

SQL

Part IV

Nested and Aggregated Queries

In-class Exercise

Announcement

- **Assignment 4 is available in Canvas**
 - SQL
 - Due time: 8am Oct 30.

SQL

- **Basic form:**

**SELECT A_1, A_2, \dots, A_n
FROM r_1, r_2, \dots, r_m
WHERE P**

- A_i represents an attribute
- r_i represents a relation
- P is a predicate

- **Nested queries**

- A select-from-where query that has another select-from-where subquery
- To perform tests for set membership, set comparisons, and set cardinality.

Aggregate Queries

- Significant extension of relational algebra.
- Aggregate operators:
 - COUNT()
 - SUM()
 - AVG()
 - MAX()
 - MIN()

GROUP BY

```
SELECT      [DISTINCT] target-list
FROM        relation-list
[WHERE      qualification]
GROUP BY    grouping-list
```

The *target-list only* contains

- (i) list of column name (**only** attributes from the *grouping-list*)
- (ii) terms with aggregate operations (e.g., MIN(*age*)).

HAVING

SELECT	[DISTINCT] <i>target-list</i>
FROM	<i>relation-list</i>
WHERE	<i>qualification</i>
GROUP BY	<i>grouping-list</i>
HAVING	<i>group-qualification</i>

- Use HAVING clause with GROUP BY clause to restrict which group-rows are returned in the result set

Exercises

- **Schema**

- Student (snum, sname, major, level, age)
- Class (cname, room, fid)
- Enrolled (snum, cname)
- Faculty (fid, fname, deptid)

- **Exercises**

- **Q1:** Find the *age of the oldest student who is enrolled in a course taught by Adam Smith* (using non-nested and nested queries);
- **Q2:** Find the name of all classes that either meet in room Babbio310 or have 2 or more students enrolled (using non-nested and nested queries);
- **Q3:** Find the name of students who have not enrolled in any class;
- **Q4:** Find the name of students who are enrolled with the maximum number of classes;
- **Q5:** Find the name of faculty members *who teach in every room in which some class is taught*.

(Breakout rooms with odd/even numbers work on questions with odd/even numbers)

Q1: Find the *age of the oldest student who is enrolled in a course taught by Adam Smith.*

Q1: Find the *age of the oldest student who is enrolled in a course taught by Adam Smith.*

- **Solution 1: Without using nested queries**

- The way of thinking

- Join necessary tables, with condition that “course is taught by Adam Smith”
- Find the age of the oldest student in the join result

```
SELECT MAX(S.age)
FROM Student S, Class C, Enrolled E, Faculty F
WHERE S.snum = E.snum AND E.cname = C.name AND
      C.fid = F.fid AND F.fname = 'Adam Smith'
```

OR

```
SELECT MAX(S.age)
FROM Student S NATURAL JOIN Enrolled E NATURAL JOIN Class C
NATURAL JOIN Faculty F
WHERE F.fname = 'Adam Smith'
```

