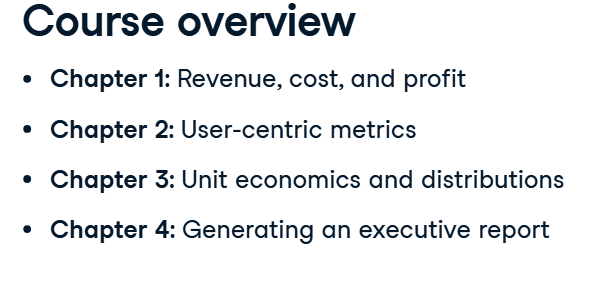
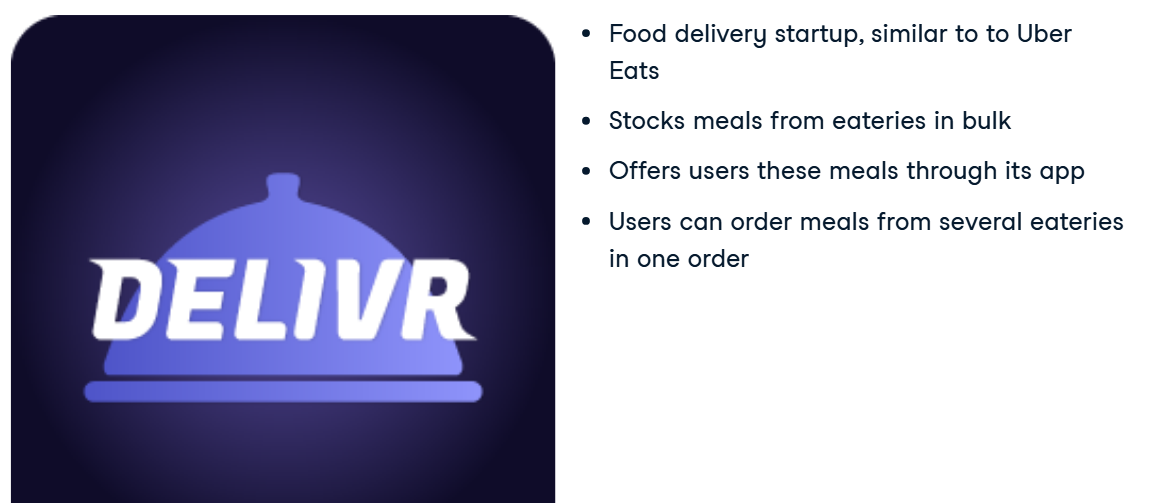
**ANALYZING BUSINESS DATA IN SQL**

