1) Write an SQL statement to create a view that will be updated daily showing all the orders for the current week including products, clients, payments and status details.

Solution

```
CREATE VIEW daily_orders_report AS
SELECT
  o.order_id,
  p.product_name,
  c.client_name,
  p.payment_amount,
  s.status_details
FROM
  orders o
JOIN
  products p ON o.product_id = p.product_id
JOIN
  clients c ON o.client_id = c.client_id
JOIN
  payments pa ON o.payment_id = pa.payment_id
JOIN
  status s ON o.status_id = s.status_id
WHERE
  o.order_date >= CURDATE() - INTERVAL WEEKDAY(CURDATE()) DAY;
```

2) Additionally, create a second view where we can visualize a weekly report with 7 columns, each column will be labeled with the day and date, and each row will show a concatenation of these elements and they will be separated by a double colon ::

    - order_uuid
    - number of products
    - order amount in cents

Solution

```
CREATE VIEW weekly_order_report AS
SELECT
  MAX(CASE WHEN DAYOFWEEK(created_on) = 1 THEN CONCAT('Sunday ',
DATE_FORMAT(created_on, '%d/%m/%Y')) END) AS Sunday,
  MAX(CASE WHEN DAYOFWEEK(created_on) = 2 THEN CONCAT('Monday ',
DATE_FORMAT(created_on, '%d/%m/%Y')) END) AS Monday,
  MAX(CASE WHEN DAYOFWEEK(created_on) = 3 THEN CONCAT('Tuesday ',
DATE_FORMAT(created_on, '%d/%m/%Y')) END) AS Tuesday,
  MAX(CASE WHEN DAYOFWEEK(created_on) = 4 THEN CONCAT('Wednesday ',
DATE_FORMAT(created_on, '%d/%m/%Y')) END) AS Wednesday,
  MAX(CASE WHEN DAYOFWEEK(created_on) = 5 THEN CONCAT('Thursday ',
DATE_FORMAT(created_on, '%d/%m/%Y')) END) AS Thursday,
  MAX(CASE WHEN DAYOFWEEK(created_on) = 6 THEN CONCAT('Friday ',
DATE_FORMAT(created_on, '%d/%m/%Y')) END) AS Friday,
  MAX(CASE WHEN DAYOFWEEK(created_on) = 7 THEN CONCAT('Saturday ',
DATE_FORMAT(created_on, '%d/%m/%Y')) END) AS Saturday,
  GROUP_CONCAT(CONCAT_WS('::', order_uuid, num_products, order_amount_cents)
SEPARATOR '::') AS order_info
FROM your_table_name
GROUP BY WEEK(created_on)
ORDER BY WEEK(created_on) ASC;
```