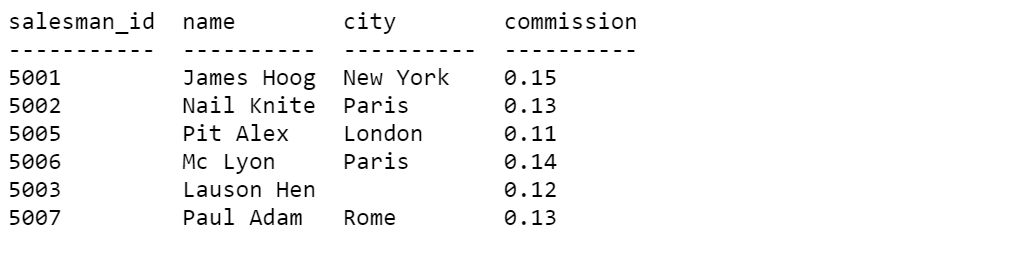
**Set 6 - Assignments on Select Statements**

**Instructions:** The tables mentioned in the questions from **question number 1 to 20** are **not** **present**. So, you should only write the select statement for those questions. Consider the table name, column names and of course data shown in each table for building the logic of select statements up to question number 20.

1. Write a SQL statement to find those salesmen with all information whose name containing the 1st character is 'N' and the 4th character is 'l' and rests may be any character.

**Sample table : Salesman**



**select \* from Salesman**

**having name like 'N\_\_\_I%'**

1. Write a SQL statement to find those rows from the table **testtable** which contain the escape character underscore ( \_ ) in its column 'col1'.

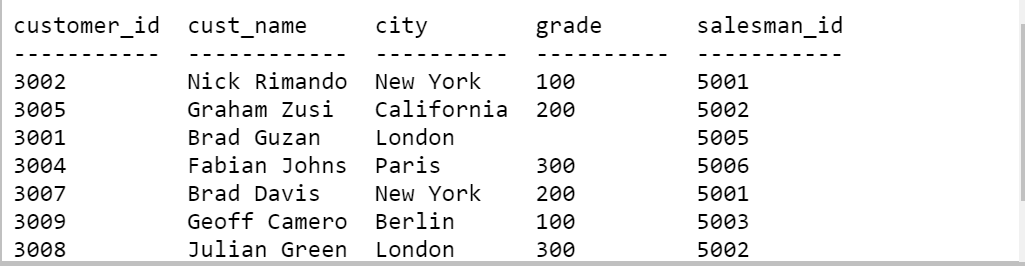


**select \* from testtable**

**where col1 like '%/\_%' Escape ‘/’**

1. Write a SQL statement find the number of customers who gets at least a gradation for his/her performance.

***Sample table*: customer**



**select COUNT (All grades)**

**from customer**

1. Write a SQL statement to find the highest purchase amount on a date '2012-08-17' for each salesman with their ID.

**Sample Table: Orders:**

**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 2 012-04-25 3002 5001

**select max(purch\_amt)**

**from Orders**

**where ord\_date = '2012-08-17'**

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

**Sample table: Orders (Refer the same table of Q.4.)**

**select customer\_id, ord\_date, purch\_amt**

**from Orders**

**where purch\_amt = (select max(purch\_amt)**

**from Orders**

**where purch\_amt between 2000 and 6000)**

1. Write a SQL statement that count the number of salesmen for whom a city is specified. Note that there may be spaces or no spaces in the city column if no city is specified.

**Sample table: salesman**

**salesman\_id name city commission**

----------- ---------- ---------- -----------------------------------------------------