# Course Overview

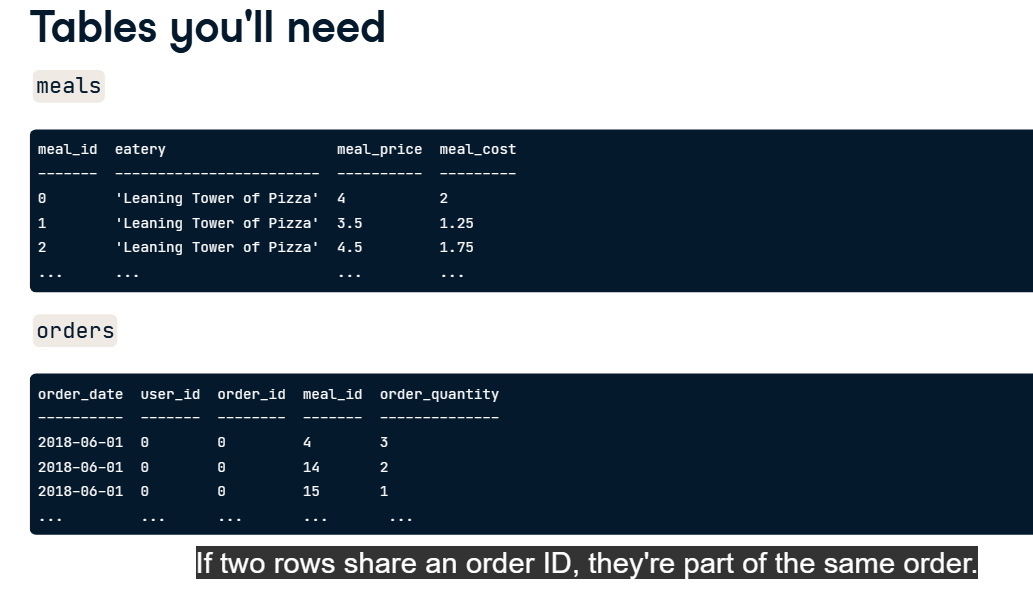




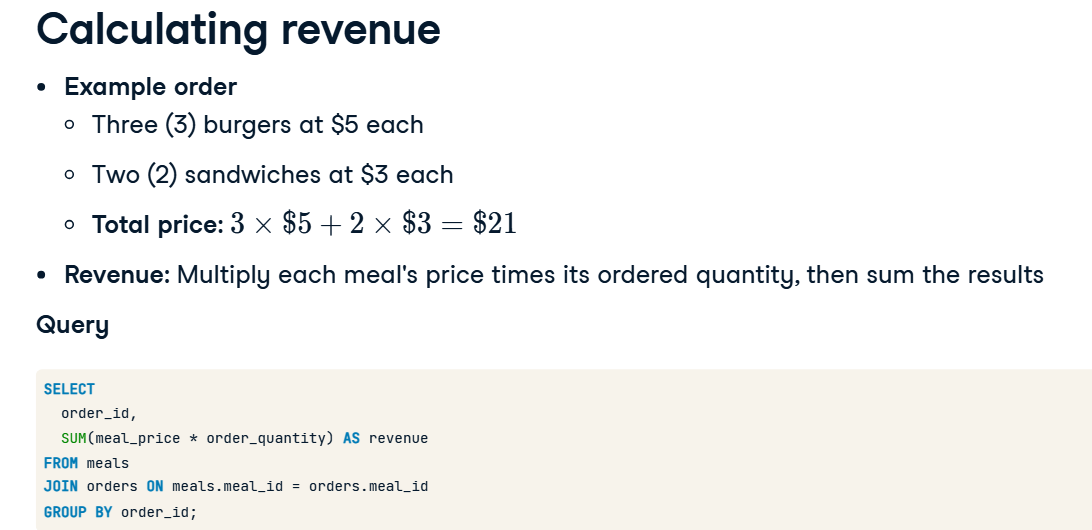
# Revenue, Cost and Profit



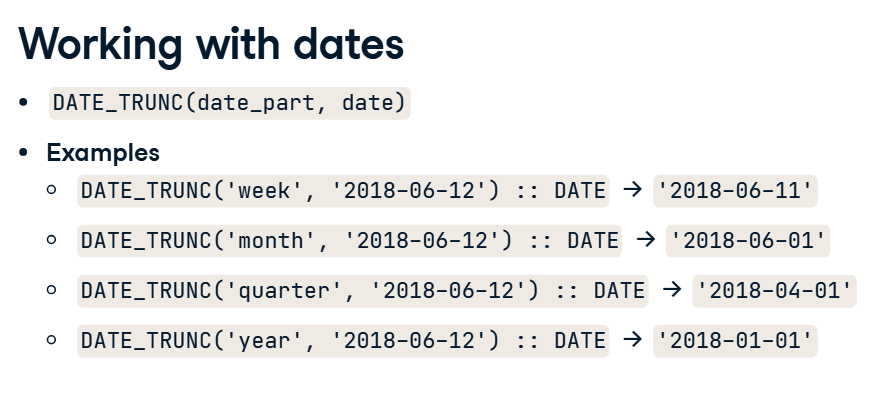
## Tables for Revenue



## Calculating Revenue



## Working with dates



## Calculate weekly revenue for June 2018

SELECT DATE\_TRUNC('week', order\_date) :: DATE AS delivr\_week,

       -- Calculate revenue

       SUM(meal\_price \* order\_quantity) AS revenue

  FROM meals

  JOIN orders ON meals.meal\_id = orders.meal\_id

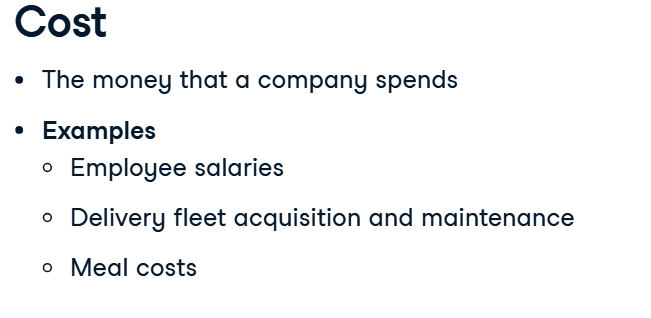
-- Keep only the records in June 2018

WHERE DATE\_TRUNC ('month', order\_date) = '2018-06-01'

GROUP BY delivr\_week

ORDER BY delivr\_week ASC;

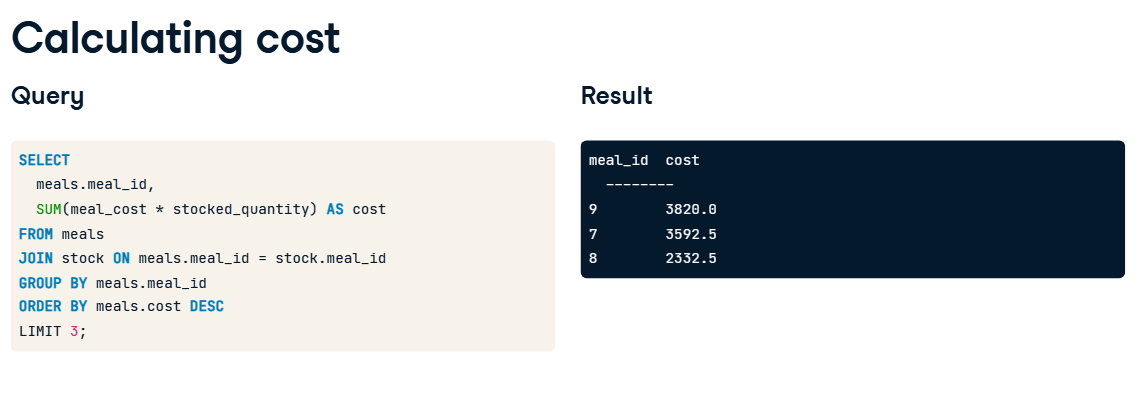
## Cost



## Tables for Cost



## Calculating Cost



## Combine Revenue and Cost - Common Table Expressions (CTEs)



## Query to calculate cost per month

SELECT

  -- Calculate cost

  DATE\_TRUNC('month', stocking\_date)::DATE AS delivr\_month,

  SUM (meal\_cost \* stocked\_quantity) AS cost

FROM meals

JOIN stock ON meals.meal\_id = stock.meal\_id

GROUP BY delivr\_month

ORDER BY delivr\_month ASC;

## How much Delivr spent per month on average during its early months (before September 2018)

-- Declare a CTE named monthly\_cost

WITH monthly\_cost AS (

  SELECT

    DATE\_TRUNC('month', stocking\_date)::DATE AS delivr\_month,

    SUM(meal\_cost \* stocked\_quantity) AS cost

  FROM meals

  JOIN stock ON meals.meal\_id = stock.meal\_id

  GROUP BY delivr\_month)

