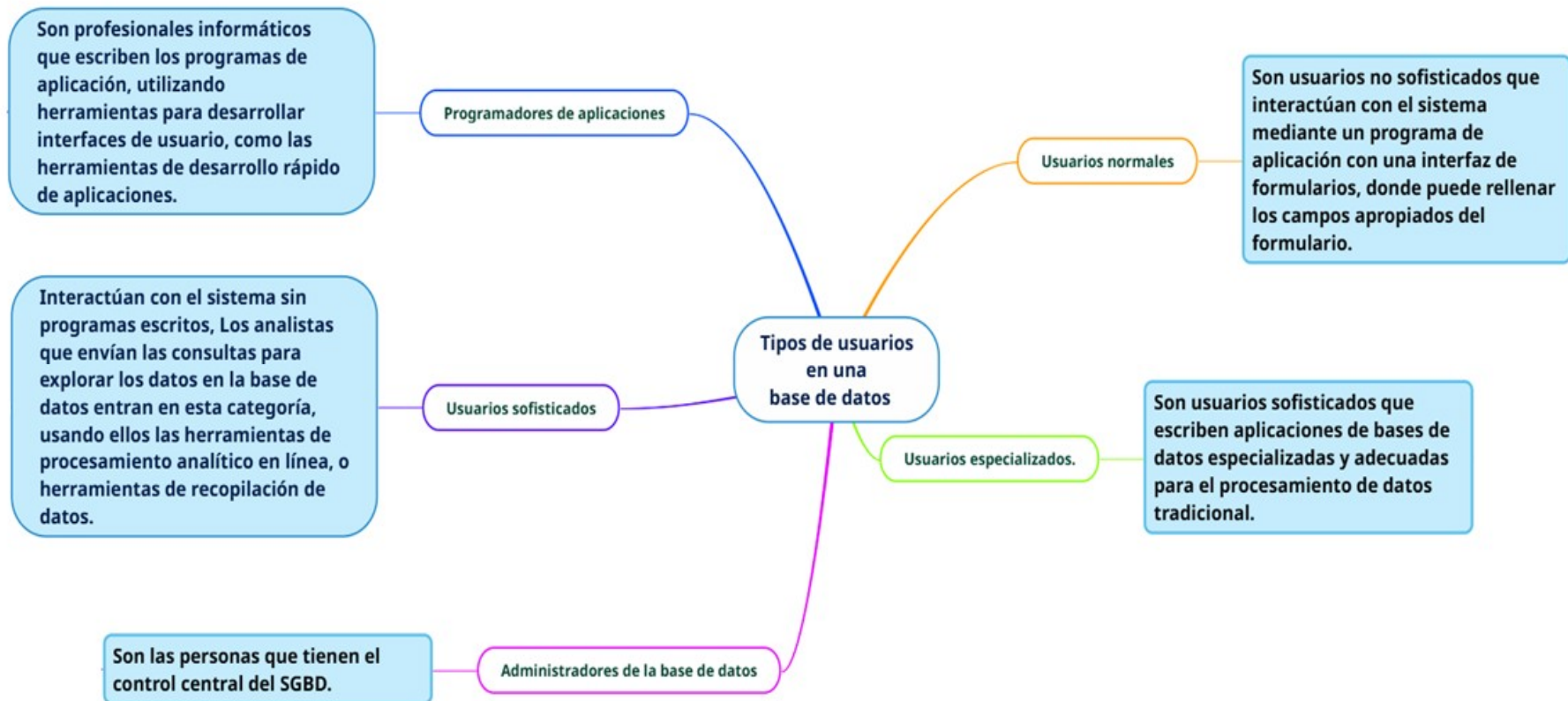


Base de Datos I

I. CONCEPTOS SOBRE BASE DE DATOS



1.4. Conceptos básicos - Usuarios de una BD



1.4. Conceptos básicos - Usuarios de una BD

Administradores de la base de datos (ABD)

Son las personas que tienen el control central del SGBD.

