FP055-Introduccion a las bases de datos.

AA3. El modelo relacional. Sentencias DDL

Contenido

[SECCIÓN A. PARTE TEÓRICA 3](#_Toc152182091)

[1.1. Elementos del modelo relacional. Completa la siguiente tabla: 3](#_Toc152182092)

[1.2 Reglas de Integridad del Modelo Relacional. Explica las siguientes reglas: 3](#_Toc152182093)

[1.3 Transformación de Relaciones. 4](#_Toc152182094)

[**2. Sentencias DDL (Data Definition Languaje) de SQL.  Para cada objeto de la primera columna** 6](#_Toc152182095)

[**3. Restricciones.** 7](#_Toc152182096)

[**4. Claves compuestas y otras propiedades.** 9](#_Toc152182097)

[5. Tipos de Datos MySQL. 11](#_Toc152182098)

[2.1 Instalar el servidor de Base de Datos MySQL y el Gestor de Bases de Datos MySQL Workbench. 12](#_Toc152182099)

[2.2 Realizar la transformación del diagrama entidad relación de la segunda actividad al modelo relacional (estructura de tablas) en MySQL Workbench. 12](#_Toc152182100)

[2.3 Crear la Base de Datos en MySQL a partir del script en tu servidor local. 13](#_Toc152182101)

[2.4 Generar un script .SQL exportando la Base de Datos recién creada. 13](#_Toc152182102)

[2 Bibliografía. 14](#_Toc152182103)

[2.1 Instalación y primera conexión MySQL Sever y Workbench Video Tutorial. 14](#_Toc152182104)

[2.2 Error Could not acquire managemment acces for adminitration Video Tutorial. 14](#_Toc152182105)

[2.3 Lista de Primeros pasos MySQL Video Tutorial. 14](#_Toc152182106)

SECCIÓN A. PARTE TEÓRICA. **1. define cada uno de los siguientes elementos y propiedades del modelo relacional e indica con qué elemento del diagrama de relación de entidades (ERD) de la actividad están asociados.**   
Ejemplo: Las tablas se asocian a las entidades.

### 1.1. Elementos del modelo relacional. Completa la siguiente tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| Modelo relacional | Sinónimos | Definición corta | Se asocian a (ERD) |
| Tabla | relación | Conjunto de datos estructurados en filas (Entidad) y columnas (Atributos). | Entidades |
| Registro | Fila | Una instancia especifica o entrada en una tabla. Representa una entidad única. | - |
| Campo | Columna | Un espacio individual para datos en una tabla, que representa los atributos de una Entidad. | - |
| Clave primaria | Primary Key | Es la forma de identificar de manera única e inequívoca a una entidad, nunca puede ser 0. | - |
| Clave candidata | Candidate  Key | Es una o conjunto de columnas (Atributos) que puede ser usa como clave primaria. No debe contener redundancias. | - |
| Clave foránea (FOREIGN KEY) | - | Columna o conjuntos de columnas que establecen una relación entre dos tablas, referenciando la clave primaria de una de las tablas. | Entidades |
| Relación | - | Asociación lógica entre tablas que define como las tablas están conectadas o relacionadas, normalmente se usa un Verbo (acción). | - |