SELECT

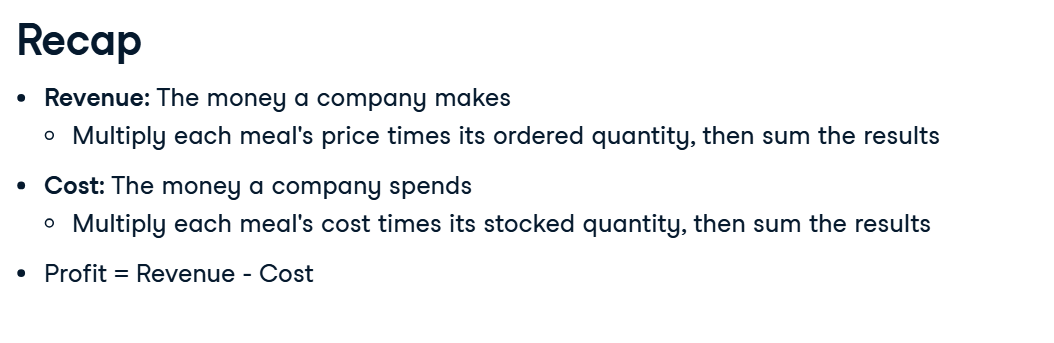
  -- Calculate the average monthly cost before September

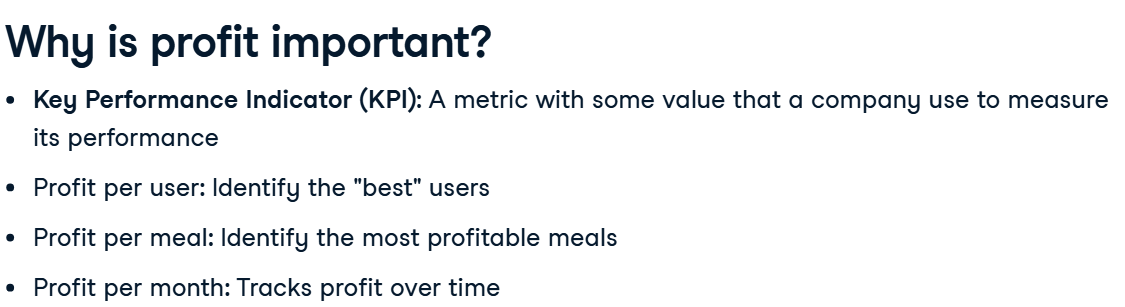
  AVG(cost)

FROM monthly\_cost

WHERE delivr\_month < '2018-09-01';

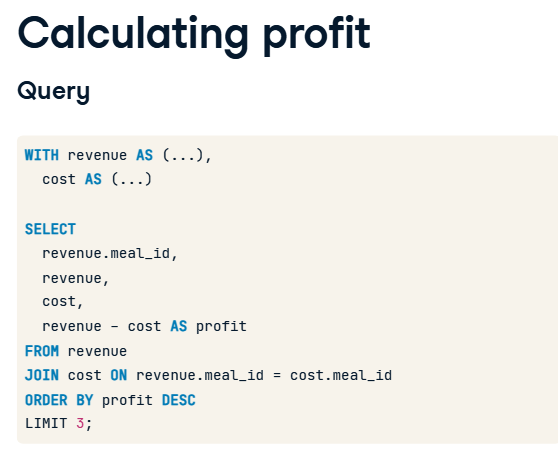
## Profit







**If you're calculating profit per some other metric, like eatery or month, make sure to group by that metric in both CTEs, since you'll join the two CTEs on that metric's column later on.**

****

## How much profit each eatery is generating

WITH revenue AS (

  -- Calculate revenue per eatery

  SELECT eatery,

         SUM(meal\_price \* order\_quantity) AS revenue

    FROM meals

    JOIN orders ON meals.meal\_id = orders.meal\_id

   GROUP BY eatery),

  cost AS (

  -- Calculate cost per eatery

  SELECT eatery,

         SUM(meal\_cost \* stocked\_quantity) AS cost

    FROM meals

    JOIN stock ON meals.meal\_id = stock.meal\_id

   GROUP BY eatery)

