Question 1:

```
create table city(
ID int,
NAME varchar(17),
COUNTRYCODE varchar(3),
DISTRICT varchar(20),
POPULATION bigint
);

insert into city values
(6,'Rotterdam','NLD','Zuid-Holland',593321),
(3878,'Scottsdale','USA','Arizona',202705),
(3965,'Corona','USA','California',124966),
(3973,'Concord','USA','California',121780),
(3977,'Cedar Rapids','USA','Iowa',120758),
(3982,'Coral Springs','USA','Florida',117549),
(4054,'Fairfield','USA','California',92256),
(4058,'Boulder','USA','Colorado',91238),
(4061,'Fall River','USA','Massachusetts',90555);

select * from city where COUNTRYCODE = 'USA' and POPULATION >  100000;
```

Question 2:

```
Select NAME from city where COUNTRYCODE = 'USA' and POPULATION >120000;
```

Question 3:

```
select * from city;
```

Question 4:

```
select * from city where ID = 1661;
```

Question 5:

```
select * from city where COUNTRYCODE = 'JPN';
```

Question 6:

```
select NAME from city where COUNTRYCODE = 'JPN';
```

Question 7:

```
SELECT CITY,STATE FROM STATION;
```

Question 8:

```
SELECT DISTINCT CITY FROM STATION WHERE MOD(ID,2)=0 ORDER BY CITY ASC;
```

Question 9:

```
SELECT
COUNT(CITY) as city_count - COUNT(DISTINCT CITY) as distinct_city_count
FROM STATION;
```

Question 10:

```
(SELECT city, LENGTH(city) FROM station ORDER BY LENGTH(city), city LIMIT 1)
UNION
(SELECT city, LENGTH(city) FROM station ORDER BY LENGTH(city) DESC, city LIMIT
1);
```

Question 11:

```
SELECT DISTINCT city
FROM station
WHERE city REGEXP '^[AEIOUaeiou]';
```

Question 12:

```
SELECT DISTINCT city
FROM station
WHERE city REGEXP '^.*[aeiouAEIOU]$';
```

Question 13:

```
SELECT DISTINCT city
FROM station
WHERE city REGEXP '^[^aeiouAEIOU]';
```

Question 14:

```
SELECT DISTINCT city
FROM station
WHERE city REGEXP '[^aeiouAEIOU]$';
```

Question 15:

```
SELECT DISTINCT city
FROM station
WHERE city REGEXP '^[^aeiouAEIOU]' OR city REGEXP '[^aeiouAEIOU]$';
```

Question 16:

```
SELECT DISTINCT city
FROM station
WHERE city REGEXP ‘^[^aeiouAEIOU].*[^aeiouAEIOU]$’;
```

Question 17:

```
create table product(product_id int primary key,
product_name varchar(30), unit_price int);

create table sales (seller_id int,
product_id int, buyer_id int, sale_date date, quantity int, price int,
Foreign key (product_id) references product(product_id));

insert into product values
(1,'S8',1000),
(2,'G4',800),
(3,'iPhone',14000);

insert into sales values
(1,1,1,'2019-01-21',2,2000),(1,2,2,'2019-02-17',1,800),
(2,2,3,'2019-06-02',1,800),(3,3,4,'2019-05-13',1,2800);
```

```
SELECT product_id,
       product_name
FROM   product
WHERE  product_id NOT IN (SELECT product_id
                          FROM   sales
                          WHERE  sale_date NOT BETWEEN
                                 '2019-01-01' AND '2019-03-31');
```

Question 18:

```
create table views
( article_id int,
author_id int,
viewer_id int,
view_date date);

insert into views values
(1,3,5,'2019-08-01'),(1,3,6,'2019-08-02'),
(2,7,7,'2019-08-01'),(2,7,6,'2019-08-02'),
(4,7,1,'2019-07-22'),(3,4,4,'2019-07-21'),
(3,4,4,'2019-07-21');
```

```
select
      author_id as id
from
      views
where
      author_id = viewer_id
group by
      author_id
order by
      author_id asc;
```

