```
==============================================================================================
===
                                             Paper 1
==============================================================================================
====

DROP TABLE IF EXISTS `customers`;

create TABLE customers (
customer_Id int(11) NOT NULL,
First_Name varchar(50) NOT NULL,
Last_Name varchar(50) NOT NULL,
phone varchar(50) NOT NULL,
creditLimit decimal(10,2) DEFAULT NULL,
PRIMARY KEY (`customer_Id`)
);
insert into customers(customer_Id,First_Name,Last_Name,phone,creditLimit) values
(103,'Atelier','Schmitt','08-78966578','21000.00'),
(112,'Signal','King','7025551838','71800.00'),
(114,'Ferguson','Peter','03-9520 4555','117300.00'),
(119,'Labrune','Janine','40.67.8555','118200.00'),
(121,'Bergulfsen','Jonas ','07-989555','81700.00'),
(124,'Nelson','Susan','4155551450','210500.00'),
(125,'Piestrzeniewicz','Zbyszek ','(26) 642-7555','0.00'),
(128,'Keitel','Roland','+49 6966 90 2555','59700.00'),
(129,'Murphy','Julie','6505555787','64600.00'),
(131,'Lee','Kwai','2125557818','114900.00'),
(141,'Freyre','Diego ','(91) 555 94 44','227600.00'),
(144,'Berglund','Christina','0921-12 3555','53100.00'),
(145,'Petersen','Jytte ','31 12 3555','83400.00'),
(146,'Saveley','Mary','78.32.5555','123900.00'),
(148,'Eric','Jecob','+65 2217555','103800.00'),
(151,'Young','Jeff','2125557413','138500.00');
mysql> select * from customers;
+-------------+-----------------+-----------+------------------+-------------+
| customer_Id | First_Name      | Last_Name | phone            | creditLimit |
+-------------+-----------------+-----------+------------------+-------------+
|         103 | Atelier         | Schmitt   | 08-78966578      |    21000.00 |
|         112 | Signal          | King      | 7025551838       |    71800.00 |
|         114 | Ferguson        | Peter     | 03-9520 4555     |   117300.00 |
|         119 | Labrune         | Janine    | 40.67.8555       |   118200.00 |
|         121 | Bergulfsen      | Jonas     | 07-989555        |    81700.00 |
|         124 | Nelson          | Susan     | 4155551450       |   210500.00 |
|         125 | Piestrzeniewicz | Zbyszek   | (26) 642-7555    |        0.00 |
|         128 | Keitel          | Roland    | +49 6966 90 2555 |    59700.00 |
|         129 | Murphy          | Julie     | 6505555787       |    64600.00 |
|         131 | Lee             | Kwai      | 2125557818       |   114900.00 |
|         141 | Freyre          | Diego     | (91) 555 94 44   |   227600.00 |
|         144 | Berglund        | Christina | 0921-12 3555     |    53100.00 |
|         145 | Petersen        | Jytte     | 31 12 3555       |    83400.00 |
|         146 | Saveley         | Mary      | 78.32.5555       |   123900.00 |
|         148 | Eric            | Jecob     | +65 2217555      |   103800.00 |
|         151 | Young           | Jeff      | 2125557413       |   138500.00 |
+-------------+-----------------+-----------+------------------+-------------+



DROP TABLE IF EXISTS `orders`;

CREATE TABLE orders(
order_Id int(11) NOT NULL,
order_Date date NOT NULL,
shipped_Date date DEFAULT NULL,
Deliver varchar(15) NOT NULL,
customer_Id int(11) NOT NULL,
PRIMARY KEY (order_Id),
FOREIGN KEY (customer_Id) REFERENCES customers(customer_Id) on delete cascade
);
insert into orders(order_Id,order_Date,shipped_Date,Deliver,customer_Id) values
(10100,'2003-01-06','2003-01-13','Shipped',114),
(10101,'2003-01-09','2003-01-18','Shipped',125),
(10102,'2003-01-10','2003-01-18','Shipped',129),
(10103,'2003-01-29','2003-02-07','Shipped',121),
(10104,'2003-01-31','2003-02-09','Shipped',141),
(10105,'2003-02-11','2003-02-21','Shipped',145);

mysql> select * from orders;
+----------+------------+--------------+---------+-------------+
| order_Id | order_Date | shipped_Date | Deliver | customer_Id |
+----------+------------+--------------+---------+-------------+
|    10100 | 2003-01-06 | 2003-01-13   | Shipped |         114 |
|    10101 | 2003-01-09 | 2003-01-18   | Shipped |         125 |
|    10102 | 2003-01-10 | 2003-01-18   | Shipped |         129 |
|    10103 | 2003-01-29 | 2003-02-07   | Shipped |         121 |
|    10104 | 2003-01-31 | 2003-02-09   | Shipped |         141 |
|    10105 | 2003-02-11 | 2003-02-21   | Shipped |         145 |
+----------+------------+--------------+---------+-------------+



