Database Systems I

More SQL

Instructor: Ouldooz Baghban Karimi

CMPT 354 - Summer 2019
More SQL

- **Subqueries**
  - **FROM** clause subqueries
  - **WHERE** clause subqueries

- SQL Review
Subqueries

• Need to compute an intermediate table only to use it later in a \texttt{SFW}

• Subqueries may appear in
  • A \texttt{WHERE} clause
    • Subqueries can return a single constant, and this constant can be compared with another value in a \texttt{WHERE} clause.
    • Subqueries can return relations that can be used in various ways in \texttt{WHERE} clauses.
  • A \texttt{FROM} clause
    • Subqueries can appear in \texttt{FROM} clauses, followed by a tuple variable that represents the tuples in the result of the subquery.

• Subqueries may have subqueries, down as many levels as we wish.
Subqueries

- Subquery: a query that is part of another query
- Such inner-outer queries are called nested queries

```sql
SELECT name
FROM MovieExec
WHERE certNum =
  ( SELECT producerCNum
      FROM Movies
      WHERE title = 'Star Wars'
  );
```
Subqueries in WHERE Clauses

• Subqueries that Produce Scalar Values

• Conditions Involving Relations

• Conditions Involving Tuples

• Correlated Subqueries
Subqueries that Produce Scalar Values

- Scalar: atomic value that can be one component of a tuple

- Comparing the result of a subquery (which is a scalar value) to a constant or attribute in **WHERE** clause
Example: Subqueries that Produce Scalar Values

Movies(title, year, length, genre, studioName, producerCNum)
StarsIn(movieTitle, movieYear, starName)
MovieExec(name, address, certNum, netWorth)

```
SELECT name
FROM Movies, MovieExec
WHERE title = 'Star Wars' AND producerCNum = certNum;
```

```
SELECT name FROM MovieExec WHERE certNum =
    ( SELECT producerCNum
      FROM Movies
      WHERE title = 'Star Wars'
    );
```
Conditions Involving Relations

- A relation $R$ must be expressed as a subquery.

**IN**

**NOT IN**

**EXISTS**

**NOT EXISTS**

**ANY**

**ALL**
Conditions Involving Relations

- **EXISTS R**: true if and only if R is not empty.

- **s IN R**: true if and only if s is equal to one of the values in unary relation R.

- **s NOT IN R**: true if and only if s is equal to no value in unary relation R.

- **s > ALL R**: true if and only if s is greater than every value in unary relation R.

- **s > ANY R**: true if and only if s is greater than at least one value in unary relation R.
Conditions Involving Tuples

• If a tuple \( t \) has the same number of components as a relation \( R \), then it makes sense to compare \( t \) and \( R \) in expressions.

```sql
SELECT name
FROM MovieExec
WHERE certNum IN
  ( SELECT producerCNum
    FROM Movies
    WHERE (title, year) IN
      ( SELECT movieTitle, movieYear
        FROM StarsIn
        WHERE starName = 'Harrison Ford'
      )
  );
```
Correlated Subqueries

• A subquery to be evaluated many times, once for each assignment of a value to some term in the subquery that comes from a tuple variable outside the subquery.

• Example: Finding movie titles that appear more than once

```sql
SELECT title
FROM Movies Old
WHERE year < ANY
  ( SELECT year
      FROM Movies
      WHERE title = Old.title
  );
```
Subqueries in FROM Clauses

- In a FROM list, instead of a stored relation, we may use a parenthesized subquery.

```sql
SELECT name
FROM MovieExec, ( SELECT producerCNum
                    FROM Movies, StarsIn
                    WHERE title = movieTitle AND
                    year = movieYear AND
                    starName = 'Harrison Ford'
                ) Prod
WHERE certNum=Prod.producerCNum;
```
More SQL

• Subqueries
  • \textbf{FROM} clause subqueries
  • \textbf{WHERE} clause subqueries

✓ SQL Review
Acknowledgements

I have used materials from the following resources in preparation of this course:

• **Database Systems: The Complete Book**
• Database Systems (Kifer, Bernstein, Lewis)
• Database System Concepts: [https://www.db-book.com](https://www.db-book.com)
• Course offerings
  • **CMPT 354 (Jiannan Wang - SFU):** [https://sfu-db.github.io/cmpt354/](https://sfu-db.github.io/cmpt354/)
  • W 4111 (Eugene Wu - Columbia): [https://w4111.github.io/](https://w4111.github.io/)
  • CS 186 (Joe Hellerstein - Berkeley): [https://sites.google.com/site/cs186fall17/](https://sites.google.com/site/cs186fall17/)