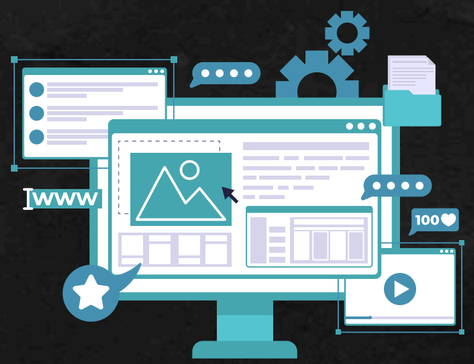
PRESENTACIÓN
DE PROYECTO

FUNDAMENTOS DE SQL Y GESTIÓN DE BASES DE DATOS

INTEGRANTES

- MABY ESTHER SANTOS
- DANIA MARITZA GOMEZ SANCHEZ
- ALEX FERNANDO MALDONADO
- JOSE WILBERTO ESPINOZA
- ADOLFO JOSE ZUNIGA SANTOS
- JESSY GERALDINA GONZALES REYES
- ANGIE CAROLINA HERNANDEZ
- JOSE NOE LEON PEREZ



INTRODUCCIÓN

Las bases de datos juegan un papel fundamental en la organización, almacenamiento y gestión de la información en distintos sectores. SQL (Structured Query Language) es el lenguaje estándar para la manipulación y gestión de bases de datos relacionales, permitiendo realizar operaciones como la creación, modificación, consulta y eliminación de datos.

OBJETIVOS

Objetivo General

- Comprender los fundamentos del lenguaje SQL y su aplicación en la gestión de bases de datos relacionales.

Objetivos Específicos

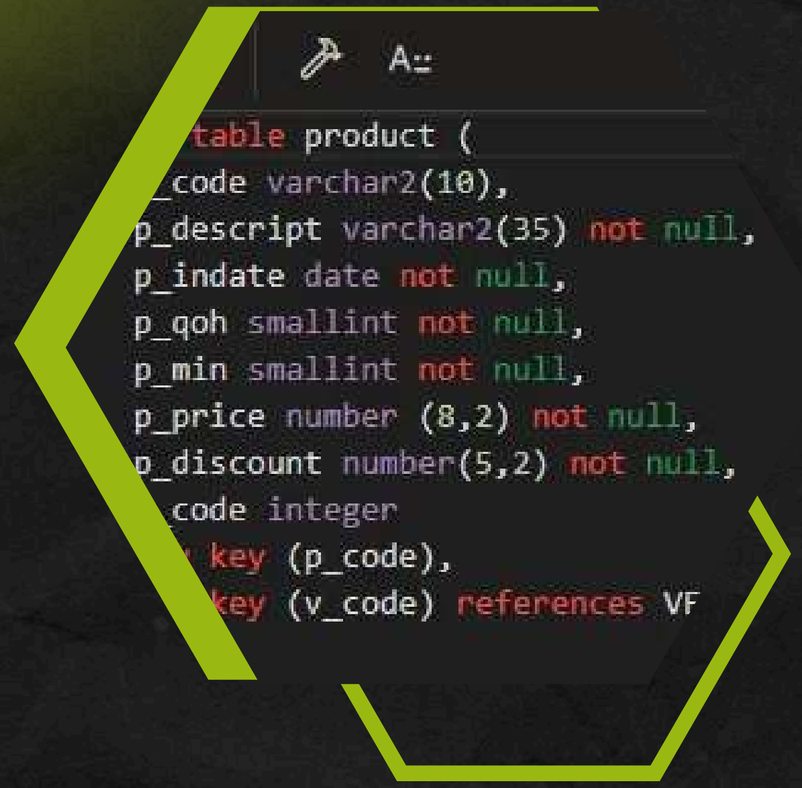
- Explicar la creación, modificación y eliminación de datos en bases de datos.
- Analizar la importancia de índices, claves primarias y foráneas en la optimización de consultas.
- Explorar las funciones de agregación y la manipulación avanzada de datos en SQL.
- Evaluar los métodos de compresión, eliminación y restauración de datos en SQL.

LENGUAJE SQL

Es la herramienta principal en la gestión de bases de datos relacionales. Se basa en un conjunto de sentencias estructuradas que permiten manipular y gestionar la información de manera eficiente.

PRINCIPALES FUNCIONALIDADES DE SQL

CREACIÓN DE TABLAS



Las bases de datos están compuestas por tablas que almacenan registros en filas y columnas. Cada tabla debe tener una clave primaria para garantizar la unicidad de los datos y evitar redundancias. Además, se pueden definir claves foráneas para establecer relaciones entre distintas tablas.

