

BASES DE DATOS

Practica 1

Manejo Lenguaje SQL

Autor

Daniel Khomyakov Trubnikov
Ekaitz Arriola Garcia
Gerardo Escudero López

Profesor

Rafael Socas Gutierrez

March 27, 2021

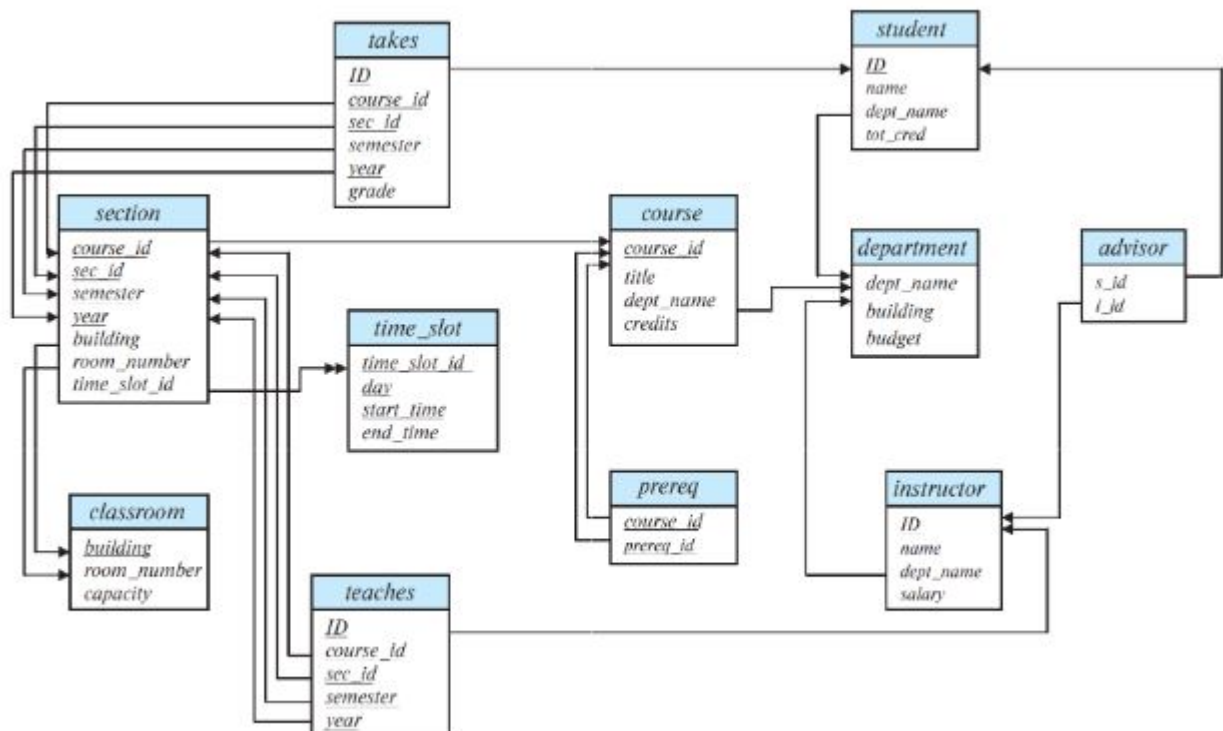
Índice

| | |
|---|----|
| Introducción | 2 |
| Identificación de los atributos de la tabla instructor | 3 |
| Confirmación de la identificación de los atributos de la tabla instructor | 3 |
| Queries con la tabla instructor | 4 |
| Queries con la tabla takes y classroom | 5 |
| Query con la tabla department | 6 |
| Query única para conseguir datos sobre el alumno "Sanchez" | 6 |
| Query única para conseguir datos sobre el departamento "Comp. Sci" | 7 |
| Query única para mostrar datos sobre la clase donde "Sanchez" recibe las clases | 7 |
| Query única para mostrar datos sobre el profesor que imparte las calses a "Sanchez" | 7 |
| Query única para mostrar datos sobre los cursos que cursa Levy | 9 |
| Bibliografía | 10 |

Introducción

EL objetivo de esta practica es reconocer y usar ciertos elementos de una base de datos y del lenguaje SQL.