5001 James Hoog New York 0.15

5002 Nail Knite Paris 0.13

5005 Pit Alex London 0.11

5006 Mc Lyon Paris 0.14

5003 Lauson Hen 0.12

5007 Paul Adam Rome 0.13

**select count(\*)**

**from salesman**

**where city is not Null**

1. Write a query to display all the orders from the orders table issued by the salesman 'Paul Adam'.

***Sample table: Salesman***

salesman\_id name city commission

----------- ---------- ---------- ----------

5001 James Hoog New York 0.15

5002 Nail Knite Paris 0.13

5005 Pit Alex London 0.11

5006 Mc Lyon Paris 0.14

5003 Lauson Hen San Jose 0.12

5007 Paul Adam Rome 0.13

***Sample table:* Orders**

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

**select o.\***

**from Orders O Join Salesman S**

**on O.Salesman\_id = S.Salesman\_id**

**where S.name = 'Paul Adam'**

1. Write a query to display all the orders which values are greater than the average order value for 10th October 2012.

***Sample table:* Salesman**

salesman\_id name city commission

----------- ---------- ---------- ----------

5001 James Hoog New York 0.15

5002 Nail Knite Paris 0.13

5005 Pit Alex London 0.11

5006 Mc Lyon Paris 0.14

5003 Lauson Hen San Jose 0.12

5007 Paul Adam Rome 0.13

***Sample table:* Orders**

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

**select \***

**from Orders**

**where purch\_amt > (select avg(purch\_amt)**

**from orders**

**where ord\_date = '2012-10-10')**

1. Write a query in SQL to find the first name and last name of employees working for departments which sanction amount is second lowest.

***Sample table:* emp\_department**

DPT\_CODE DPT\_NAME DPT\_ALLOTMENT

-------- --------------- -------------

57 IT 65000

63 Finance 15000

47 HR 240000

27 RD 55000

89 QC 75000

***Sample table:* emp\_details**

EMP\_IDNO EMP\_FNAME EMP\_LNAME EMP\_DEPT

--------- --------------- --------------- ----------

127323 Michale Robbin 57

526689 Carlos Snares 63

843795 Enric Dosio 57

328717 Jhon Snares 63

444527 Joseph Dosni 47

659831 Zanifer Emily 47

847674 Kuleswar Sitaraman 57

748681 Henrey Gabriel 47

555935 Alex Manuel 57

539569 George Mardy 27

733843 Mario Saule 63

631548 Alan Snappy 27

839139 Maria Foster 57

**SELECT emp\_fname, emp\_lname**

**FROM emp\_details**

**WHERE emp\_dept IN (SELECT dpt\_code**

**FROM emp\_department**

**WHERE dpt\_allotment = (select top 1 dpt\_allotment from emp**

**where dpt\_allotment in (select top 2 dpt\_allotment**

**from emp\_department**

**order by dpt\_allotment)**

**order by dpt\_allotment desc**

1. Write a query to display all salesmen and customer located in London.

***Sample table:* Salesman**

salesman\_id name city commission

----------- ---------- ---------- ----------

5001 James Hoog New York 0.15

5002 Nail Knite Paris 0.13

5005 Pit Alex London 0.11

5006 Mc Lyon Paris 0.14

5003 Lauson Hen 0.12

5007 Paul Adam Rome 0.13

***Sample table:* Customer**

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 Altidor Moscow 200 5007

**select name from Salesman**

**where city = 'London'**

**union**

**select cust\_name from Customer**

**where city = 'London'**

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

***Sample table:* Customer**

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 Altidor Moscow 200 5007

**select cust\_name, city, grade, 'High Rating'**

**from customer**

**where grade > 200**

**union**

**select cust\_name, city, grade, 'Low Rating'**

**from customer**

**where grade < 200**

1. Write a query to create a view to getting a count of how many customers we have at each level of a grade.

***Sample table:* customer**

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 Altidor Moscow 200 5007

**create view GradeLevel as**

**select grade, count(grades)**

**from Customer**

**group by grades**

1. Write a query to create a view to keeping track the number of customers ordering, number of salesmen attached, average amount of orders and the total amount of orders in a day.