Question 19:

```
create table Delivery (delivery_id int primary key,
customer_id int,
order_date date,
customer_pref_delivery_date date);

insert into Delivery values
(1,1,'2019-08-01',' 2019-08-02'),
(2,5,'2019-08-02',' 2019-08-02'),
(3,1,'2019-08-11',' 2019-08-11'),
(4,3,'2019-08-24',' 2019-08-26'),
(5,4,'2019-08-21',' 2019-08-22'),
(6,2,'2019-08-11',' 2019-08-13');
```

```sql
select ifnull(
    round(
      (select count(*) from Delivery where order_date =
      customer_pref_delivery_date)/
        (count(delivery_id)) * 100, 2) , 0) as immediate_percentage
from Delivery;
```

Question 20:

```sql
create table ads
(ad_id int, user_id int, action enum('Clicked', 'Viewed', 'Ignored'));

insert into ads values
(1,1,'Clicked'),(2,2,'Clicked'),(3,3,'Viewed'),(5,5,'Ignored'),(1,7,'Ignored'),
(2,7,'Viewed'),(3,5,'Clicked'),(1,4,'Viewed'),(2,11,'Viewed'),(1,2,'Clicked');
```

```sql
--Approach 1

select ad_id,
    ifnull(round(sum(action ='Clicked')/sum(action !='ignored') *100,2),0) ctr
from ads group by ad_id order by ctr desc, ad_id;

--Approach 2

select distinct ad_id, ifnull(
  round(
  sum(action='Clicked')/(sum(action='Clicked')+sum(action='Viewed'))*100,2),0)
as ctr
    from ads group by ad_id order by ctr desc, ad_id;

--Approach 3

select ad_id,
    (case when clicks+views = 0 then 0 else round(clicks/(clicks+views)*100, 2)
end) as ctr
from
    (select ad_id,
        sum(case when action='Clicked' then 1 else 0 end) as clicks,
        sum(case when action='Viewed' then 1 else 0 end) as views
    from ads group by ad_id) as t order by ctr desc, ad_id asc;
```

Question 21:

```
create table Employee (employee_id int primary key, team_id int);

insert into Employee values(1,8),(2,8),(3,8),(4,7),(5,9),(6,9);
```

```
select
      e1.employee_id, count(e2.employee_id) as team_size
from
      Employee e1
inner join
      Employee e2
on
      e1.team_id = e2.team_id
group by
      e1.employee_id, e2.team_id;
```

Question 22:

```
create table countries
(country_id int primary key,
country_name varchar(30));

create table weather(
country_id int,
weather_state int,
day date,
primary key (country_id , day));

insert into countries values
(2, 'USA'),(3, 'Australia'),(7, 'Peru'),
(5, 'China'),(8, 'Morocco'),(9, 'Spain');

insert into weather values
(2, 15, '2019-11-01'),(2, 12, '2019-10-28'),
(2, 12, '2019-10-27'),(3, -2, '2019-11-10'),
(3, 0,  '2019-11-11'),(3, 3,  '2019-11-12'),
(5, 16, '2019-11-07'),(5, 18, '2019-11-09'),
(5, 21, '2019-11-23'),(7, 25, '2019-11-28'),
(7, 22, '2019-12-01'),(7, 20, '2019-12-02'),
(8, 25, '2019-11-05'),(8, 27, '2019-11-15'),
(8, 31, '2019-11-25'),(9, 7, '2019-10-23'),(9, 3, '2019-12-23');
```

```sql
select country_name, case when avg(weather_state) <= 15 then "Cold"
                          when avg(weather_state) >= 25 then "Hot"
                          else "Warm" end as weather_type
from countries inner join weather
on countries.country_id = weather.country_id
where left(day, 7) = '2019-11'
group by country_name;
```