### 1.2 Reglas de Integridad del Modelo Relacional. Explica las siguientes reglas:

a. Reglas de Integridad de Modelo

* Reglas de Integridad de Entidades
  + Regla de Unicidad de la Clave Primaria (PRIMARY KEY).  
    **Definición:** Cada valor en la columna de clave primaria debe ser único en esa columna y no puede ser nulo.  
    **Ejemplo:** En una tabla "Clientes" con la columna "ID\_Cliente" como clave primaria, cada valor en esa columna debe ser único, y no puede haber valores nulos.
  + Regla de Entidad de la Clave Primaria. **Definición:** La clave primaria de una entidad no puede tener valores nulos, ya que representa la identificación única de cada instancia de la entidad.  
    **Ejemplo:** En una tabla "Empleados" con la columna "ID\_Empleado" como clave primaria, cada empleado debe tener un valor único en la columna "ID\_Empleado", y este valor no puede ser nulo.
* ¿Qué es Integridad Referencial? Ejemplo:  
  La integridad referencial garantiza la consistencia de las relaciones entre las tablas. Se asegura de que cualquier valor que aparece en una tabla que se refiere a otra tabla realmente exista en esa tabla.  
  **Ejemplo:** Supongamos que tenemos dos tablas, "Ordenes" y "Clientes". La tabla "Ordenes" tiene una clave foránea "ID\_Cliente" que se refiere a la clave primaria "ID\_Cliente" en la tabla "Clientes".  
  La integridad referencial garantiza que cualquier valor en la columna "ID\_Cliente" de la tabla "Ordenes" debe existir en la columna "ID\_Cliente" de la tabla "Clientes". No puede haber valores huérfanos en la tabla "Ordenes" que no tengan correspondencia en la tabla "Clientes".

b. Restricciones Semánticas. Explica las siguientes reglas:

* ¿Qué se conoce como DOMINIO en el ámbito de las Bases de Datos?  
  Se refiere al conjunto de valores que puede tomar un atributo en una tabla. Define las posibles entradas o valores que son válidos para un atributo especifico.
* ¿Qué es una restricción de Dominio definida por el usuario? Ejemplo.  
  Es una restricción que limita los valores que pueden ser insertados en una columna basándose en un conjunto de valores definido por el usuario.  
  **Ejemplo:** Supongamos que tenemos una columna "Edad" en una tabla de "Usuarios". Si queremos imponer una restricción de dominio definida por el usuario, podríamos especificar que solo se permiten valores de edad entre 18 y 99. Cualquier intento de insertar un valor fuera de este rango violaría la restricción.
* ¿Qué es una restricción de Usuario? Ejemplo.  
  Es una restricción adicional impuesta por el usuario para garantizar la integridad y calidad de los datos. Estas restricciones pueden ir más allá de las restricciones estándar proporcionadas por el sistema de gestión de bases de datos (DBMS).  
  **Ejemplo:** Supongamos que en una tabla "Empleados" queremos asegurarnos de que ningún empleado pueda tener más de 40 horas de trabajo a la semana. Podemos definir una restricción de usuario que verifique esta condición y evite que se insertan o actualicen registros que violen esta regla específica del negocio.

### 1.3 Transformación de Relaciones.

Texto

Descripción generada automáticamente

1. Describe la forma en la cual se transforman las relaciones 1:1 del Modelo Entidad-Relación al Modelo Relacional. Pon un ejemplo.  
   En una relación 1:1 del Modelo Entidad-Relación (ER), donde una entidad en un conjunto está relacionada con una entidad en otro conjunto, la transformación al Modelo Relacional implica fusionar ambas entidades en una sola tabla. La clave primaria de una entidad se convierte en una clave foránea en la otra entidad.  
   **Ejemplo:**Supongamos que tenemos dos entidades, "Persona" y "Pasaporte", con una relación 1:1. La transformación se vería así:  
   **Modelo Entidad-Relación (ER):**Entidad "Persona" (ID\_Persona, Nombre, Edad, ...).  
   Entidad "Pasaporte" (ID\_Pasaporte, Numero, FechaExpiracion, ...).  
   **Modelo Relacional:**Tabla "Persona" (ID\_Persona PRIMARY KEY, Nombre, Edad, ID\_Pasaporte FOREIGN KEY REFERENCES Pasaporte(ID\_Pasaporte)).  
   Tabla "Pasaporte" (ID\_Pasaporte PRIMARY KEY, Numero, FechaExpiracion, ...).
2. Describe la forma en la cual se transforman las relaciones 1:n del Modelo Entidad-Relación al Modelo Relacional. Pon un ejemplo.  
   En una relación 1:n, donde una entidad en un conjunto puede estar relacionada con varias entidades en otro conjunto, la transformación al Modelo Relacional implica agregar la clave primaria de la entidad del conjunto "1" como clave foránea en la entidad del conjunto "n".  
   **Ejemplo:**Supongamos que tenemos dos entidades, "Departamento" y "Empleado", con una relación 1:n. La transformación se vería así:  
   **Modelo Entidad-Relación (ER):**Entidad "Departamento" (ID\_Departamento, Nombre, ...).  
   Entidad "Empleado" (ID\_Empleado, Nombre, Salario, ID\_Departamento).  
   **Modelo Relacional:**Tabla "Departamento" (ID\_Departamento PRIMARY KEY, Nombre, ...).  
   Tabla "Empleado" (ID\_Empleado PRIMARY KEY, Nombre, Salario, ID\_Departamento FOREIGN KEY REFERENCES Departamento (ID\_Departamento)).

