# DATE\_TRUNC

SELECT DATE\_TRUNC('week', order\_date) :: DATE AS delivr\_week, SUM(meal\_price \* order\_quantity) AS revenue

FROM meals

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

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

GROUP BY delivr\_week

ORDER BY delivr\_week ASC;

# CTE - Common Table Expressions

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 avg(cost)

FROM monthly\_cost

WHERE delivr\_month < '2018-09-01';

# Window functions

# Running Total

WITH reg\_dates AS (SELECT user\_id, MIN(order\_date) AS reg\_date

FROM orders

GROUP BY user\_id),

regs AS (SELECT DATE\_TRUNC('month', reg\_date) :: DATE AS delivr\_month, COUNT(DISTINCT user\_id) AS regs

FROM reg\_dates

GROUP BY delivr\_month)

SELECT delivr\_month, sum(regs)OVER(order by delivr\_month ASC) AS regs\_rt

FROM regs

ORDER BY delivr\_month ASC;

# Window functions

# Lag

WITH mau AS (SELECT DATE\_TRUNC('month', order\_date) :: DATE AS delivr\_month, COUNT(DISTINCT user\_id) AS mau

FROM orders

GROUP BY delivr\_month)

SELECT delivr\_month, mau, COALESCE(LAG(mau) over(order by delivr\_month ASC),0) AS last\_mau

FROM mau

ORDER BY delivr\_month ASC;

#Growth Rate

WITH mau AS (SELECT DATE\_TRUNC('month', order\_date) :: DATE AS delivr\_month, COUNT(DISTINCT user\_id) AS mau

FROM orders

GROUP BY delivr\_month),

mau\_with\_lag AS (SELECT delivr\_month, mau, GREATEST(LAG(mau) OVER (ORDER BY delivr\_month ASC), 1) AS last\_mau

FROM mau)

SELECT delivr\_month, ROUND((mau - last\_mau)::NUMERIC / last\_mau,2) AS growth

FROM mau\_with\_lag

ORDER BY delivr\_month asc;

#Retention Rate

WITH user\_monthly\_activity AS (SELECT DISTINCT DATE\_TRUNC('month', order\_date) :: DATE AS delivr\_month, user\_id

FROM orders)

SELECT previous.delivr\_month,

ROUND(count(distinct current.user\_id) :: numeric /greatest(count(distinct previous.user\_id), 1),2) AS retention\_rate

FROM user\_monthly\_activity AS previous

LEFT JOIN user\_monthly\_activity AS current

ON previous.user\_id = current.user\_id

AND previous.delivr\_month = (current.delivr\_month - INTERVAL '1 month')

GROUP BY previous.delivr\_month

ORDER BY previous.delivr\_month ASC;