Para ello se dispone de la BBDD de la Universidad de Yale



classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary)
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
student(ID, name, dept_name, tot_cred)
takes(ID, course_id, sec_id, semester, year, grade)
advisor(s_ID, i_ID)
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)

Figure 1: BBDD de la universidad

Identificación de los atributos de la tabla instructor

En la tabla **instructor** de la BBDD university indique que atributos tiene y de que tipo de datos son cada uno de ellos. Al mismo tiempo indique que atributo actúa como primary key y cuales como foreign key si es que existen.

```
mysql> describe instructor
-> ;
```

| Field | Type | Null | Key | Default | Extra |
|-----------|--------------|------|-----|---------|-------|
| ID | varchar(5) | NO | PRI | NULL | |
| name | varchar(20) | NO | | NULL | |
| dept_name | varchar(20) | YES | MUL | NULL | |
| salary | decimal(8,2) | YES | | NULL | |

4 rows in set (0.21 sec)

Figure 2: Atributos de la tabla instructor

Confirmación de la identificación de los atributos de la tabla instructor

Indague en el fichero **DDL.sql** para verificar que las respuestas dadas en la pregunta anterior respecto a la tabla **instructor** corresponden a los datos obtenidos. En caso de que existiera una foreign key, indique a que tabla hace referencia.

```
create table instructor
(
  ID          varchar(5),
  name        varchar(20) not null,
  dept_name   varchar(20),
  salary      numeric(8,2) check (salary > 29000),
  primary key (ID),
  foreign key (dept_name) references department (dept_name)
  on delete set null
);
```

Figure 3: Confirmación atributos de la tabla instructor

Queries con la tabla instructor

- Los instructores que pertenecen al departamento de "Physics".
- Cuantos instructores hay.
- Los instructores en función de su salario de mayor a menor.

```
mysql> SELECT dept_name,name FROM instructor WHERE dept_name LIKE 'Physics';
+-----+-----+
| dept_name | name   |
+-----+-----+
| Physics   | Einstein |
| Physics   | Gold    |
+-----+-----+
2 rows in set (0.25 sec)
```

(a) Instructores perteneciente al departamento "Physics"

```
mysql> select count(ID) as Numero_de_instructores from instructor where ID like '%';
+-----+
| Numero_de_instructores |
+-----+
| 12 |
+-----+
1 row in set (0.00 sec)
```

(b) Cuantos instructores hay

```
mysql> SELECT name, salary FROM instructor ORDER BY salary DESC;
+-----+-----+
| name   | salary |
+-----+-----+
| Einstein | 95000.00 |
| Brandt  | 92000.00 |
| Wu      | 90000.00 |
| Gold    | 87000.00 |
| Singh   | 80000.00 |
| Kim     | 80000.00 |
| Katz     | 75000.00 |
| Crick   | 72000.00 |
| Srinivasan | 65000.00 |
| Califieri | 62000.00 |
| El Said | 60000.00 |
| Mozart  | 40000.00 |
+-----+-----+
12 rows in set (0.00 sec)
```

(c) Ordenados en función de su salario

Figure 4: Queries con la tabla instructor

Queries con la tabla takes y classroom

¿En qué semestre, en que año y en qué grado se imparte el curso ID “CS-101” (revise la tabla takes)? De las clases disponibles en la universidad, indique qué clase tiene mayor capacidad y cuál tiene menos (revise la tabla classroom).

```
mysql> SELECT semester, year, grade, course_id FROM takes WHERE course_id LIKE 'CS-101';
+-----+-----+-----+-----+
| semester | year | grade | course_id |
+-----+-----+-----+-----+
| Fall     | 2017 | A     | CS-101    |
| Fall     | 2017 | C     | CS-101    |
| Fall     | 2017 | F     | CS-101    |
| Spring   | 2018 | B+    | CS-101    |
| Fall     | 2017 | A-    | CS-101    |
| Fall     | 2017 | A     | CS-101    |
| Fall     | 2017 | C-    | CS-101    |
+-----+-----+-----+-----+
7 rows in set (0.02 sec)
```

(a) Cuando se imparte el curso ID "CS-101"

```
mysql> SELECT min(capacity), max(capacity) FROM classroom;
+-----+-----+
| min(capacity) | max(capacity) |
+-----+-----+
| 10            | 500           |
+-----+-----+
1 row in set (0.36 sec)
```

