

## Task 1 – SQL Challenge

This notebook contains the three SQL queries for the Wolt Data Science Internship assignment, structured as string outputs for easy review.

```
query1 = ''
-- Query 1: Number of users registered in Finland in the last 30 days
SELECT COUNT(*) AS users_last_30_days
FROM Users
WHERE User_country = 'Finland'
      AND User_registration_timestamp_utc >= NOW() - INTERVAL '30 days';
'''
print(query1)
```



```
-- Query 1: Number of users registered in Finland in the last 30 days
SELECT COUNT(*) AS users_last_30_days
FROM Users
WHERE User_country = 'Finland'
      AND User_registration_timestamp_utc >= NOW() - INTERVAL '30 days';
```

```
query2 = ''
-- Query 2: Users with at least one purchase in the past 30 days and more than one unique product
SELECT COUNT(DISTINCT s.User_id) AS users_multi_product
FROM Sales s
JOIN Purchases p ON s.Purchase_id = p.Purchase_id
WHERE s.Timestamp_utc >= NOW() - INTERVAL '30 days'
GROUP BY s.User_id
HAVING COUNT(DISTINCT p.Product_id) > 1;
'''
print(query2)
```



```
-- Query 2: Users with at least one purchase in the past 30 days and more than one unique product
SELECT COUNT(DISTINCT s.User_id) AS users_multi_product
FROM Sales s
JOIN Purchases p ON s.Purchase_id = p.Purchase_id
WHERE s.Timestamp_utc >= NOW() - INTERVAL '30 days'
GROUP BY s.User_id
HAVING COUNT(DISTINCT p.Product_id) > 1;
```

```
query3_method1 = ''
-- Query 3: Most recent price for each purchased product (Method 1 - using window functions)
SELECT Product_id, Price
FROM (
    SELECT Product_id, Price,
           ROW_NUMBER() OVER (PARTITION BY Product_id ORDER BY Purchase_id DESC) AS rn
    FROM Purchases
) sub
WHERE rn = 1;
'''
print(query3_method1)
```



```
-- Query 3: Most recent price for each purchased product (Method 1 - using window functions)
SELECT Product_id, Price
FROM (
    SELECT Product_id, Price,
           ROW_NUMBER() OVER (PARTITION BY Product_id ORDER BY Purchase_id DESC) AS rn
    FROM Purchases
) sub
WHERE rn = 1;
```

```
query3_method2 = ''
-- Method 2 - using JOIN
SELECT p.Product_id, p.Price
FROM Purchases p
JOIN (
    SELECT Product_id, MAX(Purchase_id) AS MaxPurchase
    FROM Purchases
    GROUP BY Product_id
) recent ON p.Product_id = recent.Product_id AND p.Purchase_id = recent.MaxPurchase;
'''
print(query3_method2)
```



```
-- Method 2 - using JOIN
SELECT p.Product_id, p.Price
FROM Purchases p
JOIN (
    SELECT Product_id, MAX(Purchase_id) AS MaxPurchase
    FROM Purchases
    GROUP BY Product_id
) recent ON p.Product_id = recent.Product_id AND p.Purchase_id = recent.MaxPurchase;
```

