

PROGRAMACION ORIENTADA A OBJETOS II

PRACTICAS DE LABORATORIO

ESTAS PRACTICAS DE LABORATORIO DEBEN SER REALIZADAS DE MANERA INDIVIDUAL. LEA DETENIDAMENTE LAS INSTRUCCIONES, PUES EL NO SEGUIRLAS TAL COMO SE INDICA PUEDE RESULTAR EN UNA PRACTICA INCORRECTA. CUANDO LA PRÁCTICA SEA TERMINADA, ASEGURESE DE QUE CUMPLE CON TODOS LOS REQUERIMIENTOS SOLICITADOS Y DE QUE FUNCIONE CORRECTAMENTE ANTES DE ENTREGARLA.

PRACTICA 5. CREACION Y MANIPULACION DE BASES DE DATOS EN MYSQL USANDO EL LENGUAJE SQL.

En esta práctica, generará un par de scripts (archivos con comandos SQL) que contendrán una serie de sentencias SQL (terminadas con ;) para crear un usuario y otorgarle permisos sobre la base de datos **controlconcursos_ad2021**, crear 3 tablas en MySQL en tal base de datos, y colocar información en tales tablas. La base de datos representará parte de la información necesaria para llevar el control de los concursos de programación tipo ICPC. Para efectos de poder realizar los comandos, deberá primero crear la base de datos con el comando siguiente:

```
CREATE DATABASE IF NOT EXISTS controlconcursos_ad2021;
```

Este comando lo dará directamente en la consola y no será incluido en los dos archivos que deberá crear en esta práctica.

Para tener los archivos necesarios, deberá crear una copia local del repositorio remoto creado en Github al aceptar la tarea. Para ello, es necesario hacer los siguientes pasos:

1) Entrar a la página cuyo URL les fue proporcionado al aceptar la tarea, en tal página dé click en el botón Code y copie el URL que aparece en el cuadro de texto de nombre ****Clone with HTTPS**** (comienza con https://)

2) En una consola de Git Bash en Windows (o en una terminal en Linux o Mac), cree una carpeta donde quiera que se contengan sus prácticas del semestre (si es que aún no la has creado) y colócate en tal carpeta. La carpeta la puedes crear desde el Git Bash o terminal Linux/Mac usando el comando ``mkdir`` (o con el explorador de archivos de su sistema operativo) y en la consola de Git Bash o terminal de Linux/Mac te puedes cambiar a la carpeta mencionada usando el comando `cd`

3) Clone el repositorio privado dando el comando `git clone URL practica05`

(donde URL es el URL que copió en el paso 1)\

Este comando creará dentro de la carpeta creada en el paso 2) una subcarpeta de nombre `practica05` donde estará una copia local del repositorio remoto.

