconds

4. All combinations of salespeople and customers who belong to the same city

QUERY:

Select s.snum, s.sname, s.city AS salesperson_city, c.cnum, c.cname, c.city AS customer_city FROM Salespeople s JOIN Customer c ON s.city = c.city;

OUTPUT:



SNUM	SNAME	SALESPERSON_CITY	CNUM	CNAME	CUSTOMER_CITY
1004	Motika	London	2001	Hoffman	London
1001	Peel	London	2001	Hoffman	London
1002	Serres	San Jose	2003	Liu	San Jose
1004	Motika	London	2006	Clemens	London
1001	Peel	London	2006	Clemens	London

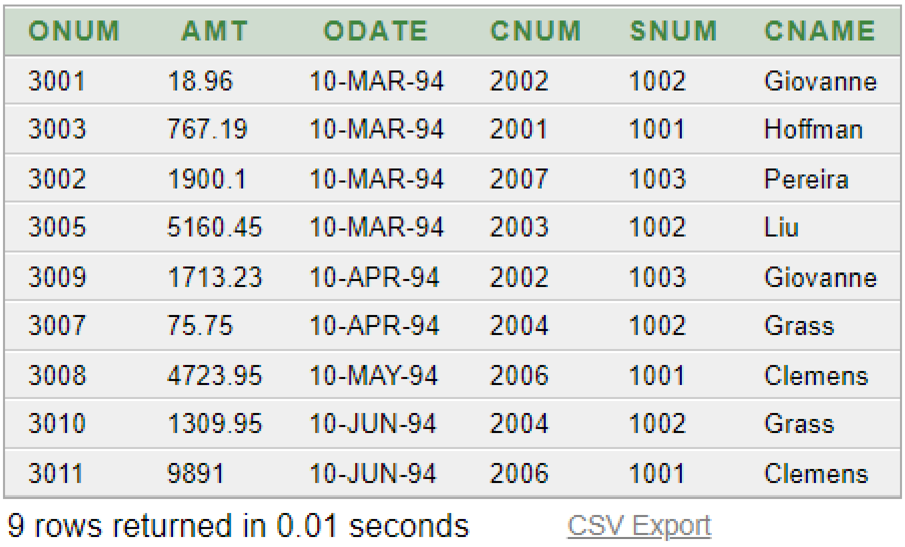
5 rows returned in 0.01 seconds

5. List each order with the name of the customer who placed the order

QUERY:

Select o.*, c.cname from "Order" o JOIN Customer c ON o.cnum = c.cnum;

OUTPUT:



ONUM	AMT	ODATE	CNUM	SNUM	CNAME
3001	18.96	10-MAR-94	2002	1002	Giovanne
3003	767.19	10-MAR-94	2001	1001	Hoffman
3002	1900.1	10-MAR-94	2007	1003	Pereira
3005	5160.45	10-MAR-94	2003	1002	Liu
3009	1713.23	10-APR-94	2002	1003	Giovanne
3007	75.75	10-APR-94	2004	1002	Grass
3008	4723.95	10-MAY-94	2006	1001	Clemens
3010	1309.95	10-JUN-94	2004	1002	Grass
3011	9891	10-JUN-94	2006	1001	Clemens

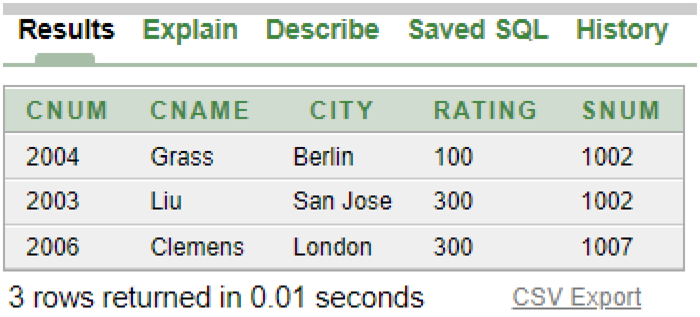
9 rows returned in 0.01 seconds

6. Produce a listing of all the customers serviced by salespeople having a commission of more than 12%

QUERY:

Select distinct c.* from Customer c JOIN Salespeople s ON c.snum = s.snum where s.comm > 0.12;

OUTPUT:



CNUM	CNAME	CITY	RATING	SNUM
2004	Grass	Berlin	100	1002
2003	Liu	San Jose	300	1002
2006	Clemens	London	300	1007

