**Q1**.

Ans: select \* from CITY

where countrycode='USA' and population>100000

**Q2.**

Ans: select distinct name from CITY

where countrycode='USA' and population>120000

**Q3.**

**Ans: select \* from CITY**

**Q4.**

**Ans: select \* from CITY where id=1661**

Q5.

Ans: **select \* from CITY** where countrycode='JPN'

Q6.

Ans: **select distinct name from CITY** where countrycode='JPN'

Q7.

Ans: select city, state from station

Q8.

Ans: select distinct city from station where id%2=0

Q9.

Ans: select count(city) - count(distinct city) from station

Q10.

Ans:

with cte as (

select city,length(city) len, dense\_rank() over (order by length(city) ) rnk from station

),

cte1 as

(

select city, len, rnk from cte

where rnk in (select min(rnk) from cte union all select max(rnk) from cte )

),

cte2 as

(

select city,len, rank() over (partition by rnk order by city) rnk from cte1

)

select city, len from cte2 where rnk=1

Q11.

Ans:

select distinct city from station where lower(substr(city,1,1)) in ('a','e','i','o','u');

Q12.

Ans:

select distinct city from station where lower(substr(city,-1,1)) in ('a','e','i','o','u');

Q13.

Ans:

select distinct city from station where lower(substr(city,1,1)) not in ('a','e','i','o','u');

Q14.

Ans:

select distinct city from station where lower(substr(city,-1,1)) not in ('a','e','i','o','u');

Q15.

Ans:

select distinct city from station where lower(substr(city,1,1)) in ('a','e','i','o','u')

or lower(substr(city,-1,1)) in ('a','e','i','o','u');

Q16.

Ans:

select distinct city from station where lower(substr(city,1,1)) in ('a','e','i','o','u')

and lower(substr(city,-1,1)) in ('a','e','i','o','u');

Q17.

Ans:

select s.product\_id, p.product\_name from sales s

inner join product p

on s.product\_id= p.product\_id

where s.sale\_date between '2019-01-01' and '2019-03-31'

and s.product\_id not in

(select distinct product\_id from sales

where sale\_date between '2019-04-01' and '2019-12-31');

Q18.

Ans:

select distinct author\_id from views

where author\_id= viewer\_id

Q19.

Ans:

select

round((sum(case

when order\_date=customer\_pref\_delivery\_date then 1

else 0

end)\*100)/ count(\*),2) immediate

from delivery;

Q20.

Ans:

with cte as

(

select

ad\_id,

sum(case when `action`='Clicked' then 1 else 0 end) ad\_total\_clicks,

sum(case when `action`='Viewed' then 1 else 0 end) ad\_total\_views

from ads

group by ad\_id )

select

ad\_id,

round(case when (ad\_total\_clicks+ad\_total\_views)=0 then 0

else (ad\_total\_clicks\*100)/(ad\_total\_clicks+ad\_total\_views)

end,2) CTR

from cte

order by ctr desc, ad\_id asc;

Q21.

Ans:

select employee\_id, count(employee\_id) over(partition by team\_id) team\_size

from employee order by employee\_id;

Q22.

Ans:

with cte as(

select country\_id, avg(weather\_state) temp

from weather

where month(day)=11

group by country\_id

)

select

cntry.country\_name,

case when cte.temp<=15 then 'Cold'

when cte.temp>=25 then 'Hot'

else 'Warm'

end weather\_type

from

cte inner join countries cntry

on cte.country\_id= cntry.country\_id

Q23.

Ans:

select

a.product\_id,

avg(a.units \* b.price)/ sum(a.units) average\_price

from unitssold a

inner join

prices b

on a.product\_id=b.product\_id and a.purchase\_date between b.start\_date and b.end\_date

group by a.product\_id;

Q24.

Ans:

select

player\_id,

min(event\_date) first\_login

from activity

group by player\_id

Q25.

Ans:

with cte as (

select

player\_id,

device\_id,

rank() over(partition by player\_id order by event\_date) rnk

from activity

)

select

player\_id,

device\_id

from cte where rnk=1

Q26.

Ans:

select

product\_name,

sum(unit) units

from

products a

left join

orders b

on a.product\_id=b.product\_id

and year(b.order\_date)=2020 and month(b.order\_date) = 2

group by product\_name

having sum(unit)>=100

Q27.

Ans:

select

\* from users

where regexp\_like(mail, '^[a-z]{1}[a-z0-9\-\_\.]\*@leetcode.com','i' )

Q28.