Imagen que contiene Diagrama

Descripción generada automáticamente

1. Describe la forma en la cual se transforman las relaciones n:m del Modelo Entidad-Relación al Modelo Relacional. Pon un ejemplo.  
   En una relación n:m, donde varias entidades en un conjunto pueden estar relacionadas con varias entidades en otro conjunto, la transformación al Modelo Relacional implica crear una tabla intermedia que contiene las claves foráneas de ambas entidades.  
   **Ejemplo:**Supongamos que tenemos dos entidades, "Estudiante" y "Curso", con una relación n:m. La transformación se vería así:  
   **Modelo Entidad-Relación (ER):**Entidad "Estudiante" (ID\_Estudiante, Nombre, ...).  
   Entidad "Curso" (ID\_Curso, Nombre, ...).  
   Relación n:m "Inscripcion" (ID\_Estudiante, ID\_Curso).  
   **Modelo Relacional:**Tabla "Estudiante" (ID\_Estudiante PRIMARY KEY, Nombre, ...).  
   Tabla "Curso" (ID\_Curso PRIMARY KEY, Nombre, ...).  
   Tabla "Inscripcion" (ID\_Estudiante FOREIGN KEY REFERENCES Estudiante(ID\_Estudiante), ID\_Curso FOREIGN KEY REFERENCES Curso(ID\_Curso), PRIMARY KEY (ID\_Estudiante, ID\_Curso)).

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

**2. Sentencias DDL (Data Definition Languaje) de SQL.  Para cada objeto de la primera columna.**

Marca con una x las sentencias que admite:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instrucción | sentencia | CREATE | ALTER | DROP | RENAME | TRUNCATE |
| Objetos | Crear | Modificar | Eliminar | Renombrar | Vaciar |
| DATABASE | SCHEMA | x | x | x | X |  |
| **TABLE \***  [TEMPORARY TABLE] | x | x | x | X | X |
| INDEX | x | x | x | X | x |
| VIEW | x | x | x | X |  |

