# WORKSHEET 2

**Student Name:** Vivek Kumar **UID:** 21BCS8129
**DOMAIN CAMP:** 16-01-2023 to 28-01-2023 **Section/Group:** DWWC-77
**Subject Name:** Database Management System

**Question1. Consider the following tables of employees for years 2018 and 2019 employees_2018**

| EMPLOYEE_ID | EMPLOYEE_NAME | JOB | MANAGER_ID | HIREDATE | SALARY | COMMISSION | DEPARTMENT_ID |
|---|---|---|---|---|---|---|---|
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 800 | - | 20 |
| 7499 | ALLEN | SALESMAN | 7698 | 20-FEB-81 | 1600 | 300 | 30 |
| 7521 | WARD | SALESMAN | 7698 | 22-FEB-81 | 1250 | 500 | 30 |
| 7566 | JONES | MANAGER | 7839 | 02-APR-81 | 2975 | - | 20 |
| 7654 | MARTIN | SALESMAN | 7698 | 28-SEP-81 | 1250 | 1400 | 30 |
| 7698 | BLAKE | MANAGER | 7839 | 01-MAY-81 | 2850 | - | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09-JUN-81 | 2450 | - | 10 |

employees_2019

| EMPLOYEE_ID | EMPLOYEE_NAME | JOB | MANAGER_ID | HIREDATE | SALARY | COMMISSION | DEPARTMENT_ID |
|---|---|---|---|---|---|---|---|
| 7788 | SCOTT | ANALYST | 7566 | 19-APR-87 | 3000 | - | 20 |
| 7839 | KING | PRESIDENT | - | 17-NOV-81 | 5000 | - | 10 |
| 7844 | TURNER | SALESMAN | 7698 | 08-SEP-81 | 1500 | 0 | 30 |
| 7876 | ADAMS | CLERK | 7788 | 23-MAY-87 | 1100 | - | 20 |
| 7900 | JAMES | CLERK | 7698 | 03-DEC-81 | 950 | - | 30 |
| 7902 | FORD | ANALYST | 7566 | 03-DEC-81 | 3000 | - | 20 |
| 7934 | MILLER | CLERK | 7782 | 23-JAN-82 | 1300 | - | 10 |
| 7369 | SMITH | CLERK | 7902 | 17-DEC-80 | 800 | - | 20 |

a) Write a sql query to find list of distinct employees from both the tables working in department having department_id=20

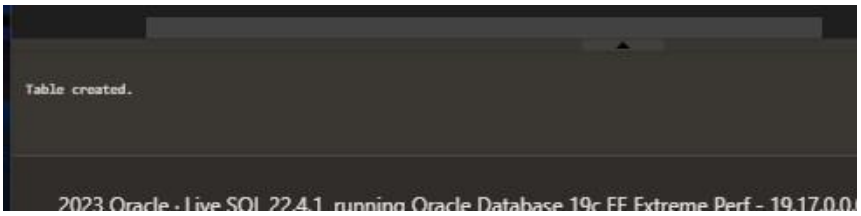
b) Write a sql query to find list of employees from both the tables working in department having department_id=20

c) Write a sql query to find the list of employees common in both the tables

d) Write a sql query to implement minus operator

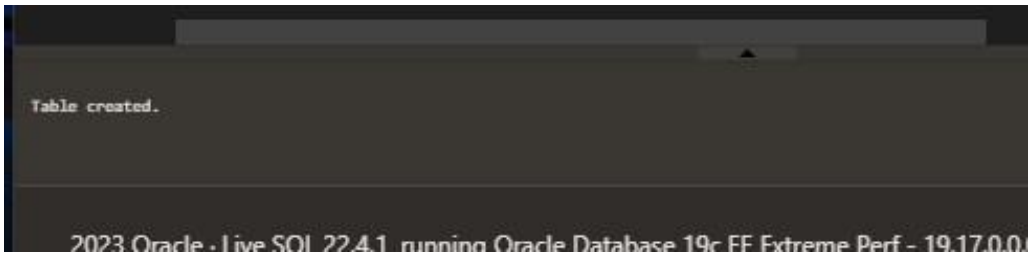
**Solution:**

CREATE TABLE employees_2018(employee_id NUMBER(4), employee_name VARCHAR2(10), job VARCHAR2(9), manager_id NUMBER(4), hiredate DATE, salary NUMBER(7,2), commission NUMBER(7,2), department_id NUMBER(2));