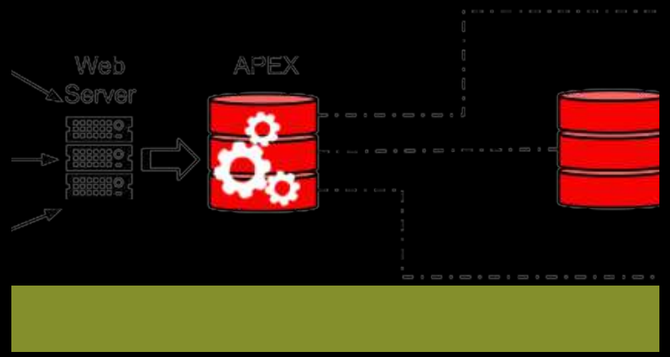
CREACIÓN DE TRIGGERS

Los triggers son procedimientos almacenados que se ejecutan automáticamente ante ciertos eventos en una tabla, como inserciones, actualizaciones o eliminaciones. Estos garantizan la integridad y coherencia de los datos.

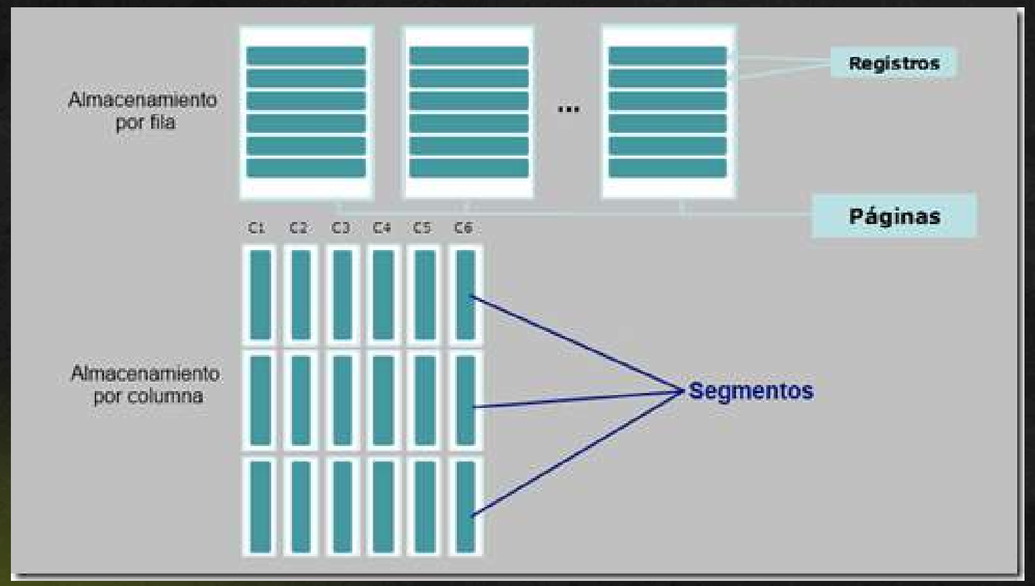
```
Create Trigger Tr_insert_cliente
on customers for Delete
as
Begin
Insert into HistorialEliminacion(fecha, accion, usuar
values(getdate(), 'Se elimino un cliente ', user)
End
Go
```

INSERCIÓN DE DATOS

SQL permite poblar la base de datos con información relevante y posteriormente recuperarla mediante la sentencia SELECT, que incluye filtros avanzados, ordenamientos y agrupaciones.



INDEXACIÓN DE COLUMNAS



Mejora el rendimiento de las consultas al acelerar la búsqueda de datos. Un índice es una estructura que almacena referencias a los registros de una tabla, permitiendo una recuperación más rápida de la información.

CONSULTAS DE DATOS

Se utilizan para recuperar información almacenada en la base de datos mediante la sentencia SELECT, que puede incluir condiciones, agrupaciones y ordenamientos para obtener resultados específicos.

