

## Laboratory work 3

Student: Almat Begaidarov

Instructor: Aibek Kuralbaev

Assistant: Bermagambet Duisek

1. Write the following queries in SQL, using the university schema:

- Find all courses worth more than 3 credits;
- Find all classrooms situated either in 'Watson' or 'Packard' buildings;
- Find all courses offered by the Computer Science department;
- Find all courses offered during fall;
- Find all students who have more than 45 credits but less than 90;
- Find all students whose names end with vowels;
- Find all courses which have course 'CS-101' as their prerequisite;

a)

```
SELECT * FROM course
WHERE credits > 3;
```

	course_id	title	dept_name	credits
1	BI0-101	Intro. to Biology	Biology	4
2	BI0-301	Genetics	Biology	4
3	CS-101	Intro. to Computer ...	Comp. Sci.	4
4	CS-190	Game Design	Comp. Sci.	4
5	PHY-101	Physical Principles	Physics	4

b)

```
SELECT * FROM classroom
WHERE building in ('Watson', 'Packard');
```

	building	room_number	capacity
1	Packard	101	500
2	Watson	100	30
3	Watson	120	50

c)

```
SELECT * FROM course
WHERE dept_name = 'Comp. Sci.';
```

	course_id	title	dept_name	credits
1	CS-101	Intro. to Computer Science	Comp. Sci.	4
2	CS-190	Game Design	Comp. Sci.	4
3	CS-315	Robotics	Comp. Sci.	3
4	CS-319	Image Processing	Comp. Sci.	3
5	CS-347	Database System Concepts	Comp. Sci.	3

d)

```
SELECT course.course_id, course.title, course.dept_name,
course.credits, section.semester FROM course, section
WHERE course.course_id = section.course_id AND section.semester =
'Fall';
```

	course_id	title	dept_name	credits	semester
1	CS-101	Intro. to Computer Science	Comp. Sci.	4	Fall
2	CS-347	Database System Concepts	Comp. Sci.	3	Fall
3	PHY-101	Physical Principles	Physics	4	Fall

e)

```
SELECT * FROM student
WHERE tot_cred BETWEEN 46 AND 89;
```

	id	name	dept_name	tot_cred
1	19991	Brandt	History	80
2	44553	Peltier	Physics	56
3	45678	Levy	Physics	46
4	54321	Williams	Comp. Sci.	54
5	76543	Brown	Comp. Sci.	58
6	76653	Aoi	Elec. Eng.	60

f)

```
SELECT * FROM student
WHERE name SIMILAR TO '%[aeuioy]';
```

	id	name	dept_name	tot_cred
1	45678	Levy	Physics	46
2	76653	Aoi	Elec. Eng.	60
3	98988	Tanaka	Biology	120

g)

```
SELECT course.course_id, prereq.prereq_id, course.title,
course.dept_name, course.credits FROM course, prereq
WHERE course.course_id = prereq.course_id AND prereq.prereq_id =
'CS-101';
```

	course_id	prereq_id	title	dept_name	credits
1	CS-190	CS-101	Game Design	Comp. Sci.	4
2	CS-315	CS-101	Robotics	Comp. Sci.	3
3	CS-319	CS-101	Image Processing	Comp. Sci.	3
4	CS-347	CS-101	Database System Concepts	Comp. Sci.	3

2. Write the following queries in SQL, using the university schema:

- For each department, find the average salary of instructors in that department and list them in ascending order. Assume that every department has at least one instructor;
- Find the building where the biggest number of courses takes place;
- Find the department with the lowest number of courses offered;
- Find the ID and name of each student who has taken more than 3 courses from the Computer Science department;
- Find all instructors who work either in Biology, Philosophy, or Music departments;
- Find all instructors who taught in the 2018 year but not in the 2017 year;

a)

```
SELECT dept_name, avg(salary) as average_salary from instructor
GROUP BY dept_name ORDER BY avg(salary) ASC;
```

	dept_name	average_salary
1	Music	40000
2	History	61000
3	Biology	72000
4	Comp. Sci.	77333.33333333333
5	Elec. Eng.	80000
6	Finance	85000
7	Physics	91000

b)

```
SELECT building, count(course.course_id) as number_of_courses FROM
course, department
WHERE course.dept_name = department.dept_name
GROUP BY building ORDER BY count(course.course_id) DESC LIMIT 1; -
- FETCH FIRST 1 ROWS ONLY
```

	building	number_of_courses
1	Taylor	6

c)

```
SELECT dept_name, count(course_id) FROM course
GROUP BY dept_name
HAVING count(course_id) = (SELECT min(cnt) FROM (select
count(course_id) as cnt FROM course GROUP BY dept_name) as
min_cnt);
```

