**SQL Assignments - Set 4**

1. Display the employee names and department names of all the employees working in the job type Salesman as well as Analyst

**select ename,**

**deptno**

**from emp e**

**where job='salesman' or job='analyst';**

1. Display the following report.

ENAME DEPARTMENT NAME

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

SMITH SMITH works in RESEARCH

ALLEN ALLEN works in SALES

WARD WARD works in SALES

JONES JONES works in RESEARCH

MARTIN MARTIN works in SALES

BLAKE BLAKE works in SALES

CLARK CLARK works in ACCOUNTING

SCOTT SCOTT works in RESEARCH

KING KING works in ACCOUNTING

TURNER TURNER works in SALES

ADAMS ADAMS works in RESEARCH

JAMES JAMES works in SALES

FORD FORD works in RESEARCH

MILLER MILLER works in ACCOUNTING

**select ename as ENAME,**

**concat(ename, ' works in ', dname) as 'DEPARTMENT NAME'**

**from emp**

**join dept**

**on emp.deptno = dept.deptno;**

1. Display the highest salary for each job within each department as per the following report –

## **Department JOB Highest Sal**

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

ACCOUNTING CLERK 1300

ACCOUNTING MANAGER 2450

ACCOUNTING PRESIDENT 5000

RESEARCH ANALYST 3000

RESEARCH CLERK 1100

RESEARCH MANAGER 2975

SALES CLERK 950

SALES MANAGER 2850

SALES SALESMAN 1600

9 rows selected.

**select dname Department,**

**job JOB,**

**max(sal) HighestSal**

**from emp**

**join dept**

**on emp.deptno = dept.deptno**

**group by dname, job;**

4. Display employee name, department name, location and job for all employees of MANAGER job having salary above 2900.

**select ename,**

**dname,**

**job**

**from emp**

**join dept**

**on emp.deptno = dept.deptno**

**where job = 'manager' and sal > 2900;**

1. Display the name of the employees reporting to KING.

**select e1.ename**

**from emp e1**

**join emp e2**

**on e1.mgr = e2.empno**

**where e2.ename = 'king';**

1. Display the count of employees reporting to KING.

**select count(e1.ename) Count**

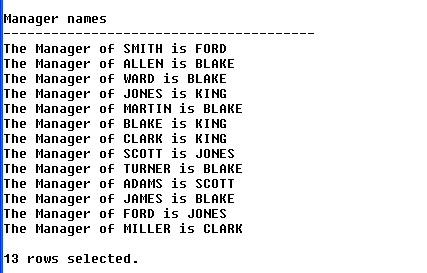
**from emp e1**

**join emp e2**

**on e1.mgr = e2.empno**

**where e2.ename = 'king';**

1. Produce the following output –



**select concat('The Manager of ', e1.ename, ' is ', e2.ename) as 'Manager Names'**

**from emp e1**

**join emp e2**

**where e1.mgr = e2.empno;**

Copy-Paste the following script about Flight Schedule in your schema –

create table customers(CustId int Primary Key, CustName varchar(40), FlightNo int Not Null);

create table flights(FlightNo int Primary key, GateNo char(3) Not Null, FlightName varchar(40), Status varchar(100), Route Varchar(100));

Insert Into Customers Values(1,'a',123);

Insert Into Customers Values(2,'b',456);

Insert Into Customers Values(3,'c',123);

Insert Into Customers Values(4, 'd',456);

Insert Into Customers Values(5, 'e', 789);

Insert Into Customers Values(6, 'f', 123);

Insert Into Customers Values(7, 'g', 456);

Insert Into Customers Values(8, 'h', 789);

Insert Into Customers Values(9, 'i', 456);

Insert Into Customers Values(10, 'j', 123);

Insert Into Customers Values(11, 'k', 125);

Insert Into Customers Values(12, 'l', 456);

Insert Into Customers Values(13, 'm', 789);

Insert Into Customers Values(14, 'n', 124);

Insert Into Customers Values(15, 'o', 125);

Insert Into Customers Values(16, 'p', 126);

Insert Into Customers Values(17, 'q', 125);

Insert Into Customers Values(18, 'r', 126);

Insert Into Customers Values(19, 's', 127);

Insert Into Customers Values(20, 't', 127);