Una consulta de base de datos puede ser una consulta de selección o una consulta de acción. La primera se utiliza para obtener datos de la base de datos que coincidan con los parámetros establecidos por el usuario. La segunda, en cambio, solicita que se realice una acción determinada sobre los datos, como actualizarlos, insertarlos o borrarlos.

```
228  ---5. Consultas de Datos (Querying Data)
229  -- Consultar todos los empleados
230  SELECT * FROM empleados;
```

Resultados | Explicar | Describir | SQL Guardado | Historial

EMPLEADO_ID	NOMBRE	APELLIDO	FECHA_CONTRATACION	SALARIO	DEPARTAMENTO_ID	CARGO
2	María	González	22/03/19	4200	2	Desarrollador Senior
4	Laura	Martínez	05/09/20	3600	4	Especialista en Marketing Digital
1	Juan	Pérez	15/01/20	3500	1	Analista de RRHH
3	Carlos	Rodríguez	10/06/21	3800	3	Contador

4 filas devueltas en 0,02 segundos Descargar

ADICION DE COLUMNAS

Es posible modificar la estructura de una tabla agregando nuevas columnas sin perder los datos existentes, lo que permite ampliar la base de datos según las necesidades del sistema.

```
24   -- Añadir la restricción de clave foránea
25   ALTER TABLE empleados
26   ADD CONSTRAINT fk_departamento
27   FOREIGN KEY (departamento_id)
28   REFERENCES departamentos(departamento_id);
29
30   -- Tabla para el historial de cambios de salario
31 > CREATE TABLE historial_salarios (
```

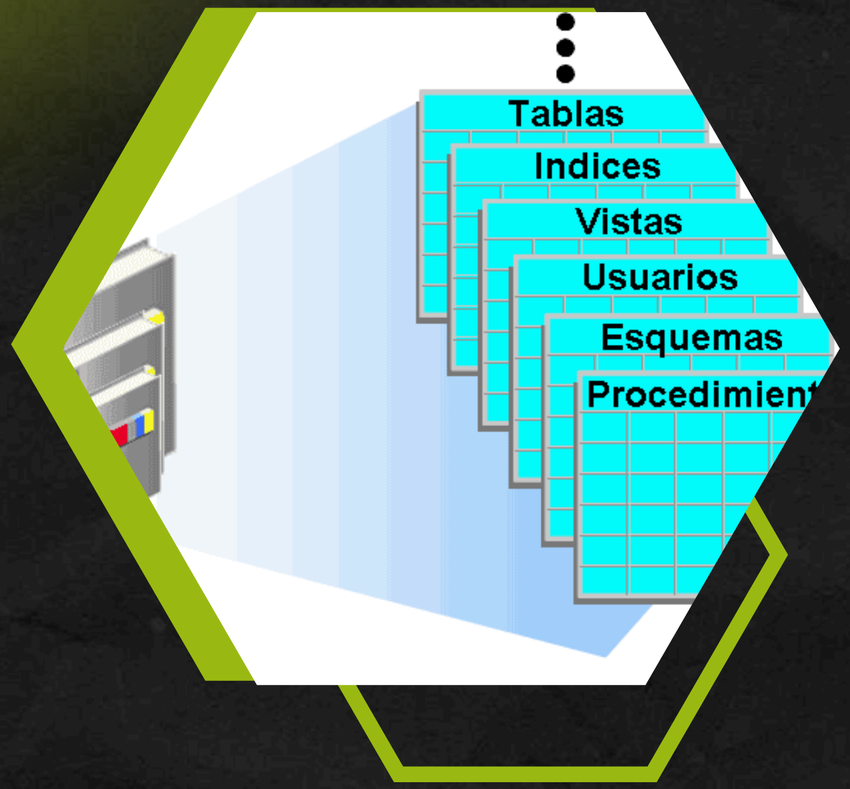
Resultados | Explicar | Describir | SQL Guardado | Historial

```
Tabla modificada.

0,08 segundos
```

Se utiliza el valor ADD COLUMN con el mandato ALTER TABLE . Al añadir las columnas, debe especificar el nombre de columna y el tipo de datos.

CONSULTA DEL DICCIONARIO DE DATOS DE ORACLE



Las bases de datos están compuestas por tablas que almacenan registros en filas y columnas. Cada tabla debe tener una clave primaria para garantizar la unicidad de los datos y evitar redundancias. Además, se pueden definir claves foráneas para establecer relaciones entre distintas tablas.

El **diccionario de datos** contiene información tal como los privilegios de los usuarios, restricciones de integridad definidas para algunas tablas, nombres y tipos de **datos** de todas las columnas y otra información acerca del espacio asignado y utilizado por los objetos de un esquema.