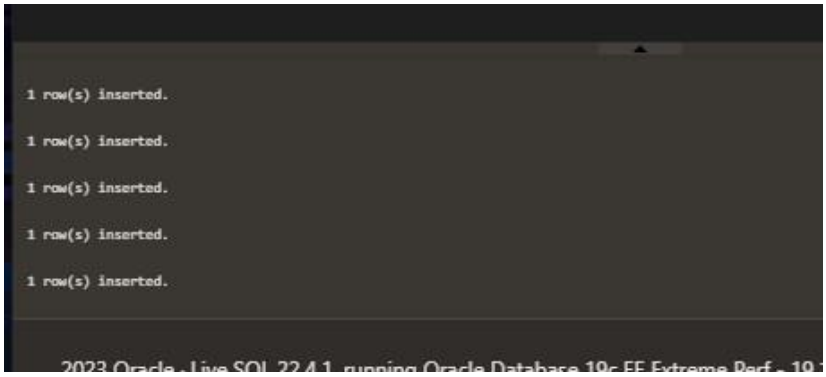



INSERT INTO employees_2018 VALUES (7369,'SMITH','CLERK',7902,to_date('17-12-1980','dd-mmyyyy'),800,NULL,20);
INSERT INTO employees_2018 VALUES (7499,'ALLEN','SALESMAN',7698,to_date('20-2-1981','dd-mmyyyy'),1600,300,30);
INSERT INTO employees_2018 VALUES (7521,'WARD','SALESMAN',7698,to_date('22-2-1981','dd-mmyyyy'),1250,500,30);
INSERT INTO employees_2018 VALUES (7566,'JONES','MANAGER',7839,to_date('2-4-1981','dd-mmyyyy'),2975,NULL,20);
INSERT INTO employees_2018 VALUES (7654,'MARTIN','SALESMAN',7698,to_date('28-9-1981','dd-mmyyyy'),1250,1400,30);
INSERT INTO employees_2018 VALUES (7698,'BLAKE','MANAGER',7839,to_date('1-5-1981','dd-mmyyyy'),2850,NULL,30);
INSERT INTO employees_2018 VALUES (7782,'CLARK','MANAGER',7839,to_date('9-6-1981','dd-mmyyyy'),2450,NULL,10);


CREATE TABLE employees_2019(employee_id NUMBER(4), employee_name VARCHAR2(10), job VARCHAR2(9), manager_id NUMBER(4), hiredate DATE, salary NUMBER(7,2), commission NUMBER(7,2), department_id NUMBER(2));




INSERT INTO employees_2019 VALUES (7788,'SCOTT','ANALYST',7566,to_date('13-JUL-87','dd-mm-rr')85,3000,NULL,20);
INSERT INTO employees_2019 VALUES (7839,'KING','PRESIDENT',NULL,to_date('17-11-1981','dd-mmyyyy'),5000,NULL,10);
INSERT INTO employees_2019 VALUES (7844,'TURNER','SALESMAN',7698,to_date('8-9-1981','dd-mmyyyy'),1500,0,30);
INSERT INTO employees_2019 VALUES (7876,'ADAMS','CLERK',7788,to_date('13-JUL-87', 'dd-mm-rr')51,1100,NULL,20);
INSERT INTO employees_2019 VALUES (7900,'JAMES','CLERK',7698,to_date('3-12-1981','dd-mmyyyy'),950,NULL,30);
INSERT INTO employees_2019 VALUES (7902,'FORD','ANALYST',7566,to_date('3-12-1981','dd-mmyyyy'),3000,NULL,20);
INSERT INTO employees_2019 VALUES (7934,'MILLER','CLERK',7782,to_date('23-1-1982','dd-mmyyyy'),1300,NULL,10);
INSERT INTO employees_2019 VALUES (7369,'SMITH','CLERK',7902,to_date('17-12-1980','dd-mmyyyy'),800,NULL,20);




SELECT employee_id,employee_name,job,salary from employees_2018 WHERE department_id = 20 UNION SELECT employee_id,employee_name,job,salary from employees_2019 WHERE department_id = 20

| EMPLOYEE_ID | EMPLOYEE_NAME | JOB | SALARY |
|---|---|---|---|
| 7369 | SMITH | CLERK | 800 |
| 7566 | JONES | MANAGER | 2975 |
| 7788 | SCOTT | ANALYST | 3000 |
| 7876 | ADAMS | CLERK | 1100 |
| 7902 | FORD | ANALYST | 3000 |

SELECT employee_id,employee_name,job,salary from employees_2018 WHERE department_id = 20 UNION ALL
SELECT employee_id,employee_name,job,salary from employees_2019 WHERE department_id = 20

| EMPLOYEE_ID | EMPLOYEE_NAME | JOB | SALARY |
|---|---|---|---|
| 7369 | SMITH | CLERK | 800 |
| 7566 | JONES | MANAGER | 2975 |
| 7788 | SCOTT | ANALYST | 3000 |
| 7876 | ADAMS | CLERK | 1100 |
| 7902 | FORD | ANALYST | 3000 |
| 7369 | SMITH | CLERK | 800 |

Download CSV

SELECT employee_id,employee_name,job,salary from employees_2018 WHERE department_id = 20 INTERSECT
SELECT employee_id,employee_name,job,salary from employees_2019 WHERE department_id = 20

| EMPLOYEE_ID | EMPLOYEE_NAME | JOB | SALARY |
|---|---|---|---|
| 7369 | SMITH | CLERK | 800 |

Download CSV

SELECT employee_id,employee_name,job,salary from employees_2018 WHERE department_id = 20 MINUS SELECT employee_id,employee_name,job,salary from employees_2019 WHERE department_id = 20

| EMPLOYEE_ID | EMPLOYEE_NAME | JOB | SALARY |
|---|---|---|---|
| 7566 | JONES | MANAGER | 2975 |

Download CSV

**Question2. For the given table**

ord_no purch_amt ord_date customer_id salesman_id

---------- ---------- ---------- ---------- ---------- 70001

150.5 2012-10-05 3005 5002

70009 270.65 2012-09-10 3001 5005

70002 65.26 2012-10-05 3002 5001

70004 110.5 2012-08-17 3009 5003

70007 948.5 2012-09-10 3005 5002

70005 2400.6 2012-07-27 3007 5001

70008 5760 2012-09-10 3002 5001

70010 1983.43 2012-10-10 3004 5006

70003 2480.4 2012-10-10 3009 5003

70012 250.45 2012-06-27 3008 5002

70011 75.29 2012-08-17 3003 5007 70013

3045.6 2012-04-25 3002 5001

a. Write a SQL statement to find the highest purchase amount ordered by the each customer with their ID and highest purchase amount.

