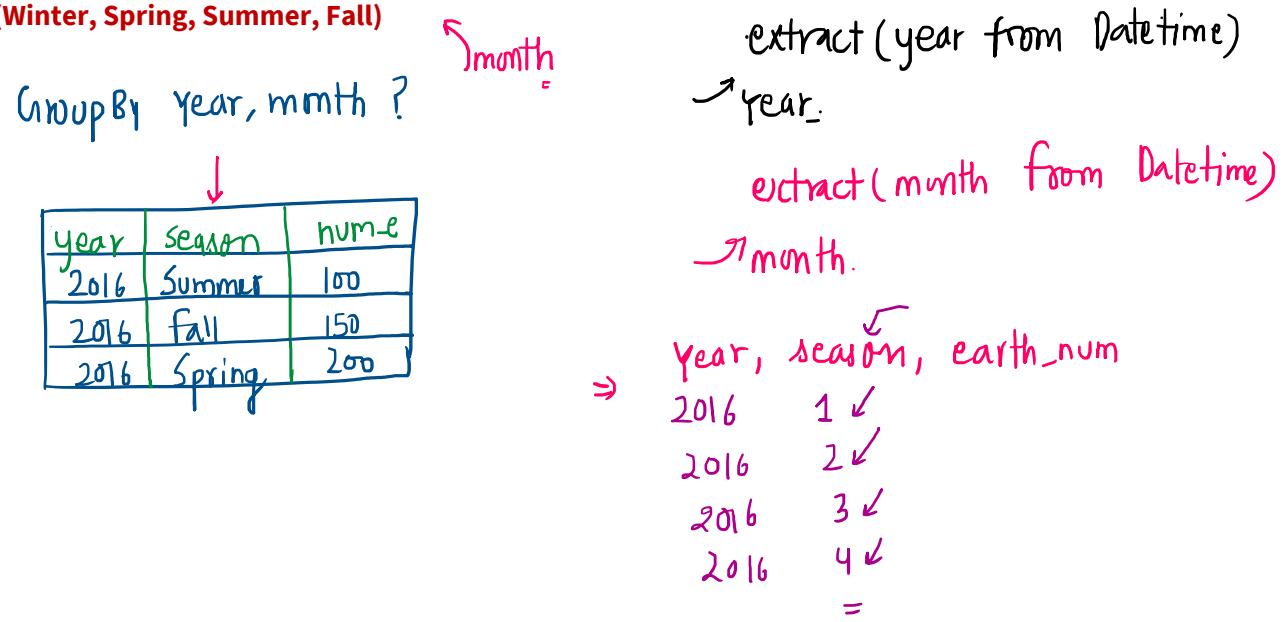
Session 14 -Applied SQL 3

22 September 2024 12:24

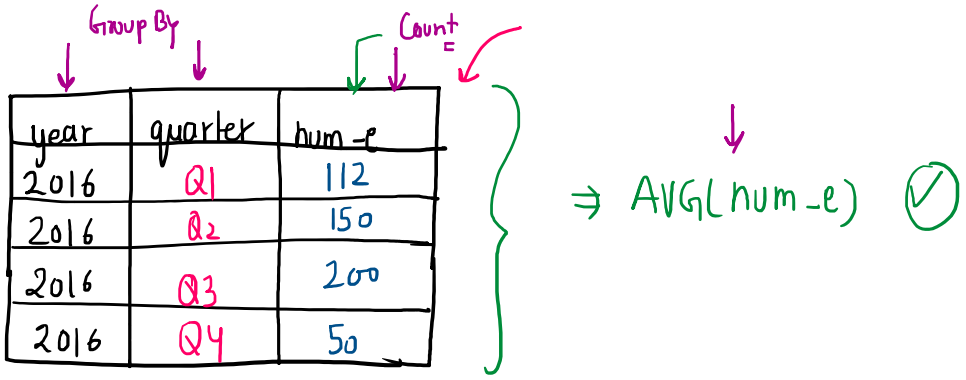
NEXT SESSION:

