

18. SELECT (5)

SELECT statement (5)

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

1

18. SELECT (5)

Sample database

```
CREATE TABLE Department(
  name      VARCHAR2(50),
  code      CHAR(5),
  total_staff_number NUMBER(2)      NOT NULL,
  Chair     VARCHAR2(50),
  budget    NUMBER(9,1)             NULL,
  CONSTRAINT dept_pkey PRIMARY KEY(name),
  CONSTRAINT dept_ckey1 UNIQUE(code),
  CONSTRAINT dept_ckey2 UNIQUE(chair),
  CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50) );
```

```
CREATE TABLE Course(
  c#        CHAR(7),
  title     VARCHAR2(200)          NOT NULL,
  credits   NUMBER(1)             NOT NULL,
  offered_by VARCHAR2(50)          NULL,
  CONSTRAINT course_pkey PRIMARY KEY(c#),
  CONSTRAINT course_check1
CHECK (credits IN (6, 12) ),
  CONSTRAINT course_fkey1 FOREIGN KEY(offered_by)
REFERENCES Department(name) );
```

© Janusz R. Getta

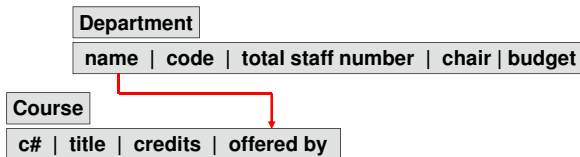
CSCI235/MCS9235 Databases, SCSSE, Spring 2009

2

18. SELECT (5)

Nested queries

Find the titles of courses offered by a department chaired by Peter



Q1: Find a department chaired by Peter

Q2: Find the titles of courses offered by a department found in Q1

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

3

18. SELECT (5)

Nested queries

Find the titles of courses offered by a department chaired by Peter

Q2

```
SELECT title
FROM Course
WHERE offered_by =
```

Q1

```
( SELECT name
  FROM Department
  WHERE chair = 'Peter' );
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

4

18. SELECT (5)

Nested queries

Find the chairpersons of all departments that offer 12 credit points course(s)

```
SELECT chair
FROM Department
WHERE name IN
( SELECT offered_by
  FROM Course
  WHERE credits = 12 );
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

5

18. SELECT (5)

Computational model (nested queries)

```
SELECT <ATTRIBUTES>
FROM <TABLE_1>
WHERE <ATTRIBUTE> IN
      (SELECT <ATTRIBUTE>
       FROM <TABLE_2>
       WHERE <CONDITION_2>);
```

```
forall rows s in <TABLE_1>
  forall rows t in <TABLE_2>
    if evaluate(<CONDITION_2>, t) then
      TEMP ← append(t.<ATTRIBUTE>)
    endif;
  endforall;
  if s.<ATTRIBUTE> in TEMP then
    output(s.<ATTRIBUTES>)
  endforall;
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

6

18. SELECT (5)

Computational model (nested queries)

```
SELECT <ATTRIBUTES>
FROM <TABLE_1>
WHERE <ATTRIBUTE> IN
      (SELECT <ATTRIBUTE>
       FROM <TABLE_2>
       WHERE <CONDITION_2>);
```

```
forall rows t in <TABLE_2>
  if evaluate(<CONDITION_2>, t) then
    TEMP ← append(t.<ATTRIBUTE>)
  endif;
endforall;
forall rows s in <TABLE_1>
  if s.<ATTRIBUTE> in TEMP then
    output(s.<ATTRIBUTES>)
  endif;
endforall;
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

7

18. SELECT (5)

Nested queries

Find the names of all students who enrolled the courses offered by the department of Physics

Course Q1: Find the courses (c#) offered by Physics

c#	title	credits	offered_by
----	-------	---------	------------

c#	s#	edate	mark
----	----	-------	------

Enrolment

Q2: Find students (s#) who enrolled courses found in Q1

s#	name	degree
----	------	--------

Student

Q3: Find the names (name) of students found in Q2

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

8

18. SELECT (5)

Nested queries

Find the names of all students who enrolled the courses offered by the department of Physics

Q3

```
SELECT name
FROM Student
WHERE s# IN
```

Q2

```
( SELECT s#
  FROM Enrolment
  WHERE c# IN
```

Q1

```
( SELECT c#
  FROM Course
  WHERE offered_by = 'Physics' ) );
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

9

18. SELECT (5)

Group functions revisited

Find the titles, and credit points of all courses that earn the largest amount of credit points

```
SELECT title, credits
FROM Course
WHERE credits =
      ( SELECT MAX(credits)
        FROM Course );
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

10

18. SELECT (5)

Group functions revisited

Find the names of departments together with a total number of courses offered by each one of them

```
SELECT offered_by, count(title)
FROM Course
GROUP BY offered_by
```

UNION

```
SELECT name, 0
FROM Department
WHERE name NOT IN
```

```
( SELECT offered_by
  FROM Course );
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

11

18. SELECT (5)

Grouping revisited

Find the names of departments that offer the largest number of courses together with the total number of courses offered

```
SELECT offered_by, count(title)
FROM Course
GROUP BY offered_by
HAVING count(title) =
      ( SELECT MAX(count(title))
        FROM Course
        GROUP BY offered_by );
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

12

Reference

Elmasri R., Navathe S. B., *Database Systems*, chapter 6

Ramakrishnan R., Gehrke J., *Database Management Systems*, chapter 5.4

<https://sai.uow.edu.au/oradocs/>
SQL Reference, **SELECT** statement