1. Write a Query to add a column package_stat to the table orders.
Answer: alter table orders Add package_stat varchar(20);
mysql> select * from orders;
+----------+------------+--------------+---------+-------------+--------------+
| order_Id | order_Date | shipped_Date | Deliver | customer_Id | package_stat |
+----------+------------+--------------+---------+-------------+--------------+
|    10100 | 2003-01-06 | 2003-01-13   | Shipped |         114 | NULL         |
|    10101 | 2003-01-09 | 2003-01-18   | Shipped |         125 | NULL         |
|    10102 | 2003-01-10 | 2003-01-18   | Shipped |         129 | NULL         |
|    10103 | 2003-01-29 | 2003-02-07   | Shipped |         121 | NULL         |
|    10104 | 2003-01-31 | 2003-02-09   | Shipped |         141 | NULL         |
|    10105 | 2003-02-11 | 2003-02-21   | Shipped |         145 | NULL         |
+----------+------------+--------------+---------+-------------+--------------+




2. Write a Query to change the package_stat column of orders table with 'not available'for
all orders.

mysql>  update orders set package_stat='not available';
mysql> select * from orders;
+----------+------------+--------------+---------+-------------+---------------+
| order_Id | order_Date | shipped_Date | Deliver | customer_Id | package_stat  |
+----------+------------+--------------+---------+-------------+---------------+
|    10100 | 2003-01-06 | 2003-01-13   | Shipped |         114 | not available |
|    10101 | 2003-01-09 | 2003-01-18   | Shipped |         125 | not available |
|    10102 | 2003-01-10 | 2003-01-18   | Shipped |         129 | not available |
|    10103 | 2003-01-29 | 2003-02-07   | Shipped |         121 | not available |
|    10104 | 2003-01-31 | 2003-02-09   | Shipped |         141 | not available |
|    10105 | 2003-02-11 | 2003-02-21   | Shipped |         145 | not available |
+----------+------------+--------------+---------+-------------+---------------+


3. Write a Query to delete a row from customers table where credit_limit is 0.00

Answer: DELETE FROM Customers WHERE creditlimit=0.00;( key problem)


===============================================================================================
===


Write SELECT statements to achieve the following:


1. Write a Query to display the first_name with the occurrence of ‘el’ in the customers
tables.
mysql> SELECT first_name FROM customers WHERE first_name LIKE '%el%';
+------------+
| first_name |
+------------+
| Atelier     |
| Nelson      |
| Keitel      |
| Saveley     |
+-------------+



2. Write a Query to prepare a list with customer name ,customer_id ,order_id for the
customers whose delivery status is shipped.

