1. Write a SQL statement to Create below table.

**Salesman\_id Name City Commision**

5001 James Hoog New York 0.15

5002 Nail Knite Paris 0.13

5005 Pit Alex London0.11

5006 Mc Lyon Paris 0.14

5003 Lauson Hen Sydney 0.12

5007 Paul Adam Rome 0.13

Query:

create table SALES(

Salesman\_id int,

Name varchar2(200),

City varchar2(200),

Comission float

);

insert into Sales(Salesman\_id,Name,City,Comission) values('5007','Paul Adam','Rome','0.13');

1. Write a SQL statement to display specific columns like Salesman\_id and Name from above table.

select SALESMAN\_ID,NAME from Sales;

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

select \* from Sales order by SALESMAN\_ID ASC;

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

select NAME, CITY from Sales WHERE CITY = 'Paris';

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

select \* from Sales WHERE CITY IN ('Paris','Rome');

1. 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.

select \* from Sales WHERE NAME LIKE 'A%' AND NAME LIKE 'K%';

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

**Customer\_id Cust\_name City Grade Salesman\_id**

3002 Nick Rimando New York 100 5001

3005 Graham Zusi California 200 5002

3001 Brad Guzan London 5005

3004 Fabian Johns Paris 300 5006

3007 Brad Davis New York 200 5001

3009 Geoff Camero Berlin 100 5003

3008 Julian Green London 300 5002

3003 Jozy Altidon Moscow 200 5007

Select \* from Customer where Grade ='200';

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

**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

Select AVG(Pro\_Price) AS Avg\_Price from PRODUCT where Pro\_com = '16';

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

Select Pro\_name,MIN( Pro\_Price) AS Min\_Price from PRODUCT GROUP BY Pro\_Name;

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

**Emp\_IDNO EMP\_Fname EMP\_Lname EMP\_Dept**

127323 Michale Robbin 57

526689 Carlos Snares 63

843795 Enric Dasio 57

328717 Jhon Snares 63

444527 Joseph Dosni 47

659831 Zanifer Emily 47

847674 Kuleswar Sitaraman 57

748681 Hanrey Gabriel 47

555935 Alex Manuel 57

539569 George Mardy 27

733843 Mario Saule 63

631548 Alan Snappy 27

839139 Maria Foster 57

Select distinct EMP\_Lname from EMPLOYEES;

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

**Customer\_id Cust\_name City Grade Salesman\_id**

3002 Nick Rimando New York 100 5001

3005 Graham Zusi California 200 5002

3001 Brad Guzan London 5005

3004 Fabian Johns Paris 300 5006

3007 Brad Davis New York 200 5001

3009 Geoff Camero Berlin 100 5003

3008 Julian Green London 300 5002

3003 Jozy Altidon Moscow 200 5007

Select \* from Customers where Grade > 100;

1. 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.

Select \* from Customers where City = 'New York' OR Grade > '100' ;

1. 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.

SELECT \* FROM customer WHERE NOT (city = 'New York' OR grade>100);

1. 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.

**ord\_no purch\_amt ord\_date custmoer\_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

SELECT \* FROM orders WHERE NOT ((ord\_date ='2012-09-10'AND salesman\_id > 505) OR purch\_amt > 1000.00);

1. 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 NOT((ord\_date ='2012-08-17' OR customer\_id>3005) AND purch\_amt<1000);