Åbo Akademi University

Advance Course on Database

Weekly Report 1

Luis Araújo(2004624)

May 27, 2021

Contents

Introduction

For this first assignment, the teacher proposed a set of exercises of Relational Algebra and SQL. The team decided to first take a look at the exercises, and try to solve them on our own. After that, we discussed the different solutions and tried to find the best one.
Once Filipe and I are leaving together the communication between us was very smooth and continuous. We also had an online session to discuss mainly the Relational Algebra part with Charles Oredola.

Chapter 1

Task 1

For the first question, you should use the relational algebra calculator available on https://dbis-uibk.github.io/relax. Choose the dataset Silberschatz - UniversityDB for this exercise.

1.1 a)

List the *course_id* and *title* of the courses in the department "Biology" that are worth 4 credit points.

$$\Pi_{course.course\_id,\ course.title}(\sigma_{course.dept\_name\ =\ 'Biology'\ \wedge\ course.credits\ \geq\ 4}(course))$$

Result Grid		
#	course_id	title
1	BIO-101	Intro. to Biology
2	BIO-301	Genetics

Review

A minor mistake made when trying to filter the course credits that are worth 4 credits, instead, we filtered the course credits that are worth 4 or more credits.

1.2 b)

List the *course_id* and *title* of the courses that the instructor "Srinivasan" teaches.

$$join1 \leftarrow (instructor) \bowtie_{instructor.ID\ =\ teaches.ID} (teaches)$$
$$join2 \leftarrow (join1) \bowtie_{teaches.course\_id\ =\ course.course\_id} (course)$$
$$where \leftarrow \sigma_{instructor.name\ =\ 'Srinivasan'}(join2)$$
$$\Pi_{(course.course\_id,\ course.title)}(where)$$

Result Grid		
#	course_id	title
1	CS-101	Intro. to Computer Science
2	CS-315	Robotics
3	CS-347	Database System Concepts

Review

Correct exercise.

1.3 c)

List the *student ID*, *name* and *dept_name* of all students that have taken courses held by the instructor "Srinivasan".

$$join1 \leftarrow (instructor) \bowtie_{instructor.ID\ =\ teaches.ID} (teaches)$$
$$join2 \leftarrow (join1) \bowtie_{teaches.course\_id\ =\ course.course\_id} (course)$$
$$where \leftarrow \sigma_{instructor.name\ =\ 'Srinivasan'}(join2)$$
$$join3 \leftarrow (student) \bowtie_{student.ID\ =\ takes.ID\_id} (takes)$$
$$join4 \leftarrow (where) \bowtie_{teaches.course\_id\ =\ takes.course\_id} (join3)$$
$$\Pi_{(course.course\_id,\ course.title)}(join4)$$

Result Grid			
#	ID	name	dept_name
1	128	Zhang	Comp. Sci.
2	12345	Shankar	Comp. Sci.
3	45678	Levy	Physics
4	54321	Williams	Comp. Sci.
5	76543	Brown	Comp. Sci.
6	98765	Bourikas	Elec. Eng.

Review

This exercise should be correct since we also obtained the 6 tuples. However, it is not the best solution, once we made one more join than necessary when connecting the students to the instructor.

1.4 d)

List the *name*, *dept_name* and *salary* of the instructors that have the highest salary.

$$\tau_{[3],\ desc}(\pi_{instructor.name,\ instructor.dept\_name,\ instructor.salary}(instructor))$$

Result Grid			
#	name	dept_name	salary
1	Einstein	Physics	95000
2	Brandt	Comp. Sci.	92000
3	Wu	Finance	90000
4	Gold	Physics	87000
...			
12	Mozart	Music	40000

Review

In this exercise, we have made the mistake of ranking teachers based on their salary instead of selecting the teachers that earned the highest salary.

