-- LESSON 14 - SUBQUERIES

-- Slide 15 - Subquery in FROM - Inner Query (97 rows)

SELECT p.name,

CASE

WHEN p.name ILIKE '%Mountain%' THEN 'mountain'

WHEN p.name ILIKE '%Road%' THEN 'road'

WHEN p.name ILIKE '%Touring%' THEN 'touring'

ELSE 'unknown'

END AS bike\_type, p.weight

FROM production.product AS p

WHERE p.productnumber ILIKE 'BK%';

-- Slide 16 - Full Query (3 rows)

SELECT subq.bike\_type, ROUND(AVG(subq.weight),2) AS avgweight

FROM(

SELECT p.name,

CASE

WHEN p.name ILIKE '%Mountain%' THEN 'mountain'

WHEN p.name ILIKE '%Road%' THEN 'road'

WHEN p.name ILIKE '%Touring%' THEN 'touring'

ELSE 'unknown'

END AS bike\_type,

p.weight

FROM production.product AS p

WHERE p.productnumber ILIKE 'BK%'

) AS subq

GROUP BY subq.bike\_type

ORDER BY avgweight ASC;

-- Slide 19 - Subquery using IN function - Inner Query (97 rows)

SELECT p.productid

FROM production.product AS p

WHERE p.productnumber ILIKE 'BK%';

-- Slide 20 - Full Query (97 rows)

SELECT p.name, od.productid, SUM(od.orderqty) as qtyordered

FROM sales.salesorderdetail AS od

LEFT JOIN production.product AS p

ON p.productid = od.productid

WHERE od.productid IN (

SELECT p.productid

FROM production.product AS p

WHERE p.productnumber ILIKE 'BK%')

GROUP BY p.name, od.productid

ORDER BY qtyordered DESC;

-- Slides 21-22 - Subquery using NOT IN (169 rows)

SELECT p.name, od.productid, SUM(od.orderqty) as qtyordered

FROM sales.salesorderdetail AS od

LEFT JOIN production.product AS p

ON p.productid = od.productid

WHERE od.productid NOT IN (

SELECT p.productid

FROM production.product AS p

WHERE p.productnumber ILIKE 'BK%')

GROUP BY p.name, od.productid

ORDER BY qtyordered DESC;

-- Slide 25 - Inner Query (504 rows)

SELECT p.listprice,

CASE

WHEN p.listprice <= 0 THEN 'free'

WHEN p.listprice <= 50 THEN 'cheap'

WHEN p.listprice <= 300 THEN 'affordable'

WHEN p.listprice > 300 THEN 'expensive'

END AS affordability

FROM production.product AS p;

-- Slide 26 - full query (4 rows)

SELECT subq.affordability, COUNT(subq.listprice) AS numberofproducts

FROM

(SELECT p.listprice,

CASE

WHEN p.listprice <= 0 THEN 'free'

WHEN p.listprice <= 50 THEN 'cheap'

WHEN p.listprice <= 300 THEN 'affordable'

WHEN p.listprice > 300 THEN 'expensive'

END AS affordability

FROM production.product AS p

) AS subq

GROUP BY subq.affordability

ORDER BY numberofproducts DESC;

-- Slide 27 inner query (504 rows)

SELECT p.listprice,

CASE

WHEN p.listprice = 0 THEN 1

ELSE 0

END AS isfree

FROM production.product AS p;

-- Slide 28 full query (39.7%)

SELECT ROUND(AVG(subq.isfree)\*100, 1) AS isfreepercentage

FROM

(SELECT p.listprice,

CASE

WHEN p.listprice = 0 THEN 1

ELSE 0

END AS isfree

FROM production.product AS p) AS subq