**Trabajo Práctico Final Gestión de Base de Datos**

**Este trabajo fue hecho en MySQL (XAMP)**

1 - Crear una función llamada "**calcular\_total\_ventas**" que tome como parámetro el mes y el año, y devuelva el total de ventas realizadas en ese mes. Verificar mediante consulta.

2 - Crear una función llamada "**obtener\_nombre\_empleado**" que tome como parámetro el ID de un empleado y devuelva su nombre completo. Verificar mediante consulta.

3 - Crear un procedimiento almacenado llamado "**obtener\_promedio**" que tome como parámetro de entrada el nombre de un curso y calcule el promedio de las calificaciones de todos los alumnos inscriptos en ese curso. Verificar mediante ejecución del procedimiento.

4 - Crear un procedimiento almacenado "**actualizar\_stock**" que tome como parámetros de entrada el código del producto y la cantidad a agregar al stock actual. El procedimiento debe actualizar el stock sumando la cantidad especificada al stock actual del producto correspondiente. Verificar mediante ejecución del procedimiento.

5 - Crear una vista que muestre el título, el autor, el precio y la editorial de todos los libros de cocina de la base pubs.

6 – Dadas las siguientes tablas:

CREATE TABLE fabricantes (

id\_fabricante INT PRIMARY KEY,

nombre\_fabricante VARCHAR(255) NOT NULL

);

INSERT INTO fabricantes (id\_fabricante, nombre\_fabricante)

VALUES(1, 'Fabricante A'),(2, 'Fabricante B'),(3, 'Fabricante C');

CREATE TABLE productos (

id\_producto INT PRIMARY KEY,

id\_fabricante INT,

nombre\_producto VARCHAR(255) NOT NULL,

fecha\_lanzamiento DATE,

FOREIGN KEY (id\_fabricante) REFERENCES fabricantes(id\_fabricante)

);

INSERT INTO productos (id\_producto, id\_fabricante, nombre\_producto, fecha\_lanzamiento)

VALUES(1, 1, 'Producto X', '2020-01-01'),(2, 2, 'Producto Y', '2019-12-01'), (3, 3, 'Producto Z', '2021-05-15');

a) Crear un índice compuesto en las columnas **id\_fabricante** y **nombre\_producto** de la tabla productos.

b) Crear un índice único en la columna **id\_producto** de la tabla productos.

c) Modificar el índice **idx\_productos\_id\_fabricante\_nombre** para que sea único en la columna **id\_fabricante**.

d) Crear un nuevo índice único en la columna **id\_fabricante**

e) Eliminar el índice **idx\_productos\_id\_fabricante** de la tabla productos

7 - Se desea modificar un sistema de gestión de empleados para incluir un mecanismo automático que transfiera a los empleados que cumplen con ciertos criterios de jubilación a una tabla especializada llamada jubilados.

Los criterios de jubilación son: los empleados deben tener 30 años o más de antigüedad y 65 años o más de edad. Además, se requiere que cualquier inserción en la tabla empleados que cumpla con estos criterios resulte en una inserción automática en la tabla jubilados.

CREATE TABLE empleados (

nombre VARCHAR(50) NOT NULL,

edad INT NOT NULL,

antiguedad INT NOT NULL

);

CREATE TABLE jubilados (

nombre VARCHAR(50) NOT NULL,

edad INT NOT NULL,

antiguedad INT NOT NULL

);

8 - Crear un procedimiento almacenado llamado **ActualizarEmpleados** que tome dos parámetros de entrada:

codigo\_empleado (VARCHAR, 10): El identificador del empleado a actualizar.

salario\_actualizado (DECIMAL): El nuevo salario del empleado.

En el procedimiento, utilizar una transacción para realizar la actualización del salario del empleado:

Obtener la información actual del empleado especificado.

Verificar si el nuevo salario es válido (no puede ser menor que el salario actual).

Si el salario es válido, realizar la actualización del salario del empleado.

Si el salario actualizado fuere menor que el salario actual, mostrar un mensaje al usuario indicando que la operación se cancela y realizar un **rollback**.

