



# **SQL SERVER IMPLEMENTACIÓN**

**Eric Gustavo Coronel Castillo**

[www.desarrollasoftware.com](http://www.desarrollasoftware.com)

[gcoronelc@gmail.com](mailto:gcoronelc@gmail.com)



**CORONEL**  
DESARROLLA SOFTWARE

# MICROSOFT SQL SERVER

## NIVEL I - IMPLEMENTACIÓN

### Mantenimiento de Datos

**Eric Gustavo Coronel Castillo**

[www.desarrollasoftware.com](http://www.desarrollasoftware.com)

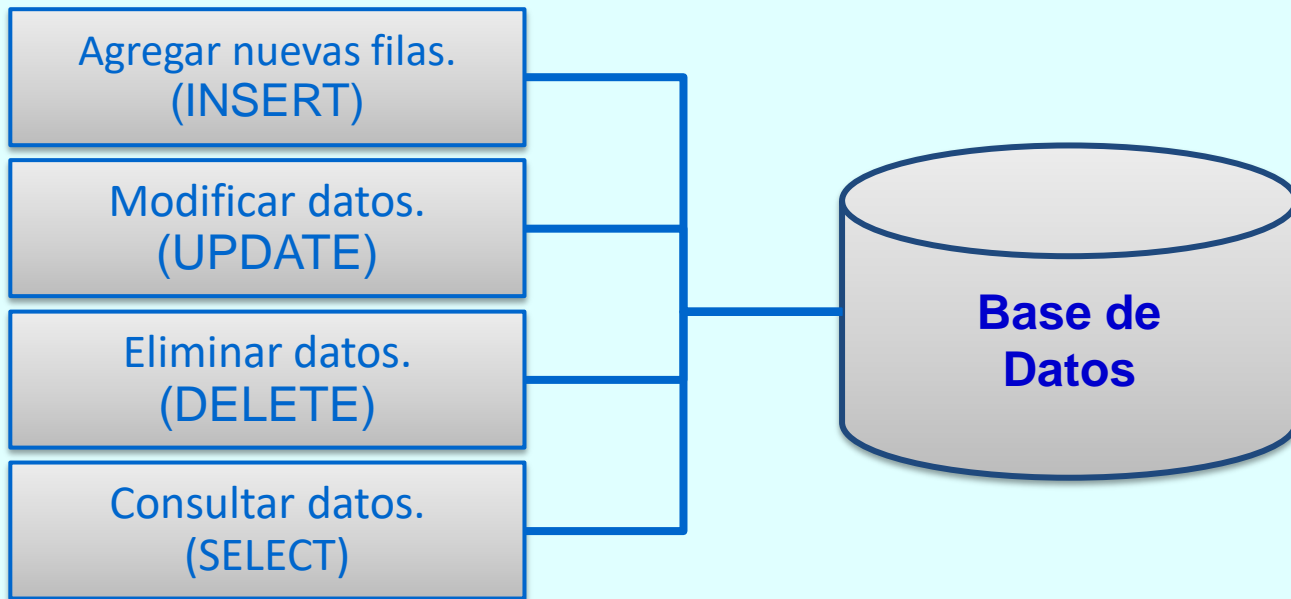
[gcoronelc@gmail.com](mailto:gcoronelc@gmail.com)



# LOGRO ESPERADO

Se espera que al finalizar la presente clase el participante se encuentre capacitado para hacer mantenimiento de datos.





# Instrucción SELECT

- **SINTAXIS**

```
SELECT columnas  
FROM nombre_tabla  
WHERE condición
```

- **EJEMPLO**

```
USE RH  
GO
```

```
SELECT * FROM dbo.empleado  
WHERE idcargo = 'C02';  
GO
```

# Instrucción INSERT

- **SINTAXIS**

```
INSERT INTO nombre_tabla [ ( columnas ) ]  
VALUES ( valores ) | Instrucción_SELECT
```

- **EJEMPLO**

```
USE RH  
GO
```

```
INSERT INTO dbo.cargo VALUES('C15','Jefe',7500.0,9000.0)  
GO
```

```
SELECT * FROM dbo.cargo  
GO
```

# Instrucción TRUNCATE TABLE

## SINTAXIS

```
TRUNCATE TABLE nombre_tabla
```

## EJEMPLO

```
USE RH  
GO
```

```
SELECT * INTO dbo.emp_c02  
FROM dbo.empleado WHERE idcargo = 'C02'  
GO
```

```
SELECT * FROM dbo.emp_c02  
GO
```

```
TRUNCATE TABLE dbo.emp_c02  
GO
```

```
SELECT * FROM dbo.emp_c02  
GO
```

# Instrucción DELETE

- **SINTAXIS**

```
DELETE FROM nombre_tabla  
WHERE condición
```

- **EJEMPLO**

```
USE RH  
GO
```

```
DELETE FROM dbo.cargo  
WHERE idcargo = 'C15'  
GO
```