select first_name,last_name,o.customer_id from customers as c,orders as o where
o.customer_id=c.customer_id and Deliver='shipped';


+------------------+-----------+-------------+
| first_name       | last_name | customer_id |
+------------------+-----------+-------------+
| Ferguson         | Peter     |         114 |
| Piestrzeniewicz  | Zbyszek   |         125 |
| Murphy           | Julie     |         129 |
| Bergulfsen       | Jonas     |         121 |
| Freyre           | Diego     |         141 |
| Petersen         | Jytte     |         145 |
+------------------+-----------+-------------+

mysql> select first_name,last_name,orders.customer_id from customers,orders where
orders.customer_id=customers.customer_id and Deliver='shipped';
+------------------+-----------+-------------+
| first_name       | last_name | customer_id |
+------------------+-----------+-------------+
| Ferguson         | Peter     |         114 |
| Piestrzeniewicz  | Zbyszek   |         125 |
| Murphy           | Julie     |         129 |
| Bergulfsen       | Jonas     |         121 |
| Freyre           | Diego     |         141 |
| Petersen         | Jytte     |         145 |
+------------------+-----------+-------------+



3. Write a Query to get the number of customers with the creditLimit greater than 50000.
mysql> select count(creditlimit) from customers where creditlimit>50000;
+--------------------+
| count(creditlimit) |
+--------------------+
|                 14 |
+--------------------+



4. Write a Query to display the customer_id, name ( first name and last name ), order_id
and deliver for all customers.

select orders.customer_id , concat (concat(First_name, ' '), last_name) as Name ,
orders.order_id , orders.Deliver from  customers , orders
where orders.customer_Id = customers.customer_Id;
+-------------+---------------------------+----------+---------+
| customer_id | Name                      | order_id | Deliver |
+-------------+---------------------------+----------+---------+
|         114 | Ferguson Peter            |    10100 | Shipped |
|         125 | Piestrzeniewicz Zbyszek   |    10101 | Shipped |
|         129 | Murphy Julie              |    10102 | Shipped |
|         121 | Bergulfsen Jonas          |    10103 | Shipped |
|         141 | Freyre Diego              |    10104 | Shipped |
|         145 | Petersen Jytte            |    10105 | Shipped |
+-------------+---------------------------+----------+---------+



5. Write a Query to customer name in order of creditLimit smallest to highest.
```

```
mysql>  select first_name,last_name,creditLimit from customers order by creditLimit;
+-----------------+-----------+-------------+
| first_name      | last_name | creditLimit |
+-----------------+-----------+-------------+
| Piestrzeniewicz | Zbyszek   |        0.00 |
| Atelier         | Schmitt   |    21000.00 |
| Berglund        | Christina |    53100.00 |
| Keitel          | Roland    |    59700.00 |
| Murphy          | Julie     |    64600.00 |
| Signal          | King      |    71800.00 |
| Bergulfsen      | Jonas     |    81700.00 |
| Petersen        | Jytte     |    83400.00 |
| Eric            | Jecob     |   103800.00 |
| Lee             | Kwai      |   114900.00 |
| Ferguson        | Peter     |   117300.00 |
| Labrune         | Janine    |   118200.00 |
| Saveley         | Mary      |   123900.00 |
| Young           | Jeff      |   138500.00 |
| Nelson          | Susan     |   210500.00 |
| Freyre          | Diego     |   227600.00 |
+-----------------+-----------+-------------+



6. Write a stored procedure by name order_day. The procedure should show the
customer_id and the day on which he had made the order.

Answer:
drop procedure if exists order_day;
drop table if exists tempp;
Create table tempp
(
customer_id int,
order_date date
);


/* ............................................................. */
delimiter //
create procedure order_day(cust int)
begin
  declare order_d date;
  declare cid date;
     select order_date into cid from orders where customer_id=cust;

        insert into tempp values(cust ,cid);
end;//
delimiter ;
/* ............................................................. */

call order_day(114);
select * from tempp;
mysql> select * from tempp;
+-------------+------------+
| customer_id | order_date |
+-------------+------------+
|         114 | 2003-01-06 |
+-------------+------------+




7. Write a stored function by the name of cutomer_search. The stored function should
return the maximum creditLimit made by any customer.