Ans:

with cte as (

select b.customer\_id, b.name, month(a.order\_date) mon,

case when sum(quantity \* price) >=100 then 1 else 0 end flag

from orders a

inner join customers b

on a.customer\_id=b.customer\_id

inner join product c

on a.product\_id=c.product\_id

and month(a.order\_date) in (6,7)

group by b.customer\_id, b.name,month(a.order\_date)

)

select customer\_id, name

from cte where flag=1

group by customer\_id, name

having count(\*)=2

Q29.

Ans:

select distinct title from content a

inner join

tvprogram b

on a.content\_id=b.content\_id

where a.content\_type='Movies' and a.Kids\_content='Y'

and b.program\_date between '2020-06-01' and '2020-06-30'

Q30.

Ans:

select b.id,b.year,b.npv from

queries a

inner join

npv b

on a.id=b.id and a.year=b.year

Q31.

Ans:

Duplicate question

Q32.

Ans:

select b.unique\_id,a.name from

employees a

left join

employeeuni b

on a.id=b.id

Q33.

Ans:

select a.name, sum(b.distance) travelled\_distance from

users a

left join

rides b

on a.id=b.user\_id

group by a.name

order by sum(b.distance) desc, a.name asc

Q34.

Ans:

select a.product\_name, sum(unit) amount from

products a

left join

orders b

on a.product\_id=b.product\_id and b.order\_date between '2020-02-01' and '2020-02-29'

group by a.product\_name;

Q35.

Ans:

Part a)

with cte as (

select a.name, count(\*) cnt from

Users a

left join

MovieRating b

on a.user\_id=b.user\_id

group by a.name

)

select \* from cte

where cnt = (select max(cnt) from cte)

order by name;

part b)

with cte as (

select a.title, avg(b.rating) avg\_rat from

movies a

left join

MovieRating b

on a.movie\_id=b.movie\_id and b.created\_at between '2020-02-01' and '2020-02-29'

group by a.name

)

select \* from cte

where avg\_rat = (select max(avg\_rat) from cte)

order by title;

Q36.

Ans: repeated question

Q37.

Ans:

Repeated question

Q38.

Ans:

select a.id, a.name from

Students a

left join

departments b

on a.department\_id=b.id

where b.name is null

Q39.

Ans:

with cte as (

select from\_id person1,to\_id person2, duration from

calls

where from\_id <to\_id

union all

select to\_id person1, from\_id person2, duration from

calls

where to\_id <from\_id

)

select person1, person2, count(\*) call\_count, sum(duration) total\_duration from cte

group by person1, person2

Q40.

Ans:

Repeated question

Q41.

Ans:

select a.name warehouse\_name, sum(a.units \* b.width \* b.length \* b.height) volume from

Warehouse a

left join

Products b

on a.product\_id=b.product\_id

group by a.name

Q42.

Ans:

select sale\_date,

sum(case when fruit='apples' then sold\_num else 0 end) -

sum(case when fruit='oranges' then sold\_num else 0 end) diff

from sales

group by sale\_date

Q43.

Ans:

with cte as

(

select

player\_id,

event\_date,

date(min(event\_date) over (partition by player\_id)) first\_login\_date

from activity

),

cte1 as

(

select count(distinct player\_id) total\_players from cte

)

select round(count(\*)/total\_players,2) fraction from cte,cte1

where datediff(event\_date,first\_login\_date)= 1

Q44.

Ans:

with cte as (

select manager.name manager, reportee.name reportee,

count(reportee.name) num

from employee manager

left join

employee reportee

on manager.id= reportee.managerId

group by manager.name

)

select distinct manager from cte

where num>=5

Q45.

Ans:

select d.dept\_name, count(student\_id) student\_number

from Department d

left join

Student s

on d.dept\_id= s.dept\_id

group by d.dept\_name

order by count(student\_id) desc, d.dept\_name

Q46.

Ans:

select distinct c.customer\_id

from product p

inner join

customer c

on p.product\_key= e.product\_key

Q47.

Ans:

with cte as (

select p.project\_id, e.employee\_id, dense\_rank() over (partition by project\_id order by experience\_years desc) ordr

from project p

left join

employee e

on p.employee\_id= e.employee\_id

)

select project\_id, employee\_id

from cte

where ordr=1

Q48.

Ans:

with cte as (

select b.name, b.available\_from, count(b.name) over (partition by b.name) num\_sold

from orders o

left join

books b

on o.book\_id= b.book\_id;

)

select distinct name from cte

where num\_sold < 10

and datediff(str\_to\_date('2019-06-23','%Y-%m-%d'), available\_from ) >30;

Q49.

Ans:

with cte as (

select \*, dense\_rank() over (partition by student\_id order by grade desc, course\_id asc) rnk

from enrollments

)

select student\_id, course\_id,grade

from cte

where rnk = 1

Q50.

Ans:

Question doesn’t match the dataset provided.