**CREATE:** Puede ser utilizada para crear todos los objetos mencionados.

**ALTER:** Puede ser utilizada para modificar (alterar) todos los objetos mencionados.

**DROP:** Puede ser utilizada para eliminar (drop) todos los objetos mencionados.

**RENAME:** Puede ser utilizada para renombrar todos los objetos mencionados.

**TRUNCATE:** Puede ser utilizada para vaciar (truncar) las tablas y las tablas temporales.

**3. Restricciones.**  
La sentencia ALTER TABLE de SQL tiene dos propósitos:

* Permite agregar, modificar o eliminar columnas en una tabla:
  + **Agregar una Columna:**
    - ALTER TABLE nombre\_tabla
    - ADD nombre\_columna tipo\_dato;
  + **Modificar una Columna:**
    - ALTER TABLE nombre\_tabla
    - MODIFY nombre\_columna nuevo\_tipo\_dato;
  + **Eliminar una Columna:**
    - ALTER TABLE nombre\_tabla
    - DROP COLUMN nombre\_columna;
* Permite agregar y eliminar restricciones (constraints) en una tabla.
  + **Agregar una Restricción (por ejemplo, PRIMARY KEY):**
    - ALTER TABLE nombre\_tabla
    - ADD CONSTRAINT nombre\_restriccion PRIMARY KEY (nombre\_columna);
  + **Eliminar una Restricción:**
    - ALTER TABLE nombre\_tabla
    - DROP CONSTRAINT nombre\_restriccion;
  + **Agregar una Restricción UNIQUE:**
    - ALTER TABLE nombre\_tabla
    - ADD CONSTRAINT nombre\_restriccion UNIQUE (nombre\_columna);
  + **Agregar una Restricción FOREIGN KEY:**
    - ALTER TABLE nombre\_tabla
    - ADD CONSTRAINT nombre\_restriccion FOREIGN KEY (nombre\_columna) REFERENCES otra\_tabla(otra\_columna);

En referencia a las restricciones, completa la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| RESTRICCIÓN | OBJETIVO | EJEMPLO |
| PRIMARY KEY | Identificar registros únicos. | CREATE TABLE Ejemplo (ID INT PRIMARY KEY, nombre VARCHAR(255)); |
|  |  |
| UNIQUE | Garantizar valores únicos. | CREATE TABLE Ejemplo (Email CRACHAR(100) UNIQUE, OtroCampo INT); |
|  |  |
| FOREIGN KEY | Mantener integridad referencial. | CREATE TABLA Detalles (ID INT, EjemploID INT. FOREIGN KEY (EjemploID) REFERENCES Ejemplo(ID)); |
|  |  |
| CHECK | Establecer condiciones especiales. | CREATE TABLE Empleados (Edad INT CHECK (Edad >= 18)); |
|  |  |
| DEFAULT | Proporcionar valor predeterminado. | CREATE TABLE Ejemplo (FechaInicio DATE DEFAULT ‘2023-11-28’); |

## **4. Claves compuestas y otras propiedades.**

Indica realizando una captura de pantalla de MySQL Workbench.

* ¿Cómo se genera una Clave Primaria, Candidata o Foránea cuando está compuesta por más de un campo?  
  PrimayKey  
  Interfaz de usuario gráfica, Aplicación

  Descripción generada automáticamente

Candidata ("UNIQUE" se refiere a una restricción en una columna de una tabla en una base de datos. Cuando se aplica la restricción "UNIQUE" a una columna, significa que los valores en esa columna deben ser únicos en todas las filas de la tabla. Esto significa que no puede haber duplicados en esa columna).  
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente  
Foranea  
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* ¿Cómo se pueden agregar las propiedades AUTOINCREMENT y NOT NULL a un campo y qué significan estas dos propiedades?  
  AI(AutoIncrement) y NN(NotNull)  
  Interfaz de usuario gráfica, Aplicación

  Descripción generada automáticamente

## 5. Tipos de Datos MySQL.

MySQL admite una gran variedad de tipos de datos. Entre todos ellos se seleccionaron los que más se utilizan para que completes la tabla siguiente:

|  |  |  |
| --- | --- | --- |
| Tipo de dato | Grupo (Numérico, Fecha, Cadena) | Capacidad, formato... |
| BIT | Numérico | 1bit |
| INT | Numérico | -2147483648 a 2147483647 |
| FLOAT | Numérico | precisión Variable |
| DECIMAL (m,n) | Numérico | precisión fija (m total, n decimales) |
| DATE | Fecha | ‘YYYY-MM-DD’ |
| DATETIME | Fecha | ‘YYYY-MM-DD HH:MI:SS’ |
| TIME | Fecha | ‘HH:MI:SS’ |
| CHAR (n) | Cadena | Cadena de longitud Fija |
| VARCHAR (n) | Cadena | Cadena de longitud variable |
| ENUM | Cadena | Lista de valores predefinidos |
| SET | Cadena | Conjunto de valores predefinidos |

6. ¿Cuál es la diferencia entre los tipos de campo FLOAT y DECIMAL y cómo se interpretan los parámetros en este último tipo de campo? ¿Y entre CHAR y VARCHAR?

