#### Trabajo Práctico 1

#### Base de Datos II - 2025

#### Fundamentos, Integridad y Concurrencia

#### Ejercicio 1: Reglas de Integridad

#### Dado un modelo de base de datos de una universidad, identificar violaciones posibles a la integridad referencial si se elimina un estudiante con cursos inscriptos. ¿Qué mecanismos usarías para evitarlo?

#### Respuesta

#### La integridad referencial es un concepto del diseño de bases de datos, cuya esencia es que la base de datos debe exhibir precisión y coherencia en los datos y que los mismos deben estar sincronizados entre las diferentes tablas que la conformen.

#### Los mecanismos que se emplean para alcanzar la integridad referencial se integran en un conjunto denominado como Normalización, cuyo contenido se conoce como Formas Normales.

#### Las Formas Normales consisten en una serie de prácticas o acciones, idealmente ejecutadas durante la etapa del diseño de la base de datos. Se considera que la simple aplicación de las tres primeras formas, soluciona la amplia mayoría de los potenciales problemas de integridad.

#### Para el caso planteado, el diseño debería tener separados los estudiantes de los cursos en tablas diferentes, que se relacionarían entre sí a partir de una tercera tabla, que contenga los alumnos y sus cursos.

#### La claves de la tercer tabla, que la unen a las tabla de estudiantes y a la tabla de cursos, son del tipo Clave Foranea, creadas para asegurar la integridad referencial. Para prevenir posibles violaciones a esta integridad, se puede definir estas relaciones con restricción de borrado. Por ejemplo, en MySQL se las definiría como ON DELETE / ON UPDATE RESTRICT.

#### Ejercicio 2: Implementación de Restricciones

#### Crear una tabla Matriculas con restricciones de clave foránea. Luego, insertar datos que violen la integridad y mostrar el error generado.

#### Respuesta

#### 

#### La tabla matriculas tiene claves foraneas a partir de su relación con las tablas alumnos y materias. La tabla matrícula tiene una clave primaria compuesta para evitar que un mismo alumno se inscriba dos veces en la misma materia.

#### Ingreso el alumno 1 y la materia 3 y, a continuación, vuelvo a intentar ingresarlo, lo cual viola la integridad, pero el diseño lo impide y me regresa un error.

#### 

#### Se adjunta ejercicio2.sql

#### Ejercicio 3: Concurrencia

#### Simular una situación donde dos usuarios intentan actualizar el mismo saldo de una cuenta bancaria. Analizar como afectan las condiciones de aislamiento (READ COMMITTED vs SERIALIZABLE).

#### Respuesta

#### 

#### El término transacción hace referencia a un conjunto de operaciones que forman una tarea única, como ser la transferencia de dinero de una cuenta a otra, que requiere de dos actualizaciones, una para cada cuenta. Esta situación es la que se conoce como concurrencia y si no se controlan las actualizaciones de los datos compartidos, existe la posibilidad de que las transacciones operen sobre estados intermedios inconsistentes.

#### Realizando el ejercicio en Mysql Workbench, es claramente evidente que el definir condiciones de aislamiento afectan la ejecución de las transacciones, en general impidiéndolas.

#### Workbench no me permitió ejecutar dos transacciones simultaneas “*cuando las mismas tenían condiciones de aislamiento definidas*”, pero no impidió que se ejecutasen cuando faltaba dicha definición. En este último caso, claramente resultaba confuso conocer el valor correcto del saldo en un momento dado.

#### Se adjuntan los archivos .sql empleados:

#### ejercicio3\_banco\_bdd.sql

#### ejercicio3\_banco\_user\_RC.sql

#### ejercicio3\_banco\_user\_SER.sql