SELECT
statement
(4)

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) );
```
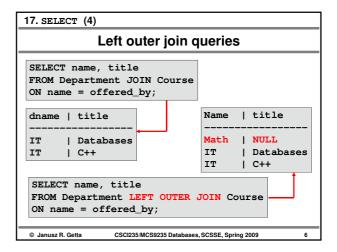

Outer join queries

Find the names of departments together with the titles of all courses offered by each department

Department

| name | code | total staff number | chair | budget |

Course

| c# | title | credits | offered by |

What about the departments that do not offer courses and what about the courses do not assigned to departments ?

```
SELECT name, title
FROM Department JOIN Course
ON name = offered_by;
```

Left outer join queries

Find the names of departments together with the titles of all courses offered by each department

Include the names of departments that offer no courses at all

```
SELECT name, title
FROM Department LEFT OUTER JOIN Course
ON name = offered_by;
```

```
SELECT name, title
FROM Department JOIN Course
ON name = offered_by(+);
```

Left outer join queries

Department

```
Name  | code  | total-staff-number | chair
--------------------------------------------------
Math  | M1    | 5                  | Ralph
IT    | I1    | 55                 | Peter
```

Course

```
C#     | title     | credits | offered-by
------------------------------------------------
CS111  | C++       | 6       | IT
CS235  | Databases | 6       | IT
```

Left outer join queries

```
SELECT name, title
FROM Department JOIN Course
ON name = offered_by;
```

```
dname | title
------------------
IT    | Databases
IT    | C++
```

```
Name  | title
------------------
Math  | NULL
IT    | Databases
IT    | C++
```

```
SELECT name, title
FROM Department LEFT OUTER JOIN Course
ON name = offered_by;
```
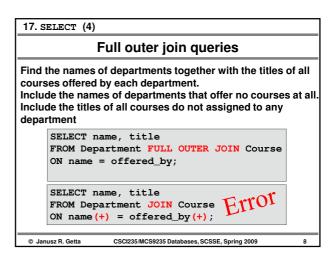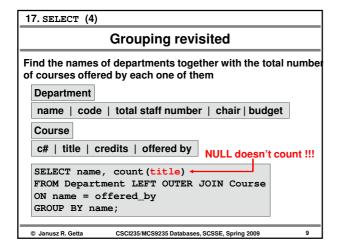
1

Right outer join queries

Find the names of departments together with the titles of all courses offered by each department.

Include the titles of all courses do not assigned to any department

```
SELECT name, title
FROM Department RIGHT OUTER JOIN Course
ON name = offered_by;
```

```
SELECT name, title
FROM Department JOIN Course
ON name(+) = offered_by;
```

Full outer join queries

**Find the names of departments together with the titles of all courses offered by each department.
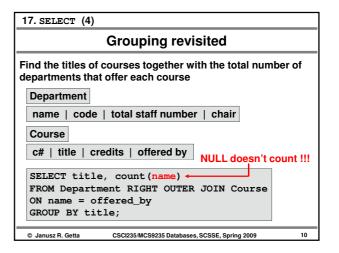Include the names of departments that offer no courses at all.
Include the titles of all courses do not assigned to any department**

```
SELECT name, title
FROM Department FULL OUTER JOIN Course
ON name = offered_by;
```

```
SELECT name, title
FROM Department JOIN Course     Error
ON name(+) = offered_by(+);
```

Grouping revisited

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

Department

| name | code | total staff number | chair | budget |

Course

| c# | title | credits | offered by |

NULL doesn't count !!!

```
SELECT name, count(title) ←
FROM Department LEFT OUTER JOIN Course
ON name = offered_by
GROUP BY name;
```

Grouping revisited

Find the titles of courses together with the total number of departments that offer each course

Department

| name | code | total staff number | chair |

Course

| c# | title | credits | offered by |

NULL doesn't count !!!

```
SELECT title, count(name) ←
FROM Department RIGHT OUTER JOIN Course
ON name = offered_by
GROUP BY title;
```

References

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

Ramakrishnan R., Gehrke J., *Database Management Systems*, chapter 5.6.4

```
https://sai.uow.edu.au/oradocs/
```
SQL Reference, SELECT statement