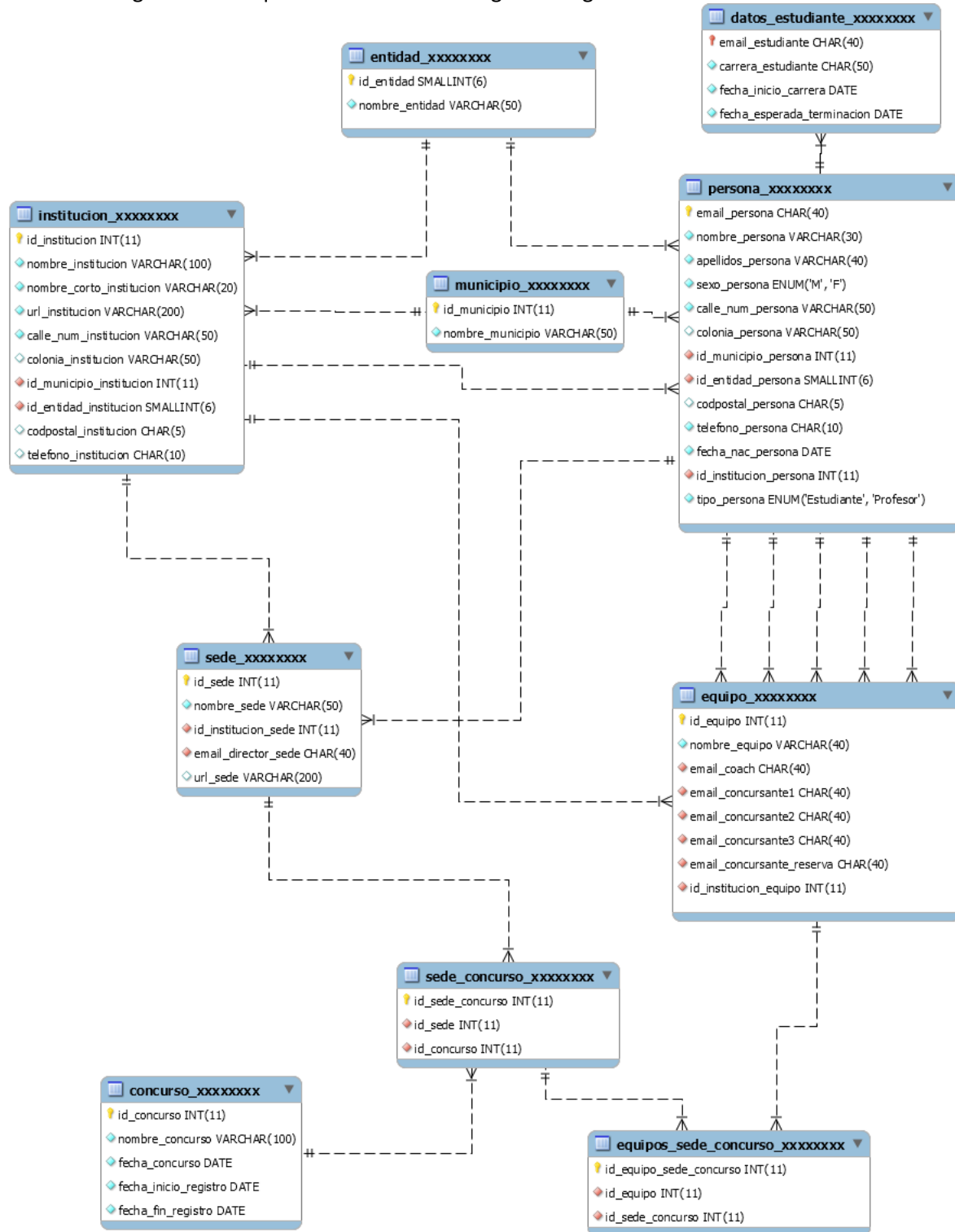
Los comandos posteriores de git tendrán que darse desde tal carpeta, por tanto, cámbiate a la carpeta usando **`cd practica05`**

Requerimientos de la serie de sentencias a generar:

- Las sentencias SQL deberán ser colocadas en dos archivos de texto con extensión `.sql` y nombrados exactamente de la siguiente manera (todos los caracteres del nombre en minúsculas): **`tablas.sql`** y **`datos.sql`**
- El primer archivo (**`tablas.sql`**) deberá cumplir con lo siguiente:
 - Se colocarán al inicio los siguientes cinco comandos (donde XXXXXXXX se sustituye por su matrícula):

```
USE controlconcursos_ad2021;  
DROP USER IF EXISTS U_XXXXXXX@'localhost';  
DROP TABLE IF EXISTS persona_XXXXXXX;  
DROP TABLE IF EXISTS datos_estudiante_XXXXXXX;  
DROP TABLE IF EXISTS equipo_XXXXXXX;
```

- Posteriormente deberá tener 1 comando para crear el usuario **U_XXXXXXX** (el cual se conectará desde localhost y con clave **UAZsw2021**) y otro comando para otorgarle a este usuario todos los permisos de acceso en la base de datos **controlconcursos_ad2021**.
- Vendrá después un conjunto de sentencias para crear 3 tablas. Las tablas por crear serán algunas de las que se muestran en la siguiente figura.



- Se mencionan los siguientes aspectos sobre las tablas mostradas en la figura:
 - Que un campo tenga una llave amarilla a la izquierda indica que es la llave primaria
 - Que tenga un rombo con relleno rojo a la izquierda indica que es una llave foránea, es decir, que su valor es la llave primaria en alguna otra tabla (**PERO NO NECESITAN INDICAR EN SU SCRIPT QUE ES DE HECHO UNA LLAVE FORÁNEA**)
 - Que un campo tenga un rombo con relleno azul a la izquierda indica que debe dársele valor al momento de insertar un registro, es decir, su valor NO puede ser NULL. Se indica para cada campo el tipo de datos correspondiente. Las tablas a crear son las siguientes: **persona_XXXXXXX**, **datos_estudiante_XXXXXXX** y **equipo_XXXXXX** (donde XXXXXXXX se sustituye por su matrícula).

