

15. SELECT (2)

SELECT statement (2)

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

1

15. SELECT (2)

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

15. SELECT (2)

Queries with a simple condition

Find the titles of all 6 credit points courses

```
SELECT title
FROM Course
WHERE credits = 6;
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

3

15. SELECT (2)

Queries with a Boolean expression

Find the titles of all 6 credit points courses offered by a department of Physics

```
SELECT title
FROM Course
WHERE (credits = 6) AND (offered_by = 'Physics');
```

Find the titles of all 6 credit points courses or the titles of all courses offered by a department of Physics

```
SELECT title
FROM Course
WHERE (credits = 6) OR (offered_by = 'Physics');
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

4

15. SELECT (2)

Queries with a Boolean expression

Find the titles of all courses other than 6 credit points courses

```
SELECT title
FROM Course
WHERE NOT (credits = 6);
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

5

15. SELECT (2)

Queries with a Boolean expression

Find the titles of all courses other than 6 credit points courses and other than courses offered by a department of Physics

```
SELECT title
FROM Course
WHERE NOT (credits = 6) AND
      NOT (offered_by = 'Physics');
```

```
SELECT title
FROM Course
WHERE NOT ( (credits = 6) OR
            (offered_by = 'Physics') );
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

6

15. SELECT (2)

Queries with a Boolean expression

Find the titles of all courses other than 6 credit points courses or other than courses offered by a department of Physics

```
SELECT title
FROM Course
WHERE NOT (credits = 6) OR
      NOT (offered_by = 'Physics');
```

```
SELECT title
FROM Course
WHERE NOT ( (credits = 6) AND
            (offered_by = 'Physics') );
```

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 7

15. SELECT (2)

Queries with a Boolean expression

Find the titles of all courses offered by a department of Physics or offered by a department of Mathematics

```
SELECT title
FROM Course
WHERE (offered_by = 'Physics') OR
      (offered_by = 'Mathematics');
```

```
SELECT title
FROM Course
WHERE offered_by IN ('Physics', 'Mathematics');
```

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 8

15. SELECT (2)

Queries with a Boolean expression

Find the same titles of all courses offered by a department of Physics and offered by a department of Mathematics

```
SELECT title
FROM Course
WHERE (offered_by = 'Physics') AND
      (offered_by = 'Mathematics');
```

WRONG !!!

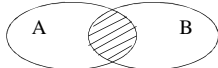
© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 9

15. SELECT (2)

Set algebra queries

Find the titles of all 6 credit points courses offered by a department of Physics

```
SELECT title
FROM Course
WHERE credits = 6
INTERSECT
SELECT title
FROM Course
WHERE offered_by = 'Physics';
```



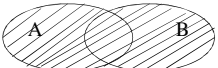
© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 10

15. SELECT (2)

Set algebra queries

Find the titles of all 6 credit points courses or the titles of courses offered by a department of Physics

```
SELECT title
FROM Course
WHERE credits = 6
UNION
SELECT title
FROM Course
WHERE offered_by = 'Physics';
```




© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 11

15. SELECT (2)

Set algebra queries

Find the titles of all 6 credit points courses and not offered by a department of Physics

```
SELECT title
FROM Course
WHERE credits = 6
MINUS
SELECT title
FROM Course
WHERE offered_by = 'Physics';
```



© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 12

15. SELECT (2)

Set algebra queries

Find the same titles of all courses offered by a department of Physics and offered by a department of Mathematics

```
SELECT title
FROM Course
WHERE offered_by = 'Physics'
INTERSECT
SELECT title
FROM Course
WHERE offered_by = 'Mathematics';
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

13

15. SELECT (2)

Computational model (set operations)

```
SELECT <ATTRIBUTES>
FROM <TABLE_1>
WHERE <CONDITION_1>
<SET OPERATION>
SELECT <ATTRIBUTES>
FROM <TABLE_2>
WHERE <CONDITION_2>;

forall rows t in <TABLE_1>
  if evaluate(<CONDITION_1>, t) then
    TEMP_1 ← append(t.<ATTRIBUTES>)
  endif;
endforall;
forall rows t in <TABLE_2>
  if evaluate(<CONDITION_2>, t) then
    TEMP_1 ← append(t.<ATTRIBUTES>)
  endif;
endforall;
<TEMP_1> <SET OPERATION> <TEMP_2>;
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

14

15. SELECT (2)

Sorting

Find the titles of all 6 credit points courses sorted in ascending order

```
SELECT title
FROM Course
WHERE credits = 6
ORDER BY title ASC;
```

Find the titles of all 6 credit point courses sorted in descending order

```
SELECT title
FROM Course
WHERE credits = 6
ORDER BY title DESC;
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

15

15. SELECT (2)

Computational model (sorting)

