De acuerdo con las siguientes tablas hipotéticas que se encuentran en un SQL Server, resuelva las siguientes preguntas:

|  |  |
| --- | --- |
| **CatAlumnos** | **CatGrupos** |
| |  |  | | --- | --- | | **Column Name** | **Data Type** | | idAlumno | Int (PK) | | nombre | varchar(255) | | apellidoPaterno | varchar(255) | | apellidoMaterno | varchar(255) | | fechaNacimiento | Date | | idGrupo | int | | estatus | char(1) | | |  |  | | --- | --- | | **Column Name** | **Data Type** | | idGrupo | Int (PK) | | grupo | varchar(255) | | estatus | char(1) | |

Los valores permitidos para la columna estatus son los caracteres ‘i’ y ‘a‘ para Activo e Inactivo respectivamente. Ambas tablas se encuentran vacías, es decir se acaban de crear en este día (hipotéticamente).

1. **¿Cómo insertas un registro a la tabla hipotética CatAlumnos con tus datos?**

****INSERT INTO CatAlumnos****

****VALUES(1, ‘Alejandra’, ‘Gorriz’,’Barrera’,’15/09/1994’,1,’a’);****

1. **Escriba la consulta para proyectar todas las columnas y todos los registros de la tabla hipotética CatAlumnos.**

****SELECT \* FROM CatAlumnos;****

1. **¿A qué se denomina Join?**

**Los joins son usados para combinar reigstros de dos o más tablas en una base de datos, se refiere a combinar los campos de las tablas usando un valor que es común para cada una de las tablas. Son necesarios para obtener detalles de datos que son referenciados desde una tabla, y crea una conexión lógica entre tablas.**

1. **¿Cuántos tipos de Join existen? Defina cada uno.**

****Existen 5 tipos de joins. Son:****

****\*Inner Join (o simplemente Join): se obtienen todos los registros que coinciden con la condición. Los que no coinciden no se muestran.****

****\*Outer Join: existen 3 tipos de outer joins, los cuales son:****

****-LEFT: este tipo de join trae todas las filas de la tabla de la izquierda, en combinación con lo que traería un inner join.****

****-RIGHT: este tipo es la contraparte de LEFT JOIN, que hace lo mismo, pero trayendo las filas de la tabla a la derecha.****

****-FULL JOIN o FULL OUTER JOIN: este tipo de join combina RIGHT con LEFT join.****

****\*Natural Join: Se realiza una igualdad con todas las columnas del mismo nombre.****

****\*Cross Join: regresa el producto cartesiano de las tablas.****

****\*Self-join: es parecido al inner join, pero es un join entre una tabla consigo misma.****

1. **Escriba la consulta para mostrar el nombre y apellidos del alumno, así como el nombre de su grupo.**

****SELECT a.nombre, a.apellidoPaterno, a.apellidoMaterno, g.grupo from CatAlumnos AS a JOIN CatGrupos AS g ON(a.idGrupo = g.idGrupo);****

1. **Escriba la consulta para obtener el número total de alumnos por grupo.**

****SELECT COUNT(idAlumno) FROM CatAlumnos GROUP BY idGrupo;****

1. **Escriba la consulta para obtener el nombre del alumno con menor edad, así como dicha fecha.**

****SELECT nombre, MAX(fechaNacimiento) FROM CatAlumnos;****

1. Escriba la consulta para crear una vista que se llame **vDatosAlumnos** en donde muestre el nombre completo del alumno en la columna nombreAlumno así como la edad que tiene cada uno de los alumnos.

**CREATE VIEW vDatosAlumnos AS**

**SELECT CONCAT(nombre, ‘ ’,apellidoPaterno, ‘ ’, apellidoMaterno) AS nombreAlumno, DATEDIFF(YEAR, GETDATE() ,fechaNacimiento) AS Edad FROM CatAlumnos;**

1. Escriba la consulta para hacer la modificación a la vista existente **vDatosAlumnos**, en la que se agregue el grupo de cada alumno.

****ALTER VIEW vDatosAlumnos AS****

**SELECT CONCAT(a.nombre, ‘ ’,a.apellidoPaterno, ‘ ’, a.apellidoMaterno) AS nombreAlumno, DATEDIFF(YEAR, GETDATE() ,a.fechaNacimiento) AS Edad, g.grupo as Grupo**

**FROM CatAlumnos AS a, CatGrupos AS g**

**WHERE a.idGrupo = g.idGrupo;**

1. **¿Cómo se agrega la columna sexo de tipo varchar de 15 caracteres a la tabla hipotética CatAlumnos?**

**ALTER TABLE CatAlumnos ADD sexo VARCHAR(15);**

1. **¿Cómo actualizo el valor ‘i’ de inactivo al alumno ‘Carlos’ en el campo ‘estatus’?**

