Practice set

1. Write a Query to add a column package_stat to the table orders.

>>alter table orders add package_stat varchar(20);

2. Write a Query to change the package_stat column of orders table with 'not available' for all orders.
>>update orders set package_stat = 'Not Available';

3. Write a Query to delete a row from customers table where credit_limit is 0.00
>> delete from customers where creditlimit=0.00;

4. Write a Query to display the first_name with the occurrence of ‘el’ in the customers tables.
select first_name from customers where first_name like '%el%';

2. Write a Query to prepare a list with customer name ,customer_id ,order_id for the customers whose delivery status is shipped.
>>select c.first_name||' '||c.last_name as customer_name,c.customer_id,o.order_id from customers c join orders o on c.customer_id=o.customer_id where deliver='Shipped';

3. Write a Query to get the number of customers with the creditLimit greater than 50000.
>>select count(*) as number_of_customers from customers where creditlimit > 50000;

4. Write a Query to display the customer_id, name (first name and last name), order_id and deliver for all customers.
>>select c.customer_id ,c.first_name||' '||c.last_name as name,o.order_id from customers c left join orders o on c.customer_id=o.customer_id ;

5. Write a Query to customer name in order of creditLimit smallest to highest.
>> select first_name||' '||last_name as name from customers order by creditlimit;

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.
>>create table a (customer_id integer , order_date date,id);
 insert into s (id) values (1);
create or replace procedure od (q integer) language plpgsql as $$ begin update s set customer_id = q where id = 1; update s set order_date = (select order_date from orders where customer_id=q) where id = 1; end ; $$;

call od (114);
CALL
cdac2=# table s;
 customer_id | order_date | id

```
------------+------------+----
        114 | 2003-01-06 |  1
(1 row)
```

7. Write a stored function by the name of cutomer_search. The stored function should return the maximum creditLimit made by any customer.
>>create or replace function customer_search() returns integer language plpgsql as max declare max integer ; begin select max(creditlimit) into max from customers ; return max; end; max ;

Display only the EMPNO and ENAME columns from EMP table
>>select empno , ename from emp;

Display all employees who are CLERKs and the MANAGERs
>>select * from emp where job in ('CLERK','MANAGER');

Display the ENAME and JOB for all employees who belong to the same DEPTNO as employee ‘KING’
>>select ename ,job from emp where deptno in (select deptno from emp where ename= 'KING');

Find the names of all employees hired in the month of February (of any year).
>>SELECT ENAME FROM EMP WHERE TO_CHAR (HIREDATE,'MONTH') LIKE '%FEB%';

Display the employees in descending order of DEPTNO
>>SELECT * FROM EMP ORDER BY DEPTNO DESC

Display the employee name and employee number of the employees with the headings as NUMBER and NAME
>>SELECT ENAME AS NAME ,EMPNO AS NUMBER FROM EMP;

Find the name of the employee who is receiving the maximum salary.
>>SELECT ENAME FROM EMP WHERE SAL IN (SELECT MAX(SAL) FROM EMP);

Display the sum of SAL for all the employees belonging to DEPTNO 10. ;
>>SELECT SUM(SAL) FROM EMP WHERE DEPTNO=10;

Display the rows where JOB column ends with the letter ‘T’
>>SELECT * FROM EMP WHERE JOB LIKE '%T';

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.
>>CREATE OR REPLACE PROCEDURE TEMP(F NUMERIC) LANGUAGE PLPGSQL AS $$ DECLARE C NUMERIC; BEGIN DROP TABLE IF EXISTS TEMPP; CREATE TABLE TEMPP (TEMP NUMERIC); C:=(F-32)*5/9; INSERT INTO TEMPP VALUES (C); END; $$;

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

>>CREATE OR REPLACE FUNCTION NUM_CODE (N INT) RETURNS INT LANGUAGE PLPGSQL AS $$ BEGIN RETURN N*N*N; END; $$;