**Desarrollo de sistemas con tecnología Java**

**Módulo 2: Manejo de bases de datos con Java**

**Práctica**

| **Nombre**: |  |  |  | **Fecha**: |  |
| --- | --- | --- | --- | --- | --- |
|  | Apellido Paterno | Apellido Materno | Nombre(s) |  |  |
|  | **Calificación:** | | | |  |

| **Objetivo**: El o la participante usará los comandos de SQL necesarios para crear estructuras de datos (tablas) y actualizar/consultar su contenido (registros), reforzando los conocimientos adquiridos en clase. |
| --- |

**Para tener derecho a la calificación, el código fuente entregado deberá ser inédito y resultado por completo del trabajo de él o la participante de manera individual e independiente.**

**Instrucciones:** Lea cuidadosamente y desarrolle correctamente cada uno de los puntos tal y como se describen a continuación:

**Parte I**

Elabore una estructura de datos (CREATE TABLE) que cubra la necesidad de un sitio web con información de películas. Considere que deberá guardar información de los siguientes 3 rubros:

1. **Películas**, con los datos generales de las películas y al menos los siguientes campos:
   1. Nombre de la película.
   2. Género.
   3. Clasificación.
   4. Idioma.
   5. Número de premios ganados.
2. **Actores**, con los datos generales de actores y actrices que participan en las películas. Contendrá al menos los siguientes campos:
   1. Nombre del actor.
   2. Nacionalidad.
   3. Edad.
   4. Número de premios Oscar ganados.
3. **Reparto**, con los actores que participaron en cada película, registrando al menos la siguiente información de cada actor/actriz contratados en las películas:
   1. Nombre del personaje dentro de la película.
   2. Sueldo pagado.

Con el modelo desarrollado en el punto anterior elabore el código DDL (en un archivo llamado **parteI.sql**) para crear las tablas en una base de datos de MariaDB. Es necesario que el código se ejecute sin errores y se implementen, al menos, las siguientes restricciones:

* El valor por default de número de premios ganados y de idioma -de las películas- será cero y “español” respectivamente.
* El valor por default del número de premios Oscar (en la tabla de actores) será de cero.
* La edad será un valor positivo siempre.
* El sueldo pagado (en la tabla de Reparto) será siempre un valor positivo.

Cuide que las tablas tengan llaves primarias y establezca las relaciones que considere adecuadas para cumplir con las primeras 3 formas normales y la integridad referencial.

**Parte II**

Ejecute el código que le fue proporcionado en **parteII.sql** (sin modificar el código y sin modificar la definición de las estructuras ahí especificadas) y genere el código SQL necesario para cumplir con los siguientes puntos (incruste los comandos utilizados en las secciones indicadas en amarillo:

1. Utilizando UPDATE, aumente 10% el precio al público en cursos de 20 horas o menos.

**UPDATE cat\_curso**

**SET precioPublico = precioPublico \* 1.1**

**WHERE horas <= 20;**

1. Añadir 1 registro en la tabla de alumnos con la siguiente información:

* Laura López, nacida el 1/enero/2000 y RFC “LOGL000101ABC”.

**INSERT INTO alumnos (boleta, nombre, primerApellido, segundoApellido, RFC, fechaNacimiento, idActividad, cursosTomados) VALUES (777, 'Laura', 'López', '', 'LOGL000101ABC', '2000-01-01', 1, 0);**

1. Añadir 2 cursos diferentes en la tabla de cursos (utilizar 2 comandos INSERT), sobre cualquier tema relativo a las bases de datos relacionales. Puede tomar como base el catálogo de cursos de la DGTIC-UNAM para llenar los registros con la información más real posible. <https://docencia.tic.unam.mx/oferta_aca_presen> .

Incruste aquí el código realizado para este punto

1. Escriba el comando SELECT para mostrar los cursos de más de 25 horas, ordenado por nombre del curso.

I**SELECT idCurso, curso, objetivo, horas, precioPublico, pagoInstructor**

**FROM cat\_curso**

**WHERE horas > 25**

**ORDER BY curso;**

**Puntos opcionales:**

1. Escriba un comando SELECT para listar los **nombres de los cursos** tomados por Arturo Morales Herrero.

**SELECT cc.curso**

**FROM alumnos a**

**JOIN calendario c ON a.boleta = c.boleta**

**JOIN cat\_curso cc ON c.idCurso = cc.idCurso**

**WHERE a.nombre = 'Arturo' AND a.primerApellido = 'Morales' AND a.segundoApellido = 'Herrero';**

1. Escriba un comando SELECT para conocer cuántos alumnos obtuvieron 10, cuántos 9, cuántos 8, etc., es decir, agrupe los cursos tomados por calificación obtenida:

**SELECT promFinal, COUNT(\*) AS numAlumnos**

**FROM calendario**

**GROUP BY promFinal**

**ORDER BY promFinal DESC;**

1. Eliminar permanentemente el alumno Arturo Morales.

**DELETE FROM alumnos**

**WHERE nombre = 'Arturo' AND primerApellido = 'Morales' AND segundoApellido = 'Herrero';**

**Entregables**

1. Archivo **parteI.sql** con el código necesario para crear la estructura solicitada en la parte I de esta actividad.
2. Este archivo (**DJava15-M2-Práctica01.docx**) con el código de la parte II en su apartado correspondiente.

Ambos archivos deberán estar dentro de un archivo .ZIP (llamado **practica1.ZIP**), mismo que deberá subir a la plataforma que ha utilizado para el Diplomado, en el apartado de la práctica.

**Criterios de evaluación**

| Elemento | Puntos |
| --- | --- |
| Parte I | |
| Incluye los comandos DDL necesarios para crear la estructura requerida en cada una de las tablas indicadas | |
| * Películas | 17 |
| * Actores | 17 |
| * Reparto | 16 |
| Parte II | |
| Incluye los comandos DML para | |
| * Subir el precio de los cursos tal y como se indica | 10 |
| * Adicionar la alumna especificada | 10 |
| * Añade dos cursos de bases de datos | 15 |
| * Mostrar los cursos de más de 25 horas | 15 |
| PUNTAJE MÁXIMO | **100** |

**Valor 100 puntos**