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

Question 28: 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); 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); 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: create table TVProgram (program\_date date, content\_id int,channel varchar(30), primary key(program\_date,content\_id)); 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'); 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: 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); 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: 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); select unique\_id, name from employee left join employeeUNI on employee.id = employeeUNI.id;

Question 33: 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); --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: 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: 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); 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: 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); select Name from employee where Id in ( select ManagerId from employee group by 1 having count(\*) >= 5 );

Question 45: 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); 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: 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); SELECT customer\_id FROM customer GROUP BY customer\_id HAVING COUNT( DISTINCT product\_key) = (SELECT COUNT(\*) FROM product);

Question 47: 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); insert into project values (1,1), (1,2), (1,3), (2,1), (2,4); 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: 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'); --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: 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); 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 : 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); 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\_score then first\_score when second\_score>first\_score then second\_score else CASE when first\_player