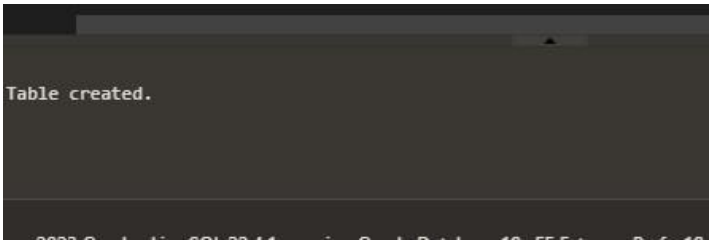
b. Write a SQL statement to find the highest purchase amount ordered by the each customer on a particular order date with their ID,

c. Write a SQL statement to find the highest purchase amount with their ID and order date, for only those customers who have highest purchase amount in a day is more than 2000.

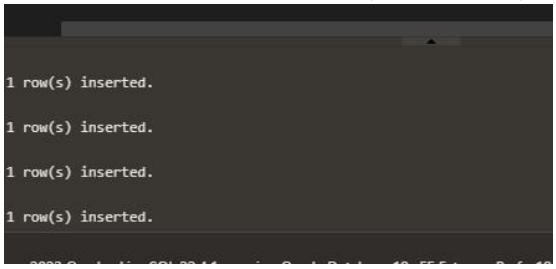
d. Write a SQL statement to find the highest purchase amount with their ID and order date, for only those customers who have a higher purchase amount in a day is within the list 2000, 3000, 5760 and 6000.

**Solution:**

Create Table orders(order_no number, purch_amt number(6,2), ord_date date, customer_id number, salesman_id number);




insert into orders values(70001, 150.5, to_date('18-05-2012', 'DD-MM-YYYY'),3005, 5002);
insert into orders values(70009, 270.65, to_date('09-10-2012', 'DD-MM-YYYY'),3001, 5005);
insert into orders values(70002, 65.26, to_date('10-05-2012', 'DD-MM-YYYY'),3002, 5001);
insert into orders values(70004, 110.5,to_date( '18-08-2012', 'DD-MM-YYYY'),3009, 5003);
insert into orders values(70007, 948.5, to_date('09-10-2012', 'DD-MM-YYYY'),3005, 5002);
insert into orders values(70005, 2400.6,to_date( '27-07-2012', 'DD-MM-YYYY'),3007, 5001);
insert into orders values(70008, 5760,to_date( '09-10-2012', 'DD-MM-YYYY'),3002, 5001); insert into orders values(70010, 1983.43,to_date( '10-10-2012','DD-MM-YYYY'), 3004, 5006); insert into orders values(70003, 2480.4, to_date('10-10-2012', 'DD-MM-YYYY'),3009, 5003); insert into orders values(70012, 250.45, to_date('27-06-2012', 'DD-MM-YYYY'),3008, 5002); insert into orders values(70011, 75.29, to_date('17-08-2012', 'DD-MM-YYYY'),3003, 5007); insert into orders values(70013, 3045.6,to_date( '25-04-2012', 'DD-MM-YYYY'),3002, 5001);




SELECT customer_id,MAX(purch_amt) FROM orders GROUP BY customer_id;

| CUSTOMER_ID | MAX(PURCH_AMT) |
|---|---|
| 3009 | 2480.4 |
| 3005 | 948.5 |
| 3002 | 5760 |
| 3004 | 1983.43 |
| 3003 | 75.29 |
| 3008 | 250.45 |
| 3001 | 270.65 |
| 3007 | 2400.6 |

SELECT customer_id,ord_date,MAX(purch_amt) FROM orders GROUP BY customer_id,ord_date;

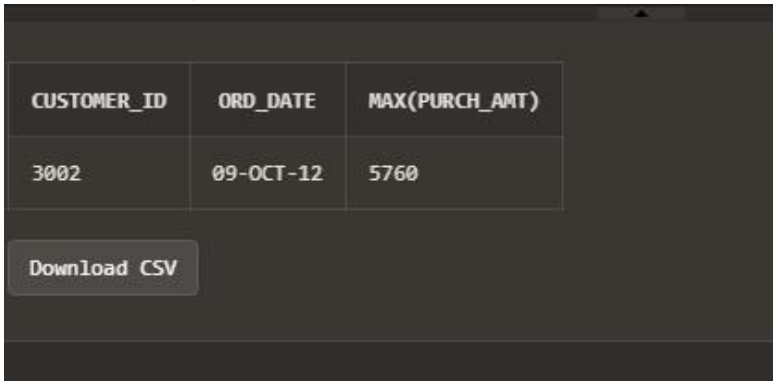
| CUSTOMER_ID | ORD_DATE | MAX(PURCH_AMT) |
|---|---|---|
| 3008 | 27-JUN-12 | 250.45 |
| 3002 | 25-APR-12 | 3045.6 |
| 3001 | 09-OCT-12 | 270.65 |
| 3003 | 17-AUG-12 | 75.29 |
| 3002 | 09-OCT-12 | 5760 |
| 3009 | 10-OCT-12 | 2480.4 |
| 3005 | 18-MAY-12 | 150.5 |
| 3002 | 10-MAY-12 | 65.26 |
| 3004 | 10-OCT-12 | 1983.43 |