***Sample table :* orders**

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

**create view Track as**

**select Distinct count(ord\_no), count(salseman\_id), avg(purch\_amt), sum(purch\_amt)**

**from orders**

1. Write a query to create a view that shows for each order the salesman and customer by name.

***Sample table:* salesman**

salesman\_id name city commission

----------- ---------- ---------- ----------

5001 James Hoog New York 0.15

5002 Nail Knite Paris 0.13

5005 Pit Alex London 0.11

5006 Mc Lyon Paris 0.14

5003 Lauson Hen 0.12

5007 Paul Adam Rome 0.13

***Sample table:* customer**

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 Altidor Moscow 200 5007

***Sample table:* orders**

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

**create view Sale\_Cust as**

**select O.ord\_no, S.name, C.cust\_name**

**from orders O inner join salsman S**

**on o.saleman\_id = S.saleman\_id**

**inner join customer C**

**on O.customer\_id = C.customer\_id**

1. Write a query to create a view that finds the salesman who has the customer with the highest order of a day.

***Sample table:* salesman**

salesman\_id name city commission

----------- ---------- ---------- ----------

5001 James Hoog New York 0.15

5002 Nail Knite Paris 0.13

5005 Pit Alex London 0.11

5006 Mc Lyon Paris 0.14

5003 Lauson Hen 0.12

5007 Paul Adam Rome 0.13

***Sample table:* orders**

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

**create view MaxSale**

**as**

**select S.name**

**from orders O join salesman S**

**on s.customer\_id = o.customer\_id**

**where O.purch\_amt = (select max(purch\_amt)**

**from orders)**

1. Write a query to create a view that shows all of the customers who have the highest grade.

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 Altidor Moscow 200 5007

**create view MaxGrade**

**as**

**select \***

**from customer**

**where grade = (select max(grade)**

**from customer)**

1. Write a SQL statement to find out the number of orders booked for each day and display it in such a format like "For 2001-10-10 there are 15 orders".

**Sample table: orders**

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

**select 'For '+cast(ord\_date as varchar)+' there are '+cast(count(ord\_no) as varchar)+' orders'**

**from orders**

**group by ord\_date**

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.

**Sample table : orders**

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

**select \***

**from orders**

**where (date <> '2012-09-10'**

**and salesman\_id = 5005)**

**or purch\_amt < 1000**

1. Write a SQL query to display order number, purchase amount, achieved, the unachieved percentage for those order which exceeds the 50% of the target value of 6000.

**Sample table: orders**

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

**select ord\_no, purch\_amt,cast((purch\_amt/60) as varchar)+'%' as "Achieved", cast((100-(purch\_amt/60)) as varchar)+'%'as "Unachieved"**

**from orders**

**where purch\_amt > purch\_amt/0.5**

1. Write a query to display the commission of all the salesmen servicing customers in Paris.

**Sample table: Salesman**

salesman\_id name city commission

----------- ---------- ---------- ----------

5001 James Hoog New York 0.15

5002 Nail Knite Paris 0.13

5005 Pit Alex London 0.11

5006 Mc Lyon Paris 0.14

5003 Lauson Hen San Jose 0.12

5007 Paul Adam Rome 0.13

***Sample table : Customer***

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 Altidor Moscow 200 5007

**select s.comm**

**from Salesman S Join Customer C**

**on S.salesman\_id = C.salesman\_id**

**where C.city = 'Paris'**

1. Copy-Paste the following SQL script to create the **7** tables and populate records. The tables getting created will be **Countries, Locations, Regions, Departments, Employees, Jobs** and **Job\_History**.

Create a new Database with the name as **Set6**. In this database create the tables.

create table countries

(country\_id char(2) not null,

country\_name varchar(40),

regiod\_id Integer);

Insert into countries

Values('CA','Canada',2);

Insert into countries

Values('DE','Germany',2);

Insert into countries

Values('UK','United Kingdom',2);

Insert into countries

Values('US','Unites States of America',2);

Create table locations

(location\_id Integer not null,

street\_address varchar(40),

postal\_code varchar(12),

city varchar(30) not null,

state\_province varchar(25),

country\_id char(2));

Insert into locations

Values(1400,'2014 Jabberwocky Rd', '26192', 'Southlake', 'Texas','US');

Insert into locations

Values(1500,'2011 Interiors Blvd', '99236', 'South San Francisco', 'California','US');

Insert into locations

Values(1700,'2004 Charade Rd', '98199', 'Seattle', 'Washington','US');

Insert into locations

Values(1800,'2460 Bloor St. W.', 'ON M5S 1XB', 'Toronto', 'Ontario','CA');

Insert into locations

Values(2500,'Magdalen Centre, The Oxford Science Park', 'OX9 9ZB', 'Oxford', 'Oxford','UK');

Create table regions

(region\_id Integer not null,

region\_name varchar(25));

Insert into regions

Values(1,'Europe' );

Insert into regions

Values(2,'Americas' );

Insert into regions

Values(3,'Asia' );

Insert into regions

Values(4,'Middle East and Africa');

create table departments

(department\_id Integer not null,

department\_name varchar(30) not null,

manager\_id Integer,

location\_id Integer);

Insert into departments

Values(10,'Administration', 200, 1700);

Insert into departments

Values(20,'Marketing', 201, 1800);

Insert into departments

Values(50,'Shipping', 124, 1500);

Insert into departments

Values(60,'IT', 103, 1400);