drop function if exists customer_search;

delimiter //

create function customer_search()
returns int
```

```
deterministic
begin
    declare x int;

    select max(creditLimit) into x from customers;
    return x;
end;//
delimiter ;

select customer_search() from dual;



================================================================================================
===
                                              Paper 2
================================================================================================
====
######################################################################
1. Create DEPT table with the following structure:-
DEPTNO INT(2)
DNAME VARCHAR (14)
LOC VARCHAR (13)


Insert the following rows in DEPT table:-
10, ACCOUNTING,NEW YORK
20, RESEARCH, DALLAS
30, SALES, CHICAGO
40, OPERATIONS, BOSTON

Answer:

drop table if exists dept ;
Create table DEPT(Deptno int,DNAME varchar(25),LOC Varchar(10));
insert into DEPT values(10,'ACCOUNTING','NEW YORK');
insert into DEPT values(20,'RESEARCH','DALLAS');
insert into DEPT values(30,'SALES','CHICAGO');
insert into DEPT values(40,'OPERATIONS','BOSTON');
select * from dept;
mysql> select * from dept;
+--------+------------+----------+
| Deptno | DNAME      | LOC      |
+--------+------------+----------+
|     10 | ACCOUNTING | NEW YORK |
|     20 | RESEARCH   | DALLAS   |
|     30 | SALES      | CHICAGO  |
|     40 | OPERATIONS | BOSTON   |
+--------+------------+----------+


================================================================================================
===

2. Create EMP table with the following structure:-
EMPNO INT(4)
ENAME VARCHAR (10)
JOB VARCHAR (9)
HIREDATE DATE
SAL FLOAT(7,2)
COMM FLOAT(7,2)
DEPTNO INT(2)

Insert the following rows in EMP table:-
7369, SMITH, CLERK, 17-DEC-80, 800, null, 20
7499, ALLEN, SALESMAN, 20-FEB-81, 1600, 300, 30
7521, WARD, SALESMAN, 22-FEB-81, 1250, 500, 30
7566, JONES, MANAGER, 02-APR-81, 2975, null, 20
7654, MARTIN, SALESMAN, 28-SEP-81, 1250, 1400, null, 30
7698, BLAKE, MANAGER, 01-MAY-81, 2850, null, 30
7782, CLARK, MANAGER, 09-JUN-81, 2450, null, 10
```

```
7788, SCOTT, ANALYST, 09-DEC-82, 3000, null, 20
7839, KING, PRESIDENT, 17-NOV-81, 5000, null, 10
7844, TURNER, SALESMAN, 08-SEP-81, 1500, 0, 30
7876, ADAMS, CLERK, 12-JAN-83, 1100, null, 20
7900, JAMES, CLERK, 03-DEC-81, 950, null, 30
7902, FORD, ANALYST, 03-DEC-81, 3000, null, 20
7934, MILLER, CLERK, 23-JAN-82, 1300, null,

Answer:

drop table if exists emp;
CREATE table EMP(EMPNO int(4),ENAME varchar(10),JOB varchar(9),HIREDATE date,SAL
float(7,2),COMM float(7,2),DEPTNO int(2));
insert into EMP values(7369, 'SMITH', 'CLERK','1980-12-17', 800, null, 20);
insert into EMP values(7499,'ALLEN','SALESMAN','1981-02-20',1600, 300, 30);
insert into EMP values(7521,'WARDv','LESMAN','1981-02-22', 1250, 500, 30);
insert into EMP values(7566,'JONES', 'MANAGER', '1981-04-02', 2975, null, 20);
insert into EMP values(7654,'MARTIN','SALESMAN','1981-09-28', 1250, 1400, null);
insert into EMP values(7698, 'BLAKE','MANAGER','1981-05-01', 2850, null, 30);
insert into EMP values(7782, 'CLARK', 'MANAGER','1981-06-09', 2450, null, 10);
insert into EMP values(7788, 'SCOTT', 'ANALYST','1982-12-09', 3000, null, 20);
insert into EMP values(7839, 'KING', 'PRESIDENT', '1981-11-17', 5000, null, 10);
