Agenda

- Tema VI
- Práctica VIII

Facultad de Ingeniería

Diseño físico de una base de datos

Tema VI

Semestre 2024-2

UNAM
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Objetivo

El alumno comprenderá y aplicará los elementos necesarios para la implementación física del diseño lógico de la base de datos a través del lenguaje SQL, así como la manipulación y uso de transacciones a través de sentencias.

Introducción

SQL, acrónimo de Structured Query Language, es un lenguaje que permite interactuar con los DBMS para realizar operaciones sobre los datos o sobre la estructura de los datos.

Historia

En los laboratorios de IBM, en los años 70’s, se trabajaba en el desarrollo de un lenguaje que se adaptara a las características del modelo relacional, produciendo un lenguaje llamado sequel.

Historia

En el año de 1986 se convierte en un estándar para la ANSI y en 1987, forma parte de la ISO.

Categorías

- **DDL (Data Definition Language): Sentencias que definen y crean objetos en la BD.**
- **DML (Data Manipulation Language): Sentencias para operar sobre los datos.**
- **DCL (Data Control Language): Sentencias orientadas hacia la administración de la BD.**

DDL

Ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Estado	Num_art	nom_art	cant	Precio
2301	23/02/11	101	Martin	Caracas	3786	Red	3	35,00
2301	23/02/11	101	Martin	Caracas	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Caracas	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Coro	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Maracay	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Maracay	3141	Funda	2	10,00

DDL

Tabla1(id_Orden , fecha, id_Cliente)

Tabla4(id_Cliente, nombre_Cliente, estado)

Tabla2 (id_orden, no_Articulo, cantidad)

Tabla3(no_Articulo, nombre_Articulo, precio)

DDL - Tipos de tablas

- Permanentes
- Temporales -> CREATE TEMPORARY TABLE
- Externas -> CREATE EXTERNAL TABLE

```
CREATE TABLE cliente (
id_cliente varchar(13) ,
nombre varchar(50) ,
ap_Pat varchar(50) ,
ap_Mat varchar(50),
estado varchar(25)
);
```

Ejercicio 6_1

Generar el código SQL de las relaciones resultantes en el ejercicio 5_3

DDL - Restricciones

- Check
- Not null
- Unique
- Primary keys
- Foreign keys

DDL - Restricciones

```
CREATE TABLE cliente (
id_cliente varchar(13) PRIMARY KEY,
nombre char(50) not null,
ap_Pat char(50) not null,
ap_Mat char(50) null,
estado varchar(25) not null
);
```

DDL - Restricciones

```
CREATE TABLE articulo (
num_Articulo int PRIMARY KEY,
nombre_Articulo varchar(30) not null,
precio numeric CHECK (precio > 0)
);
```

Nivel columna

DDL - Restricciones

```
CREATE TABLE articulo (
num_Articulo int PRIMARY KEY,
nombre_Articulo varchar(30) not null,
precio numeric CONSTRAINT verifica_Precio
CHECK (precio > 0) );
```

Nivel columna

DDL - Restricciones

```
CREATE TABLE articulo (
num_Articulo int,
nombre_Articulo varchar(30) not null,
precio numeric not null,
CONSTRAINT verifica_Precio CHECK (precio > 0),
CONSTRAINT articulo_PK PRIMARY
KEY(num_articulo,nombre_Articulo));
```

Nivel tabla

DDL - Restricciones

```
CREATE TABLE articulo (
num_Articulo int,
nombre_Articulo varchar(30) not null,
precio numeric CHECK (precio > 0),
CONSTRAINT articulo_PK PRIMARY
KEY(num_articulo, nombre_Articulo)
);
```

Nivel tabla

DDL - Restricciones

- No action
- Restrict
- Set null
- Cascade

Aplican a borrado y actualización

DDL - Restricciones

```
CREATE TABLE orden (
id_Orden int not null,
fecha date not null DEFAULT now(),
id_Cliente varchar(13),
CONSTRAINT orden_PK PRIMARY KEY(id_orden),
CONSTRAINT orden_cliente_FK FOREIGN KEY (id_Cliente)
REFERENCES cliente(id_Cliente) ON DELETE CASCADE
ON UPDATE RESTRICT);
```

DDL - Valores por defecto

```
CREATE TABLE cliente (
id_cliente varchar(13) PRIMARY KEY,
nombre char(50) not null,
ap_Pat char(50) not null,
ap_Mat char(50) null,
estado varchar(25) not null DEFAULT 'cdmx'
);
```

DDL - Valores secuenciales

```
CREATE SEQUENCE nombre_Sec [AS tipo_Dato]
[INCREMENT valor]
[MINVALUE valor]
[MAXVALUE valor]
[START valor]
[ [NO] CYCLE ];

SELECT nextval('nombre_Sec');
```

DDL - Valores secuenciales

```
CREATE SEQUENCE ejemplo
INCREMENT 1
MINVALUE 1
MAXVALUE 4
START 1
CYCLE;


SELECT nextval('ejemplo');
```

DDL - Índices

Estructura de datos que facilita el acceso a la información.