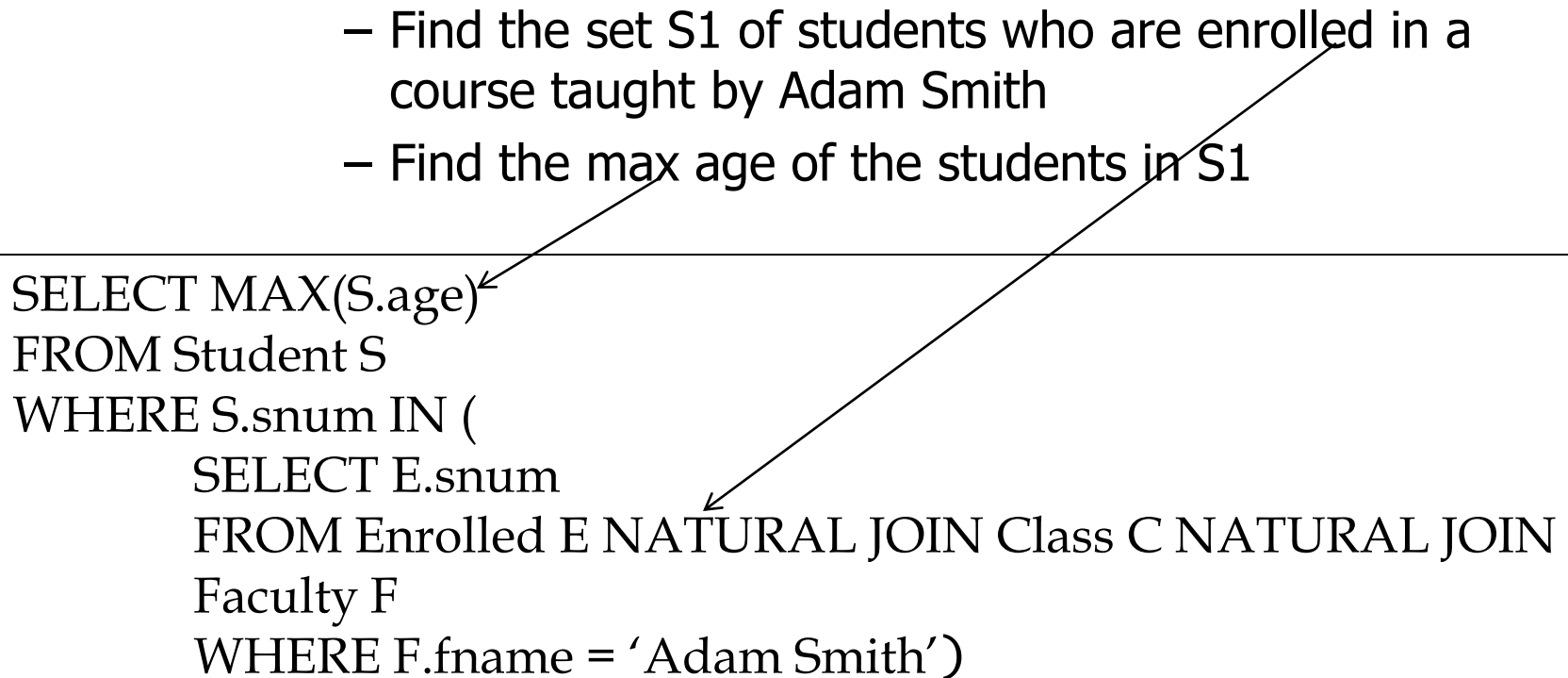
Q1: Find the *age of the oldest student who is enrolled in a course taught by Adam Smith.*

- **Solution 2: by using nested queries**

- The way of thinking

- Find the set S1 of students who are enrolled in a course taught by Adam Smith
 - Find the max age of the students in S1

```
SELECT MAX(S.age)
FROM Student S
WHERE S.snum IN (
    SELECT E.snum
    FROM Enrolled E NATURAL JOIN Class C NATURAL JOIN
    Faculty F
    WHERE F.fname = 'Adam Smith')
```



Q2: Find the name of all classes that either meet in room Babbio310 or have 2 or more students enrolled

Q2: Find the name of all classes that either meet in room Babbio310 or have 2 or more students enrolled

- **Solution 1: Without using nested queries**
 - The way of thinking
 - Find the set A of classes that meet in room Babbio310
 - Find the set B of classes that have 2 or more students enrolled
 - Return A union B

