Subqueries

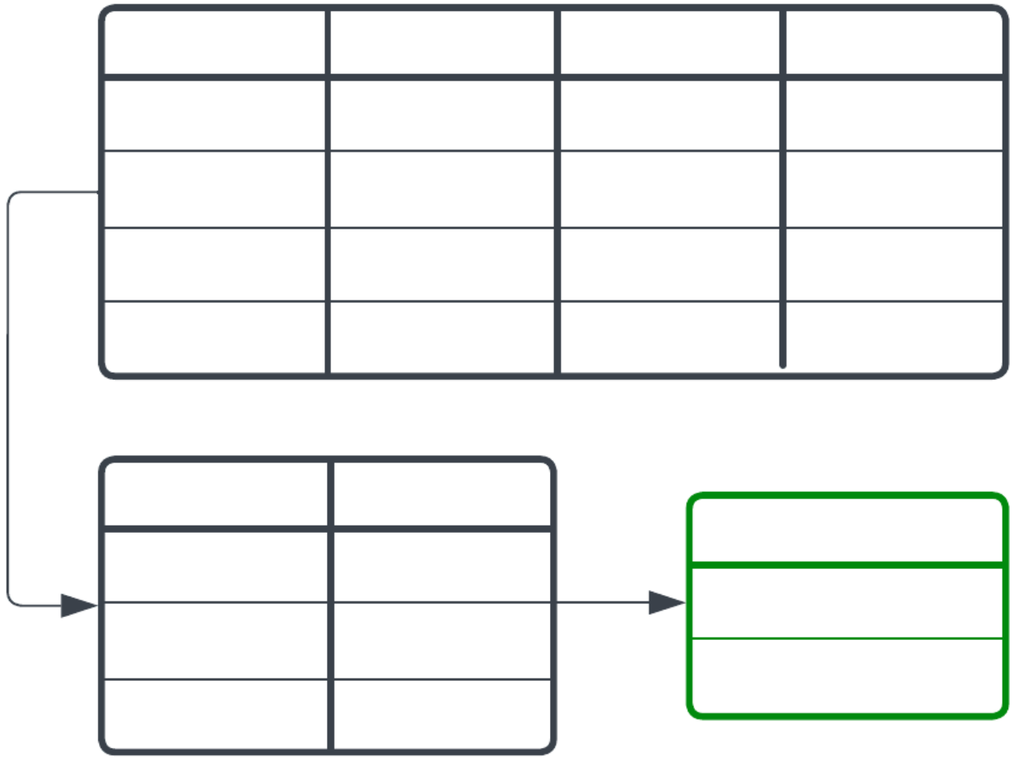
DATA MANIPULATION IN SNOWFLAKE

Jake Roach
Field Data Engineer

datacamp

What are subqueries?

Subqueries are a tool that allow for the result of one query to be used by another query.



- Combine multiple queries
- Focus on readability
- Allow for modularity
- Make data manipulation easier!

`FROM ( ... )` or `WHERE ... IN ( ... )`

Subqueries and FROM

```
SELECT
    ...
-- Pull from query, not from a table
FROM (

    -- Create a result set that will
    -- be used by the main query
    SELECT
        <fields>
    FROM <table>
    WHERE ...

);
```

Retrieve data from the result of another query rather than pulling directly from a table.

- Decouple manipulation from analysis
- Makes queries easier to read, understand
- Allows for "portability" and easier changes
- `JOIN`, `WHERE`, etc.

Before a subquery

```
SELECT
    month_num,
    -- windchill - temperature has to be used twice here. What if this changes?
    AVG(windchill - temperature) AS avg_differential
    MIN(windchill - temperature) AS most_differential
FROM weather
WHERE
    -- Filtering is taking place in the same query as aggregation/analysis
    season = 'Winter' AND
    temperature < 32
GROUP BY month_num;
```

After a subquery

```
-- Start with the subquery, then aggregate
SELECT
    month_num,
    AVG(differential) AS avg_differential
    MIN(differential) AS most_differential
FROM (

    SELECT
        month_num,
        windchill - temperature AS differential
    FROM weather
    WHERE
        season = 'Winter' AND
        temperature < 32

)
GROUP BY month_num;
```

```
| month_num | differential |
| --------- | ------------ |
|     12    |      -12     |
|     1     |      -3      |
|     1     |       0      |
|     2     |      -7      |
```

```
| month_num | avg_differential | most_differential |
| --------- | ---------------- | ----------------- |
|     12    |       -5.77      |        -14        |
|     1     |       -1.91      |        -8         |
|     2     |       -8.13      |        -22        |
```

It's easy to understand and change the analysis once the data is cleaned.

WHERE ... IN (...)

```
...

-- Filter by records with a value in
-- the subquery result set
WHERE <field> IN (

    SELECT <other-field> FROM ...

);
```

Create a small result set to be used when transforming, filtering, or manipulating data.

- Filter for records `IN` a non-constant set of results
- Can be used in other places in a query
- `AVG`, `MAX`, `MIN`, etc.

WHERE ... IN(...)

```
SELECT
    todays_date,
    temperature,
    status
FROM weather
WHERE todays_date IN (  -- Filter by all days with home games that were won

    SELECT
        game_date
    FROM game_schedule
    WHERE stadium = 'Home' AND did_win = TRUE

);
```

WHERE ... IN (...)

todays_date	temperature	status
2025-04-01	51	Rainy, Windy
2025-05-14	67	Sunny
2025-06-13	74	Thunder Storms, Windy

Let's practice!