SELECT customer_id, ord_date, MAX(purch_amt) FROM orders GROUP BY customer_id,ord_date HAVING MAX(purch_amt)>2000.00;

| CUSTOMER_ID | ORD_DATE | MAX(PURCH_AMT) |
|---|---|---|
| 3002 | 25-APR-12 | 3045.6 |
| 3002 | 09-OCT-12 | 5760 |
| 3009 | 10-OCT-12 | 2480.4 |
| 3007 | 27-JUL-12 | 2400.6 |

Download CSV

SELECT customer_id,ord_date,MAX(purch_amt) FROM orders GROUP BY customer_id,ord_date HAVING MAX(purch_amt) IN(2000 ,3000,5760, 6000);



| CUSTOMER_ID | ORD_DATE | MAX(PURCH_AMT) |
|---|---|---|
| 3002 | 09-OCT-12 | 5760 |

Download CSV

**Question3. Consider the following schema**

|CUST_CODE | CUST_NAME | CUST_CITY | WORKING_AREA | CUST_COUNTRY | GRADE | OPENING_AMT | RECEIVE_AMT | PAYMENT_AMT |OUTSTANDING_AMT| PHONE_NO | AGENT_CODE |

+----------+------------+------------+-------------+-------------+-------+------------+------------+------------+---------------+----------+------------+

| C00013 | Holmes | London | London | UK | 2 | 6000.00 | 5000.00 | 7000.00 | 4000.00 | BBBBBBB | A003 |

| C00001 | Micheal | New York | New York | USA | 2 | 3000.00 | 5000.00 | 2000.00 | 6000.00 | CCCCCCC | A008 |

| C00020 | Albert | New York | New York | USA | 3 | 5000.00 | 7000.00 | 6000.00 | 6000.00 | BBBBSBB | A008 |

| C00025 | Ravindran | Bangalore | Bangalore | India | 2 | 5000.00 | 7000.00 | 4000.00 | 8000.00 | AVAVAVA | A011 |

| C00024 | Cook | London | London | UK | 2 | 4000.00 | 9000.00 | 7000.00 | 6000.00 | FSDDSDF | A006 |

| C00015 | Stuart | London | London | UK | 1 | 6000.00 | 8000.00 | 3000.00 | 11000.00 | GFSGERS | A003 |

| C00002 | Bolt | New York | New York | USA | 3 | 5000.00 | 7000.00 | 9000.00 | 3000.00 | DDNRDRH | A008 |

| C00018 | Fleming | Brisban | Brisban | Australia | 2 | 7000.00 | 7000.00 | 9000.00 | 5000.00 | NHBGVFC | A005 |

| C00021 | Jacks | Brisban | Brisban | Australia | 1 | 7000.00 | 7000.00 | 7000.00 | 7000.00 | WERTGDF | A005 |

| C00019 | Yearannaidu | Chennai | Chennai | India | 1 | 8000.00 | 7000.00 | 7000.00 | 8000.00 | ZZZZBFV | A010 |

| C00005 | Sasikant | Mumbai | Mumbai | India | 1 | 7000.00 | 11000.00 | 7000.00 | 11000.00 | 147-25896312 | A002 |

| C00007 | Ramanathan | Chennai | Chennai | India | 1 | 7000.00 | 11000.00 | 9000.00 | 9000.00 | GHRDWSD | A010 |

| C00022 | Avinash | Mumbai | Mumbai | India | 2 | 7000.00 | 11000.00 | 9000.00 | 9000.00 | 113-12345678 | A002 |

| C00004 | Winston | Brisban | Brisban | Australia | 1 | 5000.00 | 8000.00 | 7000.00 | 6000.00 | AAAAAAA | A005 |
| C00023 | Karl | London | London | UK | 0 | 4000.00 | 6000.00 | 7000.00 | 3000.00 | AAAABAA | A006 |

| C00006 | Shilton | Torento | Torento | Canada | 1 | 10000.00 | 7000.00 | 6000.00 | 11000.00 | DDDDDDD | A004 |

| C00010 | Charles | Hampshair | Hampshair | UK | 3 | 6000.00 | 4000.00 | 5000.00 | 5000.00 | MMMMMMM | A009 |

| C00017 | Srinivas | Bangalore | Bangalore | India | 2 | 8000.00 | 4000.00 | 3000.00 | 9000.00 | AAAAAAB | A007 |

| C00012 | Steven | San Jose | San Jose | USA | 1 | 5000.00 | 7000.00 | 9000.00 | 3000.00 | KRFYGJK | A012 |

| C00008 | Karolina | Torento | Torento | Canada | 1 | 7000.00 | 7000.00 | 9000.00 | 5000.00 | HJKORED | A004 |

| C00003 | Martin | Torento | Torento | Canada | 2 | 8000.00 | 7000.00 | 7000.00 | 8000.00 | MJYURFD | A004 |

| C00009 | Ramesh | Mumbai | Mumbai | India | 3 | 8000.00 | 7000.00 | 3000.00 | 12000.00 | Phone No | A002 |

| C00014 | Rangarappa | Bangalore | Bangalore | India | 2 | 8000.00 | 11000.00 | 7000.00 | 12000.00 | AAAATGF | A001 |

| C00016 | Venkatpati | Bangalore | Bangalore | India | 2 | 8000.00 | 11000.00 | 7000.00 | 12000.00 | JRTVFDD | A007 |

| C00011 | Sundariya | Chennai | Chennai | India | 3 | 7000.00 | 11000.00 | 7000.00 | 11000.00 | PPHGRTS | A010 |