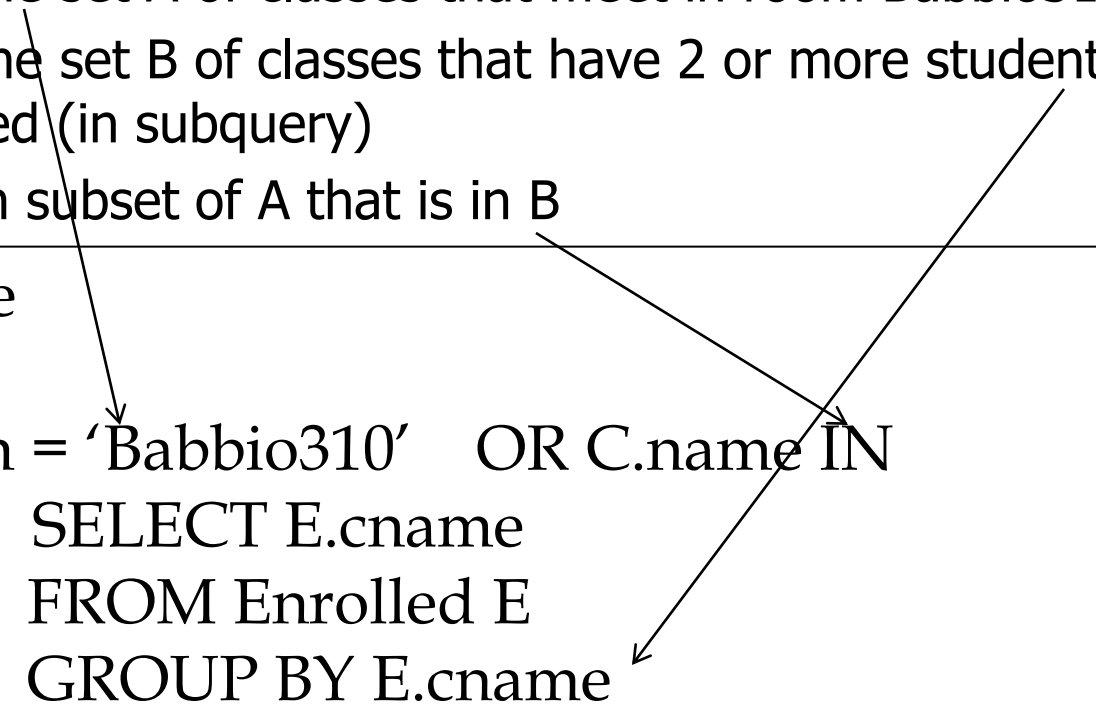
```
SELECT C.cname
FROM Class C
WHERE C.room = 'Babbio310'
UNION
SELECT E.cname
FROM Enrolled E
GROUP BY E.cname
HAVING COUNT (*) >= 2
```

Q2: Find the name of all classes that either meet in room Babbio310 or have 2 or more students enrolled.

- **Solution 2: use nested queries**

- Find the set A of classes that meet in room Babbio310
- Find the set B of classes that have 2 or more students enrolled (in subquery)
- Return subset of A that is in B

```
SELECT C.name
FROM Class C
WHERE C.room = 'Babbio310' OR C.name IN
    ( SELECT E.cname
      FROM Enrolled E
      GROUP BY E.cname
      HAVING COUNT (*) >= 2)
```



Exercises

- **Schema**

- Student (snum, sname, major, level, age)
- Class (cname, room, fid)
- Enrolled (snum, cname)
- Faculty (fid, fname, deptid)

- **Exercises**

- **Q1:** Find the *age of the oldest student who is enrolled in a course taught by Adam Smith* (using non-nested and nested queries);
- **Q2:** Find the name of all classes that either meet in room Babbio310 or have 2 or more students enrolled (using non-nested and nested queries);
- **Q3:** Find the name of students who have not enrolled in any class;
- **Q4:** Find the name of students who are enrolled with the maximum number of classes;
- **Q5:** Find the name of faculty members *who teach in every room in which some class is taught*.

(Breakout rooms with odd/even numbers work on questions with odd/even numbers)

- **Schema**

- Student (snum, sname, major, level, age)
- Class (cname, room, fid)
- Enrolled (snum, cname)
- Faculty (fid, fname, deptid)

Q3: Find the name of students who have not enrolled in any class.

Hint: Find the set A of students who have enrolled in some class; then output the name of students who are not in A.

Q3: Find the name of students who have not enrolled in any class.

- **The way of thinking**

- Find the set A of students who have enrolled in some class
- Output the name of students who are not in A

```
SELECT S.sname  
FROM Student S  
WHERE S.snum NOT IN (SELECT E.snum  
                     FROM Enrolled E )
```


- **Schema**

- Student (snum, sname, major, level, age)
- Class (cname, room, fid)
- Enrolled (snum, cname)
- Faculty (fid, fname, deptid)

Q3: Find the name of students who have not enrolled in any class.