1.5 e)

List the instructor *ID*, *name*, *department* and the total number of students that each instructor supervises.

$$join \leftarrow (advisor) \bowtie advisor.s\_id = instructor.ID(instructor)$$
$$\gamma_{instructor.ID,\ instructor.name,\ instructor.dept\_name;\ count(advisor.s\_id)\ ->\ number\_students}(join)$$

Result Grid				
#	ID	name	dept_name	number_students
1	45565	Katz	Comp. Sci.	2
2	10101	Srinivasan	Comp. Sci.	1
3	76543	Singh	Finance	1
4	22222	Einstein	Physics	2
5	98345	Kim	Elec. Eng.	2
6	76766	Crick	Biology	1

Review

Correct exercise.

1.6 f)

List the *course_id*, *title* and *name* of the instructor for all courses that were held in the building 'Packard' in 2010.

$$join1 \leftarrow (instructor) \bowtie_{instructor.ID\ =\ teaches.ID} (teaches)$$
$$join2 \leftarrow (join1) \bowtie_{teaches.course\_id\ =\ course\_id}\ course$$
$$join3 \leftarrow (join2) \bowtie_{instructor.dept\_name\ =\ department.dept\_name}\ department$$
$$where \leftarrow \sigma_{department.building\ =\ 'Packard'\ \wedge\ teaches.year\ =\ 2010}(join3)$$
$$\Pi_{course.course\_id,\ course.title,\ instructor.name}(where)$$

Result Grid			
#	course_id	title	name
1	BIO-101	Intro. to Biology	Mozart
2	BIO-301	Genetics	Mozart
3	BIO-399	Computational Biology	Mozart
4	CS-101	Intro. to Computer Science	Mozart
...			
13	PHY-101	Physical Principles	Mozart

Review

Analyzing this exercise, the only mistake done was when we used the "where". Instead of using it in the beginning, we used it at the end.

Chapter 2

Task 2

For the second question you should use a MySQL database server, either the one available at `babbage2.abo.fi` or your own MySQL server.

First and foremost we import the desired database:

```
/*Select the database 'university' in the SQL schema.*/
USE university;
```

2.1 a)

List the *course_id* and *title* of the courses in the department "Finance" that are worth 4 credit points.

```
SELECT
    course_id, title
FROM
    course
WHERE
    dept_name = 'Finance' AND credits = 4;
```

Result Grid		
#	course_id	title
1	304	Music 2 New for your Instructor
2	319	World History
3	349	Networking
4	362	Embedded Systems
5	586	Image Processing
6	781	Compiler Design
7	922	Microeconomics

Review

Correct exercise.

2.2 b)

List the *course_id* and *title* of the courses that the instructor "Dale" teaches.

```
SELECT DISTINCT
    course.course_id, title
```

```
FROM
    course
        JOIN
    teaches ON course.course_id = teaches.course_id
        JOIN
    instructor ON teaches.ID = instructor.ID
WHERE
    name = 'Dale';
```

Result Grid		
#	course_id	title
1	158	Elastic Structures
2	237	Surfing
3	496	Aquatic Chemistry
4	629	Finite Element Analysis
5	748	Tort Law
6	802	African History
7	893	Systems Software
8	927	Differential Geometry

Review

Correct exercise.

2.3 c)

Which courses have the student 'Tuomisto' passed? The result should include the *course_id*, *course title*, *grade* and number of *credits* of the course.

```
SELECT
    course.course_id, title, grade, credits
FROM
    student
        JOIN
    takes ON student.ID = takes.ID
        JOIN
    course ON takes.course_id = course.course_id
WHERE
    name = 'Tuomisto';
```

Result Grid				
#	course_id	title	grade	credits
1	239	The Music of the Ramones	A+	4
2	345	Race Car Driving	A+	4
3	362	Embedded Systems	B-	4
4	400	Visual BASIC	A	4
5	408	Bankruptcy	C	3
6	545	International Practicum	B-	3
7	571	Plastics	C	4
8	612	Mobile Computing	B	3
9	692	Cat Herding	C	3
10	702	Arabic	C+	3
11	760	How to Groom your Cat	A	3
12	795	Death and Taxes	A	3
13	808	Organic Chemistry	B+	4
14	852	World History	C+	4
15	875	Bioinformatics	B	3
16	893	Systems Software	B	3
17	974	Astronautics	A+	3

Review

In this exercise, we forgot to filter the graves above 'F' and also to remove the entries that had the value on that same column as null.

