

Introduction to Database Systems

Nested Queries

Stéphane Bressan



Nested Queries

Find the first name and last name of the customers who downloaded a game.

```
SELECT c.first_name, c.last_name  
FROM customer c, downloads d  
WHERE c.customerid = d.userid;
```

Customers are repeated...It is better done with a nested query. **outer query**

```
SELECT c.first_name, c.last_name  
FROM customer c  
WHERE c.customerid = ANY  
      (SELECT d.customerid  
       FROM downloads d);
```

inner query

There can be subqueries in the subqueries (some DBMS have a maximum level of nesting and some stop optimizing after a certain level).

ANY

Find the first name and last name of the different customers who downloaded a game.

```
SELECT c.first_name, c.last_name
FROM customers c
WHERE c.customerid = ANY (
    SELECT d.customerid
    FROM downloads d);
```

The condition in the " WHERE " clause is true if the identifier is equal to any of the results of the subquery.

" customerid " is a primary key of the "customer " table. The query now considers the different customers.

ANY

Find the first name and last name of the different customers who downloaded a game.

```
SELECT c.first_name, c.last_name  
FROM customers c  
WHERE c.customerid = (  
    SELECT d.customerid  
    FROM downloads d);
```

In most DBMS the query returns an error because there is more than one result to the subquery.

It is recommended to always use "ANY", even in cases where the result of the subquery has at most one row.

ANY and ALL

Find the name, version and price of the most expensive games.

```
SELECT g1.name, g1.version, g1.price  
FROM games g1  
WHERE g1.price >= ALL (  
    SELECT g2.price  
    FROM games g2);
```

The condition in the "WHERE" clause is true if the price is larger than or equal to all the results of the subquery.

ANY and ALL

"ANY" and "ALL" can be used with =, >, <, <= and >=.

```
SELECT g1.name, g1.version, g1.price
FROM games g1
WHERE g1.price >= ANY (
    SELECT g2.price
    FROM games g2);
```

This query returns all the games.

IN

Find the first name and last name of the different customers who downloaded a game.

```
SELECT c.first_name, c.last_name  
FROM customers c  
WHERE c.customerid IN (  
    SELECT d.customerid  
    FROM downloads d);
```

The condition in the "WHERE" clause is true if the identifier is in the results of the subquery. "`= ANY`" and "`IN`" are synonyms.

EXISTS

Find the first name and last name of the different customers who downloaded a game.

```
SELECT c.first_name, c.last_name  
FROM customers c  
WHERE EXISTS (SELECT *  
FROM downloads d  
WHERE c.customerid = d.customerid);
```

The condition in the "WHERE" clause is true if the result of the subquery is not empty. The outer query and the inner query are correlated. The condition of the subquery uses "c.customerid" from the outer query. Think of the subquery as a different query being evaluated for each row of the outer query. Attributes of a query can be used at any level of nesting in its subqueries.

Not the other way round.

<> ALL, NOT IN, NOT EXISTS

Find the identifier of the customers who never downloaded any game.

```
SELECT c.customerid
FROM customers c
WHERE u.userid <> ALL (SELECT d.customerid
FROM downloads d),
```

```
SELECT c.customerid
FROM customers c
WHERE c.customerid NOT IN (SELECT d.customerid
FROM downloads d);
```

```
SELECT c.customerid
FROM customers c
WHERE NOT EXISTS(SELECT *
FROM downloads d
WHERE c.customerid = d.customerid);
```

In the HAVING clause

Find the different identifiers of customers who downloaded the largest number of games.

```
SELECT DISTINCT c1.customerid
FROM customer c1, downloads d1
WHERE c1.customerid = d1.customerid
GROUP BY c1.customerid
HAVING COUNT(*) >= ALL (SELECT COUNT(*)
                        FROM customers c2, downloads d2
                        WHERE c2.customerid = d2.customerid
                        GROUP BY d2.customerid);
```

There can be nested queries in the "HAVING" clause.

Credits

The content of this lecture is based
on chapter 5 of the book
“Introduction to database Systems”

By
S. Bressan and B. Catania, McGraw
Hill publisher

Images and clips used in this
presentation are licensed from
Microsoft Office Online Clipart and
Media

For questions about the content of
this course and about copyrights,
please contact Stéphane Bressan

steph@nus.edu.sg

Copyright © 2017 by Stéphane Bressan

