

CS3DB3 / SE4DB3 / SE6M03 TUTORIAL

Mei Jiang

Jan 30 / Feb 1, 2013

Outline



- Connecting db2 server
- ANY, ALL, IN, EXISTS Operators
- Set Operations

Connecting db2 server

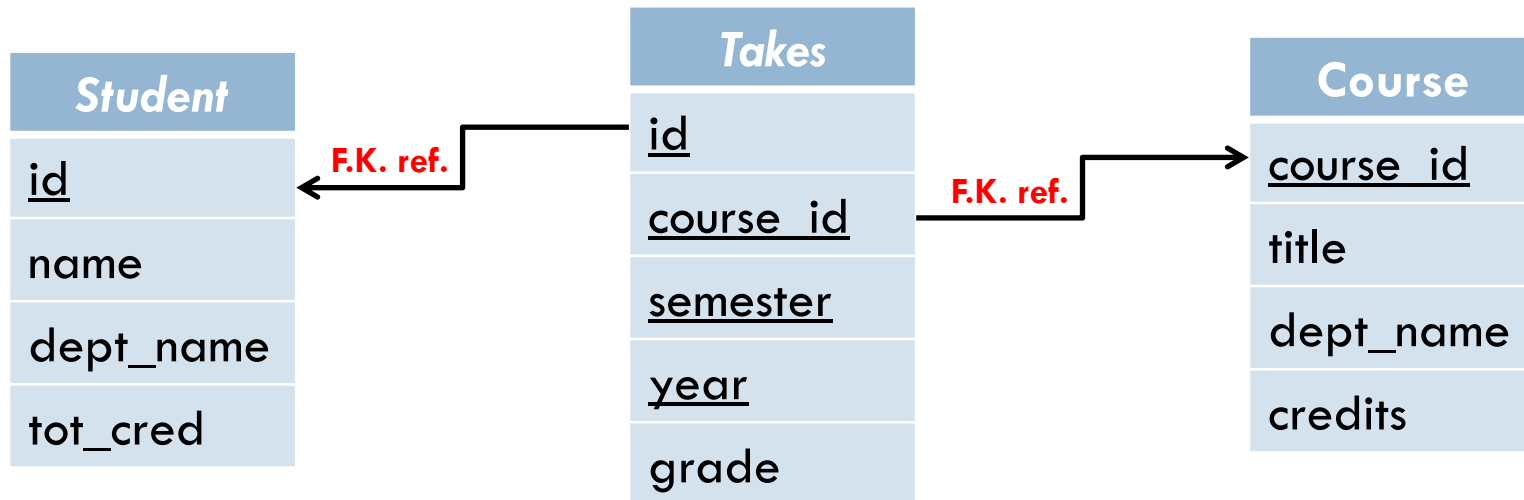
- On Campus
 - ▣ MacSecure
 - Connect directly
 - ▣ MacConnect
 - Connect as off campus
- Off Campus
 - ▣ Use VPN (see Assignment 1 FAQ)
 - ▣ Connect from a external department server
 - Connect to `mills.mcmaster.ca` (or `moore.mcmaster.ca`), then
 - `ssh loginid@db2srv2.mcmaster.ca`

Operators - ANY

- $x \langle \text{op} \rangle \text{ANY (Subquery)}$ is true *iff* there exists a r from the result of **Subquery** s.t. $x \langle \text{op} \rangle r$ is true.
- $\langle \text{op} \rangle$ can be $=, <, >, >=, <=$
- Example
 - $(6 < \text{ANY } \begin{matrix} 0 \\ 6 \end{matrix}) = \text{False}$ $(6 <> \text{ANY } \begin{matrix} 0 \\ 6 \end{matrix}) = \text{True}$
- $(= \text{ANY}) \equiv \text{IN}$, but $(<> \text{ANY}) \neq \text{NOT IN}$

Example - University Schema

- **Student** (id, name, dept_name, tot_cred)
- **Course** (course_id, title, dept_name, credits)
- **Takes** (id, course_id, semester, year, grade)



Operators - ANY (cont.)

- **Course** (course_id, title, dept_name, credits)
- Find titles of courses with credits higher than at least one course in the Biology department

```
SELECT title
FROM course
WHERE credits > ANY( SELECT credits
                        FROM course
                        WHERE dept_name = 'Biology'
                        );
```

Operators - ALL

- $x \langle op \rangle \mathbf{ALL}(\text{Subquery})$ is true *iff* for all r from the result of **Subquery**, $x \langle op \rangle r$ is true.
- $\langle op \rangle$ can be $=, \langle \rangle, >, >=, <, <=$
- Example
 - $(6 \langle \rangle \mathbf{ALL} \begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline \end{array}) = \mathbf{True}$ $(6 = \mathbf{ALL} \begin{array}{|c|} \hline 0 \\ \hline 6 \\ \hline \end{array}) = \mathbf{False}$
- $(\langle \rangle \mathbf{ALL}) \equiv \mathbf{NOT IN}$, but $(= \mathbf{ALL}) \neq \mathbf{IN}$

Operators - ALL (cont.)

