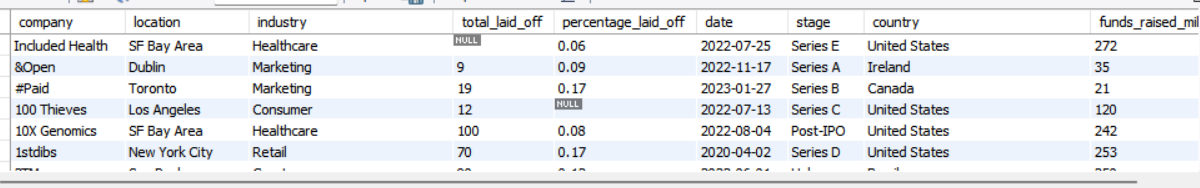
-- EDA: Use the layoff database

USE layoff\_db;

-- Display all records from layoff\_staging2

SELECT \* FROM layoff\_staging2;



-- Find the maximum values from total\_laid\_off and percentage\_laid\_off

SELECT

MAX(total\_laid\_off) AS max\_total\_laid\_off,

MAX(percentage\_laid\_off) AS max\_percentage\_laid\_off

FROM layoff\_staging2;



-- Subquery to get details where percentage\_laid\_off is maximum

SELECT \*

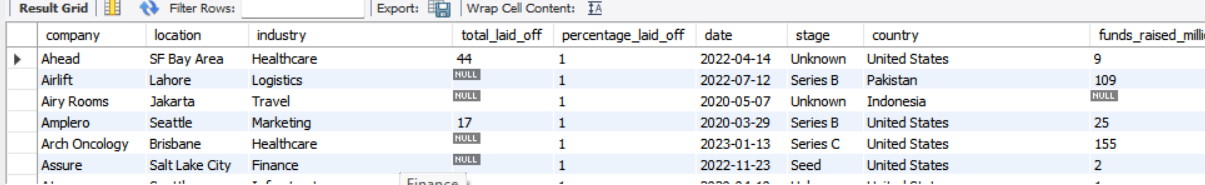
FROM layoff\_staging2

WHERE percentage\_laid\_off = (

SELECT MAX(percentage\_laid\_off)

FROM layoff\_staging2

);



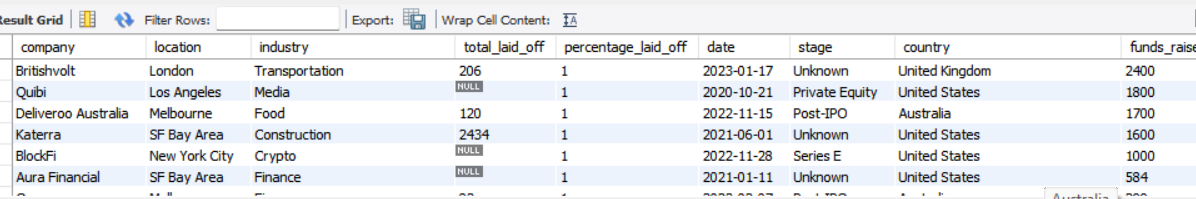
-- Get layoff details where percentage\_laid\_off is 100%, ordered by funds raised

SELECT \*

FROM layoff\_staging2

WHERE percentage\_laid\_off = 1

ORDER BY funds\_raised\_millions DESC;



-- Display all records again

SELECT \* FROM layoff\_staging2;

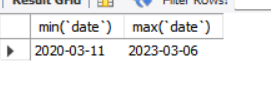
-- Find the date range of the records

SELECT

MIN(`date`) AS start\_date,

MAX(`date`) AS end\_date

FROM layoff\_staging2;



-- Company-wise total layoffs, ordered by total layoffs

SELECT

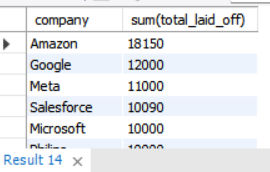
company,

SUM(total\_laid\_off) AS total\_laid\_off

FROM layoff\_staging2

GROUP BY company

ORDER BY total\_laid\_off DESC;



