

More JOIN

Can use with multiple conditions: SELECT a.x, b.y FROM a INNER JOIN b ON (a.z = b.z AND a.zz = b.zz);

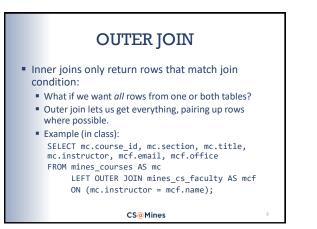
Can use with multiple tables: SELECT a.x, b.y, c.z FROM a INNER JOIN b ON (a.foo = b.foo) INNER JOIN c ON (b.bar = c.bar);

CS@Mines

INNER JOIN Notes

- The INNER keyword is optional (INNER is the default)
- Performance is same using WHERE clause or INNER JOIN
- Choice between WHERE clause or JOIN clause is completely up to you:
 - Personal preference
 - Readability (JOIN clause can get complicated with many tables)

CS@Mines





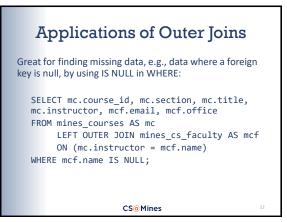
FULL OUTER JOIN

As the name implies, gives all rows from *both* tables in join, matching rows where possible.

Same example as above, replacing LEFT with FULL (OUTER is optional again).

To see effects, first have to insert a bogus faculty member into mines_cs_faculty...

CS@Mines



Oracle Outer Joins

Oracle defines a special operator that allows left and right outer joins to be created using WHERE clause (very handy!):

SELECT a.x, b.y FROM a, b WHERE a.z = b.z(+); -- right outer join!

CS@Mines



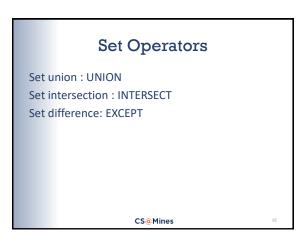
Union, Intersection, Difference

You can perform set operations on two or more SELECT query results:

SELECT course_id, title, instructor FROM mines_courses UNION SELECT 'CSCI999', 'CS Fun Fair', name FROM mines_cs_faculty;

Column names come from first SELECT query. Column types & count have to match.

CS@Mines



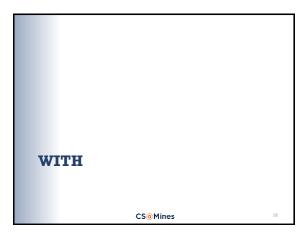
[ALL]

- By default, set operators imply DISTINCT
 - This is because sets, mathematically, contain only distinct units
- To avoid this behavior, add ALL keyword

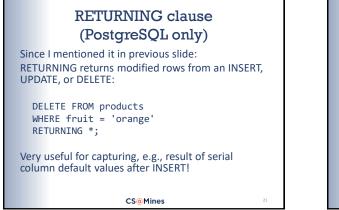
SELECT ... UNION ALL

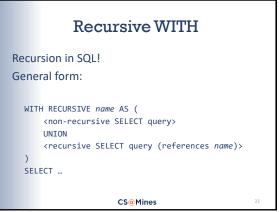
SELECT ...

CS@Mines

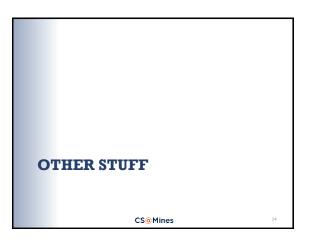


WITH (Common Table WITH (PostgreSQL extensions) **Expression**) Queries • An alternative to subqueries, also with some cool WITH in PostgreSQL has some powerful applications extensions: Can use INSERT/UPDATE/DELETE in CTE Effectively, provides a temporary named relation Gives an alternative to doing transactions (a later topic) for use in a query RETURNING clause (PostgreSQL only) also useful here Example: Can use with INSERT/UPDATE/DELETE WITH cs_courses AS (Example: SELECT * FROM mines courses WHERE course_id LIKE 'CSCI%' WITH q AS (DELETE FROM products WHERE fruit = 'apple' SELECT DISTINCT course_id, title INSERT INTO products FROM cs_courses VALUES ('apple', 'FruitCo', 3.59); WHERE instructor = 'Paone, Jeffrey'; CS@Mines CS@Mines

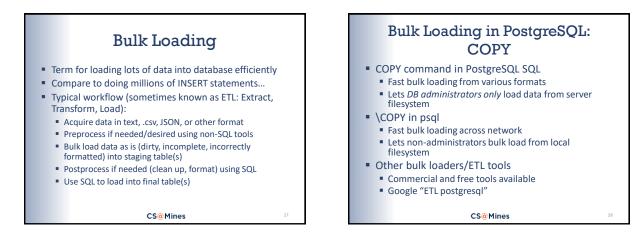


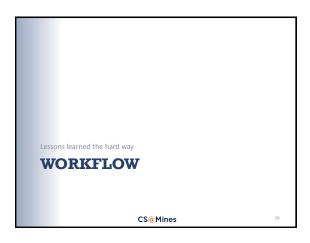


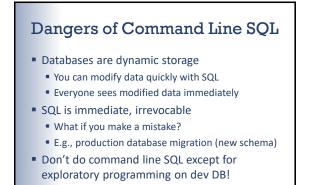
Recursive Example	
From our database (table from textbook):	
<pre>WITH RECURSIVE emp_rec (padding, first, last, ssn) AS (SELECT '', fname, lname, superssn FROM employee WHERE fname = 'loyce' AND lname = 'English' UNION ALL SELECT en.padding ' ', e.fname, e.lname, e.superssn FROM emp_rec er, employee e WHERE e.ssn = er.ssn) SELECT padding first ' ' last FROM emp_rec;</pre>	
CS@Mines 23	



OFFSET...FETCH... Get some rows starting at some offset: SELECT * FROM mines_cs_faculty ORDER BY name FETCH FIRST 5 ROWS ONLY; SELECT * FROM mines_cs_faculty ORDER BY name OFFSET 5 FETCH FIRST 5 ROWS ONLY; CS@Mines ² CS@Mines ²







CS@Mines

Scripts Are Your Friends

- Keep dev, test, production databases (at minimum)
- During development, write & test scripts on dev
- When time for a software release/database migration:
 - Clone a new db from test
 - Apply scripts
 - Fix scripts
 - Repeat as necessary
- Scripts go in your version control system (git)!!
- In theory, should be able to reproduce prod database schema (not data, necessarily) from scratch by running all scripts from beginning of project.

CS@Mines

DBAs/SysAdmins Are Also Friends

- Your production database should be backed up *nightly*
 - Not for crashes, necessarily, as SQL databases can recover from those on their own
 - Instead, need it to recover from developer mistakes ⁽²⁾
- However, make sure DBAs/sysadmins test backups regularly (DB backup is tricky!)

CS@Mines

Up Next

- Next lecture:
 - Database modeling and design
 - Entity-Relationship Diagrams (ERD)
 - Reading: Chapter 3, "Data Modeling Using the Entity-Relationship Model"

CS@Mines