- **Student** (id, name, dept_name, tot_cred)
- Find name of the student who has the highest total credit

```
SELECT name
```

```
FROM student
```

```
WHERE tot_cred >= ALL (SELECT tot_cred  
                        FROM student );
```


Operators - IN

- **x IN(Subquery)** returns true *iff* there exists a *r* from the result of **Subquery** s.t. **x = r**
- **Course** (course_id, title, dept_name, credits)
- **Takes** (id, course_id, semester, year, grade)
- Find distinct id of students who have taken courses from Computer Science department in 2010

```
SELECT DISTINCT id
FROM takes
WHERE course_id IN ( SELECT course_id
                        FROM course
                        WHERE dept_name = 'Comp. Sci.'
                        )
AND year=2010;
```

Operators - EXISTS

- **EXISTS**(Subquery) returns true *iff* the result of **Subquery** is not empty
- **Student** (id, name, dept_name, tot_cred)
- **Takes** (id, course_id, semester, year, grade)
- Find names of the students who have taken at least one course in the Spring 2010 semester

```
SELECT name
FROM student s
WHERE EXISTS ( SELECT *
                FROM takes t
                WHERE semester = 'Spring'
                    AND year= 2010
                    AND s.id = t.id
                );
```

Set Operations

- **(Subquery1) UNION (Subquery2)**
 - ▣ Returns tuples appear in **either** results of the two subqueries
- **(Subquery1) INTERSECT (Subquery2)**
 - ▣ Returns tuples appear in **both** results of the two subqueries
- **(Subquery1) EXCEPT (Subquery2)**
 - ▣ Returns tuples appear in results of subquery1, but not appears in subquery2

Set Operations (cont.)

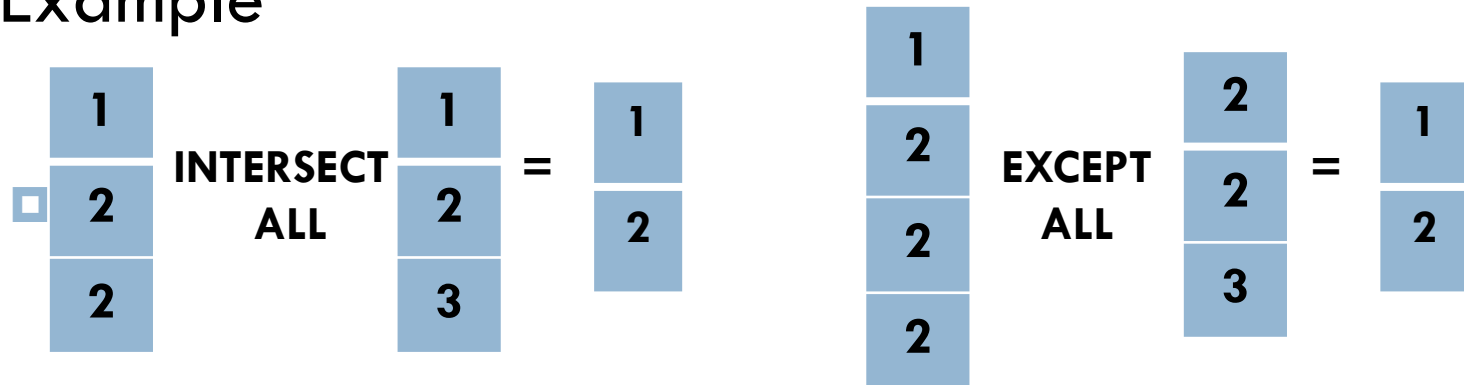
- **Takes** (id, course id, semester, year, grade)
- Find set of students id who took course in both Fall 2009 and Spring 2010
(**SELECT** id **FROM** takes **WHERE** semester = 'Fall' **AND** year = 2009)
INTERSECT
(**SELECT** id **FROM** takes **WHERE** semester = 'Spring' **AND** year = 2010);
- Find set of students id who took course in Fall 2009 but not in Spring 2010
(**SELECT** id **FROM** takes **WHERE** semester = 'Fall' **AND** year = 2009)
EXCEPT
(**SELECT** id **FROM** takes **WHERE** semester = 'Spring' **AND** year = 2010);

Set Operations - Multiset Version

- **UNION, INTERSECT, EXCEPT** operations automatically eliminate duplicates in the results
- To retain the duplicates, use the corresponding multiset versions **UNION ALL, INTERSECT ALL, EXCEPT ALL**

Set Operations - Multiset Version

□ Example



- A tuple occurs m times in **R** and n times in **S**, then it
 - $(m+n)$ times in **R UNION ALL S**
 - $\min(m, n)$ times in **R INTERSECT ALL S**
 - $\max(0, m-n)$ times in **R EXCEPT ALL S**

Set Operations -Set Containment

- We can use **NOT EXISTS** and **EXCEPT** to simulate the set containment operation
- **(B EXCEPT A) = \emptyset \Leftrightarrow B \subseteq A**
- We can write “relation A contains relation B” as **NOT EXISTS (B EXCEPT A)**

Set Operations -Example

- **Student** (id, name, dept_name, tot_cred)
- **Course** (course_id, title, dept_name, credits)
- **Takes** (id, course_id, semester, year, grade)
- Find all students id and names who have taken all courses offered in the Biology department

```
SELECT s.id, s.name
```

```
FROM student as s
```

```
WHERE NOT EXISTS ((SELECT course_id  
                   FROM course  
                   WHERE dept_name = 'Biology')
```

```
EXCEPT
```

```
(SELECT t.course_id
```

```
FROM takes as t
```

```
WHERE s.id = t.id));
```


References



- Database System Concepts (6th edition) by A. Silberschatz, H. Korth, S. Sudarshan