Joins, summeries & subqueries afleveringsløsninger

Joins & unions

Opgave 1

```
SELECT last_name, first_name, order_date, product_name, item_price,
discount_amount, quantity

FROM customers c

  JOIN orders o ON c.customer_id = o.customer_id

  JOIN order_items oi ON o.order_id = oi.order_id

  JOIN products p ON oi.product_id = p.product_id

ORDER BY last_name, order_date, product_name
```

Opgave 2

```
SELECT p1.product_name, p1.list_price

FROM products p1 JOIN products p2

    ON p1.product_id <> p2.product_id

    AND p1.list_price = p2.list_price

ORDER BY p1.product_name
```

Opgave 3

```
SELECT c.category_name, p.product_id

FROM categories c LEFT JOIN products p

  ON c.category_id = p.category_id

WHERE p.product_id IS NULL;
```

Opgave 4

```
  SELECT 'SHIPPED' AS ship_status, order_id, order_date

  FROM orders
```

```
  WHERE ship_date IS NOT NULL
UNION
  SELECT 'NOT SHIPPED', order_id, order_date
  FROM orders
  WHERE ship_date IS NULL
ORDER BY order_date
```

Summeries

Opgave 5

```
SELECT email_address, COUNT(distinct o.order_id) AS order_count,
  SUM((item_price - discount_amount) * quantity) AS order_total
FROM customers c
  JOIN orders o ON c.customer_id = o.customer_id
  JOIN order_items oi ON o.order_id = oi.order_id
GROUP BY email_address
HAVING order_count > 1
ORDER BY order_total DESC
```

Opgave 6

```
SELECT email_address, COUNT(distinct o.order_id) AS order_count,
  SUM((item_price - discount_amount) * quantity) AS order_total
FROM customers c
  JOIN orders o ON c.customer_id = o.customer_id
  JOIN order_items oi ON o.order_id = oi.order_id
WHERE item_price > 400
GROUP BY email_address
```

```
HAVING order_count > 1

ORDER BY order_total DESC;
```

Opgave 7

```
SELECT product_name, SUM((item_price - discount_amount) * quantity) AS
product_total

FROM products p

 JOIN order_items oi ON p.product_id = oi.product_id

GROUP BY product_name WITH ROLLUP
```

Opgave 8

```
SELECT email_address,

    COUNT(DISTINCT oi.product_id) AS number_of_products

FROM customers c

 JOIN orders o ON c.customer_id = o.customer_id

 JOIN order_items oi ON o.order_id = oi.order_id

GROUP BY email_address

HAVING number_of_products > 1

ORDER BY email_address
```

Subqueries

Opgave 9

```
SELECT email_address,  MAX(order_total) AS max_order_total

FROM

(

 SELECT email_address, o.order_id, SUM((item_price - discount_amount) *
quantity) AS order_total
```

```
 FROM customers c
   JOIN orders o ON c.customer_id = o.customer_id
   JOIN order_items oi ON o.order_id = oi.order_id
 GROUP BY email_address, o.order_id
) t
GROUP BY email_address
```

Opgave 10

```
SELECT product_name, discount_percent
FROM products
WHERE discount_percent NOT IN (
   SELECT discount_percent
   FROM products
   GROUP BY discount_percent
   HAVING count(discount_percent) > 1)
ORDER BY product_name;
```

Opgave 11

```
SELECT email_address, order_id, order_date
FROM customers c
 JOIN orders o ON c.customer_id = o.customer_id
WHERE order_date =
 (SELECT MIN(order_date)
  FROM orders
  WHERE customer_id = o.customer_id)
```