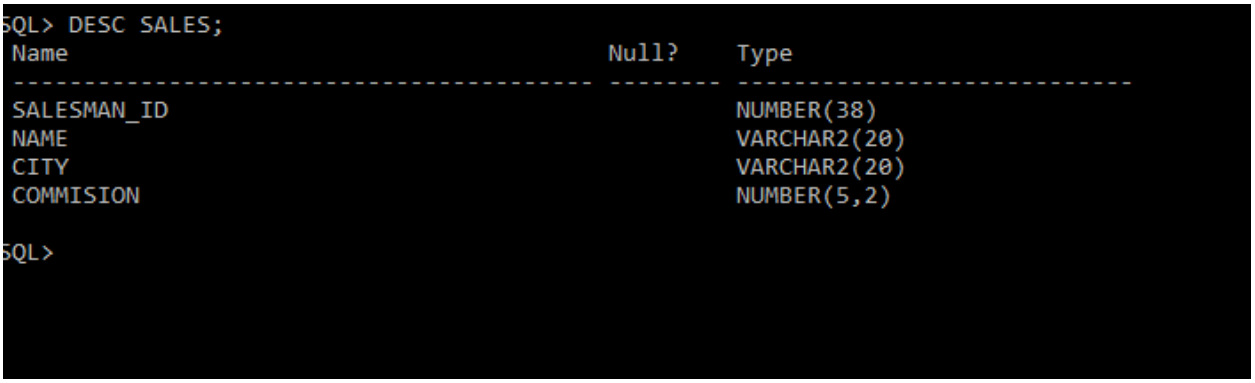
SQL FOUNDATION L1

1. Write a SQL statement to Create below table.

```
SQL>  CREATE TABLE SALES(
        SALESMAN_ID INTEGER,
        NAME VARCHAR2(20),
        CITY VARCHAR2(20),
        COMMISION NUMBER(5,2)
        );
```




```
INSERT INTO SALES VALUES(5001,'JAMES HOOG','NEW YORK',0.15);
INSERT INTO SALES VALUES(5002,'NAIL KNITE','PARIS',0.13);
INSERT INTO SALES VALUES(5005,'PIT ALEX','LONDON',0.11);
INSERT INTO SALES VALUES(5006,'MC LYON','PARIS',0.14);
INSERT INTO SALES VALUES(5003,'LAUSON HEN','SYDNEY',0.12);
INSERT INTO SALES VALUES(5007,'PAUL ADAM','ROME',0.13);

SELECT * FROM SALES;
```

```
SQL> SELECT * FROM SALES;

SALESMAN_ID NAME                 CITY                 COMMISION
----------- -------------------- -------------------- ----------
       5001 JAMES HOOG           NEW YORK                   .15
       5002 NAIL KNITE           PARIS                      .13
       5005 PIT ALEX             LONDON                     .11
       5006 MC LYON              PARIS                      .14
       5003 LAUSON HEN           SYDNEY                     .12
       5007 PAUL ADAM            ROME                       .13
```

2. Write a SQL statement to display specific columns like Salesman_id and Name from above table.

SQL> SELECT SALESMAN_ID,NAME FROM SALES;

```
SQL> SELECT SALESMAN_ID,NAME FROM SALES;

SALESMAN_ID NAME
----------- --------------------
       5001 JAMES HOOG
       5002 NAIL KNITE
       5005 PIT ALEX
       5006 MC LYON
       5003 LAUSON HEN
       5007 PAUL ADAM

6 rows selected.
```

3. Write a query to display Name column in accending order from above table.

SQL> SELECT NAME FROM SALES ORDER BY NAME;

```
SQL> SELECT NAME FROM SALES ORDER BY NAME;

NAME
--------------------
JAMES HOOG
LAUSON HEN
MC LYON
NAIL KNITE
PAUL ADAM
PIT ALEX

6 rows selected.

SQL>
```

4. Write a SQL statement for above table to diplay names and city of Salesman, who belongs to the city of Paris.

SQL> SELECT NAME,CITY FROM SALES WHERE CITY = 'PARIS';

```
SQL> SELECT NAME,CITY FROM SALES WHERE CITY = 'PARIS';

NAME                 CITY
-------------------- --------------------
NAIL KNITE           PARIS
MC LYON              PARIS

SQL>
```

5. Write a query on above table to filter those salesmen with all information who comes from any of the cities Paris and Rome.