ACTUALIZACIÓN DE DATOS

SQL permite actualizar registros existentes sin necesidad de eliminarlos y volver a insertarlos, garantizando la coherencia de la información almacenada.
La consulta de actualización en SQL modifica los datos existentes en una tabla. Este comando permite cambiar una o más columnas de una o varias filas de una tabla.
La sintaxis básica es la siguiente:

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

```
301  ---8. Actualización de Datos (Updating Data)
302  -- Actualizar el salario de un empleado (activará el trigger)
303  UPDATE empleados
304  SET salario = 3800
305  WHERE empleado_id = 1;
306
307  -- Actualizar el departamento de un empleado
308  UPDATE empleados
309  SET departamento_id = 2
310  WHERE empleado_id = 3;
311
312  -- Actualizar múltiples columnas
313  UPDATE empleados
314  SET telefono = '555-123-4567',
```

Resultados | Explicar | Describir | SQL Guardado | Historial

```
1 fila(s) actualizada(s).
```

CONSULTAS AGREGADAS

Las funciones agregadas en SQL son funciones que realizan cálculos sobre un conjunto de valores y devuelven un único valor. Ejemplos comunes son SUM(), COUNT(), AVG(), MIN(), y MAX(), que se utilizan para sumar, contar, promediar o encontrar valores mínimos y máximos en un grupo de datos.

- `COUNT(column_name | *)` devuelve el número de filas de una tabla.
- `SUM(column_name)` devuelve la suma de los valores de una columna numérica.
- `AVG(column_name)` devuelve el valor medio de una columna numérica.
- `MIN(column_name)` devuelve el valor mínimo de una columna seleccionada.
- `MAX(column_name)` devuelve el valor máximo de una columna seleccionada.

Las funciones de agregación de SQL acumulan datos de varias filas en una única fila de resumen. El valor acumulado se basa en los valores de la columna pasada como argumento. Podemos agrupar las filas utilizando una cláusula GROUP BY y filtrarlas aún más utilizando una cláusula HAVING.

Code

```
SELECT * FROM Books;
```

Id	Author	Title	Price
234	Anthony Molinaro	SQL Cookbook	20.00
235	Alan Beaulieu	Learning SQL	25.00
236	Donald Knuth	Things a Computer Scientist Rarely Talks About	25.00
237	Donald Knuth	The Art of Computer Programming	27.00

Code

```
SELECT COUNT(*) AS NumberOfBooks
FROM Books;
```

NumberOfBooks
4

COMPRESION DE DATOS

La compresión de datos en SQL es una técnica utilizada para reducir el tamaño de los datos almacenados en una base de datos. Esto se logra mediante la eliminación de redundancias y la optimización del almacenamiento de datos

```sql
-- Habilitar la compresión de filas en una tabla existente
ALTER TABLE MiTabla
REBUILD WITH (DATA_COMPRESSION = ROW);

-- Habilitar la compresión de páginas en una tabla existente
ALTER TABLE MiTabla
REBUILD WITH (DATA_COMPRESSION = PAGE);
```

ELIMINACION DE DATOS Y TABLAS

La eliminación de datos en SQL es el proceso mediante el cual se remueven registros de una base de datos. Existen diferentes formas de hacerlo dependiendo de la necesidad: DELETE permite borrar registros específicos sin afectar la estructura de la tabla, mientras que TRUNCATE elimina todos los datos de una tabla de manera más rápida y eficiente, aunque sin posibilidad de recuperación inmediata.

```sql
DELETE FROM nombre_tabla WHERE condicion;
```

```sql
TRUNCATE TABLE nombre_tabla;
```

La eliminación de tablas es un proceso irreversible que borra por completo una tabla y todos los datos almacenados en ella. La instrucción DROP TABLE se utiliza para eliminar una tabla.

```sql
DROP TABLE nombre_de_tabla;
```

```sql
DROP TABLE tabla1, tabla2, tabla3;
```

RESTAURACION DE TABLAS ELIMINADAS

Cuando se elimina una tabla en SQL usando el comando DROP TABLE, la tabla y todos sus datos se eliminan permanentemente de la base de datos. Sin embargo, hay algunas estrategias que se pueden utilizar para restaurar una tabla eliminada

Rollback

El rollback es un término utilizado en informática, especialmente en el contexto de bases de datos y sistemas de gestión de transacciones, que se refiere a la acción de revertir un conjunto de operaciones a un estado anterior. Se realiza para asegurar la integridad y consistencia de los datos en caso de que ocurra un error, fallo o cualquier condición que requiera que las operaciones se deshagan.

Si ocurre un error durante una transacción, el rollback permite revertir todas las operaciones realizadas hasta ese punto para eliminar dicho error y mantener la coherencia original en los datos.

Retroseso básico

```
START TRANSACTION;
UPDATE accounts SET balance = balance - 100 WHERE account_id = 1;
ROLLBACK;
```

Retroseso tras error

```
START TRANSACTION;
INSERT INTO orders (customer_id, product_id, quantity) VALUES (1, 2, 3);
-- Error: Duplicate entry for key 'PRIMARY'
ROLLBACK;
```

MUCHAS GRACIAS