Question 23:

```sql
create table prices
(product_id int,
start_date date,
end_date date,
price int,
primary key (product_id,start_date,end_date));

create table unitsold
(product_id int,
purchase_date date,
units int);

insert into prices values
(1, '2019-02-17', '2019-02-28', 5),
(1, '2019-03-01', '2019-03-22', 20),
(2, '2019-02-01', '2019-02-20', 15),
(2, '2019-02-21', '2019-03-31', 30);

insert into unitsold values
(1, '2019-02-25', 100),
(1, '2019-03-01', 15),
(2, '2019-02-10', 200),
(2, '2019-03-22', 30);
```

```sql
select product_id,
ifnull(round(sum(prices_sum) / sum(units), 2), 0) as average_price
    from (
        select p.product_id as product_id, units, price * units as prices_sum
            from prices p left join unitsold u
            on p.product_id = u.product_id and purchase_date between start_date
and end_date
    ) as temp
    group by product_id;
```

Question 24:

```
create table activity(
player_id int ,
device_id int,
event_date date ,
games_played int,
primary key (player_id,event_date));

insert  into activity VALUES
(1, 2, '2016-03-01', 5),
(1, 2, '2016-03-02', 6),
(2, 3, '2017-06-25', 1),
(3, 1, '2016-03-01', 0),
(3, 4, '2016-07-03', 5);
```

```
--Approach 1
SELECT
  A.player_id,
  MIN(A.event_date) AS first_login
FROM
  activity A
GROUP BY
  A.player_id;

  --Approach 2 (Using Window Function)
  SELECT
  X.player_id,
  X.event_date AS first_login
FROM
  (
    SELECT
      A.player_id,
      A.event_date,
      RANK() OVER (PARTITION BY A.player_id ORDER BY A.event_date) AS rnk
    FROM
      activity A
  ) X
WHERE
  X.rnk = 1;
```

Question 25:

```sql
create table activity(player_id int ,device_id int,
event_date date ,
games_played int,
primary key (player_id,event_date));

insert  into activity VALUES
(1, 2, '2016-03-01', 5),
(1, 2, '2016-03-02', 6),
(2, 3, '2017-06-25', 1),
(3, 1, '2016-03-01', 0),
(3, 4, '2016-07-03', 5);
```

```sql
select player_id,device_id
from
(
 select
      player_id,
      device_id,
      event_date,
      row_number() over(partition by player_id order by event_date asc)
      as rw
      from activity) a
where a.rw = 1;
```

Question 26:

```sql
create table Products (product_id int primary key,
product_namevarchar(30), product_category varchar(30));

create table orders (product_id int,order_date date,unit int);

insert into Products values
(1,'Leetcode Solutions','Book'),(2,'Jewels of Stringology','Book'),
(3,'HP','Laptop'),(4,'Lenovo','Laptop'),(5,'Leetcode Kit', 'T-shirt');

insert into orders values
(1,'2020-02-05',60),(1,'2020-02-10',70),
(2,'2020-01-18',30),(2,'2020-02-11',80),
(3,'2020-02-17',2),(3,'2020-02-24',3),
(4,'2020-03-01',20),(4,'2020-03-04',30),
(4,'2020-03-04',60),(5,'2020-02-25',50),
(5,'2020-02-27',50),(5,'2020-03-01',50);
```

```sql
select p.product_name as product_name, o.sum_unit as unit from Products p
join
(select product_id, sum(unit) as sum_unit
from orders
where order_date >= '2020-02-01' and order_date < '2020-03-01'
group by product_id) o
on p.product_id = o.product_id
where o.sum_unit >= 100;
```

Qusetion 27:

```sql
create table users
( user_id int primary key,
name varchar(30),
mail varchar(50));

insert into users values
(1, 'Winston','winston@leetcode.com'),
(2 ,'Jonathan', 'jonathanisgreat'),
(3, 'Annabelle','bella-@leetcode.com'),
(4, 'Sally','sally.come@leetcode.com'),
(5, 'Marwan','quarz#2020@leetcode.com'),
(6, 'David','david69@gmail.com'),
(7, 'Shapiro','.shapo@leetcode.com');
```

```sql
select * from users
    where mail regexp '^[a-zA-Z]+[a-zA-Z0-9_\\./\\-]{0,}@leetcode.com$'
    order by user_id;
```