La elección entre FLOAT y DECIMAL depende de la precisión requerida para los valores numéricos, mientras que la elección entre CHAR y VARCHAR depende de si se necesita una longitud fija o variable para las cadenas de texto.

* FLOAT:
  + El tipo de dato FLOAT se utiliza para representar números de punto flotante (números con decimales).
  + Almacena números en formato de punto flotante, lo que significa que la precisión y la escala pueden variar.
  + Es adecuado para valores con decimales, pero la precisión puede ser limitada, lo que puede llevar a errores de redondeo en cálculos.
  + No requiere especificar la precisión o la escala al definir el campo.
* DECIMAL:
  + El tipo de dato DECIMAL se utiliza para representar números decimales con precisión fija.
  + Almacena números con una precisión específica, donde la precisión se refiere al número total de dígitos (antes y después del punto decimal) y la escala se refiere al número de dígitos después del punto decimal.
  + Es adecuado para valores que requieren una precisión precisa, como valores monetarios o científicos.
  + Al definir un campo DECIMAL, debes especificar la precisión y la escala. Por ejemplo, DECIMAL (10, 2) significa que puede almacenar números con hasta 10 dígitos en total, con 2 de ellos después del punto decimal.

En cuanto a la diferencia entre CHAR y VARCHAR:

* CHAR:
  + El tipo de dato CHAR se utiliza para almacenar cadenas de longitud fija.
  + Requiere que todos los valores en ese campo tengan la misma longitud, lo que significa que se pueden rellenar con espacios en blanco si es necesario.
  + Es útil cuando se sabe que todas las cadenas tendrán la misma longitud, ya que puede ahorrar espacio de almacenamiento.
* VARCHAR:
  + El tipo de dato VARCHAR se utiliza para almacenar cadenas de longitud variable.
  + Permite que los valores tengan longitudes diferentes y no desperdicia espacio de almacenamiento en espacios en blanco.
  + Es adecuado para almacenar cadenas de longitud variable, como nombres de personas o descripciones.

**SECCIÓN B. PARTE PRÁCTICA**

## Instalar el servidor de Base de Datos MySQL y el Gestor de Bases de Datos MySQL Workbench.

## Realizar la transformación del diagrama entidad relación de la segunda actividad al modelo relacional (estructura de tablas) en MySQL Workbench.

Tienes que volver a crear de nuevo las tablas, campos, relaciones, pero ahora en Workbench y siguiendo el modelo relacional y asignando las Primary key y Foreign keys correctamente.  
  
Creo que esta vez entendí mejor la actividad!  
Imagen que contiene Escala de tiempo

Descripción generada automáticamente

Traslada el ERD de la AA2 a MySQL Workbench y crea el Modelo Relacional con las pautas siguientes, que permitirán apreciar mejor las propiedades de los campos de las tablas y las relaciones:  
  
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Crear la Base de Datos en MySQL a partir del script en tu servidor local.

Utilizando la función Database - Forward Engineering de MySQL Workbench, genera la Base de Datos a partir del Modelo Relacional.  
Cierra la ventana de modelos. Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Generar un script .SQL exportando la Base de Datos recién creada.

Conecta con tu servidor local desde MySQL Workbench (clicando sobre Local instance…) y exporta la base de datos .SQL generada en el paso anterior (opción Server-Data Export).

# Bibliografía.

## Instalación y primera conexión MySQL Sever y Workbench Video [Tutorial.](https://www.youtube.com/watch?v=hc9c7dwFDoI)

## Error Could not acquire managemment acces for adminitration Video [Tutorial.](https://www.youtube.com/watch?v=nFVPFCigomE)

## Lista de Primeros pasos MySQL Video [Tutorial.](https://youtube.com/playlist?list=PLgqdACsQ8US3mPVgaBu8HiCk3pkeUjF33&si=1TZxJj2UjVwsXlcL)

## 