

# SQL Subquery Review

Database Development 2021  
Cody Henrichsen

---

---

---

---

---

---

---

## Statement Order

- |                    |                    |
|--------------------|--------------------|
| ▷ Written          | ▷ Executed         |
| ▷ SELECT           | ▷ FROM/JOIN ... ON |
| ▷ FROM/JOIN ... ON | ▷ WHERE            |
| ▷ WHERE            | ▷ GROUP BY         |
| ▷ GROUP BY         | ▷ HAVING           |
| ▷ HAVING           | ▷ SELECT           |
| ▷ ORDER BY         | ▷ ORDER BY         |

---

---

---

---

---

---

---

## Subquery

- ▷ A complete query surrounded by ()
- ▷ Should be aliased
- ▷ Can go basically anywhere
  - ▷ Most common in
    - FROM/JOIN
    - SELECT
    - WHERE

---

---

---

---

---

---

---

### Subquery Types

- ▶ Scalar
  - ▶ Returns a table with a single value
- ▶ Multirow
  - ▶ Returns a table with multiple rows
- ▶ Correlated
  - ▶ References outer table in the subquery
  - ▶ Executed for each row of the table
- ▶ Inline View
  - ▶ In the FROM/JOIN clause

---

---

---

---

---

---

---

### Scalar

- ▶ Used to answer a single question
- ▶ Find all the results WHERE some field is (>, <, =) to the (AVG/SUM/MIN/MAX) of some column
- ▶ Calculated value in the SELECT clause
- ▶ Correlated queries CAN be scalar

---

---

---

---

---

---

---

### Multi-row

- ▶ Used to match many values
- ▶ Match results with IN, ANY, SOME, ALL
- ▶ No symbol operators allowed
  - ▶ ...
  - ▶ WHERE
    - field\_name IN
    - (
      - Subquery
    - )

---

---

---

---

---

---

---

### Multi-row and Multi-column

- ▶ If the subquery also has multiple columns group fields in parens and separated by commas
  - ▷ ...
  - ▷ WHERE
    - (field\_one, field\_n) IN
    - (
      - Subquery)
    - )

---

---

---

---

---

---

---

### ANY/SOME

- ▶ Used in conjunction with the relational operators
  - ▷ <, >, <=, >=
- ▶ Same operation different name
  - ▷ Pick one and stick with it
- ▶ Checks if the field and operator matches against ANY of the results from the subquery
  - ▷ field\_name operator ANY (subquery)
    - val.score < ANY ( subquery)
    - demo.value >= ANY (other subquery)

---

---

---

---

---

---

---

### ALL

- ▶ Used in conjunction with the relational operators
  - ▷ <, >, <=, >=
- ▶ Checks if the field and operator matches against ALL of the results from the subquery
  - ▷ field\_name operator ALL (subquery)
    - val.score < ALL ( subquery)

---

---

---

---

---

---

---

## Correlated

- ▶ Unlike others, this subquery executes for EVERY ROW of the outer query
  - ▷ Kinda like a nested for loop
- ▶ Easy to visually identify by the reference of the outer table in the subquery
- ▶ Subquery CANNOT run on its own because of outer/inner comparison

---

---

---

---

---

---

---

## Correlated Annotation

```
SELECT
  exp.exporter_name,
  (
    SELECT
      COUNT(*)
    FROM
      sales s
    WHERE
      s.export_id = exp.export_destination_code
  ) AS sales_count
FROM
  exporters exp
```

**Correlated Subquery**

subquery reference

Outer table reference

---

---

---

---

---

---

---

## Inline View

- ▶ The MOST POWERFUL subquery of them all!
- ▶ Any subquery that is in the FROM/JOIN clauses
- ▶ These should always be aliased for ease of use
- ▶ Use AS for column names to ease disambiguation
- ▶ Extremely helpful for complex questions
  - ▷ Build custom structures for filtering results

---

---

---

---

---

---

---