Q: What is the seasonal distribution of earthquake occurrences each year? (Winter, Spring, Summer, Fall)




```
select extract(YEAR from Datetime) AS year,
      CASE
       WHEN extract(MONTH from Datetime) IN (3,4,5) THEN "Spring"
       WHEN extract(MONTH from Datetime) IN (12,1,2) THEN "Winter"
       WHEN extract(MONTH from Datetime) IN (6,7,8) THEN "Summer"
       ELSE "Fall"
      END as season,
      count(*) AS num_earthquakes
from `neic.earthquakes` e
group by year,season
order by year desc,season;
```

Q: Based on historical data, how many earthquakes are expected in the upcoming quarter?




```
select round(AVG(t1.num_earthquake)) AS avg_earthquakes
from (
   select extract(YEAR from Datetime) as year,
```

```
        extract(QUARTER from Datetime) as quarter,
        count(ID) num_earthquake
    from `neic.earthquakes` e
    group by year, quarter
) t1;
```

Q: What is the frequency of earthquakes and their average magnitude for each quarter over the past five years?

yearquarter	num_earthqke	avg_mag
2016-Q1	112	6.52
2016-Q2	200	5.2
2016-Q3	100	6.1
2016-Q4	50	7.2

format_date ??

format_date(, Date)

"2016-12-31"

Where

```
Date > date_sub(parse_date("%Y-%m-%d", "2016-12-31"), INTERVAL 5 YEAR)
```

"2016-12-31"

date_sub(Current_date() , INTERVAL 5 YEAR)

parse_date ↓ i/p → String ("2016-12-31")

↘ o/p → Date (2016-12-31)

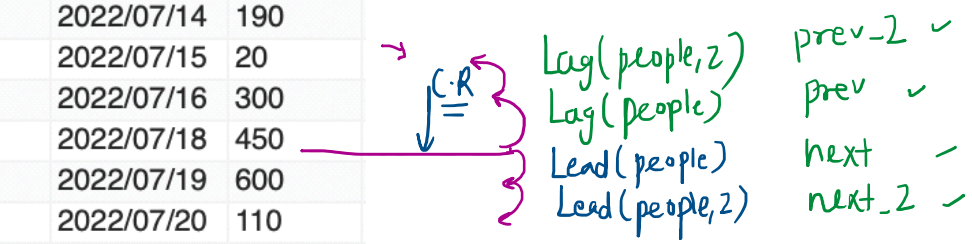
- 2024-09-23 ✓
- 09-23-2024 ✓
- 23-09-2024 ✓
- 23/09/2024 ✓ ← %d / %m / %Y

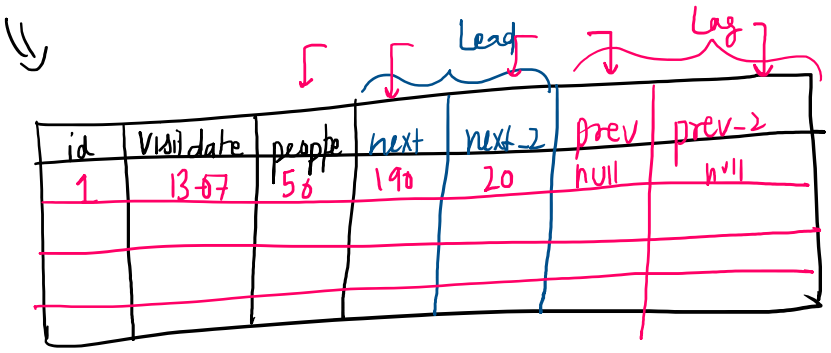
```
select format_date("%Y-Q%Q", Date) year_quarter,
      count(ID) frequency,
      round(avg(Magnitude),2) avg_mag
from  `neic.earthquakes` e
where Date > date_sub(parse_date("%Y-%m-%d", "2016-12-31"), INTERVAL 5 YEAR)
group by format_date("%Y-Q%Q", Date)
order by year_quarter desc;

-- 2016-12-31
```

Break Till : 22:16

id	visit_date	people
1	2022/07/13	50
2	2022/07/14	190
3	2022/07/15	20
4	2022/07/16	300
5	2022/07/18	450
6	2022/07/19	600
7	2022/07/20	110
8	2022/07/21	220





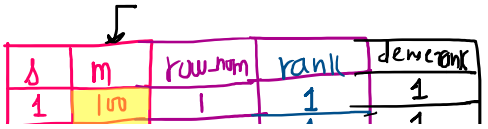
extract( )
parse_date( )
date_add
date_sub
date_diff
format_date