Question 28:

```sql

create table customer
(customer_id int primary key,
name varchar(30),
country varchar(30));

create table product
(product_id int primary key,
description varchar(50),
price int);
```

```sql
create table orders
(order_id int primary key,
customer_id int,product_id int , order_date date,
quantity int);

insert into customer values
(1,'Winston','USA'),
(2,'Jonathan','Peru'),
(3,'Moustafa','Egypt');

insert into product values
(10, 'LC Phone', 300),
(20, 'LC T-Shirt', 10),
(30, 'LC Book', 45),
(40, 'LC Keychain', 2);

insert into orders values
(1, 1, 10, '2020-06-10', 1),
(2, 1, 20, '2020-07-01', 1),
(3, 1, 30, '2020-07-08', 2),
(4, 2, 10, '2020-06-15', 2),
(5, 2, 40, '2020-07-01', 10),
(6, 3, 20, '2020-06-24', 2),
(7, 3, 30, '2020-06-25', 2),
(9, 3, 30, '2020-05-08', 2);
```

```sql
select c.customer_id, c.name
    from customer c join orders o on c.customer_id = o.customer_id
    join product p on o.product_id = p.product_id
    group by c.customer_id, c.name
    having
    sum(case when left(o.order_date, 7) = '2020-06' then p.price * o.quantity
else 0 end) >= 100
    and
    sum(case when left(o.order_date, 7) = '2020-07' then p.price * o.quantity
else 0 end) >= 100;
```

Question 29:

```sql
create table TVProgram
(program_date date, content_id int,channel varchar(30),
primary key(program_date,content_id));
```

```sql
create table Content
(content_id int primary key,
title varchar(20),
kids_content enum('Y','N'),
content_type varchar(20));

insert into TVProgram values
('2020-06-10 08:00', 1, 'LC-Channel'),
('2020-05-11 12:00', 2, 'LC-Channel'),
('2020-05-12 12:00', 3, 'LC-Channel'),
('2020-05-13 14:00', 4, 'Disney Ch'),
('2020-06-18 14:00', 4, 'Disney Ch'),
('2020-07-15 16:00', 5, 'Disney Ch');

insert into Content values
(1, 'Leetcode Movie', 'N', 'Movies'),
(2, 'Alg. for Kids', 'Y', 'Series'),
(3, 'Database Sols', 'N', 'Series'),
(4, 'Aladdin', 'Y', 'Movies'),
(5, 'Cinderella', 'Y', 'Movies');
```

```sql
select distinct title
from Content
join TVProgram using(content_id)
where kids_content = 'Y'
    and content_type = 'Movies'
    and (month(program_date), year(program_date)) = (6, 2020);
```

Question 30:

```sql
create table NPV (id int ,year int,npv int, primary key (id,year));

create table queries (id int ,year int,primary key (id,year));

insert into NPV values (1, 2018, 100),(7, 2020, 30),
(13, 2019, 40),(1, 2019, 113),
(2, 2008, 121),(3, 2009, 12),
(11, 2020, 99),(7, 2019, 0);

insert into queries values (1, 2019),(2, 2008),(3, 2009),
(7, 2018),(7, 2019),
(7, 2020),(13, 2019);
```

```sql
select
      q.id, q.year,
      ifnull(n.npv,0) as npv
from
      queries as q
left join
      NPV as n
on
      (q.id, q.year) = (n.id, n.year);
```

Question 31:

****Same as 30 in set 1**

Question 32:

```sql
create table employee
( id int primary key,name varchar(30));

create table employeeUNI
(id int, unique_id int , primary key (id,unique_id));

insert into employee values
(1, 'Alice'),
(7, 'Bob'),
(11,'Meir'),
(90, 'Winston'),
(3, 'Jonathan');

insert into employeeUNI values
(3,1),
(11,2),
(90,3);
```

```sql
select unique_id, name
from employee left join employeeUNI
on employee.id = employeeUNI.id;
```