Llamar al procedimiento **ActualizarEmpleados** con diferentes valores de **codigo\_empleado** y **salario\_actualizado**, incluyendo casos donde el salario actualizado sea menor que el salario actual.

Verificar que el procedimiento funcione correctamente y que se muestren mensajes de error y se realice un rollback cuando corresponda.

**9 -** **Gestión de Usuarios**

a) Crear un usuario sin privilegios específicos

b) Crear un usuario con privilegios de lectura sobre la base pubs

c) Crear un usuario con privilegios de escritura sobre la base pubs

d) Crear un usuario con todos los privilegios sobre la base pubs

e) Crear un usuario con privilegios de lectura sobre la tabla titles

f) Eliminar al usuario que tiene todos los privilegios sobre la base pubs

g) Eliminar a dos usuarios a la vez

h) Eliminar un usuario y sus privilegios asociados

i) Revisar los privilegios de un usuario

**10 – Gestor Mongo DB**

a) Activar la base de datos **"local"** y luego imprimir las colecciones existentes.

b) Activar la base de datos **"test"** y luego imprimir las colecciones existentes.

c) Activar la base de datos **"baseEjemplo2"**.

d) Mostrar las colecciones existentes en la base de datos **"baseEjemplo2"**.

e) Crear otra colección llamada usuarios donde almacenar dos documentos con los

campos nombre y clave.

f) Mostrar nuevamente las colecciones existentes en la base de datos **"baseEjemplo2".**

**En la base pubs**:

g) Insertar 2 documentos en la colección clientes con '\_id' no repetidos

h) Intentar insertar otro documento con clave repetida.

i) Mostrar todos los documentos de la colección libros.

j) Crear una base de datos llamada **"blog".**

k) Agregar una colección llamada **"posts"** e insertar 1 documento con una estructura a

su elección.

l) Mostrar todas las bases de datos actuales.

m) Eliminar la colección **"posts"**

n) Eliminar la base de datos **"blog"** y mostrar las bases de datos existentes.

11 - **A partir de la siguiente especificación deberá recolectar datos para poder diseñar una Base de Datos**.

a) Determinar las entidades relevantes al Sistema.

Empleados

Importadores

Componentes

Televisores

Mapas de Armado

Hojas de Trabajo

Ordenes de Compra

Componentes por Televisor

Fabricaciones

Modelos de TV

Ubicaciones de Componentes

b) Determinar los atributos de cada entidad.

Empleados

id\_empleado (PK)

nombre

apellido

puesto

Importadores

id\_importador (PK)

nombre

direccion

telefono

Componentes

id\_componente (PK)

nombre

descripcion

tipo (importado o fabricado)

id\_importador (FK, nullable if fabricated internally)

Televisores

id\_televisor (PK)

modelo

descripcion

Mapas de Armado

id\_mapa (PK)

id\_televisor (FK)

descripcion

Hojas de Trabajo

id\_hoja\_trabajo (PK)

id\_empleado (FK)

cantidad

fecha

id\_componente (FK)

Ordenes de Compra

id\_orden (PK)

fecha

id\_importador (FK)

Componentes por Televisor

id\_televisor (FK)

id\_componente (FK)

cantidad

PRIMARY KEY (id\_televisor, id\_componente)

Fabricaciones

id\_fabricacion (PK)

id\_empleado (FK)

id\_componente (FK)

fecha

cantidad

Modelos de TV

id\_modelo (PK)

nombre

descripcion

Ubicaciones de Componentes

id\_mapa (FK)

id\_componente (FK)

ubicacion

orden

PRIMARY KEY (id\_mapa, id\_componente)

c) Confeccionar el Diagrama de Entidad Relación (DER), junto al Diccionario de Datos

d) Realizar el Diagrama de Tablas e implementar en código SQL (puede utilizar cualquier Gestor) la Base de Datos.

