National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

Laboratory Manual 06:

Correlated Nested Queries

CS-220: Database Systems

Fall 2017

Class: BS(CS)-6B

Dr. Amanullah Yasin

Mr. Ahmed Shahzaib Abid Bhatti

Date:20-Oct-17

Timings:1415-1700

Introduction

- SQL DDL (Data Definition Language) commands are used to create and modify the databases. Data Manipulation Language (DML) commands are used to query the databases.

Objectives

After performing this lab students should be able to:

1. Create tables in SQL using DDL commands.
2. Perform DML operations on created tables.

Tools/Software Requirement

- MySQL Community Server 5.6
- MySQL Workbench 6.1

Description

Nested Queries/Subqueries

A nestested/subquery is a SQL query nested inside a larger query, such inner-outer queries are called nested queries
A subquery may occur in:

- A SELECT clause
- A FROM clause
- A WHERE clause

Rule of thumb: avoid writing nested queries when possible; keep in mind that sometimes it's impossible

Nested queries

- can return a single constant and this constant can be compared with another value in a WHERE clause
- Can return relations that can be used in various ways in WHERE clauses
- Can appear in FROM clauses, followed by a tuple variable that represents the tuples in the result of the subquery

- Can appear as computed values in a SELECT clause

Given the following **database schema**:

Student (snum: integer, sname: char(30), major: char(25), level: char(2), age: integer)

Faculty (fid: integer, fname: char(30), deptid: integer)

Class (cname: char(40), meets_at: char(20), room: char(10), fid: integer | fid REFS Faculty.fid)

Enrolled (snum: integer, cname: char(40) | snum REFS student.snum, cname REFS class.name)

1. **Find the name of faculty members who do not teach any course**.

```
select distinct f.fname
from faculty f
where f.fid not in
  (select c.fid from class c);
```

2. **Find the names of students who are enrolled in a course taught by I. Teach**.

```
Select s. snames
From student s
where S.snum in
(Select E.snum
 From class C, enrolled E, faculty F
 Where E.cname = C. cname and C.fid = F.fid
 and f.fname = 'I. Teach')
```

3. **Find the names of all students who are enrolled in two classes that meet at the same time**.

```
select distinct S.sname
from student S
Where S.snum in
 (select E1.snum
 from enrolled E1, enrolled E2, class C1, class C2
 where E1.snum = E2.snum and E1.cname <> E2.cname
 and E1.cname = C1.cname
 and E2.cname = C2.cname and C1.meets_at = C2.meets_at)
```

CORRELATED NESTED/SUBQUERIES

A correlated subquery is one where the inner query depends on values provided by the outer query. This means the inner query is executed repeatedly, once for each row that might be selected by the outer query. For example,

Query: **Find the names of students enrolled in any classes**

```
select distinct s.sname
from student s
where Exists (select * e.snum
from enrolled e where s.snum = e.snum)
```

The subquery in this SELECT statement cannot be resolved independently of the main query. Notice that the outer query specifies that rows are selected from the student table with an alias name of s. The inner query compares the snum (student name) column of the enrolled table to the same column of the student table. The subquery's results are correlated with each individual row of the main query – thus, the term correlated subquery.

When a subquery uses the EXISTS operator, the subquery functions as an existence test. The WHERE clause of the outer query tests for the existence of rows returned by the inner query.The subquery does not actually produce any data; rather, it returns a value of TRUE or FALSE.

The general format of a subquery WHERE clause with an EXISTS operator as shown below. Note that the NOT operator can also be used to negate the result of the EXISTS operator.

WHERE [NOT] EXISTS (subquery)

Subqueries using an EXISTS operator are a bit different from other subqueries, in the following ways:

1. The keyword EXISTS is not preceded by a column name, constant, or other expression.
2. The SELECT clause list of a subquery that uses an EXISTS operator almost always consists of an asterisk (*). This is because there is no real point in listing column names since you are simply testing for the existence of rows that meet the conditions specified in the subquery.
3. The subquery evaluates to TRUE or FALSE rather than returning any data.
4. A subquery that uses an EXISTS operator will always be a correlated subquery.

The EXISTS operator is very important, because there is often no alternative to its use. All queries that use the IN operator or a modified comparison operator (=, <, >, etc. modified by ANY or ALL) can be expressed with the EXISTS operator. However, some queries formulated with EXISTS cannot be expressed in any other way. For example,

Query: **Find the names of students enrolled in any classes**

```
select distinct s.sname
from student s
where s.snum in (select e.snum
from enrolled e);

************************************


select distinct s.sname
from student s
where Exists (select * e.snum
from enrolled e where s.snum = e.snum)
```

The NOT EXISTS operator is the mirror-image of the EXISTS operator. A query that uses NOT EXISTS in the WHERE clause is satisfied if the subquery returns no rows.

Lab Task

Write SQL expressions in correlated-nested for each of the following queries and execute them:

1. Find the names of all juniors (Level = JR) who are enrolled in a class taught by ‘Ivana Teach’.
2. Find the names of faculty members that have taught classes only in room R128.
3. Find the names of classes taught by ‘Richard Jackson’ and their times when a class meet there.
4. Retrieve the snum and sname of students who have taken classes from both ‘Ivana Teach’ and ‘Linda Davis’.
5. Find the age of the oldest students who is enrolled in a course taught by Ivana. Teach.
6. Find ages of students in ‘Database Systems’ class that are older than 20 years.
7. Find the name of faculty members that do not teach to class ‘database Systems’.
8. Find the name of faculty members who do not teach any course.
9. Find the name of faculty member, department who taught the maximum number of distinct classes.
10. Find the names of all classes and their enrollment strength that have enrollment greater than 5.

Deliverables

1. Complete your lab tasks in SQL workbench and submit a word file in with queries along with the screenshots of the results to all the questions attempted. Upload it on LMS. The marking will be based on viva/lab task submitted.