+----------+-------------+-------------+-------------+-------------+-------+------------+------------+------------+--------------+------------+

a) Write sql command to get data of 'cust_code', 'cust_name', 'cust_city', 'cust_country' and 'grade' from the 'customer' table with following conditions -

1.'cust_country' is ’UK',

2.and 'cust_city' is 'London',

3.and 'grade' of the 'customer' must be greater than 1,

b) Write SQL command to get data of 'cust_code', 'cust_name', 'cust_city', 'cust_country' and 'grade' from the 'customer' table with following conditions -

WHERE (cust_country = ’UK’ or cust_city = 'London') AND,
and GRADE <> 3

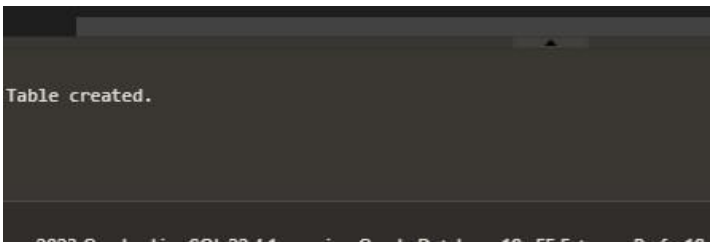
c) Write sql query to To get data of 'cust_code', 'cust_name', 'cust_city', 'cust_country' and 'grade' from the 'customer' table with following conditions -

1.'cust_country' is other than 'India',

2.and 'grade' of the 'customer' must be 3,

**Solution:**

Create TABLE customer_details(CUST_CODE varchar(15), CUST_NAME varchar(50), CUST_CITY varchar(50), WORKING_AREA varchar(50), CUST_COUNTRY varchar(50), GRADE NUMBER, OPENING_AMT float, RECEIVE_AMT float, PAYMENT_AMT float, OUTSTANDING_AMT float, PHONE_NO varchar(20), AGENT_CODE varchar(10));

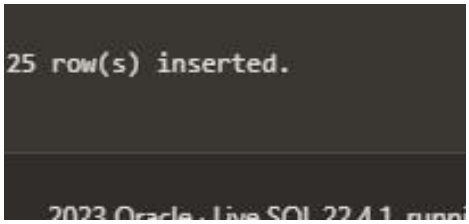



select * from customer_details;

INSERT ALL
INTO customer_details VALUES
('C00013','Holmes','London','London','UK',2,6000.00,5000.00,7000.00,4000.00,'BBBBBBB','A003')

```
 INTO customer_details VALUES ('C00001','Micheal','New York','New
York','USA',2,3000.00,5000.00,2000.00,6000.00,'CCCCCCC','A008')
 INTO customer_details VALUES ('C00020','Albert','New York','New
York','USA',3,5000.00,7000.00,6000.00,6000.00,'BBBBSBB','A008')
 INTO customer_details VALUES
('C00025','Ravindran','Bangalore','Bangalore','India',2,5000.00,7000.00,4000.00,8000.00,'AVAVAVA','A011')
INTO customer_details VALUES
('C00024','Cook','London','London','UK',2,4000.00,9000.00,7000.00,6000.00,'FSDDSDF','A006')
INTO customer_details VALUES
('C00015','Stuart','London','London','UK',1,6000.00,8000.00,3000.00,11000.00,'GFSGERS','A003')
 INTO customer_details VALUES ('C00002','Bolt','New York','New
York','USA',3,5000.00,7000.00,9000.00,3000.00,'DDNRDRH','A008')
 INTO customer_details VALUES
('C00018','Fleming','Brisban','Brisban','Australia',2,7000.00,7000.00,9000.00,5000.00,'NHBGVFC','A005')
INTO customer_details VALUES
('C00021','Jacks','Brisban','Brisban','Australia',1,7000.00,7000.00,7000.00,7000.00,'WERTGDF','A005')
INTO customer_details VALUES ('C00019
','Yearannaidu','Chennai','Chennai','India',1,8000.00,7000.00,7000.00,8000.00,'ZZZZBFV','A010')
 INTO customer_details VALUES
('C00005','Sasikant','Mumbai','Mumbai','India',1,7000.00,11000.00,7000.00,11000.00,'147-25896312','A002')
INTO customer_details VALUES
('C00007','Ramanathan','Chennai','Chennai','India',1,7000.00,11000.00,9000.00,9000.00,'GHRDWSD','A010')
 INTO customer_details VALUES ('C00022','Avinash','Mumbai','Mumbai','India',2,7000.00,11000.00,9000.00,9000.00,'
113-12345678','A002')
 INTO customer_details VALUES
('C00004','Winston','Brisban','Brisban','Australia',1,5000.00,8000.00,7000.00,6000.00,'AAAAAAA','A005')
INTO customer_details VALUES
('C00023','Karl','London','London','UK',0,4000.00,6000.00,7000.00,3000.00,'AAAABAA','A006')
INTO customer_details VALUES
('C00006','Shilton','Torento','Torento','Canada',1,10000.00,7000.00,6000.00,11000.00,'DDDDDDD','A004')
INTO customer_details VALUES
('C00010','Charles','Hampshair','Hampshair','UK',3,6000.00,4000.00,5000.00,5000.00,'MMMMMMM','A009')
INTO customer_details VALUES
('C00017','Srinivas','Bangalore','Bangalore','India',2,8000.00,4000.00,3000.00,9000.00,'AAAAAAB','A007')
 INTO customer_details VALUES ('C00012','Steven','San Jose','San
Jose','USA',1,5000.00,7000.00,9000.00,3000.00,'KRFYGJK','A012')
 INTO customer_details VALUES
('C00008','Karolina','Torento','Torento','Canada',1,7000.00,7000.00,9000.00,5000.00,'HJKORED','A004')
INTO customer_details VALUES
('C00003','Martin','Torento','Torento','Canada',2,8000.00,7000.00,7000.00,8000.00,'MJYURFD','A004')
 INTO customer_details VALUES ('C00009','Ramesh','Mumbai','Mumbai','India',3,8000.00,7000.00,3000.00,12000.00,'
113-1263343','A002')
 INTO customer_details VALUES
```