Question 33:

```sql
create table users(id int primary key, name varchar(30));

create table rides (id int primary key,user_id int,distance int);
```

```
insert into users values(1, 'Alice'),(2, 'Bob'),
(3, 'Alex'),
(4, 'Donald'),
(7, 'Lee'),
(13,'Jonathan'),
(19,'Elvis');

insert into rides values (1, 1, 120),(2, 2, 317),
(3, 3, 222),(4, 7, 100),
(5, 13, 312),
(6, 19, 50),
(7, 7, 120),
(8, 19, 400),
(9, 7, 230);
```

```
select name, sum(ifnull(distance, 0)) as travelled_distance
from rides r
right join users u
on r.user_id = u.id
group by name
order by 2 desc,1 asc;
```

Question 34:

**** Same as 29 in set 1**

Question 35:

```
create table movies (movie_id int primary key,
title varchar(30));

create table users (user_id int primary key,
name varchar(30));

create table movieRating (movie_id int,
user_id int,
rating int,
created_at date,
primary key (movie_id,user_id));

insert into movies values
(1,'Avengers'),
(2,'Frozen 2'),
(3,'Joker');
```

```
insert into users values
(1, 'Daniel'),
(2, 'Monica'),
(3, 'Maria'),
(4, 'James');

insert into movieRating values
(1, 1, 3, '2020-01-12'),
(1, 2, 4, '2020-02-11'),
(1, 3, 2, '2020-02-12'),
(1, 4, 1, '2020-01-01'),
(2, 1, 5, '2020-02-17'),
(2, 2, 2, '2020-02-01'),
(2, 3, 2, '2020-03-01'),
(3, 1, 3, '2020-02-22'),
(3, 2, 4, '2020-02-25');
```

```
SELECT user_name AS results FROM
(
SELECT a.name AS user_name, COUNT(*) AS counts FROM movieRating AS b
    JOIN users AS a
    on a.user_id = b.user_id
    GROUP BY b.user_id
    ORDER BY counts DESC, user_name ASC LIMIT 1
) first_query
UNION
SELECT movie_name AS results FROM
(
SELECT c.title AS movie_name, AVG(d.rating) AS rate FROM movieRating AS d
    JOIN movies AS c
    on c.movie_id = d.movie_id
    WHERE substr(d.created_at, 1, 7) = '2020-02'
    GROUP BY d.movie_id
    ORDER BY rate DESC, movie_name ASC LIMIT 1
) second_query;
```

Question 36:

****Same as 33 in set 1**

Question 37:

****Same as 32 in set 1**

Question 38:

```
create table Departments (id int primary key, name varchar(30));

create table students (id int primary key,
name varchar(30),
department_id int);

insert into Departments values
(1, 'Electrical Engineering'),
(7, 'Computer Engineering'),
(13, 'Business Administration');

insert into students values
(23, 'Alice', 1),
(1, 'Bob', 7),
(5, 'Jennifer', 13),
(2, 'John', 14),
(4, 'Jasmine', 77),
(3, 'Steve', 74),
(6, 'Luis', 1),
(8, 'Jonathan', 7),
(7, 'Daiana', 33),
(11, 'Madelynn', 1);
```

```
select s.id, s.name
from
      students s
left join
      Departments d
on
      s.department_id = d.id
where
      d.id is null;
```

Question 39:

```
create table calls
(from_id int,to_id int,duration int);

insert into calls values
(1, 2, 59),(2, 1, 11),(1, 3, 20),(3, 4, 100),(3, 4, 200),(3, 4, 200),(4, 3, 499);
```

```sql
--Approach 1

Select from_id as person1, to_id as person2,
    count(duration) as call_count, sum(duration) as total_duration
from (select *
      from calls
     union all
     select to_id, from_id, duration
      from calls) t1
where from_id < to_id
group by person1, person2;

--Approach 2

SELECT
    LEAST(from_id, to_id) as person1,
    GREATEST(from_id, to_id) as person2,
    COUNT(*) as call_count,
    SUM(duration) as total_duration
FROM calls
GROUP BY 1, 2;
```