Insert into departments

Values(80,'Sales', 149, 2500);

Insert into departments

Values(90,'Executive', 100, 1700);

Insert into departments

Values(110,'Accounting', 205, 1700);

Insert into departments

Values(190,'Contracting', null, 1700);

create table employees

(employee\_id Integer,

first\_name varchar(20),

last\_name varchar(25) not null,

email varchar(25) not null,

phone\_number varchar(20),

hire\_date date not null,

job\_id varchar(10) not null,

salary Numeric,

commission\_pct numeric,

manager\_id Integer,

deparment\_id Integer);

Insert into employees

Values(100, 'Steven', 'King', 'SKING','515.123.4567','17-Jun-87', 'AD\_PRES', 24000, NULL, NULL, 90);

Insert into employees

Values(101, 'Neena', 'Kochar', 'NKOCHAR','515.123.4568','21-Sep-89', 'AD\_PRES', 17000, NULL, 100, 90);

Insert into employees

Values(102, 'Lex', 'De Haan', 'NDEHAAN','515.123.4569','13-Jan-93', 'AD\_VP', 17000, NULL, 100, 90);

Insert into employees

Values(103, 'Alexander', 'Hunold', 'AHUNOLD','590.423.4567','03-Jan-90', 'IT\_PROG', 9000, NULL, 102, 60);

Insert into employees

Values(104, 'Bruce', 'Ernst', 'BERNST','590.423.4568','21-May-91', 'IT\_PROG', 6000, NULL, 103, 60);

Insert into employees

Values(107, 'Diana', 'Lorentz', 'DLORENTZ','590.423.4567','07-Feb-99', 'IT\_PROG', 4200, NULL, 103, 60);

Insert into employees

Values(124, 'Kevin', 'Mourgos', 'KMOURGOS','650.123.5234','16-Nov-99', 'ST\_MAN', 5800, NULL, 100, 50);

Insert into employees

Values(141, 'Trenna', 'Rajs', 'TRAJS','650.121.8009','17-Oct-95', 'ST\_CLERK', 3500, NULL, 124, 50);

Insert into employees

Values(142, 'Curtis', 'Davies', 'CDAVIES','650.121.2994','29-Jan-97', 'ST\_CLERK', 3100, NULL, 124, 50);

Insert into employees

Values(143, 'Randall', 'Matos', 'RMATOS','650.121.2874','15-Mar-98', 'ST\_CLERK', 2600, NULL, 124, 50);

Insert into employees

Values(144, 'Peter', 'Vargas', 'PVARGAS','650.121.2004','09-July-98', 'ST\_CLERK', 2500, NULL, 124, 50);

Insert into employees

Values(149, 'Eleni', 'ZLotkey', 'ZLOTKEY','011.44.1344.429018','29-Jan-00', 'SA\_MAN', 10500, .2, 100, 80);

Insert into employees

Values(174,'Ellen', 'Abel', 'EABEL', '011.44.1644.429267','11-May-96', 'SA\_REP', 11000, .3, 149, 80);

Insert into employees

Values(176, 'Jonathon', 'Taylor', 'JTAYLOR','011.44.1644.529265','24-MAR-98', 'SA\_REP', 8600, .2, 149, 80);

Insert into employees

Values(178, 'Kimberly', 'Grant', 'KGRANT','011.44.1644.529263','24-MAY-99', 'SA\_REP', 7000, .15, 149, NULL);

Insert into employees

Values(200, 'Jennifer', 'Whalen', 'JWHALEN','515.123.4444','17-Sep-87', 'AD\_ASST', 4400, null, 101, 10);

Insert into employees

Values(201, 'Michael', 'Hartstein', 'MHARTSTE','515.123.5555','17-FEb-96', 'MK\_MAN', 13000, null, 100, 20);

Insert into employees

Values(202, 'Pat', 'Fay', 'PFAY','603.123.6666','17-Aug-97', 'MK\_REP', 6000, null, 201, 20);

Insert into employees

Values(205, 'Shelley', 'Higgins', 'SHIGGINS','515.123.8080','07-Jun-94', 'AC\_MGR', 12000, null, 101, 110);

Insert into employees

Values(206, 'William', 'Gietz', 'WGIETZ','515.123.8181','07-Jun-94', 'AC-ACCOUNT', 8300, null, 205, 110);