```
SELECT <ATTRIBUTES>
FROM <TABLE>
WHERE <CONDITION>
ORDER BY <SORT ATTRIBUTES> <MODE>

forall rows t in <TABLE_1>
  if evaluate(<CONDITION_1>, t) then
    TEMP ← append(t.<ATTRIBUTES>)
  endif;
endforall;
Sort TEMP by <SORT ATTRIBUTES> in <MODE> way;
```

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

16

15. SELECT (2)

Queries about lack of values (NULLs)

Find the titles of all courses which are not offered now

```
SELECT title
FROM Course
WHERE offered_by IS NULL;
```

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

WRONG !!!

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

17

15. SELECT (2)

Queries about lack of values (NULLs)

Find the titles of all courses offered now

```
SELECT title
FROM Course
WHERE offered_by IS NOT NULL;
```

```
SELECT title
FROM Course
WHERE offered_by <> NULL;
```

WRONG !!!

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

18

15. SELECT (2)

Grouping

Find the names of all departments that offer any courses

```
SELECT DISTINCT offered_by
FROM Course;
```

```
SELECT offered_by
FROM Course
GROUP BY offered_by;
```

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 19

15. SELECT (2)

Grouping

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

```
SELECT offered_by, count(*)
FROM Course
GROUP BY offered_by;
```

c#	title	credits	offered_by
C235	Databases	6	Computer Science
P111	Mechanics	12	Physics
C111	C++	6	Computer Science
M111	Calculus	12	Mathematics
P312	Relativity	6	Physics

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 20

15. SELECT (2)

Grouping

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

```
SELECT offered_by, count(*)
FROM Course
GROUP BY offered_by;
```

c#	title	credits	offered_by
C235	Databases	6	Computer Science
C111	C++	6	Computer Science
M111	Calculus	12	Mathematics
P111	Mechanics	12	Physics
P312	Relativity	6	Physics

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 21

15. SELECT (2)

Grouping

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

```
SELECT offered_by, count(*)
FROM Course
GROUP BY offered_by;
```

offered_by	count (*)
Computer Science	2
Mathematics	1
Physics	2

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 22

15. SELECT (2)

Remember !!!

Find the titles and the total number of courses offered by each department

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

Wrong !!!

It is not allowed to select an attribute (**title**) which is not included in a GROUP BY clause (**offered_by**) !!!

```
SELECT title, offered_by
FROM Course
ORDER BY offered_by;
```

```
SELECT offered_by, count(*)
FROM Course
GROUP BY offered_by;
```

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 23

15. SELECT (2)

Computational model (Grouping)

```
SELECT <GROUP ATTRIBUTES>, <GROUP FUNCTIONS>
FROM <TABLE>
WHERE <CONDITION>
GROUP BY <GROUP ATTRIBUTES>
```

```
forall rows t in <TABLE>
  if evaluate(<CONDITION_1>, t) then
    TEMP ← append(t.<ATTRIBUTES>)
  endif;
endforall;
```

Sort TEMP by <GROUP ATTRIBUTES>;

```
forall groups g(<GROUP ATTRIBUTES>) in TEMP
  output(<GROUP ATTRIBUTES>, compute(g, <GROUP FUNCTIONS>));
endforall;
```

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 24

15. SELECT (2)

Grouping

Find the largest number of courses offered by any department

```
SELECT max(count(*))
FROM Course
GROUP BY offered_by;
```

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 25

15. SELECT (2)

Grouping with selection

Find the names of all departments that offer more than 1 course

```
SELECT offered_by
FROM Course
GROUP BY offered_by
HAVING count(*) > 1;
```

offered_by	count(*)
Computer Science	2
Mathematics	1
Physics	2

offered_by
Computer Science
Physics

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 26

15. SELECT (2)

Computational model (Grouping with selection)

```
SELECT <GROUP ATTRIBUTES>, <GROUP FUNCTIONS>
FROM <TABLE>
WHERE <CONDITION>
GROUP BY <GROUP ATTRIBUTES>
HAVING <GROUP CONDITION>;

forall rows t in <TABLE>
  if evaluate(<CONDITION_1>, t) then
    TEMP ← append(t.<ATTRIBUTES>)
  endif;
endforall;

Sort TEMP by <GROUP ATTRIBUTES>;

forall groups g(<GROUP ATTRIBUTES>) in TEMP
  if evaluate(g, <GROUP CONDITION>) then
    output(<GROUP ATTRIBUTES>, compute(g, <GROUP FUNCTIONS>));
  endif;
endforall;
```

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 27

15. SELECT (2)

References

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

Ramakrishnan R., Gehrke J., *Database Management Systems*, chapter 5.3, 5.5.1

<https://sai.uow.edu.au/oradocs/>

SQL Reference, SELECT statement

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009 28