   -- Calculate profit per eatery

   SELECT revenue.eatery,

          revenue - cost AS profit

     FROM revenue

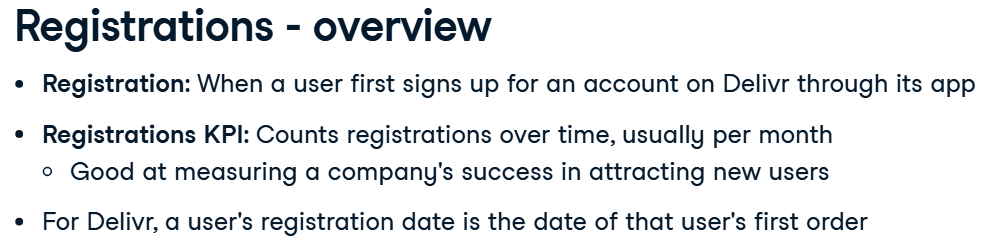
     JOIN cost ON revenue.eatery = cost.eatery

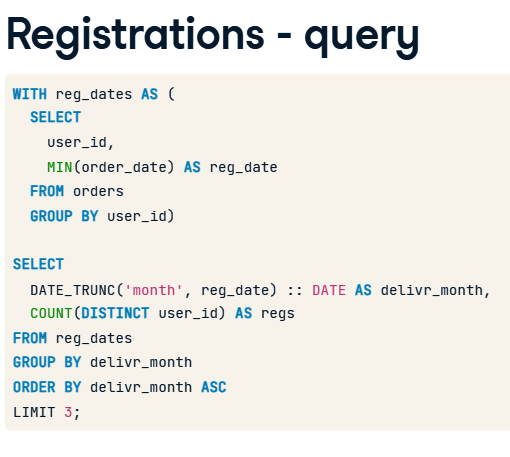
    ORDER BY profit DESC;

