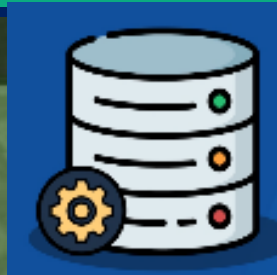




CBTis#42

Implementa el diseño físico de la base de datos.



Actividad 1-Unidad 2 **Programación**

MÓDULO IV:
DESARROLLA SOFTWARE DE APLICACIÓN
WEB CON ALMACENAMIENTO
PERSISTENTE DE DATOS.

SUBMÓDULO 1:
CONSTRUYE BASES DE DATOS PARA
APLICACIONES WEB.

Elaborado por: Abraham Castañeda Quintero
Docente: Alejandra Serrano Luna
Semestre y grupo: 5° "B"

Índice.

Contenido

Implementa el diseño físico de la base de datos.....	0
Índice.....	1
Introducción.....	2
Desarrollo.....	3
1-¿Qué es el lenguaje SQL?	3
1.1-Proceso de ejecución de sentencia SQL.	4
2-¿Qué es el lenguaje de definición de datos?	5
2.1-Operaciones básicas	5
2.1.1 ¿Para qué se utiliza el comando Create?	5
2.1.2 ¿Para qué se utiliza el comando Alter?	6
2.1.3 ¿Para qué se utiliza el comando Drop?	6
2.1.4 ¿Para qué se utiliza el comando Truncate	7
3-¿Qué es una vista en el lenguaje SQL?	8
4-¿Qué son restricciones en el lenguaje SQL?	9
4.1 NOT NULL	9
4.2 UNIQUE	9
4.3-PRIMARY KEY	10
4.4-FOREIGN KEY	10
4.5-CHECK	11
4.6-DEFAULT.....	11
Conclusión.....	12
Referencias.	13

Introducción.

En esta la actividad No. 2 unidad 2 del modulo IV de la carrera de programación, Es necesario conocer de las sentencias que se utilizan en el lenguaje de SQL para así poder desarrollar un modelo físico de base de datos para ello se pidió elaborar esta actividad donde se investigan sobre el funcionamiento de dicho lenguaje, así como también su sintaxis básica.

Para poder trabajar con el modelo físico es necesario saber como se construye una base de datos hablando de ejecutar los comandos de construcción y no solo hacerla con las herramientas graficas que ofrece por ejemplo el manejador de bases de datos MySQL con PHPmyAdmin, la intención de hacerlo de esta manera es conocer como se trabaja en una base de datos en un entorno profesional.

Desarrollo.

1-¿Qué es el lenguaje SQL?

SQL (por sus siglas en inglés Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje de dominio específico utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales. Una de sus principales características es el manejo del álgebra y el cálculo relacional para efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como realizar cambios en ellas.

Originalmente basado en el álgebra relacional y en el cálculo relacional, SQL consiste en un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de control de datos. El alcance de SQL incluye la inserción de datos, consultas, actualizaciones y borrado, la creación y modificación de esquemas y el control de acceso a los datos. También el SQL a veces se describe como un lenguaje declarativo, también incluye elementos procesales.

SQL fue uno de los primeros lenguajes comerciales para el modelo relacional de Edgar Frank Codd como se describió en su artículo de investigación de 1970 El modelo relacional de datos para grandes bancos de datos compartidos. A pesar de no adherirse totalmente al modelo relacional descrito por Codd, pasó a ser el lenguaje de base de datos más usado.

SQL pasó a ser el estándar del Instituto Nacional Estadounidense de Estándares (ANSI) en 1986 y de la Organización Internacional de Normalización (ISO) en 1987. Desde entonces, el estándar ha sido revisado para incluir más características. A pesar de la existencia de ambos estándares, la mayoría de los códigos SQL no son completamente portables entre sistemas de bases de datos diferentes sin otros ajustes.



1.1-Proceso de ejecución de sentencia SQL.