Entre las funciones del ABD se encuentran:

- Definición del esquema de la base de datos.
- Definición de la estructura y el método de acceso.
- Modificación del esquema y la organización física.
- Concesión de autorización para el acceso a los datos.
- Mantenimiento rutinario.

System Administrator y Propietario

- **sa:** (System Administrator) Tiene todos los permisos en el servidor SQL, incluyendo la capacidad de administrar el servidor completo.
- **Owner:** (Propietario) Tiene todos los permisos en una base de datos específica, pero no en el servidor completo.

1.4. Conceptos básicos - Usuarios de una BD

Definición de Usuarios, Roles y Permisos en BD.

- **Usuario:** Entidad que puede iniciar sesión y realizar acciones en la base de datos.
- **Rol:** Conjunto de permisos que pueden ser asignados a los usuarios.
- **Permiso:** Autorización para realizar una operación específica en un objeto de la base de datos.

Definición de Administrador y Propietario

ALTER LOGIN sa ENABLE;

ALTER LOGIN sa WITH PASSWORD = 'new_password';



- ALTER LOGIN sa ENABLE;; Habilita la cuenta de administrador "sa".
- ALTER LOGIN sa WITH PASSWORD = 'new_password';: Cambia la contraseña del usuario "sa" para mayor seguridad.

1.4. Conceptos básicos - Usuarios de una BD

Definición de Propietario

```
CREATE LOGIN Pedro WITH PASSWORD = 'password';
```

```
CREATE USER Pedro FOR LOGIN Pedro;
```

```
ALTER AUTHORIZATION ON DATABASE::BD_CONTROL TO Pedro;
```



- CREATE LOGIN Pedro WITH PASSWORD = 'password'; Crea un nuevo inicio de sesión llamado Pedro.
- CREATE USER Pedro FOR LOGIN Pedro; Crea un usuario en la base de datos asociado al inicio de sesión Pedro.
- ALTER AUTHORIZATION ON DATABASE::BD_CONTROL TO Pedro; Asigna la propiedad de la base de datos BD_CONTROL al usuario Pedro.

1.4. Conceptos básicos - Usuarios de una BD

Definición de Usuarios comunes

```
CREATE LOGIN Juan WITH PASSWORD = 'password';
```

```
CREATE USER Juan FOR LOGIN Juan;
```



- CREATE LOGIN Juan WITH PASSWORD = 'password';: Crea un inicio de sesión llamado Juan.
- CREATE USER Juan FOR LOGIN Juan;: Crea un usuario en la base de datos asociado al inicio de sesión Juan.

```
CREATE LOGIN Maria WITH PASSWORD = 'password';
```

```
CREATE USER Maria FOR LOGIN Maria;
```



- CREATE LOGIN Maria WITH PASSWORD = 'password';: Crea un inicio de sesión llamado María.
- CREATE USER Maria FOR LOGIN Maria;: Crea un usuario en la base de datos asociado al inicio de sesión María.

1.4. Conceptos básicos - Usuarios de una BD

Asignación de Roles y Permisos

Recordemos que los roles son conjuntos de permisos predefinidos que pueden asignarse a los usuarios.

```
ALTER ROLE db_datareader ADD MEMBER Juan;
```

```
ALTER ROLE db_datawriter ADD MEMBER Maria;
```



- `ALTER ROLE db_datareader ADD MEMBER Juan;`: Agrega a Juan al rol `db_datareader`, permitiéndole leer datos de todas las tablas.
- `ALTER ROLE db_datawriter ADD MEMBER Maria;`: Agrega a María al rol `db_datawriter`, permitiéndole escribir datos en todas las tablas..

1.4. Conceptos básicos - Usuarios de una BD

Asignación de Permisos a Objetos

Recordemos que es posible asignar permisos específicos a los usuarios para realizar operaciones en objetos específicos como tablas

```
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.CLIENTES TO Juan;
```

```
GRANT SELECT ON dbo.PRODUCTOS TO Maria;
```



- GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.CLIENTES TO Juan; Permite a Juan seleccionar, insertar, actualizar y eliminar datos en la tabla CLIENTES.
- GRANT SELECT ON dbo.PRODUCTOS TO Maria; Permite a María solo seleccionar (leer) datos en la tabla PRODUCTOS.