****UPDATE CatAlumnos SET estatus = ‘i’ WHERE nombre LIKE ‘Carlos’;****

1. **¿Qué significa hacer COMMIT?**

****Significa hacer una serie de transacciones una parte permanente de una base de datos.****

1. **¿Qué es una “Primary Key” o llave principal o primaria? ¿Cuál es su función?**

****Es un identificador único para las filas en una tabla. Este no puede ser nulo ni puede tener duplicados.****

1. **¿Qué son las “Foreign Keys” o llave externa o foránea? ¿Cuál es su función?**

**Son un dato que hace referencia a la llave primaria de otra tabla. Este puede ser repetible, a diferencia de una Primary Key, y sirve para que exista una conexión lógica entre tablas.**

1. **¿La tabla CatAlumnos tiene Foreign Keys? Si la respuesta es negativa escriba la consulta para crear la Foreign Key que se requiere.**

****Sí, el campo idGrupos es una llave foránea.****

****Si no la tuviera, la consulta para crearla sería:****

****ALTER TABLE CatAlumnos ADD CONSTRAINT FK\_grupo FOREIGN KEY (idGrupo) REFERENCES CatGrupos(idGrupo) ON DELETE CASCADE ON UPDATE CASCADE;****

1. **Listar todos los posibles valores que se pueden almacenar en un campo de datos de tipo BOOLEAN**

**TRUE, FALSE, UNKNOWN**

1. **¿Cómo trabajan o qué significan los valores NULL en SQL?**

**Estos se dan cuando hay un valor faltante en una columna.**

1. ¿**Qué es la Normalización?**

****Es una técnica para diseñar bases de datos, para que las redundancias entre datos se reduzcan al mínimo, esta consta de tres principales reglas, que también son conocidas como formas normales de Codd.****

1. **¿Qué es un Trigger? Y ¿Cuál es su función?**

**Es un objeto que se asocia a una o varias tablas, que sirve para activar una serie de acciones tras la ejecución de un evento específico.**

1. **¿Qué es DML, DDL, DCL?**

**DML: Data Manipulation Language, se usa para la gestion de datos dentro de una base de datos.**

**DDL: Data Definition Language, para definir la estructura de una base de datos.**

**DCL: Data Control Language, controla el acceso a los datos de una base.**

1. **¿Cuál de las siguientes afirmaciones son comandos DML, DCL y DDL?**

|  |  |
| --- | --- |
| **INSERT** | DML |
| **UPDATE** | DML |
| **GRANT** | DCL |
| **TRUNCATE** | DDL |
| **CREATE** | DDL |

1. **¿Cuál es la diferencia entre los comandos TRUNCATE, DELETE y DROP?**

**DELETE elimina los registros de una tabla que cumplen alguna condición.**

**DROP elimina la tabla en sí.**

**TRUNCATE elimina todos los registros de la tabla, sin borrar la tabla en sí.**

1. ¿Cómo se eliminan todos los registros de la tabla hipotética **CatAlumnos** sin escribir en el log (se hace un borrado rápido)?

**TRUNCATE TABLE CatAlumnos;**

1. ¿Cómo elimino todos los registros de alumnos inactivos de la tabla hipotética **CatAlumnos**?

**DELETE FROM CatAlumnos WHERE estatus LIKE ‘i’;**

1. ¿Qué es una **Common Table Expression** (**CTE**)? ¿para qué sirve y qué limitaciones tiene? Escriba un script de ejemplo.

**Es un conjunto de resultados temporales a los que se puede hacer referencia dentro de otras instrucciones.**

1. Para ver cómo está construida la vista **vDash\_Weekly** ¿Cuál es el comando para obtener la definición de dicha vista?

**sp\_help vDas\_Weekly;**

1. Escriba el script para crear la tabla **CatAlumnos.**

**CREATE TABLE CatAlumnos(**

**idAlumno INT PRIMARY KEY,**

**nombre VARCHAR(255),**

**apellidoPaterno VARCHAR(255),**

**apellidoMAterno VARCHAR(255),**

**fechaNacimiento DATE,**

**idGrupo INT,**

**estatus CHAR(1),**

**FOREIGN KEY(idGrupo) REFERENCES CatGrupos(idGrupos) ON DELETE CASCADE ON UPDATE CASCADE**

**);**

**Nota: para poder crear el constraint de la foreign key es necesario que primero haya sido creada la tabla CatGrupos.**

1. Escriba el script para crear la tabla **CatGrupos.**

**CREATE TABLE CatGrupos(**

**idGrupo INT PRIMARY KEY,**

**grupo VARCHAR(255),**

**estatus CHAR(1)**

**);**