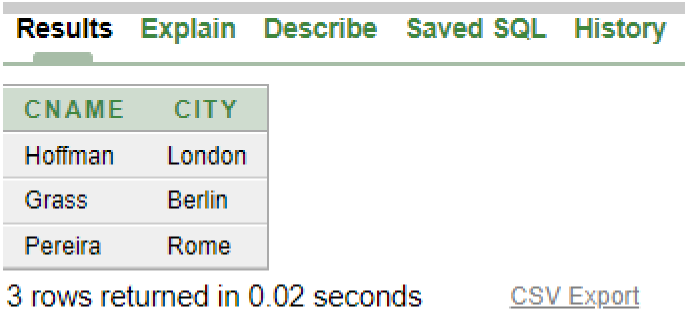
3 rows returned in 0.01 seconds

7. Produce names and cities of all customers with the same rating as Hoffman

QUERY:

Select cname, city from Customer where rating = (select rating from Customer where cname = 'Hoffman');

OUTPUT:



CNAME	CITY
Hoffman	London
Grass	Berlin
Pereira	Rome

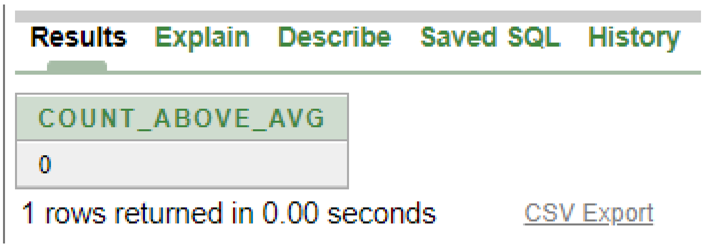
3 rows returned in 0.02 seconds

8. Count the customers with ratings above San Jose's average

QUERY:

Select COUNT(*) as count_above_avg FROM Customer where rating > (SELECT AVG(rating) FROM Customer WHERE city = 'San Jose');

OUTPUT:



COUNT_ABOVE_AVG
0

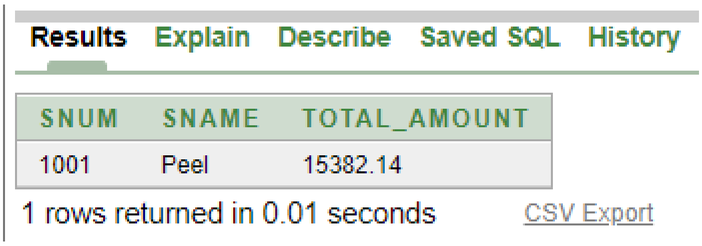
1 rows returned in 0.00 seconds

9. Find the total amount in orders for each salesperson for whom this total is greater than the amount of the largest order in the order table

QUERY:

SELECT s.snum, s.sname, SUM(o.amt) as total_amount FROM Salespeople s JOIN "Order" o ON s.snum = o.snum GROUP BY s.snum, s.sname HAVING SUM(o.amt) > (SELECT MAX(amt) FROM "Order");

OUTPUT:



SNUM	SNAME	TOTAL_AMOUNT
1001	Peel	15382.14

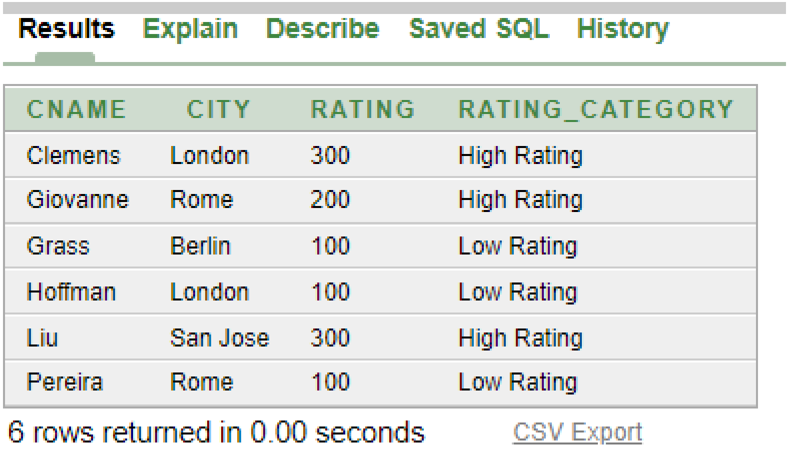
1 rows returned in 0.01 seconds

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

QUERY:

Select cname, city, rating, 'High Rating' as rating_category FROM Customer WHERE rating >= 200 UNION Select cname, city, rating, 'Low Rating' as rating_category FROM Customer where rating < 200;

OUTPUT:



CNAME	CITY	RATING	RATING_CATEGORY
Clemens	London	300	High Rating
Giovanne	Rome	200	High Rating
Grass	Berlin	100	Low Rating
Hoffman	London	100	Low Rating
Liu	San Jose	300	High Rating
Pereira	Rome	100	Low Rating

6 rows returned in 0.00 seconds

CONCLUSION:

- From is this practical I've learnt that how can we use concept of joins in a query form and to use different combinations of join to understand the concept.