Question 40:

**** Same as 23 in set 1**

Question 41:

```sql
create table warehouse
(name varchar(30),
product_id int,
units int,
primary key (name,product_id));

create table products
(product_id int primary key,
product_name varchar(30),
width int,
length int,
height int);

insert into warehouse values
('LCHouse1',1,1),
```

```
('LCHouse1',2,10),
('LCHouse1',3,5),
('LCHouse2',1,2),
('LCHouse2',2,2),
('LCHouse3',4,1);

insert into products value
(1, 'LC-TV', 5, 50, 40),
(2, 'LC-KeyChain', 5, 5, 5),
(3, 'LC-Phone', 2, 10, 10),
(4, 'LC-T-Shirt', 4, 10, 20);
```

```
select name as warehouse_name, sum(units * vol) as volume
from warehouse w
join (select product_id, Width*Length*Height as vol
     from products) p
on w.product_id = p.product_id
group by name;
```

Question 42:

```
create table sales  (sale_date date, fruit enum('apples','oranges'),
sold_num int, primary key (sale_date,fruit));

insert into sales values
('2020-05-01', 'apples', 10),
('2020-05-01', 'oranges', 8),
('2020-05-02', 'apples', 15),
('2020-05-02', 'oranges', 15),
('2020-05-03', 'apples', 20),
('2020-05-03', 'oranges', 0),
('2020-05-04', 'apples', 15),
('2020-05-04', 'oranges', 1);
```

```
select date(sale_date) as sale_date,
       sum(case when fruit = 'apples' then sold_num
                when fruit = 'oranges' then -sold_num end) as diff
from sales
group by 1 order by 1;
```

Question 43:

```sql
create table activity(player_id int ,device_id int,
event_date date ,
games_played int,
primary key (player_id,event_date));

insert  into activity VALUES
(1, 2, '2016-03-01', 5),
(1, 2, '2016-03-02', 6),
(2, 3, '2017-06-25', 1),
(3, 1, '2016-03-01', 0),
(3, 4, '2016-07-03', 5);
```

```sql
select round(
    ifnull(
        (
        select count(distinct a.player_id)
        from activity as a join activity as b
        on a.player_id = b.player_id and datediff(b.event_date, a.event_date) = 1
        where a.event_date = (
               select min(event_date) from activity where player_id = a.player_id
            )
        ) / (select count(distinct player_id) from activity  ),
    0),
2)
as fraction;
```

Question 44:

```sql
create table employee
(id int primary key,
name varchar(30),
department varchar(30),
managerId int);

insert into employee values
(101, 'John', 'A', null),
(102, 'Dan', 'A', 101),
(103, 'James', 'A', 101),
(104, 'Amy', 'A', 101),
(105, 'Anne', 'A', 101),
(106, 'Ron', 'B', 101);
```

```sql
select Name from employee
where Id in
(
  select ManagerId from employee
  group by 1
  having count(*) >= 5
);
```

Question 45:

```sql
create table department
(dept_id int primary key,
dept_name varchar(30));

create table student
(student_id int primary key,
student_name varchar(30),
gender varchar(30),
dept_id int,
Foreign key (dept_id) references department(dept_id));

insert into department values
(1,'Engineering'),
(2,'Science'),
(3,'Law');

insert into student values
(1, 'Jack', 'M', 1),
(2, 'Jane', 'F', 1),
(3, 'Mark', 'M', 2);
```

```sql
select
    a.dept_name,
    coalesce(count(student_id), 0) student_number
from
    department a
left join
    student b
on
    (a.dept_id = b.dept_id)
group by a.dept_name
order by student_number desc, a.dept_name asc;
```

Question 46:

```sql
create table customer
(customer_id int, product_key int,
foreign key (product_key) references product(product_key));

create table product
(product_key int primary key);

insert into product values
(5),(6);

insert into customer values
(1,5),
(2,6),
(3,5),
(3,6),
(1,6);
```

```sql
SELECT
    customer_id
FROM customer
GROUP BY customer_id
HAVING COUNT( DISTINCT product_key) = (SELECT COUNT(*) FROM product);
```

