Complex Queries

Database Design and Development 2023
Cody Henrichsen

1

Named query

- When you notice you need the SAME information in multiple spots of a query
- You could just rewrite the same query again and again
  - We have zero time for that nonsense
- Name the query and reference throughout your query like a variable including in subqueries!

2

Named Query Syntax

```
WITH named_query AS
(
   SELECT
   FROM
   …
)
SELECT
FROM
…
;
```

3

Question

- We want to find all the students who live in cities with less than the average number of enrolled students

4

Solution

- Named Query
  - We need to count the number of enrolled students by city
  - This requires joining student to enrollment
  - Then joining to zip to get the appropriate count
  - We use an correlated EXISTS subquery to only add enrolled students
- We need a subquery that will retrieve the cities from the stored query only if they have less than the average number of students per city (another subquery)
- Our outer query gets the distinct students by JOINing the student, enrollment, and zipcode tables again then limiting the results based on the subqueries that use the named query

5

Solution SQL

Named Query

```
WITH count_by_city AS
(
    SELECT
            COUNT(*) AS number_enrolled,
            zip.city -- The KEY for this table!
    FROM
            zipcode zip
    JOIN
            student stu
    ON
            zip.zip = stu.zip
    JOIN
            enrollment enr
    ON
            stu.student_id = enr.student_id
    WHERE EXISTS
    (
            SELECT
                    'X'
            FROM
                    student
            WHERE
                    student_id = enr.student_id
    )
    GROUP BY
            zip.city
)
```

Base Query

```
SELECT DISTINCT
    stu.student_id
FROM
    student stu
JOIN
    enrollment en
ON
    stu.student_id = en.student_id
JOIN
    zipcode zip
ON
    zip.zip = stu.zip
WHERE
    zip.city IN
    (
        SELECT
            cbc.city
        FROM
            count_by_city cbc
        WHERE
            cbc.number_enrolled <
            (
                SELECT
                    AVG(count_by_city.number_enrolled)
                FROM
                    count_by_city
            )
    )
ORDER BY
    stu.student_id DESC
;
```

6