# Instrucción UPDATE

- **SINTAXIS**

```
UPDATE nombre_tabla  
SET < columna > = < valor > [, ... ]  
WHERE < condición >
```

- **EJEMPLO**

```
USE RH  
GO
```

```
UPDATE dbo.cargo SET sueldo_min = 7000.0  
WHERE idcargo = 'C15'  
GO
```

# Casos de Integridad Referencial

- No se puede modificar una valor de clave primaria si tiene registros relacionados en una tabla hija.

```
USE RH  
GO
```

```
UPDATE dbo.cargo  
SET idcargo = 'C99'  
WHERE idcargo = 'C02'  
GO
```

**Msg 547, Level 16, State 0, Line 1**

**The UPDATE statement conflicted with the REFERENCE constraint "fk\_empleado\_cargo". The conflict occurred in database "RH", table "dbo.empleado", column 'idcargo'.**

**The statement has been terminated.**

## Casos de Integridad Referencial

- No se puede eliminar una fila cuya clave primaria tiene registros relacionados en una tabla hija.

```
USE RH  
GO
```

```
DELETE FROM dbo.cargo  
WHERE idcargo = 'C02'  
GO
```

**Msg 547, Level 16, State 0, Line 1**

**The DELETE statement conflicted with the REFERENCE constraint "fk\_empleado\_cargo". The conflict occurred in database "RH", table "dbo.empleado", column 'idcargo'.**

**The statement has been terminated.**

# Casos de Integridad Referencial

- No se puede insertar una valor de clave foránea que no existe como clave primaria en la tabla padre.

```
USE RH  
GO
```

```
SET DATEFORMAT DMY
```

```
INSERT INTO dbo.empleado ( idempleado, apellido, nombre, fecingreso, idcargo,  
iddepartamento, sueldo ) VALUES ( 'E8888', 'Ramos', 'Enrique', '30-05-2011', 'C77', 104,  
4000.0 )  
GO
```

**Msg 547, Level 16, State 0, Line 2**

**The INSERT statement conflicted with the FOREIGN KEY constraint "fk\_empleado\_cargo".  
The conflict occurred in database "RH", table "dbo.cargo", column 'idcargo'.**

The statement has been terminated.

# Transacciones

- Una transacción es una unidad única de trabajo y está conformada por una ó varias instrucciones que modifican el estado de la base de datos.
- Si una transacción tiene éxito, todas las modificaciones de los datos realizadas durante la transacción se confirman y se convierten en una parte permanente de la base de datos.
- Si una transacción encuentra errores y debe cancelarse o revertirse, se borran todas las modificaciones de los datos.



```
BEGIN TRANSACTION
INSERT INTO ...
INSERT INTO ...
UPDATE ...
UPDATE ...
DELETE ..
DELETE ..
COMMIT TRANSACTION
GO
```

## TRANSACCIONES DE CONFIRMACIÓN AUTOMÁTICA

- El modo de confirmación automática es el modo de administración de transacciones predeterminado de SQL Server.
- Cada instrucción se confirma o se revierte cuando finaliza. Si una instrucción termina correctamente, se confirma; si encuentra un error, se revierte.

- **Ejemplo**

```
UPDATE dbo.empleado  
SET sueldo = 26000.0  
WHERE idempleado = 'E0001'  
GO
```

## TRANSACCIONES EXPLICITAS

Una transacción explícita es aquella en que se define explícitamente el inicio y el final de la transacción.

- **BEGIN TRANSACTION**

Marca el punto de inicio de una transacción explicit.

- **COMMIT TRANSACTION o COMMIT WORK**

Se utiliza para finalizar una transacción correctamente si no hubo errores. Todas las modificaciones de datos realizadas en la transacción se convierten en parte permanente de la base de datos. Se liberan los recursos ocupados por la transacción.

- **ROLLBACK TRANSACTION o ROLLBACK WORK**

Se utiliza para eliminar una transacción en la que se encontraron errores. Todos los datos modificados por la transacción vuelven al estado en el que estaban al inicio de la transacción. Se liberan los recursos ocupados por la transacción.

GRACIAS  
**TOTALES**





**Gustavo Coronel**



[gcoronelc.github.io](https://gcoronelc.github.io)

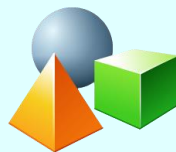
[youtube.com/DesarrollaSoftware](https://youtube.com/DesarrollaSoftware)

[facebook.com/groups/desarrollasoftware](https://facebook.com/groups/desarrollasoftware)



## **FUNDAMENTOS DE PROGRAMACIÓN CON JAVA**

Inicia tu aprendizaje, utilizando las mejores prácticas de programación



## **CURSO PROFESIONAL DE JAVA ORIENTADA A OBJETOS**

Aprende programación en capas, patrones y buenas prácticas



## **PROGRAMACIÓN DE BASE DE DATOS ORACLE CON PL/SQL**

Aprende a obtener el mejor rendimiento de tú base de datos

**JDBC**



## **PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JDBC**

Aprende a programar correctamente con JDBC