

Introducción

En la gestión de bases de datos relacionales, es esencial comprender conceptos fundamentales como el modelo entidad-relación (E-R), el lenguaje de definición de datos (DDL), el lenguaje de manipulación de datos (DML), las claves primarias (PRIMARY KEY), las claves foráneas (FOREIGN KEY), entre otros. A continuación, se proporciona información sobre estos conceptos y se abordan preguntas específicas relacionadas con consultas SQL y operaciones de conjuntos..



DDL y DML

DDL (Lenguaje de Definición de Datos): Es utilizado para definir y modificar la estructura de la base de datos, como la creación de tablas, índices, y restricciones. Ejemplo:

```
CREATE TABLE estudiantes (  
    id_estudiante INT PRIMARY KEY,  
    nombre VARCHAR(50),  
    edad INT  
);
```

DML (Lenguaje de Manipulación de Datos): Se utiliza para insertar, actualizar y eliminar datos en la base de en la base de datos. Ejemplo:

```
INSERT INTO estudiantes (id_estudiante, nombre, edad) VALUES (1, 'Juan', 20);
```

PRIMARY KEY y FOREIGN KEY

PRIMARY KEY: Es una restricción que garantiza la unicidad de los valores en una valores en una columna, y se utiliza para identificar de manera única cada fila en cada fila en una tabla.

FOREIGN KEY: Establece una relación entre dos tablas mediante la vinculación de vinculación de una columna en una tabla con la clave primaria de otra tabla. tabla.

Tabla e IDENTITY

TABLA: Es una estructura que organiza los datos en filas y columnas. Cada columna tiene un nombre único y almacena un tipo específico de datos.

IDENTITY: Es una propiedad que se utiliza para generar automáticamente valores únicos para una columna, una columna, generalmente se usa para claves primarias.

La cláusula WHERE

La cláusula WHERE se utiliza en las consultas SQL para filtrar los resultados basándose en una condición específica. Por ejemplo:

```
SELECT * FROM estudiantes WHERE edad > 18;
```




La instrucción INNER JOIN y los y los conjuntos

La instrucción INNER JOIN se utiliza para combinar filas de dos o más tablas basándose en una condición de igualdad entre columnas. Por ejemplo:

```
SELECT estudiantes.nombre, cursos.nombre_curso
FROM estudiantes
INNER JOIN inscripciones ON estudiantes.id_estudiante = inscripciones.id_estud
INNER JOIN cursos ON inscripciones.id_curso = cursos.id_curso;
```

Ejemplo de INNER JOIN

Supongamos que tenemos estas tablas :

```
CREATE TABLE empleados (  
  id_empleado INT PRIMARY KEY,  
  nombre VARCHAR(50),  
  id_departamento INT  
);  
  
CREATE TABLE departamentos (  
  id_departamento INT PRIMARY KEY,  
  nombre_departamento VARCHAR(50)  
);  
  
CREATE TABLE tareas (  
  id_tarea INT PRIMARY KEY,  
  nombre_tarea VARCHAR(50),  
  id_empleado_responsable INT  
);
```

Y queremos obtener los nombres de los empleados y los nombres de las tareas asignadas usando INNER JOIN:

```
SELECT empleados.nombre AS nombre_empleado, tareas.nombre_tarea  
FROM empleados  
INNER JOIN tareas ON empleados.id_empleado = tareas.id_empleado_responsable;
```