El Motor de base de datos de SQL procesa consultas en varias arquitecturas de almacenamiento de datos como tablas locales, tablas con particiones y tablas distribuidas en varios servidores. En los temas siguientes se trata el modo en que SQL procesa las consultas y optimiza la reutilización de consultas a través del almacenamiento en caché de los planes de ejecución.

Modos de ejecución

Motor de base de datos de SQL puede procesar las instrucciones Transact-SQL mediante dos modos de procesamiento distintos:

Ejecución del modo de fila

Ejecución del modo por lotes

Ejecución del modo de fila

La ejecución del modo de fila es un método de procesamiento de consultas que se usa con tablas RDMBS tradicionales, donde los datos se almacenan en formato de fila. Cuando se ejecuta una consulta y accede a los datos de tablas de almacén de filas, los operadores del árbol de ejecución y los operadores secundarios leen todas las filas necesarias, en todas las columnas especificadas en el esquema de tabla. De cada fila que se lee, SQL Server recupera las columnas que son necesarias para el conjunto de resultados, como se hace referencia mediante una instrucción SELECT, un predicado JOIN o un predicado de filtro.

Nota

La ejecución del modo de fila es muy eficaz para escenarios OLTP, pero puede serlo menos cuando se analizan grandes cantidades de datos, por ejemplo en escenarios de almacenamiento de datos.

Ejecución del modo por lotes

La ejecución del modo por lotes es un método de procesamiento de consultas en el que las consultas procesan varias filas a la vez (de ahí el término "por lotes"). Cada columna dentro de un lote se almacena como un vector en un área de memoria independiente, por lo que el procesamiento del modo por lotes se basa en vectores. En el procesamiento del modo por lotes también se usan algoritmos que se optimizan para las CPU de varios núcleos y el rendimiento de aumento de memoria que se encuentran en el hardware moderno.

La ejecución del modo por lotes está estrechamente integrada con el formato de almacenamiento de almacén de columnas y optimizada alrededor del mismo. El procesamiento del modo por lotes funciona en los datos comprimidos siempre que sea posible y elimina el operador de intercambio que usa el procesamiento del modo de fila. El resultado es un mayor paralelismo y un rendimiento más rápido.

Cuando una consulta se ejecuta en el modo por lotes y tiene acceso a los datos de índices de almacén de columnas, los operadores del árbol de ejecución y los operadores secundarios leen varias filas en segmentos de columna. SQL Server solo lee las columnas necesarias para el resultado, tal y como se indica en una instrucción SELECT, un predicado JOIN o un predicado de filtro.

Para más información sobre los índices de almacén de columnas, vea Arquitectura de los índices de almacén de columnas.

Nota

La ejecución del modo por lotes es muy eficaz en escenarios de almacenamiento de datos, donde se leen y se agregan grandes cantidades de datos.

Procesamiento de instrucciones SQL

La forma más básica de ejecutar instrucciones Transact-SQL en SQL Server consiste en procesar una única instrucción Transact-SQL. Los pasos que se usan para procesar una única instrucción SELECT que solo hace referencia a tablas base locales (no a vistas ni a tablas remotas) ilustran el proceso básico.

2-¿Qué es el lenguaje de definición de datos?

Un lenguaje de base de datos o lenguaje de definición de datos (Data Definition Language, DDL por sus siglas en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los programadores de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

Un DDL es un lenguaje de programación para definir una estructura de datos. El término DDL fue introducido por primera vez en relación con el modelo de base de datos CODASYL, donde el esquema de la base de datos ha sido escrito en un lenguaje de descripción de datos que describe los registros, los campos, y "conjuntos" que conforman el usuario modelo de datos. Más tarde fue usado para referirse a un subconjunto de SQL, pero ahora se utiliza en un sentido genérico para referirse a cualquier lenguaje formal para describir datos o estructuras de información, como los esquemas XML.

2.1-Operaciones básicas

2.1.1 ¿Para qué se utiliza el comando Create?