QUIZ:

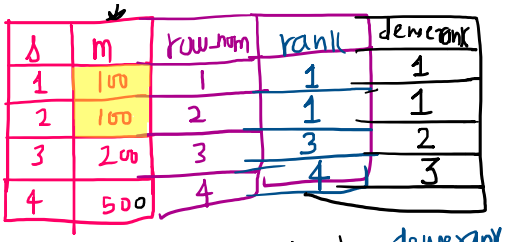
1. What is a Window Function in SQL?
   a) A function that returns a single row
   b) A function that operates over a range of rows
   c) A function that removes duplicates
   d) A function that changes data types

2. Which clause is mandatory when using a window function?
   a) GROUP BY
   b) ORDER BY
   c) PARTITION BY
   d) OVER

3. What is the difference between RANK() and ROW_NUMBER()?
   a) RANK() skips ranks when there is a tie, ROW_NUMBER() doesn't
   b) ROW_NUMBER() skips rows when there is a tie, RANK() doesn't
   c) RANK() is faster
   d) There is no difference



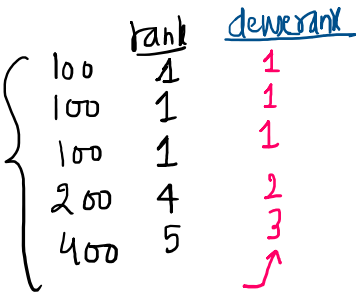
b) ROW_NUMBER() skips rows when there is a tie, RANK() doesn't
c) RANK() is faster
d) There is no difference



4. Consider the following query:

```
SELECT department, employee, salary,
RANK() OVER (PARTITION BY department ORDER BY salary DESC) as rank
FROM employees;
```

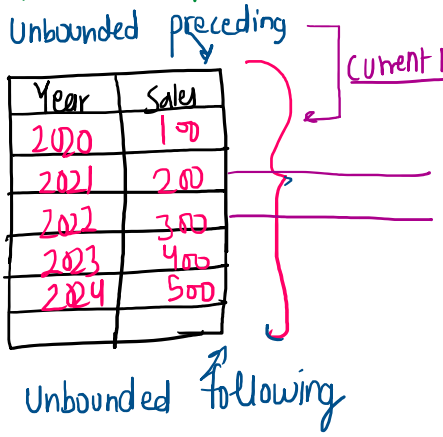
What does the query do?



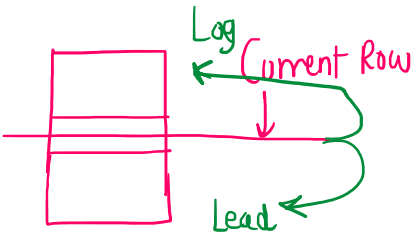
a) It ranks employees across all departments
b) It ranks employees within each department based on salary
c) It assigns a unique rank to every employee
d) It partitions salaries by department and does nothing else

5. Which window function would you use to calculate a cumulative sum?
a) RANK()
b) ROW_NUMBER()
c) SUM()
d) CUME_DIST()

① sum(revenue) over (partition by year) as revenue.
② sum(revenue) over (partition by year Order by Sales) as revenue



6. What is the difference between LAG() and LEAD() in window functions?
a) LAG() gets the previous row, LEAD() gets the next row
b) LEAD() gets the previous row, LAG() gets the next row
c) LAG() returns the highest row value
d) LEAD() returns the highest row value



7. What does NTILE(4) do in a window function?
a) Divides rows into 4 equal parts and assigns a group number
b) Returns the top 4 rows
c) Divides rows based on salary
d) Assigns the number 4 to each row

rows = 100
rows = 101 ? =>

8. Which window function would you use to find the average value within a partition?
   a) RANK()
   b) AVG()
   c) ROW_NUMBER()
   d) LEAD()

→ Select avg(salary) over (partition by department) as avg_sal

9. What will the following query return?

```
SELECT employee, salary,
SUM(salary) OVER (ORDER BY salary ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) as
running_total
FROM employees;
```

Default

   a) The total salary for all employees
   b) A running total of salaries ordered by employee names
   c) A cumulative sum of salaries in ascending order
   d) The average salary