(b) Que clase tiene mayor capacidad y cual menos

Figure 5: Queries con las tablas takes y classroom

Query con la tabla department

Este ha sido un buen año para todos los departamentos de la universidad porque han ganado varios proyectos internacionales. Esto supondrá que el presupuesto que tienen asignado crecerá un 20%. Haga una query que presente el presupuesto actual de cada departamento y cuál sería el nuevo presupuesto con el incremento del 20% ordenado de mayor a menor (revise la tabla department).

```
mysql> select budget, budget*20/100+budget as aumento from department order by budget desc;
+-----+-----+
| budget | aumento |
+-----+-----+
| 120000.00 | 144000.000000 |
| 100000.00 | 120000.000000 |
| 90000.00 | 108000.000000 |
| 85000.00 | 102000.000000 |
| 80000.00 | 96000.000000 |
| 70000.00 | 84000.000000 |
| 50000.00 | 60000.000000 |
+-----+-----+
7 rows in set (0.00 sec)
```

Figure 6: Presupuesto actual y nuevo

Query única para conseguir datos sobre el alumno "Sanchez"

Presente en una sola query los siguientes campos: name, dept_name, tot_cred (tabla student) y couse_id, semester, year, grade (table takes) para el alumno "Sanchez". Recuerde que tiene que extraer datos de dos tablas simultáneamente.

```
mysql> SELECT DISTINCT name, dept_name, tot_cred, t2.course_id, semester, year, t2.grade FROM student t1 INNER JOIN takes
t2 ON t1.ID = t2.ID WHERE name = 'Sanchez';
+-----+-----+-----+-----+-----+-----+-----+
| name | dept_name | tot_cred | course_id | semester | year | grade |
+-----+-----+-----+-----+-----+-----+-----+
| Sanchez | Music | 38 | MU-199 | Spring | 2018 | A- |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.35 sec)
```

Figure 7: Datos sobre el alumno "Sanchez"

Query única para conseguir datos sobre el departamento "Comp. Sci"

Repita la query anterior, es decir, presentar en una sola query los siguientes campos: name, dept_name, tot_cred (tabla student) y couse_id, semester, year, grade (table takes) pero en este caso para el departamento de "Comp. Sci" (Computer Science). Recuerde que tiene que extraer datos de dos tablas simultáneamente.

```
mysql> SELECT DISTINCT name, dept_name, tot_cred, t2.course_id, semester, year, t2.grade FROM student t1 INNER JOIN takes  
t2 ON t1.ID = t2.ID WHERE dept_name = 'Comp. Sci.';  
+-----+-----+-----+-----+-----+-----+-----+  
| name   | dept_name | tot_cred | course_id | semester | year | grade |  
+-----+-----+-----+-----+-----+-----+-----+  
| Zhang  | Comp. Sci. | 102      | CS-101    | Fall     | 2017 | A      |  
| Zhang  | Comp. Sci. | 102      | CS-347    | Fall     | 2017 | A-     |  
| Shankar | Comp. Sci. | 32       | CS-101    | Fall     | 2017 | C      |  
| Shankar | Comp. Sci. | 32       | CS-190    | Spring   | 2017 | A      |  
| Shankar | Comp. Sci. | 32       | CS-315    | Spring   | 2018 | A      |  
| Shankar | Comp. Sci. | 32       | CS-347    | Fall     | 2017 | A      |  
| Williams | Comp. Sci. | 54       | CS-101    | Fall     | 2017 | A-     |  
| Williams | Comp. Sci. | 54       | CS-190    | Spring   | 2017 | B+     |  
| Brown   | Comp. Sci. | 58       | CS-101    | Fall     | 2017 | A      |  
| Brown   | Comp. Sci. | 58       | CS-319    | Spring   | 2018 | A      |  
+-----+-----+-----+-----+-----+-----+-----+  
10 rows in set (0.00 sec)
```

Figure 8: Datos sobre el departamento "Comp. Sci"

Query única para mostrar datos sobre la clase donde "Sanchez" recibe las clases

Obtenga para el alumno "Sanchez" mediante una sola query que presente:

- name (tabla student)
- building, room_number, capacity (tabla classroom)

En definitiva el mostrar el edificio, el número de la clase y la capacidad de la misma donde Sánchez recibe las clases.

Ayuda

Aunque vas a extraer datos de dos tablas, por la relación de las BBDD debes actuar sobre cuatro tablas simultáneamente.

```
mysql> SELECT t1.name, t2.building, t2.room_number, t2.capacity FROM student t1 JOIN takes t3 ON t1.ID = t3.ID JOIN sec  
tion t4 ON t3.course_id = t4.course_id JOIN classroom t2 ON t4.building = t2.building WHERE t1.name='Sanchez';  
+-----+-----+-----+-----+  
| name   | building | room_number | capacity |  
+-----+-----+-----+-----+  
| Sanchez | Packard  | 101         | 500      |  
+-----+-----+-----+-----+  
1 row in set (0.36 sec)
```

Figure 9: Datos sobre la clase donde "Sanchez" recibe las clases

Query única para mostrar datos sobre el profesor que imparte las clases a "Sanchez"

Obtenga para el alumno "Sanchez" mediante una sola query que presente:

- name (tabla student)
- name, dept_name, salary (tabla instructor)

En definitiva el profesor que imparte las clases al alumno "Sanchez" donde además se muestra el departamento y su salario.