DATA MANIPULATION IN SNOWFLAKE

Common Table Expressions

DATA MANIPULATION IN SNOWFLAKE

Jake Roach
Field Data Engineer

datacamp

Common Table Expressions

Common Table Expressions (CTE's) temporarily store the results of a query to eventually be used by another query

- CTE's are defined at the beginning of a query
- `WITH <cte-name> AS ( <query> )`
- Similar to subqueries

```
-- Pass the name of the CTE, followed
-- by the query in parenthesis
WITH <cte-name> AS (
    <query>
)

SELECT
    ...
FROM <cte-name>
... ;
```

Reporting on at-risk students

```
WITH at_risk AS (
    SELECT
        student_id
        course_name,
        teacher_name,
        grade
    FROM student_courses
    WHERE grade < 70 AND is_required
)

SELECT
    students.student_name,
    at_risk.*
FROM at_risk
JOIN students ON at_risk.student_id = students.id;
```

- Temporary results stored in `at_risk`
- Select a subset of records
- Query `at_risk` to generate a report

Reporting on at-risk students

student_id	course_name	teacher_name	grade
821930	Algebra	Mrs. Walker	61
636133	Biology	Mr. Casey	67
097165	History	Ms. Grimes	59

student_name	student_id	course_name	teacher_name	grade
L. Holt	821930	Algebra	Mrs. Walker	61
J. Barnes	636133	Biology	Mr. Casey	67
E. Yang	097165	History	Ms. Grimes	59

Comparing Subqueries and CTE's

```
SELECT
    month_num,
    AVG(differential) AS avg_differential
    MIN(differential) AS most_differential
FROM (

    SELECT
        month_num,
        windchill - temperature AS differential
    FROM weather
    WHERE
        season = 'Winter' AND
        temperature < 32

)
GROUP BY month_num;
```

```
WITH daily_temperature_differential AS (
    SELECT
        month_num,
        windchill - temperature AS differential
    FROM weather
    WHERE
        season = 'Winter' AND
        temperature < 32
)

SELECT
    month_num,
    AVG(differential) AS avg_differential
    MIN(differential) AS most_differential
FROM daily_temperature_differential
GROUP BY month_num;
```

Finding temperature differential

```
WITH daily_temperature_differential AS (
    SELECT
        month_num,
        windchill - temperature AS differential
    FROM weather
    WHERE
        season = 'Winter' AND
        temperature < 32
)

SELECT
    month_num,
    AVG(differential) AS avg_differential
    MIN(differential) AS most_differential
FROM daily_temperature_differential
GROUP BY month_num;
```

```
| month_num | differential |
| --------- | ------------ |
|     12    |      -12     |
|     1     |      -3      |
|     1     |       0      |
|     2     |      -7      |
```

The progression is more natural than subqueries.

```
| month_num | avg_differential | most_differential |
| --------- | ---------------- | ----------------- |
|     12    |       -5.77      |        -14        |
|     1     |       -1.91      |        -8         |
|     2     |       -8.13      |        -22        |
```

Let's practice!

DATA MANIPULATION IN SNOWFLAKE

Advanced Common Table Expressions

DATA MANIPULATION IN SNOWFLAKE

Jake Roach
Field Data Engineer

datacamp

Defining multiple Common Table Expressions

More than one common table expression can be defined using a single `WITH` clause

- Arbitrary # of CTE's can be defined
- CTE's can be `JOIN` 'd together
- Can still perform complex operations in a CTE
- Could even use the results of one CTE in another

```
WITH <cte-name> AS (

    <query>

), <another-cte-name> (

    -- Add another query!
    <another-query>
)

-- These CTE's could be JOIN'd
SELECT ... ;
```

Top-performing courses

```
WITH active_courses AS (
    SELECT
        id,
        course_name,
        teacher_name
    FROM courses
    WHERE is_active
), course_avgs (
    SELECT
        course_id,
        AVG(grade) AS avg_grade
    FROM student_courses
    GROUP BY course_id
)
```

```
SELECT
    a.course_name,
    a.teacher_name,
    c.avg_grade
FROM active_courses AS a

-- JOIN these CTEs together
JOIN course_avgs AS c
    ON a.id = c.course_id

ORDER BY avg_grade DESC;
```

Query becomes easier to understand!

Top-performing courses

id	course_name	teacher_name
1145	Algebra	Mrs. Walker
1672	Biology	Mr. Casey
0985	History	Ms. Grimes

course_id	avg_grade
1145	87.77
1672	81.32
0985	91.56

course_name	teacher_name	avg_grade
Algebra	Mrs. Walker	87.77
Biology	Mr. Casey	81.32
History	Ms. Grimes	91.56

Tenured teachers

```
WITH active_courses AS (
    SELECT
        id,
        course_name,
        teacher_name,
        teacher_tenure
    FROM courses

    -- JOIN the teachers table to the courses
    -- table to get teacher_tenure
    JOIN teachers
        ON courses.teacher_id = teachers.id

    WHERE is_active
),
...
```

```
...
), course_avgs (
    SELECT
        course_id,
        AVG(grade) AS avg_grade
    FROM student_courses
    GROUP BY course_id
)

SELECT
    a.teacher_name,
    a.teacher_tenure
    MAX(c.avg_grade) AS highest_grade
FROM active_courses AS a
JOIN course_avgs AS c ON a.id = c.course_id
GROUP BY a.teacher_name, a.teacher_tenure;
```

Tenured teachers

teacher_name	teacher_tenure	highest_grade
Mrs. Walker	11 Years	92.89
Mr. Casey	27 Years	87.11
Ms. Grimes	2 Years	94.48

Let's practice!

DATA MANIPULATION IN SNOWFLAKE

Wrapping up!

DATA MANIPULATION IN SNOWFLAKE

Jake Roach
Field Data Engineer

datacamp

Let's practice!

DATA MANIPULATION IN SNOWFLAKE