1.4. Conceptos básicos - Usuarios de una BD

Prueba de Permisos asignados a Objetos

Recordemos que para probar los permisos, los usuarios deben iniciar sesión y realizar las operaciones permitidas en las tablas correspondientes como tablas

```
INSERT INTO dbo.CLIENTES (Nombre, Dirección) VALUES ('Cliente1', 'Dirección1');
```

```
SELECT * FROM dbo.CLIENTES;
```

```
SELECT * FROM dbo.PRODUCTOS;
```



- Juan intenta insertar un nuevo cliente en la tabla CLIENTES.
- María intenta seleccionar productos de la tabla PRODUCTOS.

1.4. Conceptos básicos - Usuarios de una BD

Usuario	Rol	Permiso	Objeto	Ejemplo de Comando SQL
Ronald	sa	Todos	Todos	SELECT * FROM dbo.CLIENTES; ALTER DATABASE BD_CONTROL MODIFY NAME = BD_CONTROL2;
Pedro	Owner	Todos en BD_CONTROL	Todos en BD_CONTROL	SELECT * FROM dbo.PRODUCTOS; ALTER TABLE dbo.CLIENTES ADD Email VARCHAR(255);
Juan	db_datareader	SELECT	CLIENTES	SELECT * FROM dbo.CLIENTES;
Juan	db_datawriter	INSERT, UPDATE, DELETE	CLIENTES	INSERT INTO dbo.CLIENTES (Nombre, Dirección) VALUES ('Cliente1', 'Dirección1');
Juan	GRANT ALTER ON dbo.CLIENTES TO Juan;	ALTER	CLIENTES	ALTER TABLE dbo.CLIENTES ADD Email VARCHAR(255);
Juan	GRANT EXECUTE ON dbo.spActualizarCliente TO Juan;	EXECUTE	Procedimientos Almacenados	EXEC spActualizarCliente 'Cliente1', 'Dirección2';
María	db_datareader	SELECT	PRODUCTOS	SELECT * FROM dbo.PRODUCTOS;
María	db_datawriter	INSERT, UPDATE, DELETE	PRODUCTOS	INSERT INTO dbo.PRODUCTOS (Nombre, Precio) VALUES ('Producto1', 100);
María	GRANT EXECUTE ON dbo.fnCalcularPrecioConDescuento TO Maria;	EXECUTE	Funciones	SELECT dbo.fnCalcularPrecioConDescuento(100, 10);
María	GRANT ALTER ON dbo.PRODUCTOS TO Maria;	ALTER	PRODUCTOS	ALTER TABLE dbo.PRODUCTOS ADD Descripción VARCHAR(255);

1.5. Ambiente de base de datos - Entidades

Una entidad en una base de datos representa un objeto o concepto del mundo real sobre el cual se desea almacenar información.

Ejemplo: Un cliente en un sistema de ventas.

Entidad Cliente:

Atributos: ID, Nombre, Dirección, Teléfono.

Representación:

ID: 1

Nombre: Juan Pérez

Dirección: Calle Falsa 123

Teléfono: 555-1234

Una tabla es una estructura de datos que organiza información en filas y columnas. Cada fila representa una entidad y cada columna representa un atributo de esa entidad.

Ejemplo: Tabla CLIENTES



ID	Nombre	Dirección	Teléfono
1	Juan Pérez	Calle Falsa 123	555-1234
2	María López	Av. lindavista	555-5678

1.5. Ambiente de base de datos - Consultas

Una consulta es una solicitud de información específica de una base de datos. Se utiliza para obtener datos según ciertos criterios.