('C00014','Rangarappa','Bangalore','Bangalore','India',2,8000.00,11000.00,7000.00,12000.00,' AAAATGF','A001')
INTO customer_details VALUES
('C00016','Venkatpati','Bangalore','Bangalore','India',2,8000.00,11000.00,7000.00,12000.00,' JRTVFDD','A007')
INTO customer_details VALUES
('C00011','Sundariya','Chennai','Chennai','India',3,8000.00,11000.00,7000.00,11000.00,' PPHGRTS','A010')  select * from dual;

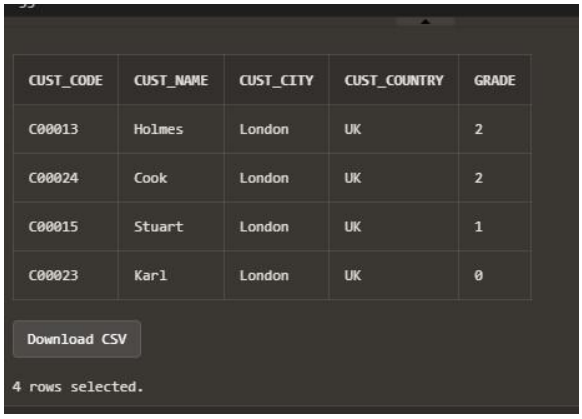



SELECT cust_code,cust_name,cust_city,cust_country,grade FROM customer_details WHERE cust_country='UK' AND cust_city='London' AND  grade>1;

| CUST_CODE | CUST_NAME | CUST_CITY | CUST_COUNTRY | GRADE |
|---|---|---|---|---|
| C00013 | Holmes | London | UK | 2 |
| C00024 | Cook | London | UK | 2 |

Download CSV

SELECT cust_code, cust_name, cust_city, cust_country, grade FROM customer_details WHERE (cust_country = 'UK' OR cust_city = 'London') AND grade < > 3;

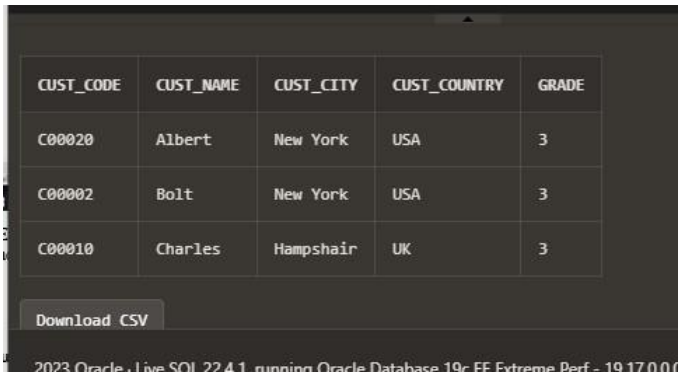


| CUST_CODE | CUST_NAME | CUST_CITY | CUST_COUNTRY | GRADE |
|---|---|---|---|---|
| C00013 | Holmes | London | UK | 2 |
| C00024 | Cook | London | UK | 2 |
| C00015 | Stuart | London | UK | 1 |
| C00023 | Karl | London | UK | 0 |

Download CSV

4 rows selected.

SELECT cust_code, cust_name, cust_city, cust_country, grade FROM customer_details WHERE NOT cust_country = 'India' AND grade = 3;



| CUST_CODE | CUST_NAME | CUST_CITY | CUST_COUNTRY | GRADE |
|---|---|---|---|---|
| C00020 | Albert | New York | USA | 3 |
| C00002 | Bolt | New York | USA | 3 |
| C00010 | Charles | Hampshair | UK | 3 |

Download CSV

**Question4. Implement flashback and purge command**

**Solution:**

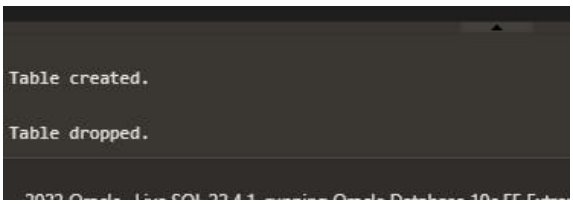
a) Flashback command

It is used to recover a dropped table. Drop table <tablename>. Flashback table <tablename> to before drop.