NOTAS:

1. Recuerde sustituir las XXXXXXXX por su matrícula en el nombre de las tablas.
2. El campo **id_equipo** de la tabla **equipo_XXXXXXX** deberá tener la característica de ser autoincrementables (para lo cual se agrega la palabra AUTO_INCREMENT al definir la columna en la sentencia SQL de creación de la tabla).
3. Recuerde que los campos con rombo relleno indican que no pueden ser NULL, por lo cual se les debe poner NOT NULL como atributo en la definición de la columna correspondiente.
4. La tabla **persona_XXXXXXX** contendrá los siguientes campos y tipos:
 - **email_persona** de tipo CHAR(40)
 - **nombre_persona** de tipo VARCHAR(30)
 - **apellidos_persona** de tipo VARCHAR(40)
 - **sexo_persona** de tipo ENUM('M','F')
 - **calle_num_persona** de tipo VARCHAR(50)
 - **colonia_persona** de tipo VARCHAR(50)
 - **id_municipio_persona** de tipo INT
 - **id_entidad_persona** de tipo SMALLINT
 - **codpostal_persona** de tipo CHAR(5)
 - **telefono_persona** de tipo CHAR(10)
 - **fecha_nac_persona** de tipo DATE
 - **id_institucion_persona** de tipo INT
 - **tipo_persona** de tipo ENUM ("Estudiante","Profesor")
 - La llave primaria será el campo **email_persona**. Todos los campos excepto **colonia_persona** y **codpostal_persona** deben tener un valor al agregar un registro, es decir, deben tener el atributo **NOT NULL**.
5. La tabla **datos_estudiante_XXXXXXX** contendrá los siguientes campos y tipos:
 - **email_estudiante** de tipo CHAR(40)
 - **carrera_estudiante** de tipo CHAR(50)
 - **fecha_inicio_carrera** de tipo DATE
 - **fecha_esperada_terminacion** de tipo DATE
 - La llave primaria será el campo **email_estudiante**. Todos los campos deben tener un valor al agregar un registro, es decir, deben tener el atributo **NOT NULL**.

6. La tabla **equipo_XXXXXX** contendrá los siguientes campos y tipos:

- **id_equipo** de tipo INT y con característica **AUTO_INCREMENT**
- **nombre_equipo** de tipo VARCHAR(40)
- **email_coach** de tipo CHAR(40)
- **email_concursante1** de tipo CHAR(40)
- **email_concursante2** de tipo CHAR(40)
- **email_concursante3** de tipo CHAR(40)
- **email_concursante_reserva** de tipo CHAR(40)
- **id_institucion_equipo** de tipo INT
- La llave primaria será el campo **id_equipo**. Todos los campos excepto **email_concursante_reserva** deben tener un valor al agregar un registro, es decir, deben tener el atributo **NOT NULL**.

- Después de los comandos para crear las tablas ponga los siguientes comandos al final del archivo tablas.sql (recuerde sustituir los XXXXXXXX por su matrícula):

```
SHOW TABLES LIKE '%XXXXXXX';  
SELECT user,host FROM mysql.user;
```

- El segundo archivo (**datos.sql**) iniciará con los siguientes cuatro comandos (recuerde sustituir los XXXXXXXX por su matrícula):

```
USE controlconcursos_ad2021;  
DELETE FROM persona_XXXXXXX;  
DELETE FROM datos_estudiante_XXXXXXX;  
DELETE FROM equipo_XXXXXXX;
```

Seguido de comandos para realizar los siguientes pasos:

