**WEEK 1 SQL CHALLENGE – LEVEL -EASY**

--1-Write a query that calculates the difference between the highest salaries found in the marketing and engineering departments. Output just the absolute difference in salaries.

select ABS( MAX(CASE WHEN department = 'engineering' THEN salary END)

- MAX(CASE WHEN department = 'marketing' THEN salary END)

) as Difference

from db\_employee as a

join db\_dept as b on a.department\_id = b.id

--2 Find the current salary of each employee assuming that salaries increase each year. Output their id, first name, last name, department ID, and current salary. Order your list by employee ID in ascending order.

Select id,first\_name,last\_name,department\_id,

MAX(salary) as Current\_salary

from ms\_employee\_salary

Group by id,first\_name,last\_name,department\_id

Order by id

--3-Find all inspections which are part of an inactive program.

select \* from los\_angeles\_restaurant\_health\_inspections

where program\_status = 'Inactive'

--4-Output the National Olympics Committee (NOC) name along with the desired year.Sort records by the year and the NOC in ascending order.

select noc,year

from olympics\_athletes\_events

Order by noc,year ASC

--5-Find the total cost of each customer's orders. Output customer's id, first name, and the total order cost. Order records by customer's first name alphabetically.

select C.id,C.first\_name,SUM(O.total\_order\_cost) as total\_cost

from customers C

inner join orders O

on C.id = O.id

Group by C.id,C.first\_name

Order by C.first\_name ASC

--6 Output both the bike number and the date-timestamp of the bike's last use (i.e., the date-time the bike was returned). Order the results by bikes that were most recently used.

select bike\_number, max(end\_time) AS time

from dc\_bikeshare\_q1\_2012

group by bike\_number;

--7-Find the gender that has made the most number of doctor appointments. Output the gender along with the corresponding number of appointments.

select gender,COUNT(appointmentday) as total\_appt

from medical\_appointments

Group by gender

Order by total\_appt DESC

**WEEK 1 SQL CHALLENGE – LEVEL -EASY**

--8-Find the number of rows for each review score earned by 'Hotel Arena'. Output the hotel name (which should be 'Hotel Arena'), review score along with the corresponding number of rows with that score for the specified hotel.

SELECT hotel\_name, reviewer\_score, COUNT(\*) AS count

FROM hotel\_reviews

WHERE hotel\_name = 'Hotel Arena'

GROUP BY hotel\_name, reviewer\_score;

--9-Count the number of movies that Abigail Breslin was nominated for an oscar.

select Count(movie)

from oscar\_nominees

where nominee = 'Abigail Breslin'

--10-Find all posts which were reacted to with a heart. For such posts output all columns from facebook\_posts table.

select distinct P.post\_id,P.poster,P.post\_text,P.post\_keywords,P.post\_date

from facebook\_reactions R

Inner Join facebook\_posts P

on R.post\_id = P.post\_id

where reaction = 'heart'

--11-find the average popularity of the Hack per office location. Output the location along with the average popularity.

select E.location,avg ( cast ( popularity as float ) )

from facebook\_employees E

Inner join facebook\_hack\_survey H

on E.id = H.employee\_id

Group by E.location

--12-Find all Lyft drivers who earn either equal to or less than 30k USD or equal to or more than 70k USD. Output all details related to retrieved records.

select \* from lyft\_drivers

where yearly\_salary <= 30000

or yearly\_salary >= 70000

--13- Find how many times each artist appeared on the Spotify ranking list Output the artist name along with the corresponding number of occurrences. Order records by the number of occurrences in descending order.

select artist,Count(artist) as total\_occurance

from spotify\_worldwide\_daily\_song\_ranking

Group by artist

Order by Count(artist) DESC

--14-Find the base pay for Police Captains. Output the employee name along with the corresponding base pay.

select employeename,basepay

from sf\_public\_salaries

where jobtitle like 'Captain%'

**WEEK 1 SQL CHALLENGE – LEVEL -EASY**

--15-Find libraries who haven't provided the email address in circulation year 2016 but their notice preference definition is set to email. Output the library code.

select distinct home\_library\_code from library\_usage

where circulation\_active\_year = 2016

and provided\_email\_address = 'False'

and notice\_preference\_definition = 'email'

--16- Compare each employee's salary with the average salary of the corresponding department. Output the department, first name, and salary of employees along with the average salary of that department.

select department,first\_name,salary,

AVG(cast(salary as float)) over(partition by department) as dptavgsalary

from employee;

--17-Find order details made by Jill and Eva. Consider the Jill and Eva as first names of customers. Output the order date, details and cost along with the first name. Order records based on the customer id in ascending order.