SQL>SELECT * FROM SALES WHERE CITY ='PARIS' OR CITY = 'ROME';

```
SQL> SELECT * FROM SALES WHERE CITY ='PARIS' OR  CITY = 'ROME';

SALESMAN_ID NAME                 CITY                 COMMISION
----------- -------------------- -------------------- ----------
       5002 NAIL KNITE           PARIS                       .13
       5006 MC LYON              PARIS                       .14
       5007 PAUL ADAM            ROME                        .13

SQL>
```

6. Write a SQL statement to find those salesmen with all the other information and name started with any letter wining ‘A’ and ‘K’ from above table.

SQL> SELECT * FROM SALES WHERE NAME LIKE 'A%' OR NAME LIKE 'K%';

```
SQL> SELECT * FROM SALES WHERE NAME LIKE 'A%' OR NAME LIKE 'K%';

no rows selected

SQL>
```

7.Write a SQL statement to display all the information for those customers with a Grade of 200.

CREATE TABLE CUSTOMER

(CUSTOMER_ID NUMBER(10) NOT NULL,

CUSTOMER_NAME VARCHAR(50) NOT NULL,

CITY VARCHAR2(50),

GRADE VARCHAR2(50),

SALESMAN_ID NUMBER(10)

);

INSERT INTO CUSTOMER VALUES (3002,'Nick Rimando','New York',100,5001);
INSERT INTO CUSTOMER VALUES (3005,'Graham Zusi','California',200,5002);

INSERT INTO CUSTOMER (CUSTOMER_ID,CUSTOMER_NAME,CITY,SALESMAN_ID)VALUES (3001,'Brad Guzan','London',5005);
INSERT INTO CUSTOMER VALUES (3004,'Fabian Johns','Paris',300,5006);
INSERT INTO CUSTOMER VALUES (3007,'Brad Davis','New York',200,5001);
INSERT INTO CUSTOMER VALUES (3009,'Geoff Camero','Berlin',100,5003);
INSERT INTO CUSTOMER VALUES (3008,'Julian Green','London',300,5002);
INSERT INTO CUSTOMER VALUES (3003,'Jozy Altidon','Moscow',200,5007);

SQL>SELECT * FROM CUSTOMER WHERE GRADE=200;

```
SQL> SELECT * FROM CUSTOMER WHERE GRADE=200;

CUSTOMER_ID CUSTOMER_NAME                                    CITY                                               GRADE
                                SALESMAN_ID
----------- ---------------------------------------------------- -------------------------------------------------- -----
----------------------------------------- -----------
       3005 Graham Zusi                                        California                                         200
                                    5002
       3007 Brad Davis                                         New York                                           200
                                    5001
       3003 Jozy Altidon                                       Moscow                                             200
                                    5007
```

CREATION

```
 CREATE TABLE MANUFACTURER(
  2  PRO_ID NUMBER(4),
  3  PRO_NAME VARCHAR2(15),
  4  PRO_PRICE NUMBER(4),
  5  PRO_COM NUMBER(3));
```

INSERTION

```
BEGIN
INSERT INTO MANUFACTURER VALUES(101,'Mother Board',3200,15);
INSERT INTO MANUFACTURER VALUES(102,'Key Board',450,16);
INSERT INTO MANUFACTURER VALUES(103,'Zip drive',250,14);
INSERT INTO MANUFACTURER VALUES(104,'Speaker',550,16);
INSERT INTO MANUFACTURER VALUES(105,'Monitor',5000,11);
INSERT INTO MANUFACTURER VALUES(106,'DVD drive',900,12);
INSERT INTO MANUFACTURER VALUES(107,'CD drive',800,12);
INSERT INTO MANUFACTURER VALUES(108,'Printer',2600,13);
INSERT INTO MANUFACTURER VALUES(109,'Refill cartridge',350,13);
END;
```

8. Write a SLQ query to calculate the average price of all products of the manufacturer which code is 16.

SQL> SELECT AVG(PRO_PRICE) FROM MANUFACTURER WHERE PRO_COM=16;

```
SQL> SELECT * FROM MANUFACTURER;

    PRO_ID PRO_NAME                       PRO_PRICE    PRO_COM
---------- ------------------------------ ---------- ----------
       101 Mother Board                         3200         15
       102 Key Board                             450         16
       103 Zip drive                             250         14
       104 Speaker                               550         16
       105 Monitor                              5000         11
       106 DVD drive                             900         12
       107 CD drive                              800         12
       108 Printer                              2600         13
       109 Refill cartridge                      350         13

9 rows selected.

SQL> SELECT AVG(PRO_PRICE) FROM MANUFACTURER WHERE PRO_COM=16;

AVG(PRO_PRICE)
--------------
           500
```