Create table jobs

(job\_id varchar(10) not null,

job\_title varchar(35) not null,

min\_salary Integer,

max\_salary Integer);

Insert into jobs

Values('AD\_PRES', 'President', 20000, 40000);

Insert into jobs

Values('AD\_VP', 'Administration Vice President', 15000, 30000);

Insert into jobs

Values('AD\_ASST', 'Administration Assistant', 3000, 6000);

Insert into jobs

Values('AC\_MGR', 'Accounting Manager', 8200, 16000);

Insert into jobs

Values('AC\_ACCOUNT', 'Public Accountant', 4200, 9000);

Insert into jobs

Values('SA\_MAN', 'Sales Manager', 10000, 20000);

Insert into jobs

Values('SA\_REP', 'Sales Representative', 6000, 12000);

Insert into jobs

Values('ST\_MAN', 'Stock Manager', 5500, 8500);

Insert into jobs

Values('ST\_CLERK', 'Stock Clerk', 2000, 5000);

Insert into jobs

Values('IT\_PROG', 'Programmer', 4000, 10000);

Insert into jobs

Values('MK\_MAN', 'Marketing Manager', 9000, 15000);

Insert into jobs

Values('MK\_REP', 'Marketing Representative', 4000, 9000);

Create table job\_history

(employee\_id Integer not null,

start\_date date not null,

end\_date date not null,

job\_id varchar(10) not null,

department\_id Integer);

Insert into job\_history

Values(102, '13-jan-93', '24-jul-98', 'IT\_PROG', 60);

Insert into job\_history

Values(101, '21-Sep-89', '27-Oct-93', 'AC\_ACCOUNT', 110);

Insert into job\_history

Values(101, '28-oct-93', '15-mar-97', 'AC\_MGR', 110);

Insert into job\_history

Values(201, '17-Feb-96', '19-dec-99', 'MK\_REP', 20);

Insert into job\_history

Values(114, '24-Mar-98', '31-dec-99', 'ST\_CLERK', 50);

Insert into job\_history

Values(122, '01-jan-99', '31-dec-99', 'ST\_CLERK', 50);

Insert into job\_history

Values(200, '17-sep-87', '17-jun-93', 'AD\_ASST', 90);

Insert into job\_history

Values(176, '24-Mar-98', '31-dec-98', 'SA\_REP', 80);

Insert into job\_history

Values(176, '01-Jan-99', '31-Dec-99', 'SA\_MAN', 80);

Insert into job\_history

Values(200, '01-Jul-94', '31-Dec-98', 'AC\_ACCOUNT', 90);

Once the above tables get created in Set6 database with records do the following:

21.1) Create **ERD** of all the tables.

21.2) Using select statement display **Employeeid, employee name, department id, department name,**

**Start date of job, end date of job, job title, country name, region name, Difference in the maximum salary and the Employee’s salary**.

For the above question (i.e. 21.2) write two select statements showing the same output.

21.2.1) In the first select statement use the **ANSI SQL 89** syntax of joins.

**Select E.employee\_id, (E.first\_name+e.last\_name) as "Name", D.department\_id, D.department\_name,**

**JH.start\_date, JH.end\_date, J.job\_title, C.country\_name, R.region\_name,**

**(max(J.max\_salary) over()) - E.salary as "Difference"**

**from employees E, departments D, job\_history JH, jobs J, locations L, countries C, regions R**

**where E.deparment\_id = D.department\_id**

**and D.department\_id = JH.department\_id**

**and JH.job\_id = J.job\_id**

**and D.location\_id = L.location\_id**

**and L.country\_id = C.country\_id**

**and C.region\_id = R.region\_id**

21.2.2) In the second select statement use the **ANSI SQL 92** syntax of joins.

**Select E.employee\_id, (E.first\_name+e.last\_name) as "Name", D.department\_id, D.department\_name,**

**JH.start\_date, JH.end\_date,J.job\_title, C.country\_name, R.region\_name,**

**(max(J.max\_salary) over()) - E.salary as "Difference"**

**from employees E**

**join departments D on D.department\_id = E.deparment\_id**

**join job\_history JH on D.department\_id = JH.department\_id**

**join jobs J on JH.job\_id = J.job\_id**

**join locations L on D.location\_id = L.location\_id**

**join countries C on L.country\_id = C.country\_id**

**join regions R on C.region\_id = R.region\_id**

**Ensure that the output is same for both the select statements.**