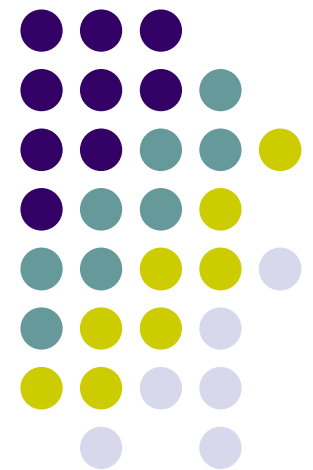


Past errors and future strategies

**CIS 331:
Introduction to
Database Systems**





Topics

- **Common errors I discovered while grading homeworks and lab reports for database courses.**

Hopefully this comes in handy before the midterm.



Group By errors

- How many students does each professor have?

```
SELECT P.first_name, P.last_name, count(*)
      FROM professors P, professors_classes
      PC, students_classes SC
      WHERE P.ssn = PC.professor
            AND PC.class = SC.class
            GROUP BY P.ssn
;

```

- This didn't work. Why?



Group By errors

- But if we change it slightly:

```
SELECT P.first_name, P.last_name, count (*)
      FROM professors P, professors_classes PC,
      students_classes SC
      WHERE P.ssn = PC.professor
            AND PC.class = SC.class
            GROUP BY P.first_name, P.last_name
;
;
```

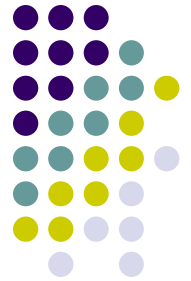


Where vs. Having

- List all the classes which attend more than 3 students:

```
SELECT class, count(*)  
    FROM students_classes  
    WHERE count(*) > 3  
    GROUP BY class  
;
```

- This didn't work. Why?



Where vs. Having

- And the correct solution is:

```
SELECT class, count(*)  
      FROM students_classes  
      GROUP BY class  
      HAVING count(*) > 3  
;
```



Outer join errors

- How many students does each professor have? Report even the professors who do not have any students.

```
SELECT P.first_name, P.last_name, count (*)
      FROM professors P, professors_classes PC,
students_classes SC
      WHERE P.ssn = PC.professor (+)
      AND PC.class = SC.class
      GROUP BY P.first_name, P.last_name
;
```

- This didn't work. Why?

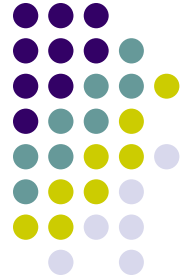


Outer join errors

```
SELECT P.first_name, P.last_name, count (*)
      FROM professors P, professors_classes PC,
      students_classes SC
      WHERE P.ssn = PC.professor (+)
      AND PC.class (+) = SC.class (+)
      GROUP BY P.first_name, P.last_name
;

```

- This did not work either. Why?



Outer join errors

```
SELECT P.first_name, P.last_name, count (*)
      FROM professors P,
           (SELECT SC.student, PC.professor
            FROM professors_classes PC,
                 students_classes SC
            WHERE PC.class = SC.class
           ) PCS
      WHERE P.ssn = PCS.professor (+)
      GROUP BY P.first_name, P.last_name
;
```

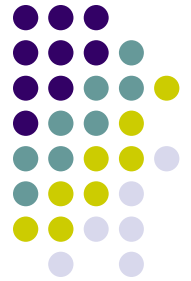
- Almost there... ;-)



Outer join errors

```
SELECT P.first_name, P.last_name, count (PCS.student)
FROM professors P,
      (SELECT SC.student, PC.professor
       FROM professors_classes PC,
            students_classes SC
        WHERE PC.class = SC.class
       ) PCS
WHERE P.ssn = PCS.professor (+)
GROUP BY P.first_name, P.last_name
;
```

- If there is any conclusion to this example, that would be that in order to find the solution, you will very often have to reiterate before you manage to solve the problem...



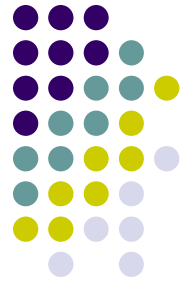
Intersect vs. the Double And

- List all the students who are taking CIS525 and CIS526:

```
SELECT S.first_name, S.last_name
       FROM students S, students_classes SC
       WHERE S.ssn = SC.student
             AND SC.class = 'CIS525'
             AND SC.class = 'CIS526'
;

```

- This will not work. Why?



Intersect vs. the Double And

- And the solution is:

```
SELECT S.first_name, S.last_name
      FROM students S, students_classes SC
      WHERE S.ssn = SC.student
      AND SC.class = 'CIS525'
INTERSECT
SELECT S.first_name, S.last_name
      FROM students S, students_classes SC
      WHERE S.ssn = SC.student
      AND SC.class = 'CIS526'
;
```



Set logic errors

- List all classes that students haven't taken already.

```
SET PAGESIZE 10;  
CLEAR BRAKES;  
TTITLE OFF;
```

```
COLUMN FNAME NOPRINT NEW_VALUE FNAME;  
COLUMN LNAME NOPRINT NEW_VALUE LNAME;
```

```
TTITLE CENTER 'List of classes' SKIP -  
          CENTER FNAME ' ' LNAME ' did not take:'
```

```
BREAK ON FNAME SKIP PAGE ON FNAME
```

```
SELECT DISTINCT S.first_name FNAME, S.last_name LNAME, SC.class  
          FROM students S, students_classes SC  
          WHERE S.ssn != SC.student  
          ORDER BY S.first_name, S.last_name  
;
```

- This will not work. Why?



Set logic errors

- And the solution is:

```
SELECT DISTINCT S.first_name FNAME, S.last_name LNAME,  
SC.class
```

```
    FROM students S, students_classes SC
```

MINUS

```
SELECT DISTINCT S.first_name, S.last_name,  
SC.class
```

```
    FROM students S, students_classes SC
```

```
    WHERE S.ssn = SC.student
```

```
;
```