Ejemplo: Obtener todos los clientes con el nombre "Juan".

```
SELECT * FROM CLIENTES WHERE Nombre = 'Juan Pérez';
```

ID	Nombre	Dirección	Teléfono
1	Juan Pérez	Calle Falsa 123	555-1234

Una vista es una consulta almacenada que presenta datos de una o más tablas de una manera específica. Es como una "tabla virtual".

Ejemplo: Vista de clientes con su dirección.



```
CREATE VIEW VistaClientes AS  
SELECT Nombre, Dirección FROM CLIENTES;
```

```
SELECT * FROM VistaClientes;
```

1.5. Ambiente de Base de Datos - Instrucción

Una instrucción en SQL es una operación que realiza una acción específica sobre la base de datos. Incluye comandos como SELECT, INSERT, UPDATE y DELETE.

Ejemplo: Insertar un nuevo cliente en la tabla CLIENTES.

```
INSERT INTO CLIENTES (ID, Nombre, Dirección, Teléfono)  
VALUES (3, 'Carlos Méndez', 'Av. Principal', '555-6789');
```

Una clausula de condición : esto sirve para traer información especifica en base a una condición usando la clausula where.

Ejemplo: si queremos condicionar la información de CLIENTES.

```
SELECT * FROM CLIENTES WHERE Nombre "Carlos"
```



1.5. Ambiente de Base de Datos - Módulos

Un procedimiento almacenado es un conjunto de instrucciones SQL que pueden ser ejecutadas como una sola unidad. Se utiliza para encapsular lógica de negocio.

Ejemplo: Procedimiento para actualizar un cliente.

```
CREATE PROCEDURE spActualizarCliente
@ID INT,
@Nombre VARCHAR(100),
@Dirección VARCHAR(255),
@Teléfono VARCHAR(15)
AS
BEGIN
    UPDATE CLIENTES
    SET Nombre = @Nombre, Dirección = @Dirección,
    Teléfono = @Teléfono WHERE ID = @ID;END;
```



```
EXEC spActualizarCliente 1, 'Juan Pérez', 'Calle Nueva 456', '555-4321';
```

Un módulo en SQL Server puede referirse a una colección de procedimientos y funciones almacenados que realizan tareas relacionadas. Agrupa lógica de negocio en una base de datos.

Ejemplo: Módulo de Gestión de Clientes.

- **Procedimientos:** spAgregarCliente, spActualizarCliente, spEliminarCliente.
- **Funciones:** fnObtenerCliente, fnCalcularEdadCliente;

1.5. Ambiente de Base de Datos - Relaciones

Relaciones

El modo de relacionar registros entre tablas es por tanto mediante referencias, para lo cual se usan los identificadores definidos como claves primarias y foráneas.

Clave primaria

En el diseño de bases de datos relacionales, se llama clave primaria (Primary Key) a un campo o a una combinación de campos que identifica de forma única a cada fila de una tabla. Una clave primaria comprende de esta manera una columna o conjunto de columnas. No puede haber dos filas en una tabla que tengan la misma clave primaria.

Ejemplos son DNI (asociado a una persona) o ISBN (asociado a un libro). Sin embargo las guías telefónicas y diccionarios no pueden usar nombres o palabras o números del sistema decimal como claves candidatas, porque no identifican unívocamente números de teléfono o palabras.

1.5. Ambiente de Base de Datos - Relaciones

Clave foránea

En el contexto de Bases de Datos relacionales, una clave foránea o clave ajena o Foreign Key (FK) es una limitación referencial entre dos tablas.

Identifica una columna o grupo de columnas en una tabla (tabla hija o referenciada) que se refiere a una columna o grupo de columnas en otra tabla referendo o padre. Las columnas en la tabla referendo deben ser la clave primaria u otra clave candidata en la tabla referenciada.

Los valores en una fila de las columnas referendo deben existir solo en una fila en la tabla referenciada, así, una fila en la tabla referendo no puede contener valores que no existen en la tabla referenciada.

De esta forma, las referencias pueden ser creadas para vincular o relacionar información. Esto es una parte esencial de la normalización de base de datos.