Is this solution correct?

```
SELECT S1.sname
FROM Student S1
WHERE S1.sname NOT IN (SELECT S2.sname FROM
Student S2 NATURAL JOIN Enrolled E);
```

- **Schema**

- Student (snum, sname, major, level, age)
- Class (cname, room, fid)
- Enrolled (snum, cname)
- Faculty (fid, fname, deptid)

Q3: Find the name of students who have not enrolled in any class.

Is this solution correct?

```
SELECT S1.sname  
FROM Student S1, Enrolled E  
WHERE S1.snum <> E.snum;
```

- **Schema**
 - Student (snum, sname, major, level, age)
 - Class (cname, room, fid)
 - Enrolled (snum, cname)
 - Faculty (fid, fname, deptid)

Q4: Find the name of students who are enrolled with the maximum number of classes.

Hint: Rephrase (change “maximum ...” to “greater than or equal to all others ...”)

- Find the name of students whose number of enrolled classes is greater than or equal to that of all other students.

Q4: Find the name of students enrolled in the maximum number of classes.

```
SELECT DISTINCT S.sname
FROM Student S
WHERE S.snum IN (SELECT E.snum
                  FROM Enrolled E
                  GROUP BY E.snum
                  HAVING COUNT (*) >= ALL
                        (SELECT COUNT (*)
                         FROM Enrolled E2
                         GROUP BY E2.snum))
```

- **Schema**
 - Student (snum, sname, major, level, age)
 - Class (cname, room, fid)
 - Enrolled (snum, cname)
 - Faculty (fid, fname, deptid)

Q5: Find the name of faculty members *who teach in every room in which* some class is taught

Q5: Find the name of faculty members *who teach in every room in which* some class is taught

- **Schema**

- Student (snum, sname, major, level, age)
- Class (cname, room, fid)
- Enrolled (snum, cname)
- Faculty (fid, fname, deptid)

- **Solution 1:**

- Way of thinking:
 - For each faculty member:
 - Find set A: all the rooms in which some class is taught
 - Find set B: all the rooms in which he/she has taught a class
 - Take the set difference $A - B$
 - Return the faculty member if $A - B$ is empty

Q5: Find the name of faculty members *who teach in every room in which* some class is taught

Solution 1:

```
SELECT F.fname
FROM Faculty F
WHERE NOT EXISTS (
    ( SELECT C.room FROM Class C )
    EXCEPT
    (SELECT DISTINCT C1.room FROM Class C1
    WHERE C1.fid = F.fid )
)
```

Note: DISTINCT can be omitted in the subquery, as *EXCEPT* removes duplicates automatically.

Q5: Correlations between Main and Sub-queries

- **Is this solution correct (note: no correlation between main and subquery)?**

```
SELECT F.fname
FROM Faculty F
WHERE NOT EXISTS (
    ( SELECT DISTINCT C.room FROM Class C )
    EXCEPT
    (SELECT DISTINCT C1.room
     FROM Class C1, Faculty F1
     WHERE C1.fid = F1.fid )
)
```


Q5: Correlations between Main and Sub-queries

```
SELECT F.fname
FROM Faculty F
WHERE NOT EXISTS (
    ( SELECT DISTINCT C.room
      FROM Class C )
  EXCEPT
  (SELECT DISTINCT C1.room
   FROM Class C1, Faculty F1
   WHERE C1.fid = F1.fid )
)
```

- **All faculty records are checked in the same way by the subquery**
 - If each class record has a non-empty fid (i.e., subquery returns empty set), all faculty names will be output;
 - otherwise, no faculty name will be output

Lesson: Always keep correlations between main & sub query.

Q5: Find the name of faculty members *who teach in every room in which* some class is taught

- **Solution 2:**

- Way of thinking:

- For each faculty member:

- Count number X of rooms he/she taught a class in

- Count number Y of rooms in which some class is taught

- Return the name of this faculty member if $X=Y$

Q5: Find the name of faculty members *who teach in every room in which* some class is taught

Solution 2:

```
SELECT F.fname
FROM Faculty F NATURAL JOIN Class C
GROUP BY F.fname, F.fid
HAVING COUNT(DISTINCT C.room) = (
    SELECT COUNT(DISTINCT C1.room)
    FROM Class C1)
```

Question:

- Can we remove two DISTINCTs in HAVING clause?
- Do we need to include fname in GROUP BY class?