9. Write a SQL query to find the name and price of the cheapest item from above table.

SELECT PRO_NAME,PRO_PRICE FROM MANUFACTURER WHERE PRO_PRICE=(SELECT MIN(PRO_PRICE) FROM MANUFACTURER);

```
SQL> SELECT PRO_NAME,PRO_PRICE FROM MANUFACTURER WHERE PRO_PRICE=(SELECT MIN(PRO_PRICE) FROM MANUFACTURER);

PRO_NAME                       PRO_PRICE
------------------------------ ----------
Zip drive                             250

SQL>
```

10. Write a query in SQL to find the last name of all employees, without duplicates.

```
CREATE TABLE EMPLOYEES
(
EMP_IDNO NUMBER(10),
EMP_FNAME VARCHAR2(30),
EMP_LNAME VARCHAR2(30),
EMP_DEPT NUMBER(5)
);
```

INSERTION INTO EMPLOYEES TABLE

BEGIN

```
INSERT INTO EMPLOYEES VALUES(843795,'Enric','Dasio',57);
INSERT INTO EMPLOYEES VALUES(328717,'Jhon','Snares',63);
INSERT INTO EMPLOYEES VALUES(444527,'Joseph','Dosni',47);
INSERT INTO EMPLOYEES VALUES(659831,'Zanifer','Emily',47);
INSERT INTO EMPLOYEES VALUES(847674,'Kuleswar','Sitaraman',57);
INSERT INTO EMPLOYEES VALUES(748681,'Hanrey','Gabriel',47);
INSERT INTO EMPLOYEES VALUES(555935,'Alex','Manuel',57);
INSERT INTO EMPLOYEES VALUES(539569,'George','Mardy',27);
INSERT INTO EMPLOYEES VALUES(733843,'Mario','Saule',63);
INSERT INTO EMPLOYEES VALUES(631548,'Alan','Snappy',27);
INSERT INTO EMPLOYEES VALUES(839139,'Maria','Foster',57);
END;
```

SQL>SELECT DISTINCT(EMP_LNAME) FROM EMPLOYEES;

```
SQL> SELECT DISTINCT(EMP_LNAME) FROM EMPLOYEES;

EMP_LNAME
------------------------------
Saule
Robbin
Dasio
Snappy
Emily
Manuel
Gabriel
Mardy
Dosni
Sitaraman
Foster
Snares

12 rows selected.
```

11. Write a query to display all customers with a grade above 100

SQL>SELECT * FROM CUSTOMER WHERE GRADE>100;

```
SQL> SELECT * FROM CUSTOMER WHERE GRADE>100;

CUSTOMER_ID CUSTOMER_NAME                                   CITY                                               GRADE
                                SALESMAN_ID
----------- ---------------------------------------------- -------------------------------------------------- -----
-------------------------------------------- -----------
       3005 Graham Zusi                                    California                                         200
                                       5002
       3004 Fabian Johns                                   Paris                                              300
                                       5006
       3007 Brad Davis                                     New York                                           200
                                       5001
       3008 Julian Green                                   London                                             300
                                       5002
       3003 Jozy Altidon                                   Moscow                                             200
                                       5007
```

12.Write a SQL statement to display all customers, who are either belongs to the city New York or had a grade above 100 from above table.

SQL>SELECT * FROM CUSTOMER WHERE CITY ='New York' OR GRADE>100;

```
SQL> SELECT * FROM CUSTOMER WHERE CITY ='New York' OR GRADE>100;

CUSTOMER_ID CUSTOMER_NAME                                   CITY                                               GRADE
                                SALESMAN_ID
----------- ---------------------------------------------- -------------------------------------------------- -----
-------------------------------------------- -----------
       3002 Nick Rimando                                   New York                                           100
                                       5001
       3005 Graham Zusi                                    California                                         200
                                       5002
       3004 Fabian Johns                                   Paris                                              300
                                       5006
       3007 Brad Davis                                     New York                                           200
                                       5001
       3008 Julian Green                                   London                                             300
                                       5002
       3003 Jozy Altidon                                   Moscow                                             200
                                       5007

6 rows selected.
```

13. Write a SQL query to display those customers who are neither belongs to the city New York nor grade values is more than 100 from above table.

