TALLER INVESTIGACION

JERICOTH JOSE CASTELLAR BOLAÑO

FUNDACION UNIVERSITARIA COLOMBO INTERNACIONAL.

TECNOLOGIA EN DESARROLLO DE SOFTWARE

SEMESTRE V

CARTAGENA DE INDIAS DT y C.

2025.

Contenido

[CHECK 3](#_Toc195393299)

[SUBCONSULTAS 4](#_Toc195393300)

[JOIN 5](#_Toc195393301)

[INTEGRIDAD REFERENCIAL 6](#_Toc195393302)

CHECK

un CHECK es una restricción (constraint) que se usa para limitar los valores que se pueden insertar en una columna. Sirve para garantizar que los datos cumplen ciertas condiciones antes de ser aceptados en la base de datos.

Un ejemplo seria este código para una tabla de empleados.

CODIGO:

CREATE TABLE empleados (

id INT PRIMARY KEY,

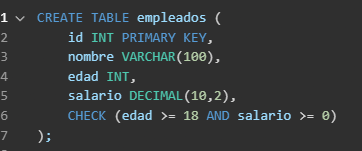
nombre VARCHAR(100),

edad INT,

salario DECIMAL(10,2),

CHECK (edad >= 18 AND salario >= 0)

);



¿Qué hace esto?

Evita que se inserten empleados con edad menor a 18.

Tambien es posible agregar un check en una tabla ya existente.

CODIGO:

ALTER TABLE empleados

ADD CONSTRAINT chk\_edad CHECK (edad >= 18);



SUBCONSULTAS

Una subconsulta (también llamada consulta anidada) es una consulta dentro de otra consulta. Se usa cuando necesitas que un resultado intermedio alimente a la consulta principal.

EJEMPLOS:

**WHERE**

CODIGO:

SELECT nombre

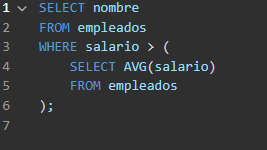
FROM empleados

WHERE salario > (

SELECT AVG(salario)

FROM empleados

);



Hay varios tipos de subconsultas posibles.

Del tipo Escalar (scalar) que se usan en SELECT, WHERE y son un solo valor

Tambien las de columna o tabla las cuales serian IN, EXIST, FROM que son de valores únicos.

JOIN

se usa para combinar filas de dos o más tablas basadas en una relación entre columnas (normalmente una clave primaria y una clave foránea).

INNER JOIN: Devuelve solo las coincidencias entre ambas tablas.

LEFT JOIN: Devuelve todas las filas de la tabla izquierda, y las coincidentes de la derecha (si hay).

RIGHT JOIN: Igual al LEFT, pero empieza por la tabla de la derecha.

FULL OUTER JOIN: Devuelve todas las filas de ambas tablas, coincidan o no.

CROSS JOIN: Devuelve el producto cartesiano (todas las combinaciones posibles).

INTEGRIDAD REFERENCIAL

La integridad referencial garantiza que las relaciones entre tablas sean válidas. Es decir:

Si una tabla tiene una clave foránea el valor que pone allí debe existir como clave primaria (primary key) en otra tabla.

Ejemplo:

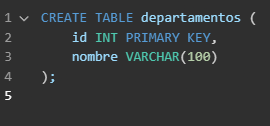
CODIGO:

CREATE TABLE departamentos (

id INT PRIMARY KEY,

nombre VARCHAR(100)

);



CREATE TABLE empleados (

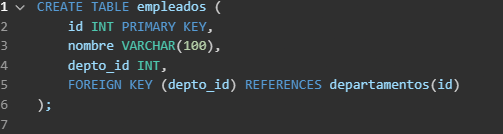
id INT PRIMARY KEY,

nombre VARCHAR(100),

depto\_id INT,

FOREIGN KEY (depto\_id) REFERENCES departamentos(id)

);



¿Qué nos permite esto?

No poder insertar un empleado con un depto\_id que no exista en departamentos.

No poder eliminar un departamento si hay empleados que lo usan (a menos que lo configures para que sí se pueda, con acciones como ON DELETE).