insert into EMP values(7844, 'TURNER', 'SALESMAN', '1981-09-08', 1500, 0, 30);
insert into EMP values(7876,'ADAMS', 'CLERK', '1983-01-12', 1100, null, 20);
insert into EMP values(7900, 'JAMES', 'CLERK', '1981-12-03', 95, null, 30);
insert into EMP values(7902, 'FORD', 'ANALYST', '1981-12-03', 3000, null, 20);
insert into EMP values(7934, 'MILLER','CLERK', '1982-01-23', 1300, null, 10);

mysql> select * from emp;
+-------+--------+-----------+------------+---------+---------+--------+
| EMPNO | ENAME  | JOB       | HIREDATE   | SAL     | COMM    | DEPTNO |
+-------+--------+-----------+------------+---------+---------+--------+
|  7369 | SMITH  | CLERK     | 1980-12-17 |  800.00 |    NULL |     20 |
|  7499 | ALLEN  | SALESMAN  | 1981-02-20 | 1600.00 |  300.00 |     30 |
|  7521 | WARDv  | LESMAN    | 1981-02-22 | 1250.00 |  500.00 |     30 |
|  7566 | JONES  | MANAGER   | 1981-04-02 | 2975.00 |    NULL |     20 |
|  7654 | MARTIN | SALESMAN  | 1981-09-28 | 1250.00 | 1400.00 |   NULL |
|  7698 | BLAKE  | MANAGER   | 1981-05-01 | 2850.00 |    NULL |     30 |
|  7782 | CLARK  | MANAGER   | 1981-06-09 | 2450.00 |    NULL |     10 |
|  7788 | SCOTT  | ANALYST   | 1982-12-09 | 3000.00 |    NULL |     20 |
|  7839 | KING   | PRESIDENT | 1981-11-17 | 5000.00 |    NULL |     10 |
|  7844 | TURNER | SALESMAN  | 1981-09-08 | 1500.00 |    0.00 |     30 |
|  7876 | ADAMS  | CLERK     | 1983-01-12 | 1100.00 |    NULL |     20 |
|  7900 | JAMES  | CLERK     | 1981-12-03 |   95.00 |    NULL |     30 |
|  7902 | FORD   | ANALYST   | 1981-12-03 | 3000.00 |    NULL |     20 |
|  7934 | MILLER | CLERK     | 1982-01-23 | 1300.00 |    NULL |     10 |
+-------+--------+-----------+------------+---------+---------+--------+
14 rows in set (0.00 sec)




==============================================================================================
===



Write SELECT statements to achieve the following:-
1. Display only the EMPNO and ENAME columns from EMP table.
   mysql>  select empno,ename from emp;
+-------+--------+
| empno | ename  |
+-------+--------+
|  7369 | SMITH  |
|  7499 | ALLEN  |
|  7521 | WARDv  |
|  7566 | JONES  |
|  7654 | MARTIN |
|  7698 | BLAKE  |
|  7782 | CLARK  |
|  7788 | SCOTT  |
|  7839 | KING   |
|  7844 | TURNER |
|  7876 | ADAMS  |
|  7900 | JAMES  |
|  7902 | FORD   |
|  7934 | MILLER |
+-------+--------+

2. Display all employees who are CLERKs and the MANAGERs.
 mysql> select * from emp where job='Clerk' or job='Manager' order by job;
+-------+--------+---------+------------+---------+------+--------+
| EMPNO | ENAME  | JOB     | HIREDATE   | SAL     | COMM | DEPTNO |
+-------+--------+---------+------------+---------+------+--------+
|  7369 | SMITH  | CLERK   | 1980-12-17 |  800.00 | NULL |     20 |
|  7876 | ADAMS  | CLERK   | 1983-01-12 | 1100.00 | NULL |     20 |
|  7900 | JAMES  | CLERK   | 1981-12-03 |   95.00 | NULL |     30 |
|  7934 | MILLER | CLERK   | 1982-01-23 | 1300.00 | NULL |     10 |
|  7566 | JONES  | MANAGER | 1981-04-02 | 2975.00 | NULL |     20 |
|  7698 | BLAKE  | MANAGER | 1981-05-01 | 2850.00 | NULL |     30 |
|  7782 | CLARK  | MANAGER | 1981-06-09 | 2450.00 | NULL |     10 |
+-------+--------+---------+------------+---------+------+--------+

3. Display the ENAME and JOB for all employees who belong to the same DEPTNO as
employee 'KING'.
