SQL Nested Queries

Slides based on “Database Management Systems” 3rd ed, Ramakrishnan and Gehrke
Nested Queries

- An SQL query can be used to help the evaluation of another query
  - E.g., a condition may need to be evaluated on a computed relation, not one readily available
  - Multiple levels of nesting are possible
  - Semantics similar to those of nested loops

- Nested queries do not appear in relational algebra
  - But it is possible to write relational algebra expressions to obtain same result
  - Using nested queries leads to more concise solutions
Connecting queries and subqueries

- Depends on what the subquery returns:
  - A scalar value (1x1 table) – can appear in a query in the same place where a constant appears
  - A relation

- Where can subqueries appear?
  - Most often in **WHERE** clause of parent query
  - Also used in **FROM** clause followed by range variable
    - … FROM Sailors, (SELECT bid FROM Boats) Bids …
  - In **HAVING** clauses
    - Will discuss later on
### Example Schema

#### Sailors

<table>
<thead>
<tr>
<th>sid</th>
<th>sname</th>
<th>rating</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>dustin</td>
<td>7</td>
<td>45.0</td>
</tr>
<tr>
<td>31</td>
<td>lubber</td>
<td>8</td>
<td>55.5</td>
</tr>
<tr>
<td>58</td>
<td>rusty</td>
<td>10</td>
<td>35.0</td>
</tr>
</tbody>
</table>

#### Boats

<table>
<thead>
<tr>
<th>bid</th>
<th>name</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>interlake</td>
<td>red</td>
</tr>
<tr>
<td>103</td>
<td>clipper</td>
<td>green</td>
</tr>
</tbody>
</table>

#### Reserves

<table>
<thead>
<tr>
<th>sid</th>
<th>bid</th>
<th>day</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>101</td>
<td>10/10/96</td>
</tr>
<tr>
<td>58</td>
<td>103</td>
<td>11/12/96</td>
</tr>
</tbody>
</table>
Subqueries that return a constant

- Also referred to as subqueries that return a scalar
- Most easy case to understand

```sql
SELECT S.sname 
FROM Sailors S 
WHERE S.sid = (SELECT R.sid 
               FROM Reserves R 
               WHERE R.bid=103)
```

- If subquery returns more than one value or zero values, a runtime error occurs! FRAGILE, AVOID!

- Next, we focus on subqueries that return relations
Conditions involving relations

- Test that a relation satisfies some condition
  ... WHERE EXISTS (SELECT ...) - TRUE if subquery result is not empty
  ... WHERE UNIQUE (SELECT ...) - TRUE if subquery result has no duplicates

SELECT S.sname
FROM Sailors S
WHERE EXISTS (SELECT *
  FROM Reserves R
  WHERE R.bid=103 AND S.sid=R.sid)

- Find names of sailors who’ve reserved boat #103

- Subquery is CORRELATED with parent query
Conditions involving relations and tuples

- Typically have some sort of set operations semantics
  - ...WHERE field \textbf{IN} (SELECT ... )
  - ... WHERE field \textbf{op} \textbf{ANY} (SELECT ... )
  - ... WHERE field \textbf{op} \textbf{ALL} (SELECT ... )

  \begin{verbatim}
  SELECT S.sname
  FROM  Sailors S
  WHERE S.sid \textbf{IN} (SELECT R.sid
                   FROM  Reserves R
                   WHERE  R.bid=103)
  \end{verbatim}

- \textit{Find names of sailors who’ve reserved boat #103}
Conditions involving relations and tuples

- Typically have some sort of set operations semantics
  - ...WHERE field IN (SELECT … )
  - ... WHERE field op ANY (SELECT …)
  - ... WHERE field op ALL (SELECT …)

  \[
  \text{SELECT} \ S.sname \\
  \text{FROM} \ \text{Sailors} \ S \\
  \text{WHERE} \ S\text{.rating} > \text{ANY (SELECT S1.rating} \\
  \text{FROM} \ \text{Sailors} \ S1, \ \text{Reserves} \ R1 \\
  \text{WHERE} \ S1\text{.sid}=R1\text{.sid} \ \text{AND} \\
  \text{R1.bid}=103) \\
  \]

- Find names of sailors whose rating is higher than the minimum rating among sailors who reserved boat 103
Conditions involving relations and tuples

- Typically have some sort of set operations semantics
  ...  WHERE field IN (SELECT ... )
  ...  WHERE field op ANY (SELECT ...)
  ...  WHERE field op ALL (SELECT ...)

  SELECT S.sname
  FROM Sailors S
  WHERE S.age >= ALL (SELECT S1.age
                    FROM   Sailors S1)

- Find names of sailors with maximum age
Subqueries in the FROM clause

SELECT  SQ.sname, SQ.bname
FROM ( SELECT S.sname, B.name AS bname
        FROM Sailors S, Boats B, Reserves R
        ) SQ
WHERE  SQ.bname=‘interlake’;

- Find names of sailors who reserved ‘interlake’
Rewriting INTERSECT Queries Using IN

Find sid’s of sailors who’ve reserved both a red and a green boat:

SELECT S.sid
FROM Sailors S, Boats B, Reserves R
WHERE S.sid=R.sid AND R.bid=B.bid AND B.color=‘red’
    AND S.sid IN (SELECT S2.sid
                   FROM Sailors S2, Boats B2, Reserves R2
                   WHERE S2.sid=R2.sid AND R2.bid=B2.bid
                   AND B2.color=‘green’)

• Similarly, EXCEPT queries re-written using NOT IN.
Nested Queries - Review

- **Nested queries returning a constant**
  - Typically constant is compared with other value in the WHERE clause
  - … WHERE field = (SELECT bid FROM …) …

- **Nested queries returning a relation**
  - in WHERE clause
    - … WHERE EXISTS|UNIQUE (SELECT bid FROM …) …
    - … WHERE field IN (SELECT bid FROM …) …
    - … WHERE field op ANY|ALL (SELECT bid FROM …) …
  - in FROM clause followed by range variable
    - … FROM Sailors, (SELECT bid FROM Boats) Bids …
Find sids of sailors who’ve reserved only red boats

SELECT R.sid
FROM Reserves R, Boats B
WHERE B.bid = R.bid AND B.color='red' AND R.sid NOT IN
(SELECT R1.sid FROM Reserves R1, Boats B1
WHERE B1.color<>‘red’ AND B1.bid = R1.bid)
Find sid of sailors who’ve reserved only red boats

SELECT R.sid FROM Reserves R
WHERE NOT EXISTS(
    SELECT * from Reserves R1, Boats B
    WHERE B.bid=R1.bid AND R1.sid=R.sid
    AND B.color <> ‘red’
)

Example Query 2 Answer

Find the name(s) of sailor(s) who have the highest rating

- Sailors
  - sid
  - sname
  - rating
  - age
- Boats
  - bid
  - name
  - color
- Reserves
  - sid
  - bid
  - day

SELECT S1.sname FROM Sailors S1
WHERE S1.sid NOT IN (SELECT S2.sid FROM Sailors S2, Sailors S3
WHERE S2.rating < S3.rating)
Find the name(s) of sailor(s) who have the highest rating

SELECT S.sname FROM Sailors S
WHERE S.rating >= ALL (SELECT rating FROM Sailors)