- Agregar 4 registros a la tabla persona_XXXXXX (con al menos la información que es obligatoria) donde 3 registros deben ser marcados como estudiante, poniéndole el valor "Estudiante" al campo tipo_persona y el quinto registro debe ser marcado como profesor, poniéndole el valor "Profesor" al campo tipo_persona. Los campos que guardan el nombre, apellidos, email y sexo deben ser reales, los valores de los otros campos pueden ser inventados. De los 3 registros que debe crear como estudiante, **el primer registro debe contener sus datos propios, y los otros 2 de los compañeros que estén antes y después en la lista de ESTUDIANTES del Moodle (ordenada por apellido), tomándolo de manera circular, es decir, como si antes del primer estudiante de la lista estuviera el último y después del último, el primero. El email está en el Moodle,** los demás datos pueden ser inventados. En los campos **id_institucion_persona**, **id_entidad_persona** e **id_municipio_persona** ponga 1, 32 y 32056 respectivamente en todos los registros a insertar. **NOTA:** Los valores para campos de tipo DATE se especifican con una cadena de formato "YYYY-MM-DD" donde YYYY es el año, MM es el mes y DD es el día, por ejemplo "2021-09-13" representa el 13 de septiembre del 2021.
- Agregar 3 registros a la tabla datos_estudiante_XXXXXXX que representan los datos de los 3 estudiantes que agregó en la tabla persona_XXXXXXX (por lo cual el email es el mismo que esa otra tabla), **SI CONOCE LAS FECHAS DE INICIO DE CARRERA Y FECHA ESPERADA DE TERMINACION, USE TALES DATOS, DE OTRA MANERA USE DATOS FICTICIOS (PERO EN UN RANGO REAL**

DE ACUERDO A LA DURACION NORMAL DE LA CARRERA QUE ES DE CINCO AÑOS).

- Agregar 1 registro a la tabla equipo_XXXXXXX especificando el nombre de equipo que deseen, el email del profesor que agregaron en la tabla persona_XXXXXXX se usara como valor para el campo email_coach, el email de los 3 estudiantes que agregó en la tabla persona_XXXXXXX pónganlos en email_concursante1, email_concursante2 y email_concursante3 y en el campo email_reserva ponga un NULL.

Después de estos comandos de inserción de registros ponga los siguientes comandos al final del archivo (recuerde sustituir los XXXXXXXX por su matrícula):

```
SELECT email_persona, nombre_persona, apellidos_persona,
sexo_persona, tipo_persona FROM persona_XXXXXXX;
SELECT * FROM datos_estudiante_XXXXXXX;
SELECT * FROM equipo_XXXXXXX;
```

En resumen, el archivo tablas.sql deberá contener 12 comandos SQL y el archivo datos.sql deberá contener 15 comandos SQL.

ANTES DE ENTREGAR LOS SCRIPTS VERIFIQUEN QUE ESTOS FUNCIONAN CORRECTAMENTE CONECTANDOSE A LA CONSOLA DE MYSQL Y EJECUTANDOLOS CON EL COMANDO SOURCE.

NOTAS IMPORTANTES

1. Recuerda que de acuerdo con lo visto en las prácticas anteriores, el proceso que debes estar haciendo es:

- a) Haz cambios en los archivo .sql
- b) Verifica que funcionan correctamente ejecutándolos en MySQL (trata de ir haciendo cambios incrementales, por ejemplo, agrega algunas de las sentencias SQL que debe tener uno de los archivos, y prueba que funcione antes de agregar más sentencias SQL)
- c) Para avisar a tu repositorio local que registre los cambios emite los comandos `git add NNNNN.sql` (donde NNNNN es el nombre del archivo que modificaste) y `git commit -m "MENSAJE"` donde MENSAJE es un texto que describe brevemente, pero de manera clara los cambios que realizaste desde el último commit.

2. Después de haber hecho todos los commits que completan tus scripts, súbelos al repositorio remoto dando `git push`

3. Cada vez que hagas `git push` se realizarán automáticamente pruebas sobre tus scripts para verificar si funcionan correctamente. Para esta práctica en particular no se te proporcionara una calificación de manera inmediata pues debe ser revisada con mayor detalle. Verifica en todo caso que las pruebas que se hacen no marquen errores. Recuerda que en la página de tu repositorio en la sección **Pull Requests**, se encuentra una subsección de nombre **Feedback**, donde podrás encontrar los resultados de las pruebas en la pestaña denominada **Check** (expandiendo la parte que dice **Run education/autograding@v1**), y cualquier comentario general que el profesor tenga sobre tu código en la pestaña **Conversation**.