SELECT c.first\_name, o.order\_date, o.order\_details, o.total\_order\_cost

FROM customers c INNER JOIN orders o

ON c.id = o.cust\_id

WHERE c.first\_name IN ('Jill', 'Eva')

ORDER BY cust\_id ;

--18- Find the details of each customer regardless of whether the customer made an order. Output the customer's first name, last name, and the city along with the order details. You may have duplicate rows in your results due to a customer ordering several of the same items. Sort records based on the customer's first name and the order details in ascending order.

select first\_name, last\_name, city, order\_details

from customers c

left join orders o

on c.id = o.cust\_id

--19-Find the number of workers by department who joined in or after April.Output the department name along with the corresponding number of workers.Sort records based on the number of workers in descending order.

select DEPARTMENT, COUNT(WORKER\_ID) as Total\_workers

from worker

WHERE MONTH(JOINING\_DATE) >=4

GROUP BY DEPARTMENT

order by COUNT(WORKER\_ID) DESC

--20-Find the number of employees working in the Admin department that joined in April or later.

select COUNT(worker\_id) as total\_workers

from worker

Where department = 'Admin' and month(joining\_date) >= 4

**WEEK 1 SQL CHALLENGE – LEVEL -EASY**

--20-Find the activity date and the pe\_description of facilities with the name 'STREET CHURROS' and with a score of less than 95 points.

select activity\_date,pe\_description

from los\_angeles\_restaurant\_health\_inspections

where facility\_name like '%STREET CHURROS%' and score < 95

--21- Find the most profitable company from the financial sector. Output the result along with the continent.

select top 1 company,continent

from forbes\_global\_2010\_2014

where sector = 'Financials'

order by profits DESC

--22-Count the number of user events performed by MacBookPro users. Output the result along with the event name. Sort the result based on the event count in the descending order.

select Count(user\_id) as no\_of\_users,event\_name

from playbook\_events

where device = 'macbook pro'

group by event\_name

order by no\_of\_users DESC

--23-Find the average number of bathrooms and bedrooms for each city’s property types. Output the result along with the city name and the property type.

select property\_type,city,AVG(cast(bathrooms as float)) as AVG\_bathrooms,AVG(cast(bedrooms as float)) as avg\_bedrooms

from airbnb\_search\_details

group by property\_type,city

--24-You have been asked to find the 5 most lucrative products in terms of total revenue for the first half of 2022 (from January to June inclusive).Output their IDs and the total revenue.

select top 5 product\_id,sum(cost\_in\_dollars \*units\_sold ) as total\_revenue

from online\_orders

where month(date) >=1 AND month(date) < 7

group by product\_id

Order by total\_revenue DESC

--25-Write a query that will calculate the number of shipments per month. The unique key for one shipment is a combination of shipment\_id and sub\_id. Output the year\_month in format YYYY-MM and the number of shipments in that month.

select format(cast(shipment\_date as date),'yyyy-MM'),count(\*)

from amazon\_shipment

group by format(cast(shipment\_date as date),'yyyy-MM')

--26-Write a query that returns the number of unique users per client per month

select count(distinct user\_id) as no\_userids,client\_id,month(time\_id) AS MONTH

from fact\_events

group by client\_id,month(time\_id)

**WEEK 1 SQL CHALLENGE – LEVEL -EASY**

--27- Write a SQL statement that displays all the information about all salespeople.

SELECT \* FROM salesman;

--28- Write a query identifying the type of each record in the TRIANGLES table using its three side lengths. Output one of the following statements for each record in the table:

SELECT CASE WHEN (A + B > C) AND (A + C > B) AND (B + C > A)

THEN CASE WHEN A = B

THEN CASE WHEN A = C

THEN 'Equilateral'

ELSE 'Isosceles' END

ELSE CASE WHEN B = C

THEN 'Isosceles'

ELSE CASE WHEN A = C

THEN 'Isosceles' ELSE 'Scalene'

END END END

ELSE 'Not A Triangle'

END

FROM TRIANGLES

--28-Query the two cities in STATION with the shortest and longest CITY names, as well as their respective lengths (i.e.: number of characters in the name). If there is more than one smallest or largest city, choose the one that comes first when ordered alphabetically.