Este comando permite crear objetos de datos, como nuevas bases de datos, tablas, vistas y procedimientos almacenados.

```
CREATE TABLE clientes;
```

```
CREATE [OR REPLACE] {DATABASE | SCHEMA} [IF NOT EXISTS] db_name  
    [create_specification] ...
```

create_specification:

```
    [DEFAULT] CHARACTER SET [=] charset_name  
    | [DEFAULT] COLLATE [=] collation_name  
    | COMMENT [=] 'comment'
```


2.1.2 ¿Para qué se utiliza el comando Alter?

Este comando permite modificar la estructura de una tabla u objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc.

```
ALTER TABLE alumnos ADD edad INT UNSIGNED;
```

```
ALTER [ONLINE] [IGNORE] TABLE [IF EXISTS] tbl_name
    [WAIT n | NOWAIT]
    alter_specification [, alter_specification] ...

alter_specification:
    table_option ...
    | ADD [COLUMN] [IF NOT EXISTS] col_name column_definition
      [FIRST | AFTER col_name ]
```

2.1.3 ¿Para qué se utiliza el comando Drop?

Este comando elimina un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier objeto que el motor de la base de datos soporte. Se puede combinar con la sentencia ALTER.

```
DROP TABLE alumnos;
```

```
DROP DATABASE bufg;
Query OK, 0 rows affected (0.39 sec)

DROP DATABASE bufg;
ERROR 1008 (HY000): Can't drop database 'bufg'; database doesn't exist

\W
Show warnings enabled.

DROP DATABASE IF EXISTS bufg;
Query OK, 0 rows affected, 1 warning (0.00 sec)
Note (Code 1008): Can't drop database 'bufg'; database doesn't exist
```

2.1.4 ¿Para qué se utiliza el comando Truncate

Este comando solo aplica a tablas y su función es borrar el contenido completo de la tabla especificada. La ventaja sobre el comando DELETE, es que si se quiere borrar todo el contenido de la tabla, es mucho más rápido, especialmente si la tabla es muy grande. La desventaja es que TRUNCATE solo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula WHERE. Si bien, en un principio, esta sentencia parecería ser DML (Lenguaje de Manipulación de Datos), es en realidad una DDL, ya que internamente, el comando TRUNCATE borra la tabla y la vuelve a crear y no ejecuta ninguna transacción.

```
TRUNCATE TABLE nombre_tabla;
```

```
SELECT TRUNCATE(1.223,1);
```

```
+-----+
| TRUNCATE(1.223,1) |
+-----+
|           1.2 |
+-----+
```

```
SELECT TRUNCATE(1.999,1);
```

```
+-----+
| TRUNCATE(1.999,1) |
+-----+
|           1.9 |
+-----+
```

```
SELECT TRUNCATE(1.999,0);
```

```
+-----+
| TRUNCATE(1.999,0) |
+-----+
|           1 |
+-----+
```


3-¿Qué es una vista en el lenguaje SQL?

Una vista es una tabla derivada de otras tablas (virtuales).

Una vista se caracteriza porque:

- Una vista es una tabla virtual (no tiene correspondencia a nivel físico)
- Se puede consultar como cualquier tabla básica.

Las vistas ("views") en SQL son un mecanismo que permite generar un resultado a partir de un pedido (query) almacenado, y ejecutar nuevos pedidos sobre este resultado como si fuera una tabla normal.

Por ejemplo, si tenemos unas tablas que representan autores y libros, y queremos hacer un listado plano de libros y sus autores, podemos ejecutar un query que haga una junta (join) entre estas dos tablas. Pero si posteriormente queremos pedir solo unas líneas de este resultado a partir de otro filtro, vamos a tener que re-ejecutar el query completo, agregando nuestro filtro. Obviamente es posible, pero también implica repetir operaciones anteriores. En el caso de tener pedidos complejos, esto puede resultar en una pérdida de eficiencia grande, y mucho trabajo adicional para el desarrollador.