Insert Into Flights Values('123', 'G1', 'Spice Jet', 'Departure On Time', 'Pune-Delhi');

Insert Into Flights Values('456', 'G2', 'Indigo', 'Departure Late by 45 Minutes', 'Pune-Hyderabad');

Insert Into Flights Values('789', 'G1', 'Jet Lite', 'Departed', 'Pune-Bangalore');

Insert Into Flights Values('124', 'G3', 'Indian Airlines', 'Arrival late by 30 Minutes', 'Chennai-Pune-Baroda');

Insert Into Flights Values('125', 'G4', 'King Fisher', 'Departure On Time', 'Pune-Kolkata');

Insert Into Flights Values('126', 'G7', 'King Fisher', 'Departure On Time', 'Pune-Chandigad');

Insert Into Flights Values('127', 'G9', 'Air Deccan', 'Departure On Time', 'Pune-Chandigad');

1. Display the names of the customers who are going to depart on time.

**select custname**

**-> from customers**

**-> join flights**

**-> on customers.flightno = flights.flightno**

**-> where flights.status = 'Departure On Time';**

1. Display the names of those customers who are not flying from Pune to Bangalore

**select custname**

**-> from customers**

**-> join flights**

**-> on customers.flightno = flights.flightno**

**-> where flights.route <> 'Pune-Bangalore'’;**

1. Display the names of those customers who all are going from Gate number 1.

**select custname**

**from customers**

**join flights**

**on customers.flightno = flights.flightno**

**where flights.gateno = 'g1';**

1. Display the names of those customers who are going to Chandigad through King Fisher airlines only.’’

**select custname**

**from customers**

**join flights**

**on customers.flightno = flights.flightno**

**where flightname = 'King fisher' and route regexp 'chandigad$';**

Copy-Paste the following script about Billing Cycle in your schema –

create table Customer\_Details

(CustId Char(3), CName Varchar(20));

create table Payment\_Details

(CustId Char(3), PaymentId char(3), Amount Int, Status varchar(20));

Insert Into Customer\_Details Values('C1', 'John');

Insert Into Customer\_Details Values('C2','Roger');

Insert Into Customer\_Details Values('C3','Smith');

Insert Into Customer\_Details Values('C4','Martin');

Insert Into Customer\_Details Values('C5','Luther');

Insert Into Customer\_Details Values('C6', 'King');

Insert Into Customer\_Details Values('C7','Allen');

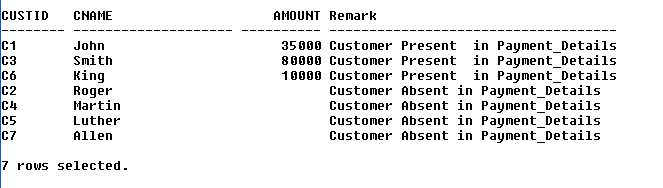
Insert Into Payment\_Details Values('C1', 'P1', 35000, 'Paid');

Insert Into Payment\_Details Values('C3','P2',80000,'Paid');

Insert Into Payment\_Details Values('C16','P3',90000,Null);

Insert Into Payment\_Details Values('C6','P4',10000,'Paid');

1. Generate the following report –



**select c.custid, c.cname, ifnull(p.amount,'') Amount,**

**concat('Customer ', case status**

**when 'paid' then 'Present'**

**else 'Absent'**

**end,**

**' in Payment\_Details') Remark**

**from customer\_details c**

**left outer join payment\_details p**

**on c.custid = p.custid**

**order by Remark desc;**

1. Display the customer id customer name, status and amount of the customers who have paid as well as those who have not paid.

**select c.custid,**

**c.cname,**

**p.status,**

**p.amount**

**from customer\_details c**

**left outer join payment\_details p**

**on c.custid = p.custid;**

Create the following 3 tables and insert the records in each table by copy pasting the below SQL script.

create table continent\_master

(ContinentID Varchar(15),

Name\_of\_Continent Varchar(20),

Continent\_Amount\_Allocated Int

);

create table country\_master

(ContinentID Varchar(15),

CountryID Varchar(15),

Name\_of\_Country Varchar(50),

Country\_Amount\_Allocated Int

);

create table transactions

(TrID Varchar(10),

CountryID Varchar(15),

Name\_of\_City Varchar(40),

Amount\_Spent Int

);