SQL>SELECT * FROM CUSTOMER WHERE CITY !='New York' AND GRADE<100;

```
SQL> SELECT * FROM CUSTOMER WHERE CITY !='New York' AND GRADE<100;

no rows selected
```

CREATE TABLE ORDERS(

ORD_NO NUMBER,

PURCH_AMT NUMBER(10,2),

ORD_DATE DATE,

Customer_id NUMBER,

salesman_id NUMBER);

```
SQL> CREATE TABLE ORDERS(ORD_NO NUMBER,PURCH_AMT NUMBER(10,2),ORD_DATE DATE,Customer_id NUMBER,salesman_id NUMBER);

Table created.
```

INSERT INTO ORDERS VALUES(70001,150.5,TO_DATE('2012-10-05', 'YYYY-MM-DD'),3005,5002);

INSERT INTO ORDERS VALUES(70009,270.65,TO_DATE('2012-09-10', 'YYYY-MM-DD'),3001,5005);

INSERT INTO ORDERS VALUES(70002,65.26,TO_DATE('2012-10-05', 'YYYY-MM-DD'),3002,5001);

INSERT INTO ORDERS VALUES(70004,110.5,TO_DATE('2012-08-17', 'YYYY-MM-DD'),3009,5003);

INSERT INTO ORDERS VALUES(70007,948.5,TO_DATE('2012-09-10', 'YYYY-MM-DD'),3005,5002);

INSERT INTO ORDERS VALUES(70005,2400.6,TO_DATE('2012-07-27', 'YYYY-MM-DD'),3007,5001);

INSERT INTO ORDERS VALUES(70008,5760,TO_DATE('2012-09-10', 'YYYY-MM-DD'),3002,5001);

INSERT INTO ORDERS VALUES(70010,1983.43,TO_DATE('2012-10-10', 'YYYY-MM-DD'),3004,5006);

INSERT INTO ORDERS VALUES(70003,2480.4,TO_DATE('2012-10-10', 'YYYY-MM-DD'),3009,5003);

INSERT INTO ORDERS VALUES(70012,250.45,TO_DATE('2012-06-27', 'YYYY-MM-DD'),3008,5002);

INSERT INTO ORDERS VALUES(70011,75.29,TO_DATE('2012-08-17', 'YYYY-MM-DD'),3003,5007);

INSERT INTO ORDERS VALUES(70013,3045.6,TO_DATE('2012-04-25', 'YYYY-MM-DD'),3002,5001);

14. 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 505 and below or those orders which purchase amount is 1000.00 and below.

SQL>SELECT * FROM ORDERS WHERE (ORD_DATE NOT IN TO_DATE('2012-09-10','YYYY-MM-DD') AND SALESMAN_ID <=5005) OR PURCH_AMT<=1000;

```
SQL> SELECT * FROM ORDERS WHERE (ORD_DATE NOT IN TO_DATE('2012-09-10','YYYY-MM-DD') AND SALESMAN_ID <=5005) OR PURCH_AMT<=1000;

    ORD_NO  PURCH_AMT ORD_DATE  CUSTOMER_ID SALESMAN_ID
---------- ---------- --------- ----------- -----------
     70001      150.5 05-OCT-12        3005        5002
     70009     270.65 10-SEP-12        3001        5005
     70002      65.26 05-OCT-12        3002        5001
     70004      110.5 17-AUG-12        3009        5003
     70007      948.5 10-SEP-12        3005        5002
     70005     2400.6 27-JUL-12        3007        5001
     70003     2480.4 10-OCT-12        3009        5003
     70012     250.45 27-JUN-12        3008        5002
     70011      75.29 17-AUG-12        3003        5007
     70013     3045.6 25-APR-12        3002        5001

10 rows selected.
```

15. Write a SQL statement for above table where i) order dates are anything but not 2012-08-17 or customer_id is not greater than 3005. Ii) and purchase amount is not below 1000.

SELECT * FROM ORDERS WHERE (ORD_DATE NOT IN TO_DATE('2012-08-17','YYYY-MM-DD') OR Customer_id <= 3005) AND purch_amt>=1000 ;

```
SQL> SELECT * FROM ORDERS WHERE (ORD_DATE NOT IN TO_DATE('2012-08-17','YYYY-MM-DD') OR Customer_id <= 3005) AND purch_amt>=1000 ;

    ORD_NO  PURCH_AMT ORD_DATE  CUSTOMER_ID SALESMAN_ID
---------- ---------- --------- ----------- -----------
     70005     2400.6 27-JUL-12        3007        5001
     70008       5760 10-SEP-12        3002        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

SQL>
```