En muchos sistemas de gestión de bases de datos, usar vistas es algo muy simple. Por ejemplo:

```
CREATE VIEW biblio AS SELECT a.id as aid, a.name  
as author, b.id as bid, b.title as book  
FROM author a, book b WHERE b.author_id = a.id AND a.id > 100;
```

4-¿Qué son restricciones en el lenguaje SQL?

Las restricciones en SQL son reglas y restricciones predefinidas que se aplican en una sola columna o en varias columnas, relacionados a los valores permitidos en las columnas, para mantener la integridad, precisión y confiabilidad de los datos de esa columna. En otras palabras, si los datos insertados cumplen con la regla de restricción, se insertarán con éxito. Si los datos insertados violan la restricción definida, la operación de inserción se cancelará.

Las restricciones en SQL se pueden considerar a nivel de columna, donde se especifica como parte de la definición de columna y se aplicarán solo a esa columna, o se declararán de forma independiente a nivel de tabla. En este caso, las reglas de restricción se aplicarán a más de una columna en la tabla especificada. La restricción se puede crear dentro del comando CREATE TABLE T-SQL al crear la tabla o agregarse usando el comando ALTER TABLE T-SQL después de crear la tabla. Al agregar la restricción después de crear la tabla, se verificará previamente la regla de restricción en los datos existentes antes de crear esa restricción.

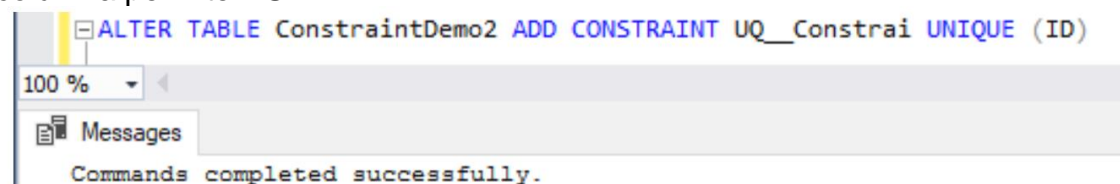
4.1 NOT NULL

Por defecto, las columnas pueden contener valores NULL. Se usa una restricción NOT NULL en SQL para evitar insertar valores NULL en la columna especificada, considerándolo entonces como un valor no aceptado para esa columna. Esto significa que debe proporcionar un valor válido SQL NOT NULL a esa columna en las instrucciones INSERT o UPDATE, ya que la columna siempre contendrá datos.

```
ALTER TABLE ConstraintDemo1 ALTER COLUMN [Name] VARCHAR(50) NOT NULL;
```

4.2 UNIQUE

La restricción UNIQUE en SQL se utiliza para garantizar que no se inserten valores duplicados en una columna específica o combinación de columnas que participen en la restricción UNIQUE y no formen parte de la CLAVE PRIMARIA. En otras palabras, el índice que se crea automáticamente cuando define una restricción ÚNICA garantizará que no haya dos filas en esa tabla que puedan tener el mismo valor para las columnas que participan en ese índice, con la capacidad de insertar solo un valor NULL único en estas columnas, esto si la columna permite NULL.



4.3-PRIMARY KEY

La restricción PRIMARY KEY consta de una columna o varias columnas con valores que identifican de forma única cada fila de la tabla.

La restricción PRIMARY KEY de SQL se combina entre las restricciones UNIQUE y SQL NOT NULL, donde la columna o el conjunto de columnas que participan en PRIMARY KEY no pueden aceptar el valor NULL. Si la CLAVE PRIMARIA se define en varias columnas, entonces se puede insertar valores duplicados en cada columna individualmente, pero es importante mencionar que los valores de combinación de todas las columnas de CLAVE PRIMARIA deben ser únicos. Tenga en cuenta que solo puede definir una CLAVE PRIMARIA por cada tabla, y se recomienda utilizar columnas pequeñas o INT en la CLAVE PRIMARIA.

Además de proporcionar un acceso rápido a los datos de la tabla, el índice que se crea automáticamente, al definir la CLAVE PRIMARIA de SQL, impondrá cualidad de la unicidad de los datos. La CLAVE PRIMARIA se usa principalmente para imponer la integridad de la entidad de la tabla. La integridad de la entidad garantiza que cada fila de la tabla sea de forma única una entidad identificable.