Insert into continent\_master values('Con1', 'Asia', 50000);

Insert into continent\_master values('Con2', 'Africa', 30000);

Insert into continent\_master values('Con3', 'Europe', 80000);

Insert into continent\_master values('Con4', 'Australia', 85000);

Insert into continent\_master values('Con5', 'Americas', 90000);

Insert into country\_master values('Con1','Cu1','India',10000);

Insert into country\_master values('Con1','Cu2','China',20000);

Insert into country\_master values('Con1','Cu3','UAE',15000);

Insert into country\_master values('Con2','Cu4','Sudan',15000);

Insert into country\_master values('Con2','Cu5','Kenya',15000);

Insert into country\_master values('Con3','Cu6','France',20000);

Insert into country\_master values('Con3','Cu7','Germany',20000);

Insert into country\_master values('Con3','Cu8','Italy',20000);

Insert into country\_master values('Con4','Cu9','Australia', 12000);

Insert into country\_master values('Con4','Cu10','New Zealand', 9000);

Insert into country\_master values('Con5','Cu11','USA', 50000);

Insert into country\_master values('Con5','Cu12','Canada', 35000);

Insert into country\_master values('Con5','Cu13','Brazil', 20000);

Insert into country\_master values('Con5','Cu14','Argentina', 15000);

Insert into Transactions values('T1', 'Cu1','Mumbai',200);

Insert into Transactions values('T2', 'Cu1','Mumbai',150);

Insert into Transactions values('T3', 'Cu1','Delhi',50);

Insert into Transactions values('T4', 'Cu1','Delhi',70);

Insert into Transactions values('T5','Cu2','Beijing',300);

Insert into Transactions values('T6','Cu2','Beijing',200);

Insert into Transactions values('T7','Cu2','Shanghai',700);

Insert into Transactions values('T8','Cu2','Shanghai',900);

Insert into Transactions values('T9','Cu3','Dubai',1000);

Insert into Transactions values('T10','Cu3','Dubai',1500);

Insert into Transactions values('T11', 'Cu3','Abu Dhabi',800);

Insert into Transactions values('T12', 'Cu3','Abu Dhabi',600);

Insert into Transactions values('T13','Cu4','Khartoum',80);

Insert into Transactions values('T14','Cu4','Khartoum',20);

Insert into Transactions values('T15','Cu4','Kassala',50);

Insert into Transactions values('T16','Cu4','Kassala',10);

Insert into Transactions values('T17','Cu5','Nairobi',90);

Insert into Transactions values('T18','Cu5','Nairobi',70);

Insert into Transactions values('T19','Cu5','Mombasa', 300);

Insert into Transactions values('T20','Cu5','Mombasa', 2000);

Insert into Transactions values('T21','Cu6','Paris', 900);

Insert into Transactions values('T22','Cu6','Paris', 1100);

Insert into Transactions values('T23','Cu6','Lyon', 2100);

Insert into Transactions values('T24','Cu6','Lyon', 200);

Insert into Transactions values('T25','Cu7','Berlin', 500);

Insert into Transactions values('T26','Cu7','Berlin', 700);

Insert into Transactions values('T27','Cu7','Munich', 1500);

Insert into Transactions values('T28','Cu7','Munich', 1700);

Insert into Transactions values('T29','Cu8','Rome', 3000);

Insert into Transactions values('T30','Cu8','Rome', 2000);

Insert into Transactions values('T31','Cu8','Milan', 500);

Insert into Transactions values('T32','Cu8','Milan', 800);

Insert into Transactions values('T33','Cu9','Sydney', 2800);

Insert into Transactions values('T34','Cu9','Sydney', 3100);

Insert into Transactions values('T35','Cu9','Melbourne', 1400);

Insert into Transactions values('T36','Cu9','Melbourne', 2400);

Insert into Transactions values('T37','Cu10','Auckland', 600);

Insert into Transactions values('T38','Cu10','Auckland', 800);

Insert into Transactions values('T39','Cu10','Wellington', 2100);

Insert into Transactions values('T40','Cu10','Wellington', 2700);

Insert into Transactions values('T41','Cu11','Sand City', 5000);

Insert into Transactions values('T42','Cu11','Sand City', 6000);

Insert into Transactions values('T43','Cu11','Dublin', 2000);

Insert into Transactions values('T44','Cu11','Dublin', 1000);