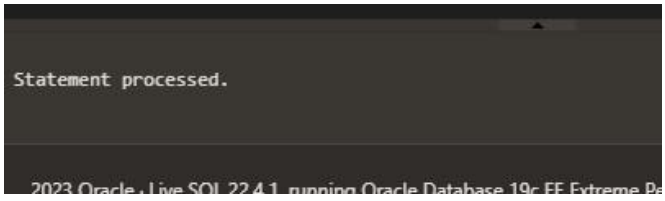
create table sample

(name varchar(20)); drop

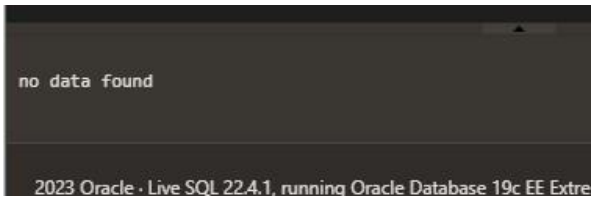
table sample




flashback table sample to before drop




select * from sample

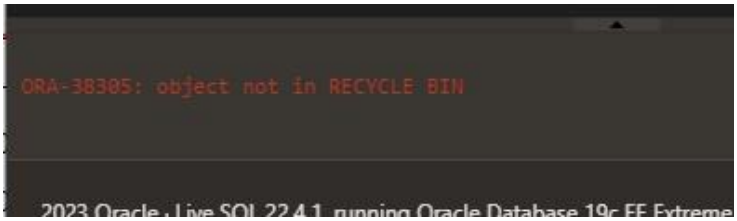



b)

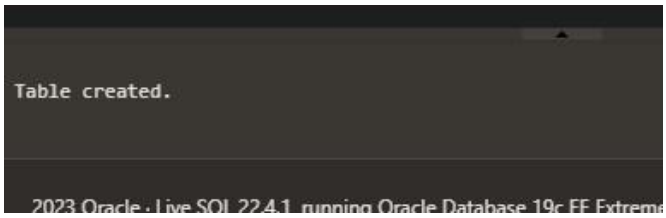
Purge command

To permanently delete a table. Table will not be saved in recycle bin from where otherwise it could have been recovered using flashback command. Drop table <tablename> purge drop table sample purge

flashback table sample to before drop;



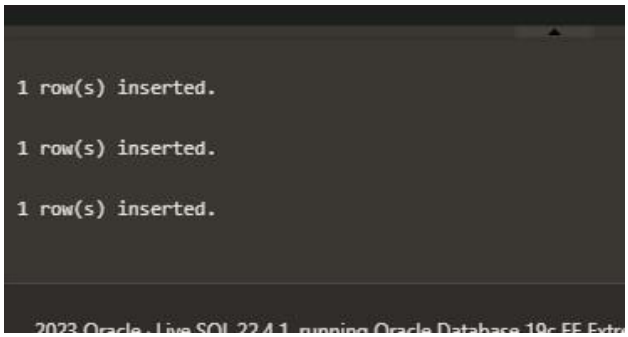

c) Timestamp create table sample(name varchar(20));



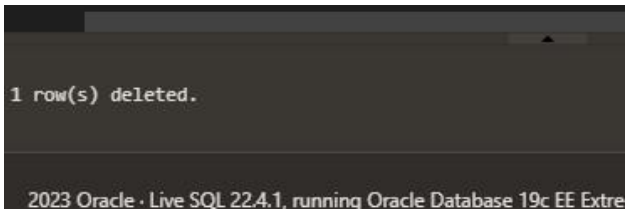

insert into sample values('Armaan');

insert into sample values('Mehak'); insert

into sample values('Nirvaan');

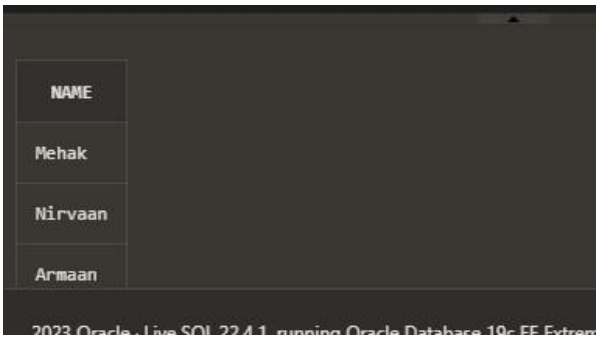



delete from sample where name = 'Armaan'




**To insert the deleted record 2 mins back, we use**

INSERT INTO sample (SELECT * FROM sample AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '10' SECOND)) MINUS (SELECT * FROM sample); **Deleted record recovered** select * from sample