- **Clustered**
- **Non clustered**

DDL - Índices

```
CREATE TABLE student
(
    id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    gender VARCHAR(50) NOT NULL,
    DOB datetime NOT NULL,
    total_score INT NOT NULL,
    city VARCHAR(50) NOT NULL
 )
```

DDL - Índices

```
INSERT INTO student

VALUES
(6, 'Kate', 'Female', '03-JAN-1985', 500, 'Liverpool'),
(2, 'Jon', 'Male', '02-FEB-1974', 545, 'Manchester'),
(9, 'Wise', 'Male', '11-NOV-1987', 499, 'Manchester'),
(3, 'Sara', 'Female', '07-MAR-1988', 600, 'Leeds'),
(1, 'Jolly', 'Female', '12-JUN-1989', 500, 'London'),
(4, 'Laura', 'Female', '22-DEC-1981', 400, 'Liverpool'),
(7, 'Joseph', 'Male', '09-APR-1982', 643, 'London'),
(5, 'Alan', 'Male', '29-JUL-1993', 500, 'London'),
(8, 'Mice', 'Male', '16-AUG-1974', 543, 'Liverpool'),
(10, 'Elis', 'Female', '28-OCT-1990', 400, 'Leeds');
```

DDL - Índices

```
SELECT * FROM student
```

Los registros serán recuperados en el siguiente orden:

id	name	gender	DOB	total_score	city
1	Jolly	Female	1989-06-12 00:00:00.000	500	London
2	Jon	Male	1974-02-02 00:00:00.000	545	Manchester
3	Sara	Female	1988-03-07 00:00:00.000	600	Leeds
4	Laura	Female	1981-12-22 00:00:00.000	400	Liverpool
5	Alan	Male	1993-07-29 00:00:00.000	500	London
6	Kate	Female	1985-01-03 00:00:00.000	500	Liverpool
7	Joseph	Male	1982-04-09 00:00:00.000	643	London
8	Mice	Male	1974-08-16 00:00:00.000	543	Liverpool
9	Wise	Male	1987-11-11 00:00:00.000	499	Manchester
10	Elis	Female	1990-10-28 00:00:00.000	400	Leeds

DDL - Índices

```
CREATE NONCLUSTERED INDEX IX_tblStudent_Name
ON student(name ASC)
```

DDL - Índices

```
SELECT * FROM student
```

Los registros serán recuperados en el siguiente orden:

id	name	gender	DOB	total_score	city
1	Jolly	Female	1989-06-12 00:00:00.000	500	London
2	Jon	Male	1974-02-02 00:00:00.000	545	Manchester
3	Sara	Female	1988-03-07 00:00:00.000	600	Leeds
4	Laura	Female	1981-12-22 00:00:00.000	400	Liverpool
5	Alan	Male	1993-07-29 00:00:00.000	500	London
6	Kate	Female	1985-01-03 00:00:00.000	500	Liverpool
7	Joseph	Male	1982-04-09 00:00:00.000	643	London
8	Mice	Male	1974-08-16 00:00:00.000	543	Liverpool
9	Wise	Male	1987-11-11 00:00:00.000	499	Manchester
10	Elis	Female	1990-10-28 00:00:00.000	400	Leeds

DDL - Índices

name	Row Address
Alan	Row Address
Elis	Row Address
Jolly	Row Address
Jon	Row Address
Joseph	Row Address
Kate	Row Address
Laura	Row Address
Mice	Row Address
Sara	Row Address
Wise	Row Address

DDL - Índices

- Unique
- Non unique
- Compuestos
- Basados en funciones: CREATE INDEX first_name_idx ON user_data (UPPER(first_name));

DDL - Índices

CREATE [UNIQUE] INDEX nombre_Ind ON nombre_tabla [USING tipo] (columnas)

CLUSTERED nombre_Tab USING nombre_Ind;

DDL - Sinónimos

Los sinónimos proporcionan independencia de datos y transparencia de ubicación.

CREATE SYNONYM nombre_Sin
FOR origen

DDL - Sinónimos

```
CREATE SYNONYM employees_view
FOR employes_view;
```

DDL - Vistas

Es una especie de tabla virtual, ya que está compuesta por filas y columnas, y puede estar formada por toda la información de una(s) tabla(s) o parte de ella(s).

DDL - Vistas

usuario	contrasenia	fecha_registro	contrasenia_previa	activo
profesor1	prof1	2022-02-01		T
profesor2	prof2	2022-02-02	1234	T
profesor3	prof3	2022-01-30	3540	F
profesor4	prof2	2022-02-02		T

DDL - Vistas

Vista donde sólo mostramos algunas columnas y las observaciones con activo = T

usuario	fecha_registro	activo
profesor1	2022-02-01	T
profesor2	2022-02-02	T
profesor4	2022-02-02	T

DDL - Vistas

CREATE [OR REPLACE] [TEMP | TEMPORARY] VIEW nombre_Vist [nombres_Columnas] AS consulta;

DDL - Mod. estructura

- Alter: Permite realizar diversas modificaciones a un objeto de la base de datos, por ejemplo, agregar/editar/eliminar una columna a una tabla, editar/agregar/eliminar constraints, modificar almacenamiento, etc.

DDL - Mod. estructura

- **Drop: Permite eliminar objetos en la base de datos.**

DROP TIPO_OBJETO nombre_Objeto;

DDL - Mod. estructura

ALTER TABLE nombre_Tabla ACCION

Tarea 12

Investigar:

- **Niveles de aislamiento en bases de datos relaciones**
- **Propiedades ACID**