

## Laboratorio No. 4

### Descripción general

A continuación se definen los lineamientos sobre modelos conceptuales: Lenguaje de consulta (SQL). Algunos ejercicios y actividades basadas en los libros guía propuestos para el curso, otros son propios.

### Objetivos

Durante el desarrollo de las actividades se logra:

1. Trabajar en los lenguajes relacionales formales Álgebra Relacional y Cálculo Relacional de Tuplas, y el lenguaje estándar SQL

### Antes de empezar

Realice la lectura de:

- Database Systems - Concepts, Languages and Architectures [\[Link\]](#) [\[Link\]](#)
  - Chapter 3: Relational algebra and calculus
  - Chapter 4: SQL
- Database Management Systems [\[Link\]](#)
  - Chapter 4: Relational algebra and calculus
  - Chapter 5: Queries, constraints, triggers

Cree un documento PDF a partir de las siguientes actividades

### Actividad No. 1 [20%]

Sea (E) el esquema de una relación que posee tres atributos ( $A_1$ ,  $A_2$ ,  $A_3$ ), sea  $R_1$  y  $R_2$  relaciones sobre el esquema (E) y sean  $\langle t, p, q \rangle$ ,  $\langle a, b, c \rangle$ ,  $\langle a, b, p \rangle$ ,  $\langle q, b, c \rangle$  tuplas con los valores  $t, p, q, a, b, c$  en orden con respecto a los atributos del esquema (E). En el archivo PDF **empareje y justifique**.

- |  |   |
|--|---|
| 1. $\Pi_{A_3}(R_1)$                                  | I. $\{ \langle a, b, c \rangle \mid \langle a, b, c \rangle \in R_1 \vee \langle a, b, c \rangle \in R_2 \}$                    |
| 2. $\sigma_{A_1 = 15}(R_2)$                          | II. $\{ \langle a, b, c \rangle \mid \langle a, b, c \rangle \in R_2 \wedge a = 15 \}$  |
| 3. $R_1 \cup R_2$                                    | III. $\{ \langle a, b, c \rangle \mid \exists p, q (\langle a, b, p \rangle \in R_1 \wedge \langle q, b, c \rangle \in R_2) \}$ |
| 4. $R_2 \cap R_1$                                    | IV. $\{ \langle a, b, c \rangle \mid \langle a, b, c \rangle \in R_1 \wedge \langle a, b, c \rangle \notin R_2 \}$              |
| 5. $R_1 - R_2$                                       | V. $\{ \langle q \rangle \mid \exists t, p (\langle t, p, q \rangle \in R_1) \}$  |
| 6. $\Pi_{A_1, A_2}(R_1) \bowtie \Pi_{A_2, A_3}(R_2)$ | VI. $\{ \langle a, b, c \rangle \mid \langle a, b, c \rangle \in R_1 \wedge \langle a, b, c \rangle \in R_2 \}$                 |

### Actividad No. 2 [30%]

Sea (F) el esquema de una relación que posee dos atributos ( $A_1$ ,  $A_2$ ) y (D) el esquema de una relación que posee dos atributos ( $B_1$ ,  $B_2$ ), sea  $R_1$  y  $R_2$  relaciones sobre los anteriores esquemas respectivamente. Las

tuplas  $\langle a, b \rangle$ ,  $\langle c, d \rangle$  contiene los valores  $a, b, c, d$  en el orden de los atributos de los esquemas. En el archivo PDF **escriba la consulta usando álgebra relacional o notación para conjuntos**.

1.  $\sigma_{B2 = \text{"Andres"} \text{ OR } B2 = \text{"Eduardo"}} (R2)$
2.  $\{ \langle a \rangle \mid \exists b (\langle a, b \rangle \in R1 \wedge a = 17) \}$
3.  $\{ \langle a, b, c \rangle \mid \langle a, b \rangle \in R1 \wedge \langle a, c \rangle \in R2 \}$
4.  $R1 \bowtie_{A1 < B1} R2$

### Actividad No. 3 [50%]

Considere el siguiente esquema relacional para la oficina de registro de una universidad. La oficina mantiene datos sobre cada clase, incluido el instructor, la cantidad de estudiantes inscritos, la hora, entre otros datos.

