

Contenidos teóricos

Sitio: Agencia de Aprendizaje a lo largo de la Vida
Curso: CAC Big Data MODULO I 2C
Libro: Contenidos teóricos

Imprimido por: Eduardo Moreno
Día: lunes, 14 de agosto de 2023, 16:30

Tabla de contenidos

- 1. Introducción a SQL
- 2. Primary key y foreign key
- 3. Estructura de la Base de datos

1. Introducción a SQL

SQL

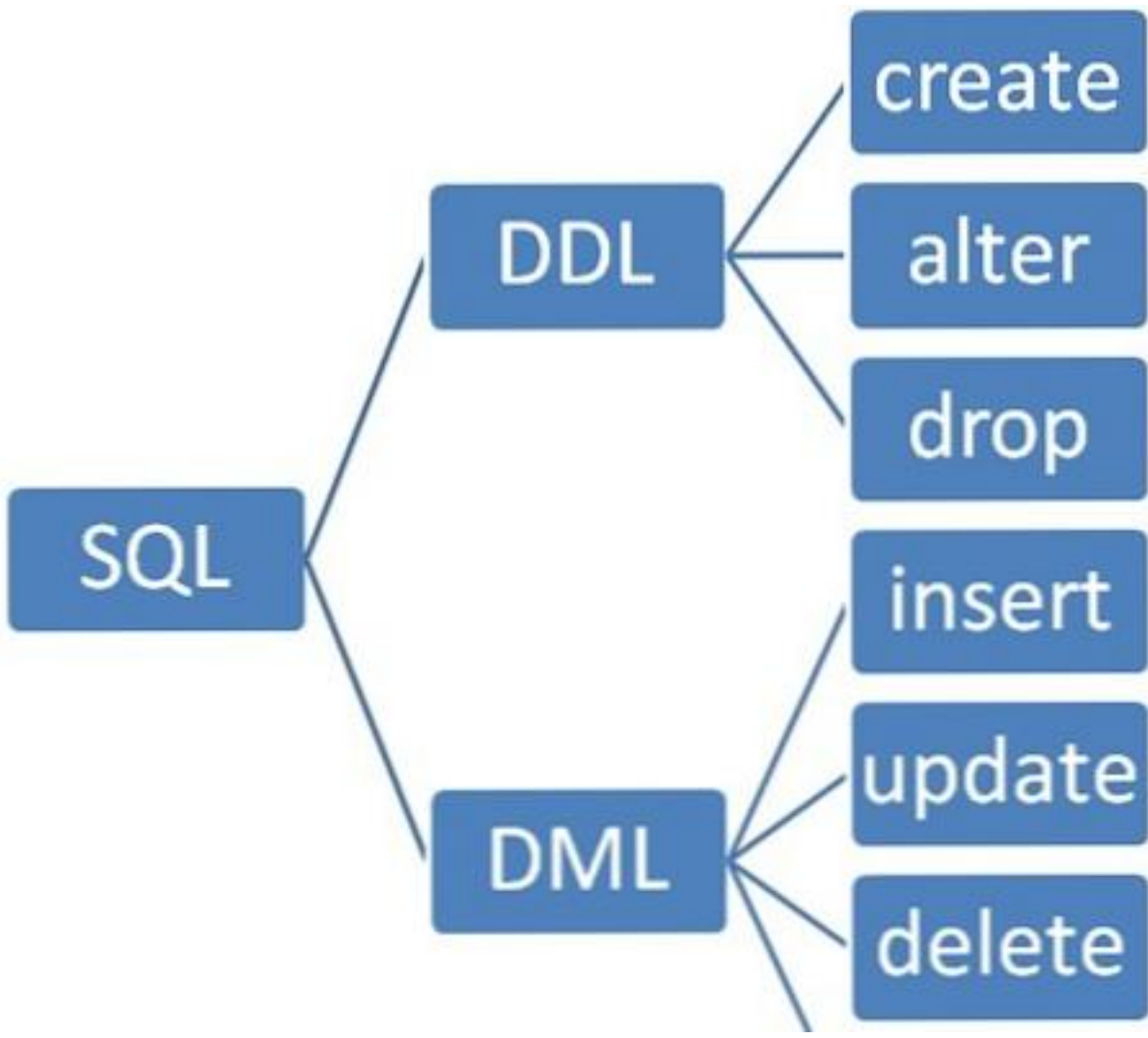
(Structured Query Language – Lenguaje de consulta estructurado)

Es el lenguaje por excelencia para crear y manipular bases de datos relacionales. Permite ser ejecutado directamente dentro de un entorno de trabajo, o puede ser incrustado dentro del código de un programa escrito en otro lenguaje.