Question 47:

```sql
create table employee
(employee_id int primary key,
name varchar(30),
experience_years int);

create table project
(project_id int ,
employee_id int,
primary key (project_id, employee_id),
foreign key (employee_id) references employee(employee_id));

insert into employee values
(1,'Khaled',3),
(2,'Ali',2),
(3,'John',3),
(4,'Doe',2);
```

```sql
insert into project values
(1,1),
(1,2),
(1,3),
(2,1),
(2,4);
```

```sql
select project_id, project.employee_id
from project inner join employee
on project.employee_id = employee.employee_id
where (project_id, experience_years) in
    (select project_id, max(experience_years) as years
    from project inner join employee
    on project.employee_id = employee.employee_id
    group by project_id);
```

Question 48:

```sql
create table Books (
book_id int,
name varchar(20),
available_from date
);

insert into Books values
(1,"Kalila And Demna",'2010-01-01'),
(2,"28 Letters",'2012-05-12'),
(3,"The Hobbit",'2019-06-10'),
(4,"13 Reasons Why",'2019-06-01'),
(5,"The Hunger Games",'2008-09-21');

create table Orders(
order_id int,
book_id int,
quantity int,
dispatch_date date
);

insert into Orders values
(1,1,2,'2018-07-26'),(2,1,1,'2018-11-05'),
(3,3,8,'2019-06-11'),(4,4,6,'2019-06-05'),
(5,4,5,'2019-06-20'),(6,5,9,'2009-02-02'),
(7,5,8,'2010-04-13');
```

```sql
--First condition

SELECT DISTINCT b.book_id, b.name
FROM Books b
WHERE available_from < '2019-05-23'
AND book_id NOT IN
(SELECT book_id
 FROM Orders
 WHERE dispatch_date between '2018-06-23' and '2019-06-23'
 GROUP BY book_id
 HAVING SUM(quantity) >= 10);

--Second condition

SELECT DISTINCT b.book_id, b.name
FROM Books b
WHERE available_from < '2019-05-23'
AND book_id  IN
(SELECT book_id
 FROM Orders
 WHERE dispatch_date > '2018-06-23'
 GROUP BY book_id
 HAVING SUM(quantity) < 10);
```

Question 49:

```sql
create table enrollments
(student_id int,
course_id int,
grade int,
primary key (student_id,course_id));

insert into enrollments values
(2, 2, 95),
(2, 3, 95),
(1, 1, 90),
(1, 2, 99),
(3, 1, 80),
(3, 2, 75),
(3, 3, 82);
```

```sql
select e1.student_id, min(e1.course_id) as course_id, e1.grade
from enrollments e1
where e1.grade =
    (select max(grade) as max_grade
    from enrollments e2;
    where e1.student_id = e2.student_id)
group by e1.student_id,e1.grade
order by e1.student_id;
```

Question 50 :

```sql
create table Players
(player_id   int,
group_id   int);

insert into Players values
(15,1),(25,1),(30,1),(45,1),(10,2),(35,2),(50,2),(20,3),(40,3);

create table Matches
(match_id      int,
first_player  int,
second_player int,
first_score   int,
second_score  int);

insert into Matches values
(1,15,45,3,0),(2,30,25,1,2),(3,30,15,2,0),(4,40,20,5,2),(5,35,50,1,1);
```

```sql
select group_id,player_id from(
 select *,dense_rank() over (partition by group_id order by
 winningscore desc) rnk
 from Players p join
 (select
case when
first_score> second_score then first_player
when second_score>first_score then second_player
else
 CASE
   when first_player<second_player then first_player
   when second_player<first_player then second_player
   end
end as winner,
case when
```

```
first_score> second_score then first_score
when  second_score>first_score then second_score
else
  CASE
   when first_player<second_player then first_player
   when second_player<first_player then second_player
   end
end as winningscore
 from Matches) w on p.player_id = w.winner
 order by player_id)a where a.rnk =1;
```