-- Industry-wise total layoffs, ordered by total layoffs

SELECT

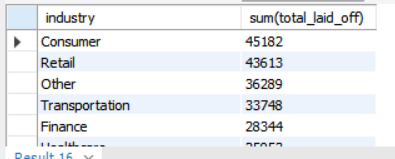
industry,

SUM(total\_laid\_off) AS total\_laid\_off

FROM layoff\_staging2

GROUP BY industry

ORDER BY total\_laid\_off DESC;



-- Country-wise total layoffs, ordered by total layoffs

SELECT

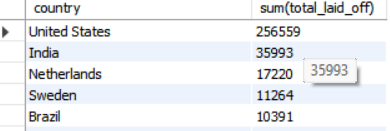
country,

SUM(total\_laid\_off) AS total\_laid\_off

FROM layoff\_staging2

GROUP BY country

ORDER BY total\_laid\_off DESC;



-- Year-wise total layoffs, ordered by total layoffs

SELECT

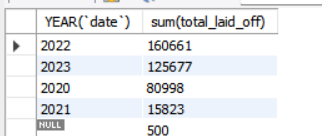
YEAR(`date`) AS year,

SUM(total\_laid\_off) AS total\_laid\_off

FROM layoff\_staging2

GROUP BY year

ORDER BY total\_laid\_off DESC;



-- Stage-wise total layoffs, ordered by total layoffs

SELECT

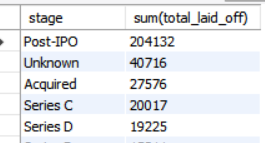
stage,

SUM(total\_laid\_off) AS total\_laid\_off

FROM layoff\_staging2

GROUP BY stage

ORDER BY total\_laid\_off DESC;



-- Company-wise average percentage layoffs, ordered by average percentage

SELECT

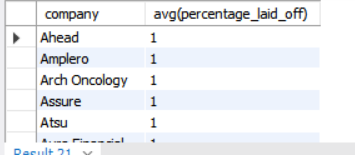
company,

AVG(percentage\_laid\_off) AS avg\_percentage\_laid\_off

FROM layoff\_staging2

GROUP BY company

ORDER BY avg\_percentage\_laid\_off DESC;



-- Stored procedure to get rolling sum of total\_laid\_off by date for a specific company

DELIMITER $$

CREATE PROCEDURE company\_layoff(IN com TEXT)

BEGIN

WITH cte\_rolling AS (

SELECT

company,

`date`,

total\_laid\_off,

SUM(total\_laid\_off) OVER (PARTITION BY company ORDER BY `date`) AS rolling\_sum

FROM layoff\_staging2

)

SELECT

company,

`date`,

total\_laid\_off,

rolling\_sum

FROM cte\_rolling

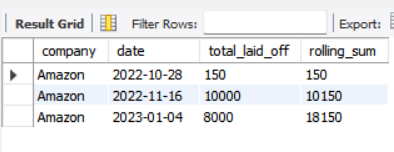
WHERE company = com;

END $$

DELIMITER ;

-- Call the stored procedure for Amazon

CALL company\_layoff('Amazon');



-- CTE to find month-wise rolling sum of total\_laid\_off

WITH cte\_rolling\_sum AS (

SELECT

SUBSTRING(`date`, 1, 7) AS `Month`,

SUM(total\_laid\_off) AS total\_laidoff

FROM layoff\_staging2

WHERE SUBSTRING(`date`, 1, 7) IS NOT NULL

GROUP BY `Month`

ORDER BY `Month`

)

SELECT

`Month`,

total\_laidoff,

SUM(total\_laidoff) OVER (ORDER BY `Month`) AS roll\_sum

FROM cte\_rolling\_sum;