Las sentencias SQL pueden ser clasificadas en dos grupos:

DDL (Data Definition Languaje - lenguaje de definición de datos): las sentencias DDL son aquellas utilizadas para la creación de una base de datos y todos sus componentes: tablas, índices, relaciones. Se utilizan para darle estructura a las tablas de la base de datos.

DML (Data Manipulation Languaje - lenguaje de manipulación de datos): las sentencias DML son aquellas utilizadas para insertar, borrar, modificar y consultar los datos de una base de datos.



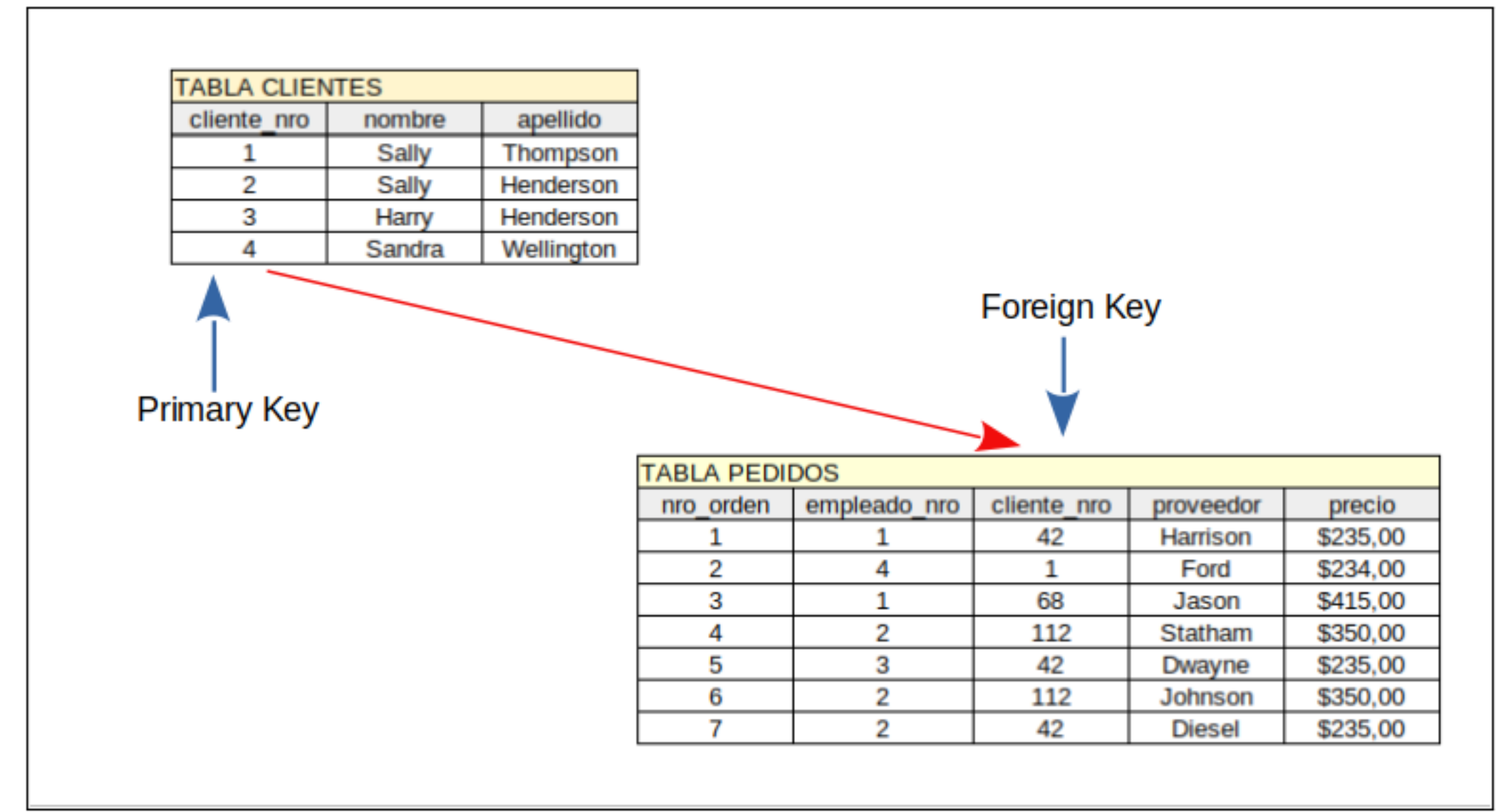
2. Primary key y Foreign key

Primary key y Foreign key

Las claves primarias (Primary Key) son las columnas que contienen valores que identifican de manera única a cada fila o registro de una tabla, esto quiere decir que no se puede repetir. Por ejemplo: un DNI, un código de producto, etc.

Una clave foránea (Foreign Key) es un campo de una tabla "X" que sirve para enlazar o relacionar entre sí con otra tabla "Y" en la cual el campo de esta tabla es una llave primaria (Primary Key). Para que sea una clave foránea un campo, esta tiene que ser una llave primaria en otra tabla.

Por ejemplo, en la tabla clientes la columna cliente_nro es una primary key, pero en una tabla de pedidos representa a quién pertenece ese determinado pedido.



3. Estructura de la Base de datos

SQL: Instrucciones DDL

Creando la estructura de la base de datos

Crear Base de Datos:

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
```

Borrar Base de Datos:

```
DROP DATABASE db_name
```

Usar Base de Datos:

```
USE db_name
```

Crear Tabla

La sentencia CREATE TABLE crea una tabla con el nombre especificado.

Sintaxis:

```
CREATE TABLE nombre_tabla (
```

```
nombre_columna1 [definición_columna][PRIMARY KEY],
```