**Question5. Consider the schema**

ord_no purch_amt ord_date customer_id salesman_id

---------- ---------- ---------- ---------- ---------- 70001

150.5 2012-10-05 3005 5002

70009 270.65 2012-09-10 3001 5005

70002 65.26 2012-10-05 3002 5001

70004 110.5 2012-08-17 3009 5003

70007 948.5 2012-09-10 3005 5002

70005 2400.6 2012-07-27 3007 5001

70008 5760 2012-09-10 3002 5001

70010 1983.43 2012-10-10 3004 5006

70003 2480.4 2012-10-10 3009 5003

70012 250.45 2012-06-27 3008 5002

70011 75.29 2012-08-17 3003 5007 70013

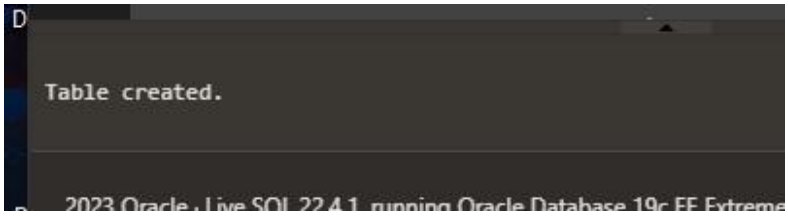
3045.6 2012-04-25 3002 5001

a. Write a SQL statement to display either those orders which are not issued on date 2012-09-10 and issued by the salesman whose ID is 5005 and below or those orders which purchase amount is 1000.00

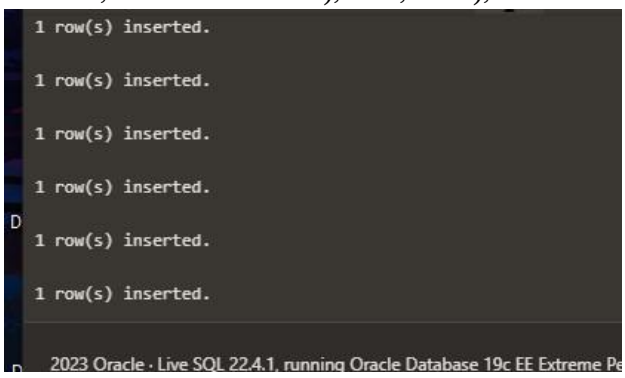
b. Write a SQL statement to exclude the rows which satisfy 1) order dates are 2012-08-17 and purchase amount is below 1000 2) customer id is greater than 3005 and purchase amount is below 1000.

**Solution:**

Create Table orders(order_no number, purch_amt number(6,2), ord_date date, customer_id number, salesman_id number);




insert into orders values(70001, 150.5, to_date('18-05-2012', 'DD-MM-YYYY'),3005, 5002); insert into orders values(70009, 270.65, to_date('09-10-2012', 'DD-MM-YYYY'),3001, 5005); insert into orders values(70002, 65.26, to_date('10-05-2012', 'DD-MM-YYYY'),3002, 5001); insert into orders values(70004, 110.5,to_date( '18-08-2012', 'DD-MM-YYYY'),3009, 5003); insert into orders values(70007, 948.5, to_date('09-10-2012', 'DD-MM-YYYY'),3005, 5002); insert into orders values(70005, 2400.6,to_date( '27-07-2012', 'DD-MM-YYYY'),3007, 5001); insert into orders values(70008, 5760,to_date( '09-10-2012', 'DD-MM-YYYY'),3002, 5001); insert into orders values(70010, 1983.43,to_date( '10-10-2012','DD-MM-YYYY'), 3004, 5006); insert into orders values(70003, 2480.4, to_date('10-10-2012', 'DD-MM-YYYY'),3009, 5003); insert into orders values(70012, 250.45, to_date('27-06-2012', 'DD-MM-YYYY'),3008, 5002); insert into orders values(70011, 75.29, to_date('17-08-2012', 'DD-MM-YYYY'),3003, 5007); insert into orders values(70013, 3045.6,to_date( '25-04-2012', 'DD-MM-YYYY'),3002, 5001); select * from orders;




SELECT * FROM orders WHERE NOT (ord_date ='09-OCT-2012') AND salesman_id<=5005 OR purch_amt=1000.00;

| ORDER_NO | PURCH_AMT | ORD_DATE | CUSTOMER_ID | SALESMAN_ID |
|---|---|---|---|---|
| 70001 | 150.5 | 18-MAY-12 | 3005 | 5002 |
| 70002 | 65.26 | 10-MAY-12 | 3002 | 5001 |
| 70004 | 110.5 | 18-AUG-12 | 3009 | 5003 |
| 70005 | 2400.6 | 27-JUL-12 | 3007 | 5001 |
| 70003 | 2480.4 | 10-OCT-12 | 3009 | 5003 |
| 70012 | 250.45 | 27-JUN-12 | 3008 | 5002 |
| 70013 | 3045.6 | 25-APR-12 | 3002 | 5001 |

Download CSV

SELECT * FROM orders WHERE NOT ord_date ='09-OCT-2012' AND NOT purch_amt<1000;

| ORDER_NO | PURCH_AMT | ORD_DATE | CUSTOMER_ID | SALESMAN_ID |
|---|---|---|---|---|
| 70005 | 2400.6 | 27-JUL-12 | 3007 | 5001 |
| 70010 | 1983.43 | 10-OCT-12 | 3004 | 5006 |
| 70003 | 2480.4 | 10-OCT-12 | 3009 | 5003 |
| 70013 | 3045.6 | 25-APR-12 | 3002 | 5001 |

Download CSV

4 rows selected.

SELECT * FROM orders WHERE NOT customer_id>3005 AND NOT purch_amt<1000;

| ORDER_NO | PURCH_AMT | ORD_DATE | CUSTOMER_ID | SALESMAN_ID |
|---|---|---|---|---|
| 70008 | 5760 | 09-OCT-12 | 3002 | 5001 |
| 70010 | 1983.43 | 10-OCT-12 | 3004 | 5006 |
| 70013 | 3045.6 | 25-APR-12 | 3002 | 5001 |

Download CSV