	dept_name	count
1	Finance	1
2	History	1
3	Physics	1
4	Music	1
5	Elec. Eng.	1

d)

```
SELECT student.id, name, count(course_id) FROM student, takes
WHERE student.id = takes.id AND student.dept_name = 'Comp. Sci.'
GROUP BY student.id, student.name
HAVING count(course_id) > 3;
```

	id	name	count
1	12345	Shankar	4

e)

```
SELECT * FROM instructor
WHERE dept_name in ('Philosophy', 'Biology', 'Music');
```

	id	name	dept_name	salary
2	76766	Crick	Biology	72000.00

f)

```
SELECT * FROM instructor
WHERE name not in (SELECT instructor.name FROM instructor, teaches
WHERE instructor.id = teaches.id AND teaches.year = 2017 AND
teaches.year <> 2018);
```

	id	name	dept_name	salary
1	12121	Wu	Finance	90000.00
2	15151	Mozart	Music	40000.00
3	32343	El Said	History	60000.00
4	33456	Gold	Physics	87000.00
5	45565	Katz	Comp. Sci.	75000.00
6	58583	Califieri	History	62000.00
7	76543	Singh	Finance	80000.00

3. Write the following queries in SQL, using the university schema:

- Find all students who have taken Comp. Sci. course and got an excellent grade (i.e., A, or A-) and sort them alphabetically;
- Find all advisors of students who got grades lower than B on any class;
- Find all departments whose students have never gotten an F or C grade;
- Find all instructors who have never given an A grade in any of the courses they taught;
- Find all courses offered in the morning hours (i.e., courses ending before 13:00);

a)

```
SELECT student.id, student.name, student.dept_name, course.title,
takes.grade FROM student, takes, course
WHERE student.id = takes.id AND takes.course_id = course.course_id
AND course.dept_name = 'Comp. Sci.' AND takes.grade in ('A-', 'A')
ORDER BY student.name ASC;
```

	id	name	dept_name	title	grade
1	76543	Brown	Comp. Sci.	Image Processing	A
2	76543	Brown	Comp. Sci.	Intro. to Computer Science	A
3	12345	Shankar	Comp. Sci.	Database System Concepts	A
4	12345	Shankar	Comp. Sci.	Game Design	A
5	12345	Shankar	Comp. Sci.	Robotics	A
6	54321	Williams	Comp. Sci.	Intro. to Computer Science	A-
7	00128	Zhang	Comp. Sci.	Database System Concepts	A-
8	00128	Zhang	Comp. Sci.	Intro. to Computer Science	A

b)

```
SELECT DISTINCT advisor.s_id, advisor.i_id, student.name FROM
advisor, student, takes
WHERE takes.id = advisor.s_id AND student.id = takes.id AND
takes.grade not in ('B', 'B+', 'A-', 'A');
```

	s_id	i_id	name
1	23121	76543	Chavez
2	12345	10101	Shankar
3	76653	98345	Aoi
4	98765	98345	Bourikas
5	45678	22222	Levy
6	44553	22222	Peltier

c)

```
SELECT DISTINCT dept_name FROM course
WHERE dept_name not in (SELECT dept_name FROM course, takes
WHERE course.course_id = takes.course_id AND takes.grade in
('F','C'));
```

	dept_name
1	History
2	Physics
3	Music
4	Biology

d)

```
ELECT DISTINCT * FROM instructor
WHERE name not in (SELECT instructor.name FROM takes, teaches,
instructor
WHERE takes.course_id = teaches.course_id AND teaches.id =
instructor.id AND takes.grade = 'A');
```

	id	name	dept_name	salary
1	12121	Wu	Finance	90000.00
2	15151	Mozart	Music	40000.00
3	22222	Einstein	Physics	95000.00
4	32343	El Said	History	60000.00
5	33456	Gold	Physics	87000.00
6	58583	Califieri	History	62000.00
7	76543	Singh	Finance	80000.00
8	98345	Kim	Elec. Eng.	80000.00

e)

```
SELECT DISTINCT course.title, time_slot.start_hr,  
time_slot.start_min, time_slot.end_hr, time_slot.end_min FROM  
course, section, time_slot  
WHERE course.course_id = section.course_id AND  
section.time_slot_id = time_slot.time_slot_id AND  
section.time_slot_id in ('A','B','C');
```

	title	start_hr	start_min	end_hr	end_min
1	Database System Concepts	8	0	8	50
2	Game Design	8	0	8	50
3	Genetics	8	0	8	50
4	Image Processing	9	0	9	50
5	Image Processing	11	0	11	50
6	Intro. to Biology	9	0	9	50
7	Intro. to Digital Systems	11	0	11	50
8	Investment Banking	9	0	9	50
9	Physical Principles	8	0	8	50
10	World History	11	0	11	50