```
nombre_columna2 [definición_columna]...
```

```
nombre_columnaN [definición_columna
```

```
FOREIGN KEY(nombre_de_otra_tabla) nombre_de_otra_tabla(nombre_columna_pk_otra_tabla)
```

```
);
```

Parámetros opcionales:

IF NOT EXISTS: Evita que se produzca un error si la tabla existe. Sin embargo, no se verifica que la tabla existente tenga una estructura idéntica a la indicada por el CREATE TABLE.

LIKE: CREATE TABLE ... LIKE crea una tabla vacía basada en la definición de otra tabla, incluidos los atributos e índices de columna definidos en la tabla original:

```
CREATE TABLE destino_tbl LIKE origen_tbl;
```

NOT NULL | NULL: especifica si se acepta o no que el dato quede vacío. Si no se especifica NULL ni NOT NULL, la columna se trata como si se hubiera especificado NULL.

DEFAULT: Especifica un valor predeterminado para una columna.

AUTO_INCREMENT: El valor de la columna se incrementa automáticamente a medida que se suman registros.

PRIMARY KEY: se declara implícitamente como NOT NULL.

UNIQUE: Un índice UNIQUE crea una restricción tal que todos los valores del índice deben ser distintos. Se produce un error si intenta agregar una nueva fila con un valor clave que coincide con una fila existente. Un índice UNIQUE sólo permite múltiples valores NULL para las columnas que pueden contener NULL.

FOREIGN KEY: Permiten realizar referencias cruzadas de datos relacionados entre tablas, que ayudan a mantener la coherencia de estos datos dispersos.

Borrar Tabla

```
DROP TABLE [IF EXISTS] tbl_name
```

Modificar Tabla

Agregar o borrar columnas:

```
ALTER TABLE nombre_de_tabla ADD nombre_de_columna tipo_de_dato;
```

```
ALTER TABLE nombre_de_tabla DROP COLUMN nombre_de_columna;
```

Ver la estructura de una tabla

```
DESCRIBE nombre_de_tabla;
```