CSCB20 – Week 5

Introduction to Database and Web Application Programming

> *Anna Bretscher* Winter 2016

Last Week

- Creating tables, setting constraints...
- Inserting and updating tables
- More query commands

 HAVING clause
 LIKE clause
 IN clause
 UNION, INTERSECT
 CASE

This Week

- Revisit creating tables
- Creating views
- Outer Joins
 - \circ Left
 - o Right
 - o Full
- More on NULL values
- E-R Model basics

Views

- A view is a virtual relation.
- A view is defined in terms of stored tables (called base tables) and other views.
- Access a view like any base table.
- Materialized views exist, but are actually constructed and stored. Expensive to maintain!
- We'll use only virtual views.

Creating Views

CREATE VIEW view_name AS SELECT STATEMENT;

CREATE VIEW view_name(col_nam₁, col_name₂, ..., col_name_{k)} AS SELECT STATEMENT;

CREATE VIEW faculty AS SELECT ID, name, dept_name FROM instructor;

We can now use view faculty as we would a table.

Every time the view is used, it is reconstructed.

Why Use Views

Allow us to break down a large query.

Make available specific category of data a particular user.

Gives another way to think about the data.

Q. Why is it good that views are virtual?

A. If a table is changed the corresponding view is changed appropriately.

Outer Joins

What does the following query return?

SELECT * FROM student INNER JOIN takes ON student.id = takes.id;

We would like it to return every student and the courses they are taking.

Q. What about students who have not yet taken any courses?

A. They are left out.

Dangling Tuples

When JOINs require some attributes to match, tuples lacking a match are left out.

These tuples are said to be "dangling".

OUTER JOINs preserve dangling tuples by padding them with NULL in the other relation.

INNER JOINs do not pad with NULL.

Outer Joins

Use OUTER JOINS to prevent this loss of information.

The LEFT OUTER JOIN preserves tuples only in the relation to the left of the JOIN.

The **RIGHT OUTER JOIN** preserves tuples only in the relation to the right of the JOIN.

The FULL OUTER JOIN preserves tuples in both relations.*

* MySQL does not support FULL OUTER JOIN, but we can emulate by doing the UNION of a LEFT and a RIGHT.





В	С
2	3
6	7

S

R NATURAL JOIN S

Α	В	С
1	2	3





R

S

R NATURAL LEFT JOIN S

Α	В	С
1	2	3
4	5	NULL





R

S

R NATURAL RIGHT JOIN S

Α	В	С
1	2	3
NULL	6	7





R

S

R NATURAL FULL JOIN S

OR

(R NATURAL LEFT JOIN S) UNION (R NATURAL RIGHT JOIN S)

Α	В	С
1	2	3
4	5	NULL
NULL	6	7

JOIN Recap

A JOIN B ON C

inner join

A {LEFT | RIGHT | FULL} JOIN B ON C outer join

A NATURAL JOIN B

natural inner join

A NATURAL {LEFT | RIGHT | FULL} JOIN B natural outer join

NULL

We can check for NULL values using:

IS NULL

IS NOT NULL

Because we have NULL, we need three truth values for comparisons:

TRUE, FALSE and UNKNOWN

If one or both operands is NULL, the comparison always evaluates to UNKNOWN.

Otherwise, comparisons evaluate to TRUE and FALSE.

Booleans and UNKNOWN

What is NOT UNKNOWN? UNKNOWN.

What is TRUE AND UNKNOWN? UNKNOWN.

What is TRUE OR UNKNOWN? TRUE.

WHAT IS FALSE AND UNKNOWN? FALSE.

WHAT IS FALSE OR UNKNOWN? UNKNOWN.

NULL and Aggregation

	Some NULLS in A	All NULLS in A
MIN(A)		
MAX(A)	Ignore the NULLS	NULL
SUM(A)		
AVG(A)		
COUNT(A)		0
COUNT(*)	All tuples count	