CREATE TABLE empleados (

id\_empleado INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(50),

apellido VARCHAR(50),

puesto VARCHAR(50)

);

CREATE TABLE importadores (

id\_importador INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100),

direccion VARCHAR(255),

telefono VARCHAR(50)

);

CREATE TABLE componentes (

id\_componente INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100),

descripcion TEXT,

tipo ENUM('importado', 'fabricado'),

id\_importador INT,

FOREIGN KEY (id\_importador) REFERENCES importadores(id\_importador)

);

CREATE TABLE televisores (

id\_televisor INT AUTO\_INCREMENT PRIMARY KEY,

modelo VARCHAR(50),

descripcion TEXT

);

CREATE TABLE mapas\_de\_armado (

id\_mapa INT AUTO\_INCREMENT PRIMARY KEY,

id\_televisor INT,

descripcion TEXT,

FOREIGN KEY (id\_televisor) REFERENCES televisores(id\_televisor)

);

CREATE TABLE hojas\_de\_trabajo (

id\_hoja\_trabajo INT AUTO\_INCREMENT PRIMARY KEY,

id\_empleado INT,

cantidad INT,

fecha DATE,

id\_componente INT,

FOREIGN KEY (id\_empleado) REFERENCES empleados(id\_empleado),

FOREIGN KEY (id\_componente) REFERENCES componentes(id\_componente)

);

CREATE TABLE ordenes\_de\_compra (

id\_orden INT AUTO\_INCREMENT PRIMARY KEY,

fecha DATE,

id\_importador INT,

FOREIGN KEY (id\_importador) REFERENCES importadores(id\_importador)

);

CREATE TABLE componentes\_por\_televisor (

id\_televisor INT,

id\_componente INT,

cantidad INT,

PRIMARY KEY (id\_televisor, id\_componente),

FOREIGN KEY (id\_televisor) REFERENCES televisores(id\_televisor),

FOREIGN KEY (id\_componente) REFERENCES componentes(id\_componente)

);

CREATE TABLE fabricaciones (

id\_fabricacion INT AUTO\_INCREMENT PRIMARY KEY,

id\_empleado INT,

id\_componente INT,

fecha DATE,

cantidad INT,

FOREIGN KEY (id\_empleado) REFERENCES empleados(id\_empleado),

FOREIGN KEY (id\_componente) REFERENCES componentes(id\_componente)

);

CREATE TABLE modelos\_de\_tv (

id\_modelo INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100),

descripcion TEXT

);

CREATE TABLE ubicaciones\_de\_componentes (

id\_mapa INT,

id\_componente INT,

ubicacion VARCHAR(255),

orden INT,

PRIMARY KEY (id\_mapa, id\_componente),

FOREIGN KEY (id\_mapa) REFERENCES mapas\_de\_armado(id\_mapa),

FOREIGN KEY (id\_componente) REFERENCES componentes(id\_componente)

);

e) Crear al menos 2 consultas relacionadas para poder probar la Base de Datos.

SELECT c.nombre AS componente, f.cantidad, f.fecha

FROM fabricaciones f

JOIN componentes c ON f.id\_componente = c.id\_componente

WHERE f.id\_empleado = 1

AND f.fecha BETWEEN '2023-01-01' AND '2023-12-31';

SELECT t.modelo, c.nombre AS componente, cp.cantidad

FROM televisores t

JOIN componentes\_por\_televisor cp ON t.id\_televisor = cp.id\_televisor

JOIN componentes c ON cp.id\_componente = c.id\_componente

WHERE t.id\_televisor = 1;

Esta empresa se encuentra ubicada en Hong Kong y se dedica a la fabricación de Smart TV.

Las componentes de los TV pueden ser comprados a un importador, en tal caso la compra viene acompañada de una orden, otros componentes son fabricados en la empresa, para lo cual esos componentes tienen asignado un empleado que se dedica exclusivamente a un tipo de componente, aunque un componente puede ser fabricado por más de un empleado, el empleado completa una hoja de trabajo con la cantidad fabricada y la fecha.

Los diferentes modelos de Smart TV tienen de 275 a 430 componentes, aunque un componente puede estar incorporado en más de un TV, existe un mapa de armado para cada modelo de TV donde se indica la ubicación y el orden de los componentes.