# User-centric metrics

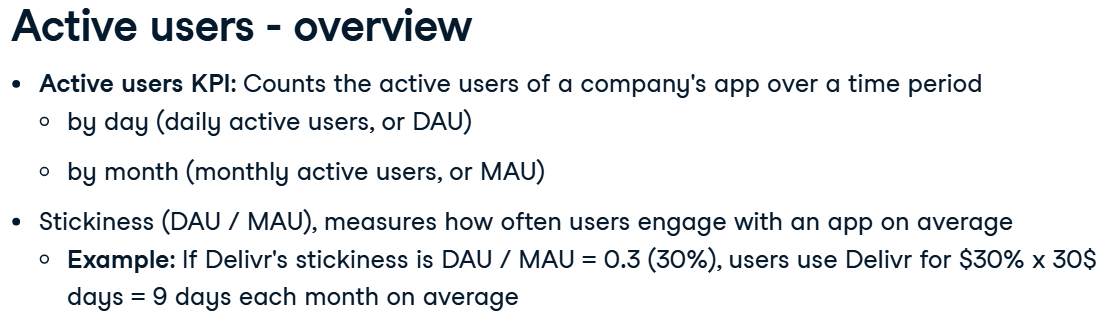


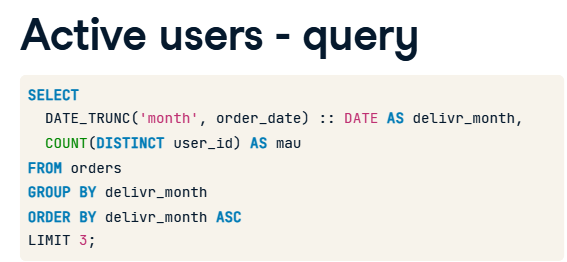
## Registrations



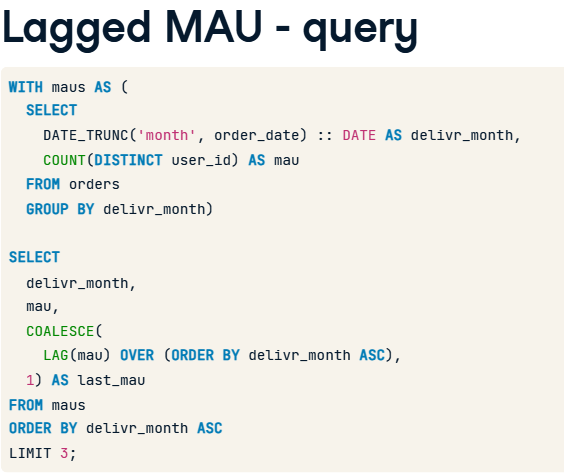


## Active users

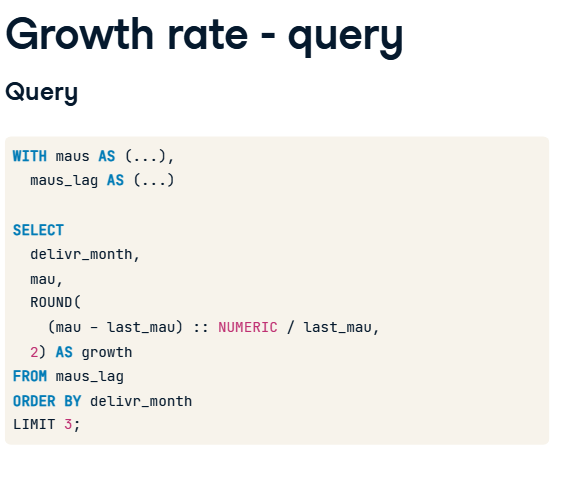
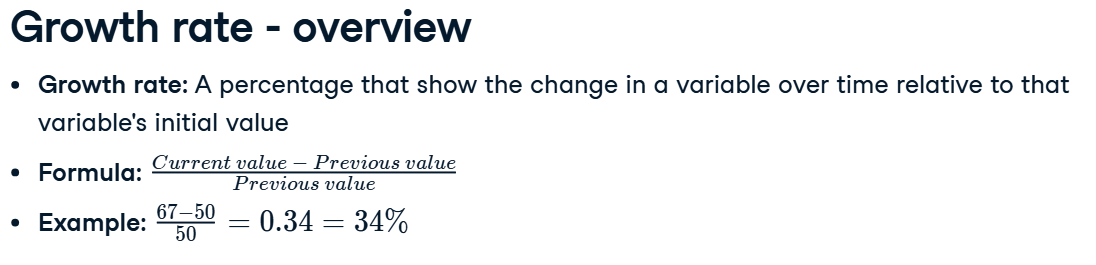
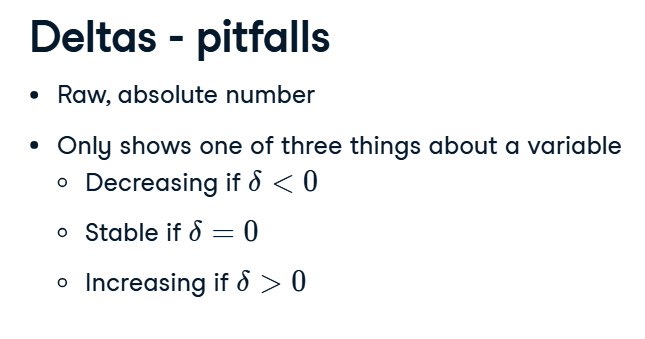




## Window Functions – Running Total and Lag



## Growth Rate and Delta



## Return a table of MoM order growth rates.

WITH orders AS (

  SELECT

    DATE\_TRUNC('month', order\_date) :: DATE AS delivr\_month,

    --  Count the unique order IDs

    COUNT(DISTINCT order\_id) AS orders

  FROM orders

  GROUP BY delivr\_month),

  orders\_with\_lag AS (

  SELECT

    delivr\_month,

    -- Fetch each month's current and previous orders

    orders,

    COALESCE(

      LAG(orders) OVER (ORDER BY delivr\_month ASC),

    1) AS last\_orders

  FROM orders)

SELECT

  delivr\_month,

  -- Calculate the MoM order growth rate

  ROUND(

    (orders - last\_orders) :: NUMERIC / last\_orders,

  2) AS growth

FROM orders\_with\_lag

ORDER BY delivr\_month ASC;