Ayuda

Aunque vas a extraer datos de dos tablas, por la relación de las BBDD debes actuar sobre cinco tablas simultáneamente.

```
mysql> SELECT  t1.name AS 'alumno',
->             t5.name AS 'Profesor', t5.dept_name, t5.salary
->
-> FROM      section t3
->           INNER JOIN teaches t4 ON
->                 t3.course_id = t4.course_id      AND
->                 t4.sec_id     = t3.sec_id         AND
->                 t4.semester   = t3.semester       AND
->                 t4.year       = t3.year
->
->           INNER JOIN instructor t5  ON t4.ID = t5.ID
->           INNER JOIN takes t2 ON
->                 t2.course_id = t3.course_id      AND
->                 t2.sec_id     = t3.sec_id         AND
->                 t2.semester   = t3.semester       AND
->                 t2.year       = t3.year
->           INNER JOIN student t1    ON t1.ID = t2.ID
->
-> WHERE t1.name='Sanchez'
-> ORDER BY t3.semester
-> ;
+-----+-----+-----+-----+
| alumno | Profesor | dept_name | salary |
+-----+-----+-----+-----+
| Sanchez | Mozart  | Music     | 40000.00 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Figure 10: Datos sobre el profesor que imparte las clases a "Sanchez"

Query única para mostrar datos sobre los cursos que cursa Levy

Obtenga para el alumno “Levy” mediante una sola query que presente:

- name, tot_cred (tabla student)
- semester, year, grade (tabla takes)
- title, dept_name , credits (tabla course)

En definitiva el mostrar el/los cursos que está haciendo el alumno Levy indicando también el semestre, año, etc.

Ayuda

Aunque vas a extraer datos de tres tablas, por la relación de las BBDD debes actuar sobre cuatro tablas simultáneamente.

```
mysql> SELECT  t1.name, t1.tot_cred,
->             t2.semester, t2.year, t2.grade,
->             t4.title, t4.dept_name, t4.credits
->
-> FROM      course t4
->          INNER JOIN section t3  ON t4.course_id = t3.course_id
->          INNER JOIN takes t2    ON t3.course_id = t2.course_id
->          INNER JOIN student t1  ON t2.ID = t1.ID
->
-> WHERE t1.name = 'Levy';
```

| name | tot_cred | semester | year | grade | title | dept_name | credits |
|------|----------|----------|------|-------|----------------------------|------------|---------|
| Levy | 46 | Fall | 2017 | F | Intro. to Computer Science | Comp. Sci. | 4 |
| Levy | 46 | Fall | 2017 | F | Intro. to Computer Science | Comp. Sci. | 4 |
| Levy | 46 | Spring | 2018 | B+ | Intro. to Computer Science | Comp. Sci. | 4 |
| Levy | 46 | Spring | 2018 | B+ | Intro. to Computer Science | Comp. Sci. | 4 |
| Levy | 46 | Spring | 2018 | B | Image Processing | Comp. Sci. | 3 |
| Levy | 46 | Spring | 2018 | B | Image Processing | Comp. Sci. | 3 |

6 rows in set (0.18 sec)

Figure 11: Datos sobre los cursos que cursa Levy

Índice de figuras

| | | |
|----|--|---|
| 1 | BBDD de la universidad | 2 |
| 2 | Atributos de la tabla instructor | 3 |
| 3 | Confirmación atributos de la tabla instructor | 3 |
| 4 | Queries con la tabla instructor | 4 |
| 5 | Queries con las tablas takes y classroom | 5 |
| 6 | Presupuesto actual y nuevo | 6 |
| 7 | Datos sobre el alumno "Sanchez" | 6 |
| 8 | Datos sobre el departamento "Comp. Sci" | 7 |
| 9 | Datos sobre la clase donde "Sanchez" recibe las clases | 7 |
| 10 | Datos sobre el profesor que imparte las clases a "Sanchez" | 8 |
| 11 | Datos sobre los cursos que cursa Levy | 9 |

Bibliografía

- [1] MySQL Tutorial
- [2] BBDD de la universidad