

Practica

SQL

Crea un repositorio publico en alguna plataforma Git (de preferencia **GitHub**) con el nombre: **campus-practice-sql**.

Crea los scripts (.sql) correspondientes a cada uno de los siguientes puntos:

1. **create_tables.sql**

Crea las siguientes tablas:

1.1. Crea una tabla "Company" con las siguientes columnas y restricciones

- 1.1.1. Id (entero, unico, identificador)
- 1.1.2. Name (unico, texto de longitud no mayor a 150).
- 1.1.3. Address (texto, con limite de longitud maxima).

1.2. Crea una tabla "Status" con las siguientes columnas y restricciones

- 1.2.1. Id (entero, identificador unico)
- 1.2.2. Name (texto unico con longitud no mayor a 50).

1.3. Crea una tabla "Projects" con las siguientes columnas y restricciones

- 1.3.1. Id (entero, identificador unico).
- 1.3.2. Name (unico, texto de longitud no mayor a 150).
- 1.3.3. StartDate (fecha de inicio).
- 1.3.4. Deadline (fecha limite de entrega).
- 1.3.5. FinishedOn (fecha en que se entrega el proyecto)
- 1.3.6. StatusId (entero, identificador del *Status*).

1.4. Crea una tabla "Employees" con las siguientes columnas y restricciones

- 1.4.1. Id (entero, unico, identificador)
- 1.4.2. FirstName (texto de longitud no mayor a 50).
- 1.4.3. LastName (texto de longitud no mayor a 50).
- 1.4.4. Email (unico, texto de longitud no mayor a 100).
- 1.4.5. Phone (unico, texto de longitud no mayor a 12).
- 1.4.6. Salary (decimal).

2. **relate_tables.sql**

Relaciones entre tablas (crea tablas pivote o agrega columnas a las tablas entides segun sea necesaria).

2.1. Relaciona la tabla "Company" con "Employees", de tal forma que un empleado pertenezca a una compania *obligatoriamente*.

2.2. Relaciona la table "Projects" con "Employees" de tal forma que un empleado pueda pertenecer a varios proyectos, y varios empleados puedan pertenecer a un mismo proyecto.

3. **insert_data.sql**

Agrega los registros **anexos en las tablas de este documento**, a las tablas creadas anteriormente (considera que todas la companias tengan empleados relacionados, y todos los proyectos tengan al menos 1 empleado.):

3.1. Da de alta 5 status (en proceso, pendiente, cancelado, finalizado, en pausa).

3.2. Inserta 3 companias.

3.3. Inserta 5 proyectos.

3.4. Inserta 10 empleados.

4. **queries.sql**

Realiza las siguientes consultas (sin retornar las llaves foraneas):

4.1. Devuelve todas las *companias*.

4.2. Devuelve todos los *empleados*.

4.3. Devuelve los proyectos que hayan sido finalizados antes del deadline.

4.4. Devuelve los proyectos que no hayan empezado a partir de la fecha actual (en que se aplica esta practica).

4.5. Devuelve los empleados con salario mayor a 10,000.

4.6. Devuelve los empleados de Atoz.

4.7. Devuelve los empleados que no son de Disney.

4.8. Devuelve los empleados con sus respectivas companies, ordenados primeramente por el nombre de sus companies y Luego por sus apellidos.

4.9. Devuelve los empleados que pertenezcan a proyectos que esten en proceso.

4.10. Devuelve los empleados sin proyectos pendientes o en proceso.

5. **triggers.sql**

Crea los triggers necesarios para las siguientes indicaciones:

5.1. Generar un error para prevenir "inserts" de proyectos cuyo Deadline sea antes que la fecha de inicio.

5.2. Al insertar proyectos, actualizar sus estatus a "pendientes", o "en proceso" en caso que la fecha actual sea posterior a StartDate y antes del deadline.

6. **functions.sql**

6.1. Crea una funcion "GetLastProjetIdByEmployeeName()" que reciba el nombre de un empleado y retorne el Id del proyecto mas reciente del empleado ingresando.

6.2. Crea una funcion "HasAnyProject()" que reciba el nombre de un empleado y retorne un bit, indicando 1 en caso de que el empleado pertenezca a mas de 1 proyecto o 0 en caso contrario.

7. **store-procedures.sql**

6.1. Crea un store procedure que reciba el nombre de una compania y el nombre de un proyecto, el store procedure debera realizar la siguientes operaciones por cada empleado perteneciente a la compania:

- Debera utilizar la funcion HasAnyProject() para verificar que el empleado pertenezca a algun proyecto, en caso que asi sea entonces se deberá eliminar al empleado de cada unos de sus proyectos.
- Posteriormente, el empleado sera asignado al proyecto segun el nombre ingresado.

Finalmente, el repositorio GitHub de nombre **campus-practice-sql**, sube los archivos ".sql".

Anexo

Tabla Status.

Name
Pendiente
En
proceso
Cancelado
Finalizado
En pausa

Tabla Employees

FirstName	LastName	Email	Phone	Salary
Juan	Perez	juan@jmail.com	9991808182	9,500
Paco	Ochoa	paco@jmail.com	9991808183	8,000
Pedro	Fernandez	pedro@jmail.com	9991808184	12,500
Sofi	Hernandez	sofi@jmail.com	9991808185	11,000
Isabella	Smith	isabella@jmail.com	9991808186	9,000
Eduardo	Jimenez	eduardo@jmail.com	9991808187	11,000
Jose	Pavon	jose@jmail.com	9991808188	12,000
Pancho	Fernandez	pancho@jmail.com	9991808189	12,500
Francisco	Fernandez	francisco@jmail.com	9991808190	25,000
Diego	Olivarez	diego@jmail.com	9991808191	9,000

Tabla Companies

Name	Address
	5151 W 29th St #2201
IMC	Greeley, Colorado(CO), 80634
	2007 Ardmore Hwy
Atoz	Ardmore, Tennessee(TN), 38449
	4226 Highgate Dr
Disnei	Horn Lake, Mississippi(MS), 38637

Tabla Projects

Name	StartDate	Deadline	FinishedOn	Status
				En
Dainler Learning	2/7/1995	2/22/2050	-	proceso
Provident Software	9/15/2022	2/28/2023	-	Pendiente
DataAnlysis	1/31/2023	10/5/2023	-	Pendiente
SoftCentral				
Migration	5/2/2021	7/25/2022	1/1/2022	Finalizado
Atoz Insight	12/30/2022	10/1/2024	-	Cancelado