2.4 d)

List all students that have taken courses by the instructor 'Morris'. The result should contain the student *ID* and *name*, and the *course_id* and course *title*, and be ordered alphabetically on student *name*.

```
SELECT
    student.id, student.name, course.course_id, course.title
FROM
    instructor
        JOIN
    teaches ON instructor.id = teaches.id
        JOIN
    course ON teaches.course_id = course.course_id
        JOIN
    takes ON teaches.course_id = takes.course_id
        JOIN
    student ON takes.id = student.id
WHERE
    instructor.name = 'Morris'
ORDER BY student.name;
```

Result Grid				
#	ID	name	course_id	title
1	5144	Abdellatif	795	Death and Taxes
2	5144	Abdellatif	313	International Trade
3	78858	Abdul-Rahman	242	Rock and Roll
4	20244	Abu-B	791	Operating Systems
5	83622	Achilles	696	Heat Transfer
...				
1475	38121	Zuyev	313	International Trade

In order to check if this query was correct I split it in two queries. The first get the courses that the instructor is 'Morris' *ID*'s and second get the requested information based on the *ID*s

```
SELECT DISTINCT
    course.course_id
FROM
    course
        JOIN
    teaches ON course.course_id = teaches.course_id
        JOIN
    instructor ON teaches.ID = instructor.ID
WHERE
    name = 'Morris';
```

Result Grid	
#	course_id
1	242
2	313
3	696
4	791
5	795

```
SELECT student.ID, name, course.course_id , course.title
FROM
    student
        JOIN
    takes ON student.ID = takes.ID
        JOIN
    course ON takes.course_id = course.course_id
WHERE
    course.course_id IN ('242','313','696','791','795')
ORDER BY name;
```

Result Grid				
#	ID	name	course_id	title
1	5144	Abdellatif	795	Death and Taxes
2	5144	Abdellatif	313	International Trade
3	78858	Abdul-Rahman	242	Rock and Roll
4	20244	Abu-B	791	Operating Systems
5	83622	Achilles	696	Heat Transfer
...				
1475	38121	Zuyev	313	International Trade

Review

Correct exercise.

2.5 e)

List the *name*, *dept_name* and *salary* of the instructors that have the highest salary.

```
SELECT
    name, dept_name, salary
FROM
    instructor
ORDER BY salary DESC;
```

Result Grid			
#	name	dept$_{n}ame$	salary
1	Wieland	Pol. Sci.	124651.41
2	Voronina	Physics	121141.99
3	Mird	Marketing	119921.41
4	Sakurai	English	118143.98
5	Bietzk	Cybernetics	117836.50
...			
50	Lembr	Accounting	32241.56

Review

In this exercise we have made the same mistake from the task 1d, of ranking teachers based on their salary instead of selecting the teachers that earned the highest salary.

2.6 f)

List the instructor *ID*, *name*, *department* and the total number of students that each instructor supervises.

```
SELECT
    instructor.ID,
    name,
    dept_name,
    COUNT(takes.ID) AS number_students
FROM
    instructor
        JOIN
    teaches ON instructor.ID = teaches.ID
        JOIN
    takes ON teaches.course_id = takes.course_id
GROUP BY instructor.ID;
```

Later I found the table advisor that connects directly students and instructors, so where is an alternative way to do it:

```
SELECT
    instructor.ID,
    name,
    dept_name,
    COUNT(advisor.s_ID) AS number_students
FROM
```

```
    instructor
        JOIN
    advisor ON instructor.ID = advisor.i_ID
GROUP BY instructor.ID;
```

Result Grid				
#	ID	name	dept_name	number_students
1	14365	Lembr	Accounting	870
2	15347	Bawa	Athletics	266
3	19368	Wieland	Pol. Sci.	910
4	22591	DAgostino	Psychology	5391
5	25946	Liley	Languages	338
...				
31	99052	Dale	Cybernetics	3658

Review

Correct exercise.

Chapter 3

Conclusion

Having an overview of the assignment, I could say that the exercises helped me improve my skills in relational algebra and SQL. Even though we had some mistakes, we don't think that they were major ones, having only minor failures.