- **student** (student\_id, name, program)
- **instructor** (instructor\_id, name, dept, title)
- **course** (course\_id, title, syllabus, credits)
- **course\_offering** (course\_id\*, sec\_id, year, semester, time, classroom)
- **enrols** (student\_id\*, course\_id\*, sec\_id, semester, year, grade)
- **teaches** (course\_id\*, sec\_id, semester, year, instructor\_id\*)
- **requires** (main\_course\*, prerequisite\*)

Claves primarias con subrayadas y (\*) para claves ajenas.

1. Cree el archivo con el nombre *createdb.sql* con las instrucciones para crear las tablas/relaciones y tenga en cuenta
  - a. El atributo *student\_id* inicia en 7488 con un incremento de 168 cada vez que se inserta un estudiante.
  - b. El atributo *course\_id* inicia en 837827 con un incremento de 23.
  - c. El tipo de dato del atributo *grade* de la tabla *enrolls* es NUMERIC con dos decimales, superior a 1.00 y menor a 5.00
2. Cree un archivo con el nombre *insert.sql* que contenga las instrucciones para insertar los datos a cada una de las tablas de la BD. Tenga en cuenta:
  - a. Ejecute el script *createdb.sql* para llevar el esquema a PostgreSQL
  - b. Cada tabla debe contener al menos 5 filas. Se recomienda el uso de generadores de datos ficticios (*dummy*).
  - c. Se pueden considerar datos simples o inventados como "Profe 1", "Profe 2", "Curso I", "Curso II" u otros.

3. Cree un archivo con el nombre *queries.sql* que contenga las instrucciones para obtener registros previamente consignados en la tabla de datos.

a. Encuentre el dept, title de los instructores registrados en la base de datos.

b. Indique el nombre y programa del estudiante con student\_id = 7656

c. Encuentre los nombres de todos los estudiantes que se han matriculado en el curso con course\_id = 837873

d. Cree una vista llamada *better\_students* que presente los estudiantes que obtuvieron las notas más altas por cada semestre entre los años 1900 y 2018

4. Cree un archivo con el nombre *tandplpg.sql* con los siguientes procedimientos (procedures) y disparadores (triggers)

a. Cree uno o varios disparadores (triggers) que implemente los siguiente requerimientos para la relación enrolls

- i. Al agregar una tupla en enrolls, en caso de que la nota sea negativa, cero (0.0) o mayor de 5.00 se debe generar una excepción indicando que el valor a guardar en grade es incorrecto o invalido.

- ii. Durante la actualización de un registro si el valor grade es modificado, usando RAISE NOTICE se debe presentar un mensaje indicando el cambio, si es igual al valor grade en la tabla se debe indicar que el valor no ha sido modificado. Si el grade a actualizar es negativo, cero o mayor de cinco use RAISE EXCEPTION.

b. Cree un procedimiento *create\_teaches* que automáticamente agregue un registro a *teaches*. Este recibe dos argumentos un identificador de instructor *instructor\_id* y un identificador de *course\_id*. Se asume que ambos existen en la base de datos.

- i. Este procedimiento debe verificar que el curso exista en la oferta de cursos.

- ii. Use course\_id, sec\_id, year y semester de la oferta de curso y instructor\_id el para insertar en *teaches*.

5. Cree un archivo con el nombre *drops.sql* con las instrucciones necesarias para hacer el borrado de todas las tablas, secuencias, disparadores, etc, que se crearon en los anteriores puntos de este laboratorio. (No utilice la instrucción *CASCADE*)
6. Comprima todos los archivos *sql* en un archivo LABNO1-APELLIDO1-APELLIDO2 (.zip), (.rar) o (.7z) y junto con un informe en PDF describa las soluciones realizadas en cada punto de este laboratorio. (Utilice este documento como plantilla para el informe)

### Observaciones

- La entrega se debe realizar en equipos de (3), en las fechas establecidas en nuestra bitácora. Se penaliza con (1.5) a los quienes entreguen sin tener en cuenta esta observación.
- Si no entiende el enunciado de alguna de las actividades no dude en escribir a [jefferson.amado.pena@correounivalle.edu.co](mailto:jefferson.amado.pena@correounivalle.edu.co)