Syntax Error
→ Consumple sum

sum(_) over(order by <col> rows b/w unbounded preceding & Unbounded following) ⇒

10. How would you reset a rank within a window function?
   a) Use ORDER BY
   b) Use GROUP BY
   c) Use PARTITION BY
   d) Use WHERE

11. Which of the following is true about DENSE_RANK()?
   a) It behaves the same as RANK()
   b) It skips rank numbers when there is a tie
   c) It assigns consecutive rank numbers, even in the case of ties
   d) It gives the highest rank to the first row

12. What will the following query return?

```
SELECT employee, salary,
LEAD(salary, 1) OVER (ORDER BY salary) AS next_salary
FROM employees;
```

Lead(salary) ≈ Lead(Salary, 1)

Salary

   a) The salary of the next employee in the salary order
   b) The salary of the previous employee
   c) The sum of all salaries
   d) The cumulative salary of all employees

d) The cumulative salary of all employees

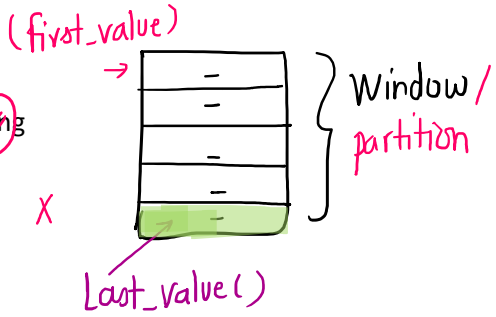
13. What does the following query achieve?

```
SELECT department, employee, salary,
PERCENT_RANK() OVER (PARTITION BY department ORDER BY salary DESC) AS pct_rank
FROM employees;
```

a) It ranks employees within each department based on their salary, as a percentage between 0 and 1
b) It gives employees a random percentage based on salary
c) It assigns an exact rank to each employee
d) It gives the percentage of total salary for each department

14. What does the FIRST_VALUE() function do in SQL?

a) Returns the first row from a result set
b) Returns the first value in a partition based on the ORDER BY clause
c) Returns the first alphabetically ordered value
d) Returns the first value that appears in a column, regardless of partitioning

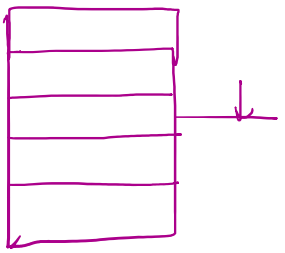


15. What does the LAST_VALUE() function do in SQL?

a) Returns the last row from a result set
b) Returns the last value in a partition based on the ORDER BY clause
c) Returns the highest value in a set of rows
d) Returns the last value that appears in a column, regardless of partitioning

16. How can you calculate a rolling average of the last 3 rows in a result set?

a) AVG() OVER (ROWS BETWEEN 2 PRECEDING AND CURRENT ROW)
b) AVG() OVER (ROWS BETWEEN 3 PRECEDING AND CURRENT ROW)
c) AVG() OVER (ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
d) AVG() OVER (PARTITION BY employee ORDER BY salary)



17. Which of the following window functions provides a relative rank between 0 and 1?

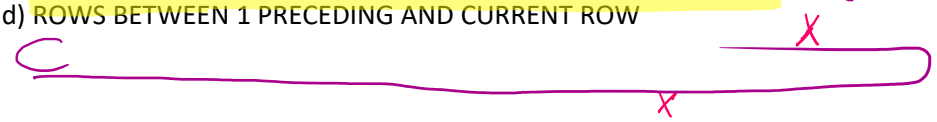
a) NTILE()
b) PERCENT_RANK()
c) CUME_DIST()
d) ROW_NUMBER()

18. What does CUME_DIST() return?

a) The relative rank of a row within a partition as a percentage
b) The cumulative distribution of a value within a set
c) The number of rows before the current row
d) The sum of the distribution values in a partition

19. What is the default window frame in a window function when ROWS is not specified?

a) ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
b) ROWS BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING
c) ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING
d) ROWS BETWEEN 1 PRECEDING AND CURRENT ROW



20. What will the following query return?

```
SELECT employee, salary,
SUM(salary) OVER (PARTITION BY department ORDER BY salary) as cumulative_salary
FROM employees;
```

a) The total salary of all employees
b) The cumulative salary per department, ordered by salary
c) The sum of all salaries ordered by department
d) The average salary per department