Correlated Subqueries

Topic Cheat Sheet

DATABASE STAR

What is it?

A correlated subquery is a subquery that references a column from the outer query.

How It Works

```
SELECT customerid, customer_name,
(
  SELECT MAX(order_date)
  FROM orders o
  WHERE o.customer_id = c.customer_id
) AS last_order
FROM customers c;
```

Correlated subquery

Common Use Cases

- Latest record per group (e.g. most recent order per customer)
- Aggregate Filter (e.g. customers with above-average spend)
- Existence check (with EXISTS or NOT EXISTS)

Performance Impact

The subquery executes for every outer query row:
10,000 rows x 1 subquery row
= 10,000 executions
OK for small tables. Avoid on large tables.

Faster Alternatives

Latest Record Per Group

Correlated (Slow)

```
SELECT c.name,
  (SELECT MAX(o.dt)
   FROM orders o
   WHERE o.cust_id = c.cust_id)
FROM customers c;
```

Wndow function (fast)

```
SELECT name, dt AS last_order
FROM (
  SELECT c.name, o.dt,
    ROW_NUMBER() OVER (
      PARTITION BY o.cust_id
      ORDER BY o.dt DESC) AS rn
  FROM customers c
  JOIN orders o
    ON o.cust_id = c.cust_id
) t
WHERE rn = 1;
```

Above Average Filter

Correlated (Slow)

```
SELECT * FROM orders o
WHERE o.amount >
  (SELECT AVG(amount)
   FROM orders o2
   WHERE o2.cust_id
         = o.cust_id);
```

Derived Table (fast)

```
SELECT o.* FROM orders o
JOIN (
  SELECT cust_id,
    AVG(amount) AS avg_amt
  FROM orders
  GROUP BY cust_id
) a ON a.cust_id = o.cust_id
WHERE o.amount > a.avg_amt;
```

What About EXISTS?

EXISTS stops as soon as it finds one match, which is often faster than a correlated subquery. It's usually OK, even on larger tables