mysql> select ename,job from emp where deptno=(select deptno from emp where ename='KING');
+--------+-----------+
| ename  | job       |
+--------+-----------+
| CLARK  | MANAGER   |
| KING   | PRESIDENT |
| MILLER | CLERK     |
+--------+-----------+


4. Find the names of all employees hired in the month of February (of any year).
 Select * FROM emp WHERE Substr(Hiredate,4,2)='02';
 mysql> Select * FROM emp WHERE Substr(Hiredate,6,2)='02';
+-------+-------+----------+------------+---------+--------+--------+
| EMPNO | ENAME | JOB      | HIREDATE   | SAL     | COMM   | DEPTNO |
+-------+-------+----------+------------+---------+--------+--------+
|  7499 | ALLEN | SALESMAN | 1981-02-20 | 1600.00 | 300.00 |     30 |
|  7521 | WARDv | LESMAN   | 1981-02-22 | 1250.00 | 500.00 |     30 |
+-------+-------+----------+------------+---------+--------+--------+

5. Display the employees in descending order of DEPTNO.
mysql> select * from emp as e,dept as d where e.deptno=d.deptno order by e.deptno desc;
+-------+--------+-----------+------------+---------+--------+--------+--------+------------+-
----------+
| EMPNO | ENAME  | JOB       | HIREDATE   | SAL     | COMM   | DEPTNO | Deptno | DNAME      |
| LOC      |
+-------+--------+-----------+------------+---------+--------+--------+--------+------------+-
----------+
|  7499 | ALLEN  | SALESMAN  | 1981-02-20 | 1600.00 | 300.00 |     30 |     30 | SALES      |
| CHICAGO  |
|  7521 | WARDv  | LESMAN    | 1981-02-22 | 1250.00 | 500.00 |     30 |     30 | SALES      |
| CHICAGO  |
|  7698 | BLAKE  | MANAGER   | 1981-05-01 | 2850.00 |   NULL |     30 |     30 | SALES      |
| CHICAGO  |
|  7844 | TURNER | SALESMAN  | 1981-09-08 | 1500.00 |   0.00 |     30 |     30 | SALES      |
| CHICAGO  |
|  7900 | JAMES  | CLERK     | 1981-12-03 |   95.00 |   NULL |     30 |     30 | SALES      |
| CHICAGO  |
|  7369 | SMITH  | CLERK     | 1980-12-17 |  800.00 |   NULL |     20 |     20 | RESEARCH   |
| DALLAS   |
|  7566 | JONES  | MANAGER   | 1981-04-02 | 2975.00 |   NULL |     20 |     20 | RESEARCH   |
| DALLAS   |
|  7788 | SCOTT  | ANALYST   | 1982-12-09 | 3000.00 |   NULL |     20 |     20 | RESEARCH   |
| DALLAS   |
|  7876 | ADAMS  | CLERK     | 1983-01-12 | 1100.00 |   NULL |     20 |     20 | RESEARCH   |
| DALLAS   |
|  7902 | FORD   | ANALYST   | 1981-12-03 | 3000.00 |   NULL |     20 |     20 | RESEARCH   |
| DALLAS   |
|  7782 | CLARK  | MANAGER   | 1981-06-09 | 2450.00 |   NULL |     10 |     10 | ACCOUNTING
| NEW YORK |
|  7839 | KING   | PRESIDENT | 1981-11-17 | 5000.00 |   NULL |     10 |     10 | ACCOUNTING
| NEW YORK |
|  7934 | MILLER | CLERK     | 1982-01-23 | 1300.00 |   NULL |     10 |     10 | ACCOUNTING
| NEW YORK |
+-------+--------+-----------+------------+---------+--------+--------+--------+------------+-
---------+


6. Display the employee name and employee number of the employees with the
headings as NUMBER and NAME.