Se puede advertir que La restricción PRIMARY KEY difiere de la restricción UNIQUE en eso; usted puede crear múltiples restricciones ÚNICAS en una tabla, con la capacidad de definir solo una CLAVE PRIMARIA SQL por cada tabla. Otra diferencia es que la restricción UNIQUE permite un valor NULL, pero la PRIMARY KEY no permite valores NULL.

```
USE SQLShackDemo
GO
CREATE TABLE ConstraintDemo3
(
    ID INT PRIMARY KEY,
    Name VARCHAR(50) NULL
);
```

4.4-FOREIGN KEY

Una clave externa es una clave que se utiliza para vincular dos tablas. A veces, esto también se denomina clave de referencia.

Una clave externa es una columna o una combinación de columnas cuyos valores coinciden con una clave principal en una tabla diferente.

La relación entre 2 tablas coincide con la clave principal en una de las tablas con una clave externa en la segunda tabla.

Si una tabla tiene una clave primaria definida en cualquier campo, entonces no puede tener dos registros que tengan el mismo valor de ese campo.

```
CREATE TABLE ORDERS (  
    ID          INT          NOT NULL,  
    DATE        DATETIME,  
    CUSTOMER_ID INT references CUSTOMERS(ID),  
    AMOUNT      double,  
    PRIMARY KEY (ID)  
);  
ALTER TABLE ORDERS  
    ADD FOREIGN KEY (Customer_ID) REFERENCES CUSTOMERS (ID);
```

4.5-CHECK

La restricción CHECK se utiliza para limitar el rango de valores que se puede colocar en una columna.

Si define una restricción CHECK en una sola columna, solo permite ciertos valores para esta columna.

Si define una restricción CHECK en una tabla, puede limitar los valores en ciertas columnas en función de los valores en otras columnas de la fila.

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CHECK (Age>=18)  
);
```

4.6-DEFAULT

La restricción DEFAULT se utiliza para proporcionar un valor predeterminado para una columna.

El valor predeterminado se agregará a todos los registros nuevos Si no se especifica ningún otro valor.

```
CREATE TABLE Orders (  
    ID int NOT NULL,  
    OrderNumber int NOT NULL,  
    OrderDate date DEFAULT GETDATE()  
);
```

Conclusión.

En conclusión, en esta actividad aprendí sobre el funcionamiento del lenguaje SQL y como se debe aplicar, aprendí que es en si el lenguaje además de como se ejecuta, junto con algunos comandos básicos como CREATE, ALTER, DROP, TRUNCATE con su respectiva definición y funcionamiento; Con el comando de view podemos visualizar la estructura de una tabla mas aparte se pidió investigar sobre las restricciones y como se aplican.

Referencias.

Cai, S. (s. f.). MicrosoftDocs/sql-docs.es-es. GitHub. Recuperado 3 de noviembre de 2020, de <https://github.com/MicrosoftDocs/sql-docs.es-es/blob/live/docs/relational-databases/query-processing-architecture-guide.md>

colaboradores de Wikipedia. (2020a, marzo 4). Lenguaje de definición de datos. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Lenguaje_de_definici%C3%B3n_de_datos

colaboradores de Wikipedia. (2020b, octubre 25). SQL. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/>

SQLComandos SQL Básicos. (s. f.). MariaDB KnowledgeBase. Recuperado 3 de noviembre de 2020, de <https://mariadb.com/kb/es/basic-sql-statements/>

Yaseen, A. (2019, 16 diciembre). Restricciones en SQL Server: SQL NOT NULL, UNIQUE y SQL PRIMARY KEY. SQL Shack - articles about database auditing, server performance, data recovery, and more. <https://www.sqlshack.com/es/restricciones-en-sql-server-sql-not-null-unique-y-sqlprimarykey/#:%7E:text=Las%20restricciones%20en%20SQL%20Server,los%20datos%20de%20esa%20columna>