Insert into Transactions values('T45','Cu12','Toronto', 2200);

Insert into Transactions values('T46','Cu12','Toronto', 3300);

Insert into Transactions values('T47','Cu12','Ottawa', 2100);

Insert into Transactions values('T48','Cu12','Ottawa', 1300);

Insert into Transactions values('T49','Cu13','Sao Paulo', 900);

Insert into Transactions values('T50','Cu13','Sao Paulo', 400);

Insert into Transactions values('T51','Cu13','Rio de Janeiro', 700);

Insert into Transactions values('T52','Cu13','Rio de Janeiro', 800);

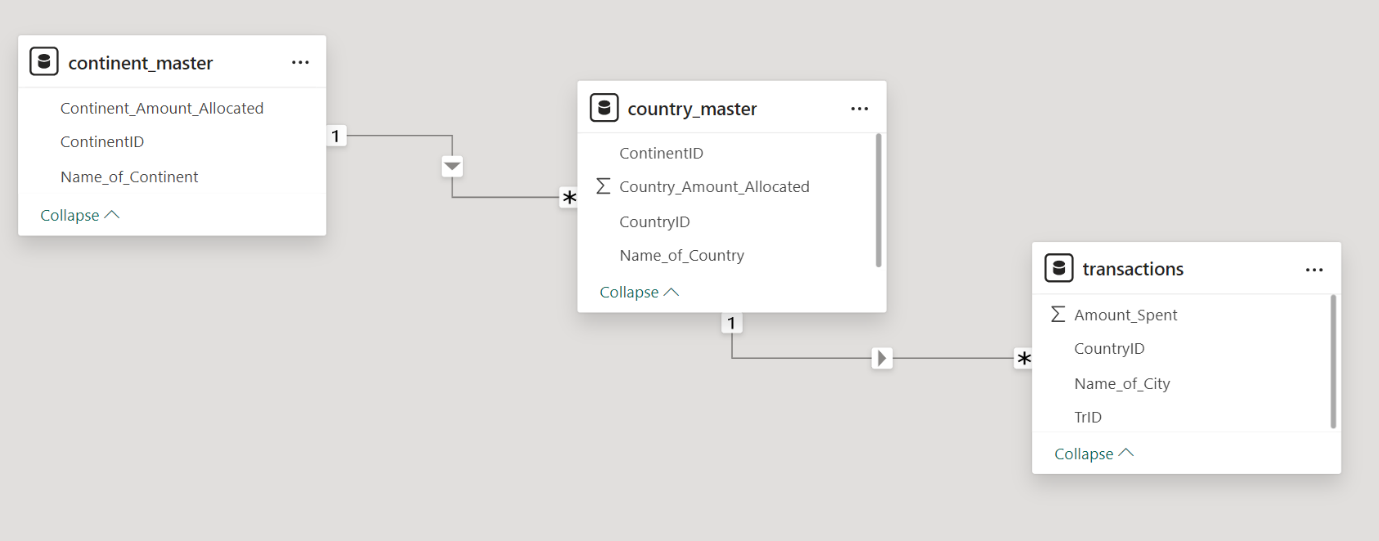
Insert into Transactions values('T53','Cu14','Buenos Aires', 200);

Insert into Transactions values('T54','Cu14','Buenos Aires', 400);

Insert into Transactions values('T55','Cu14','Rosario', 2300);

Insert into Transactions values('T56','Cu14','Rosario', 4400);

**ERD:**



1. Display transaction id, continent name, country name, city name, amount allocated and amount spent.

**select t.trid,**

**conti.name\_of\_continent,**

**contr.name\_of\_country,**

**t.name\_of\_city,**

**conti.continent\_amount\_allocated,**

**contr.country\_amount\_allocated,**

**t.amount\_spent**

**from transactions t**

**join country\_master contr**

**on t.countryid = contr.countryid**

**join continent\_master conti**

**on conti.continentid = contr.continentid;**

1. Display the total amount spent for each city, within each country from each continent.

**select continent.name\_of\_continent,**

**country.name\_of\_country,**

**t.name\_of\_city,**

**t.amount\_spent**

**from transactions t**

**join country\_master country**

**on t.countryid = country.countryid**

**join continent\_master continent**

**on continent.continentid = country.continentid;**