mysql> select empno "Number",ename "NAME" from emp;
+--------+--------+
| Number | NAME   |
+--------+--------+
|   7369 | SMITH  |
|   7499 | ALLEN  |
|   7521 | WARDv  |
|   7566 | JONES  |
|   7654 | MARTIN |
|   7698 | BLAKE  |
|   7782 | CLARK  |
|   7788 | SCOTT  |
|   7839 | KING   |
|   7844 | TURNER |
|   7876 | ADAMS  |
|   7900 | JAMES  |
|   7902 | FORD   |
|   7934 | MILLER |
+--------+--------+




7. Find the names of all employees who were hired on the last day of the month.
mysql> select last_day(Hiredate) from emp;
+--------------------+
| last_day(Hiredate) |
+--------------------+
| 1980-12-31         |
| 1981-02-28         |
| 1981-02-28         |
| 1981-04-30         |
| 1981-09-30         |
| 1981-05-31         |
| 1981-06-30         |
| 1982-12-31         |
| 1981-11-30         |
| 1981-09-30         |
| 1983-01-31         |
| 1981-12-31         |
| 1981-12-31         |
| 1982-01-31         |
+--------------------+


8. Find the name of the employee who is receiving the maximum salary.
mysql> select ename,max(sal) from emp;
+-------+----------+
| ename | max(sal) |
+-------+----------+
| SMITH |  5000.00 |
+-------+----------+


9. Display the sum of SAL for all the employees belonging to DEPTNO 10.
mysql> select sum(sal) from emp where deptno=10;
+----------+
| sum(sal) |
+----------+
|  8750.00 |
+----------+
```

10. Display the rows where JOB column ends with the letter 'T'.

```
mysql> select * from emp where job like '%T';
+-------+--------+-----------+------------+---------+------+--------+
| EMPNO | ENAME  | JOB       | HIREDATE   | SAL     | COMM | DEPTNO |
+-------+--------+-----------+------------+---------+------+--------+
|  7788 | SCOTT  | ANALYST   | 1982-12-09 | 3000.00 | NULL |     20 |
|  7839 | KING   | PRESIDENT | 1981-11-17 | 5000.00 | NULL |     10 |
|  7902 | FORD   | ANALYST   | 1981-12-03 | 3000.00 | NULL |     20 |
+-------+--------+-----------+------------+---------+------+--------+
================================================================================================
===
```

11. Write a stored procedure to convert a temperature in Fahrenheit (F) to its equivalent in Celsius (C). The required formula is:- C= (F-32)*5/9
Insert the temperature in Centigrade into TEMPP table. Calling program for the stored procedure need not be written.

```
drop procedure if exists abc;
drop table if exists tempp;
Create table tempp
(
Fahrenheit float,
Temp_Celsius float

);


/* ............................................................. */
delimiter //
create procedure abc(Fahrenheit float)
begin
  declare Temp_Celsius float;

        set Temp_Celsius=(Fahrenheit-32)*5/9;
        insert into tempp values(Fahrenheit,Temp_Celsius);
end;//
delimiter ;
/* ............................................................. */


call abc(100);
call abc(99);
call abc(98);
select * from tempp;
mysql> select * from tempp;
+------------+--------------+
| Fahrenheit | Temp_Celsius |
+------------+--------------+
|        100 |      37.7778 |
|         99 |      37.2222 |
|         98 |      36.6667 |
+------------+--------------+


================================================================================================
===
```

12. Write a stored function by the name of Num_cube. The stored function should return the cube of a number 'N'. The number 'N' should be passed to the stored function as a parameter. Calling program for the stored function need not be written.

Answer:

```
drop function if exists Num_cube;
drop PROCEDURE if exists pqr;
drop table if exists tempp;
Create table tempp
(
Num varchar(20),
Num_cube int
);


/* ............................................................ */
delimiter //
create function Num_cube(N int)
   returns int
   deterministic
   begin
        return N*N*N;
   end; //
delimiter ;

/* ............................................................ */

select Num_cube(10) from dual;
mysql> select Num_cube(10) from dual;
+--------------+
| Num_cube(10) |
+--------------+
|         1000 |
+--------------+
===============================================================================================
===
```