--------------Q1------------------

create table orders(order\_id int not null,customer\_id int not null,order\_date timestamp,item\_id int,quantity int)

select \* from orders

insert into orders values(2,1,'2020-06-08',2,10),(3,2,'2020-06-02',1,5),

(4,3,'2020-06-03',3,5),

(5,4,'2020-06-04',4,1),

(6,4,'2020-06-05',5,5),

(7,5,'2020-06-05',1,10),

(8,5,'2020-06-14',4,5),

(9,5,'2020-06-21',3,5)

create table item(item\_id int not null,item\_name varchar(100),item\_category varchar(100))

insert into item values(1,'LC Alg. Book','Book'),

(2,'LC DB. Book','Book'),

(3,'LC Smartphone','Phone'),

(4,'LC Phone 2020','Phone'),

(5,'LC SmartGlass','Glasses'),

(6,'LC T-Shirt XL','T-Shirt')

select \* from item

create temporary table t1 as (

SELECT DISTINCT item\_category,

CASE WHEN extract(dow from order\_date) = 1 THEN SUM(quantity) OVER main\_window ELSE 0

END AS Monday,

CASE WHEN extract(dow from order\_date) = 2 THEN SUM(quantity) OVER main\_window ELSE 0

END AS Tuesday,

CASE WHEN extract(dow from order\_date) = 3 THEN SUM(quantity) OVER main\_window ELSE 0

END AS Wednesday,

CASE WHEN extract(dow from order\_date) = 4 THEN SUM(quantity) OVER main\_window ELSE 0

END AS Thursday,

CASE WHEN extract(dow from order\_date) = 5 THEN SUM(quantity) OVER main\_window ELSE 0

END AS Friday,

CASE WHEN extract(dow from order\_date) = 6 THEN SUM(quantity) OVER main\_window ELSE 0

END AS Saturday,

CASE WHEN extract(dow from order\_date) = 7 THEN SUM(quantity) OVER main\_window ELSE 0

END AS Sunday

FROM orders

RIGHT JOIN item

USING (item\_id)

WINDOW main\_window AS (PARTITION BY item\_category, TO\_CHAR(order\_date, 'Day'))

)

select \* from t1

drop table t1

SELECT item\_category,

SUM(Monday) AS Monday,

SUM(Tuesday) AS Tuesday,

SUM(Wednesday) AS Wednesday,

SUM(Thursday) AS Thursday,

SUM(Friday) AS Friday,

SUM(Saturday) AS Saturday,

SUM(Sunday) AS Sunday

FROM t1

GROUP BY item\_category

ORDER BY item\_category

------------------Q2----------------------

create table employee(id int,name varchar(100),salary int, dept\_id int)

select \* from employee

insert into employee values(1,'Joe',85000,1),

(2,'Henry',80000,2),

(3,'Sam',60000,2),

(4,'Max',90000,1),

(5,'Janet',69000,1),

(6,'Randy',85000,1),

(7,'Will',70000,1)

create table department(id int, name varchar(100))

select \* from department

insert into department values(1,'IT'),

(2,'Sales')

-- department, Employee, salary---

select \* from employee

select \* from department

select a.department\_name,

a.employee\_name,

a.salary from (select distinct employee.name as employee\_name,salary,employee.dept\_id,

department.name as department\_name,

dense\_rank () over(partition by department.name ORDER BY salary DESC) as Rank

from employee inner join department

on employee.dept\_id=department.id ) As a

where a.rank <

------------Q3--------------

create table customer(

customer\_id integer,

name varchar(100),

visited\_on timestamp,

amount integer

)

insert into customer

values

(1,'John','2019-01-01',100),

(2,'Daniel','2019-01-02',110),

(3,'Jade','2019-01-03',120),

(4,'Khaled','2019-01-04',130),

(5,'Winston','2019-01-05',110),

(6,'Elvis','2019-01-06',140),

(7,'Anna','2019-01-07',150),

(8,'Maria','2019-01-08',80),

(9,'Jaze','2019-01-09',110),

(1,'John','2019-01-10',130),

(3,'Jade','2019-01-10',150)

select \* from customer

SELECT visited\_on,

SUM(amount) OVER(ORDER BY visited\_on ROWS 6 PRECEDING),

ROUND(AVG(amount) OVER(ORDER BY visited\_on ROWS 6 PRECEDING),2)

FROM (

SELECT visited\_on, SUM(amount) AS amount

FROM customer

GROUP BY visited\_on

ORDER BY visited\_on

) AS a

ORDER BY visited\_on OFFSET 6 ROWS

----------Q4---------------

create table coordinates(X int, Y int)

insert into coordinates values(-1,-1),

(0,0),

(-1,-2)

SELECT p1.x,p1.y,p2.x,p2.y,SQRT(POW((p1.x-p2.x),2)+POW((p1.y-p2.y),2)) as shortest

FROM coordinates AS p1

CROSS JOIN coordinates AS p2

WHERE p1.x!=p2.x OR p1.y!=p2.y

ORDER BY SQRT(POW((p1.x-p2.x),2)+POW((p1.y-p2.y),2))

LIMIT 1

------------- Q5---------------------

create table a5(id int,num int)

insert into a5 values(1,1),

(2,1),

(3,1),

(4,2),

(5,1),

(6,2),

(7,2)

select \* from a5

SELECT DISTINCT(num) as ConsecutiveNums from a5

where num < 2

SELECT DISTINCT a.num AS ConsecutiveNums

FROM(

SELECT num,

LAG(num) OVER() as prev,

LEAD(num) OVER() as next

FROM a5) AS a

WHERE a.num = a.prev AND a.num = a.next