SELECT DISTINCT CITY, LENGTH(CITY)

FROM STATION

ORDER BY LENGTH(CITY) ASC

LIMIT 1;

SELECT CITY, LENGTH(CITY)

FROM STATION

ORDER BY LENGTH(CITY) DESC

LIMIT 1;

-29--Write a query calculating the amount of error (i.e.: average monthly salaries), and round it up to the next integer.

SELECT CEILING( AVG( CAST(Salary AS decimal)) - AVG( CAST(REPLACE(Salary, 0, '') AS decimal)))

FROM Employees

--30-Query the following two values from the STATION table: The sum of all values in LAT\_N rounded to a scale of decimal places. The sum of all values in LONG\_W rounded to a scale of decimal places. Input Format The STATION table is described as follows:

select FORMAT(round(SUM(LAT\_N),2),'0.00'),FORMAT(round(SUM(LONG\_W),2),'0.00')

from STATION;

--31-Query the sum of Northern Latitudes (LAT\_N) from STATION having values greater than 38.7880 and less than 137.2345 . Truncate your answer to decimal places.

select FORMAT(SUM(LAT\_N),'0.0000')

from STATION

where LAT\_N > 38.7880 and LAT\_N < 137.2345

**WEEK 1 SQL CHALLENGE – LEVEL -EASY**

--

--32-Query the greatest value of the Northern Latitudes (LAT\_N) from STATION that is less than . Truncate your answer to 4 decimal places.

select FORMAT(MAX(LAT\_N),'0.0000')

from STATION

where LAT\_N < 137.2345

--33-Query the Western Longitude (LONG\_W) for the largest Northern Latitude (LAT\_N) in STATION that is less than 137.2345 . Round your answer to decimal places.

select format(round(LONG\_W,4),'0.0000')

from station where LAT\_N=(select max(LAT\_N) from station where LAT\_N < 137.2345)

--34-Query the smallest Northern Latitude (LAT\_N) from STATION that is greater than 38.7880 . Round your answer to 4 decimal places

select FORMAT(MIN(LAT\_N),'0.0000') as Smallest\_LAT\_N from STATION

where LAT\_N > 38.7780

--35- Query the Western Longitude (LONG\_W)where the smallest Northern Latitude (LAT\_N) in STATION is greater than . Round your answer to decimal places.

select FORMAT((LONG\_W),'0.0000')

from STATION where LAT\_N = (select Min(Lat\_N) from STATION where lat\_N > 38.7780 )

--36-Query the Manhattan Distance between points and and round it to a scale of 4 decimal places.

Select FORMAT((MAX(LAT\_N) - MIN(LAT\_N)) + (Max(LONG\_W) + MIN(LONG\_W)),'0.0000')

from STATION

--37-Given the CITY and COUNTRY tables, query the sum of the populations of all cities where the CONTINENT is 'Asia'.

select SUM(city.population)

from city

Inner Join country

on CITY.CountryCode = COUNTRY.Code

where CONTINENT = 'Asia'

--38-Given the CITY and COUNTRY tables, query the names of all cities where the CONTINENT is 'Africa'.

select (city.name)

from city

Inner Join country

on CITY.CountryCode = COUNTRY.Code

where CONTINENT = 'Africa'

**WEEK 1 SQL CHALLENGE – LEVEL -EASY**

--39-Given the CITY and COUNTRY tables, query the names of all the continents (COUNTRY.Continent) and their respective average city populations (CITY.Population) rounded down to the nearest integer.

select country.Continent,ROund(AVG(CITY.Population),2)

from city

inner join country

on CITY.CountryCode = COUNTRY.Code

group by country.Continent

--40- Write a query to print the pattern P(20

DECLARE @count INT = 10;

WHILE @count > 0

BEGIN PRINT(REPLICATE('\* ', @count)) SET @count = @count -1 END

-- Output

\* \* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*