

# Unidad 2: Funciones y Procedimientos (Parte 3)

Dr. Rogelio Mamani Ramos, Ph.D.

Abril, 2025



## 2.2.7. Operadores

- Dentro de las funciones en PostgreSQL, se utilizan una amplia gama de operadores para realizar diversas operaciones con datos.
- los operadores son similares a los que se utilizan en las consultas SQL estándar, pero también hay algunos específicos para el entorno procedural.

Base de  
Datos II

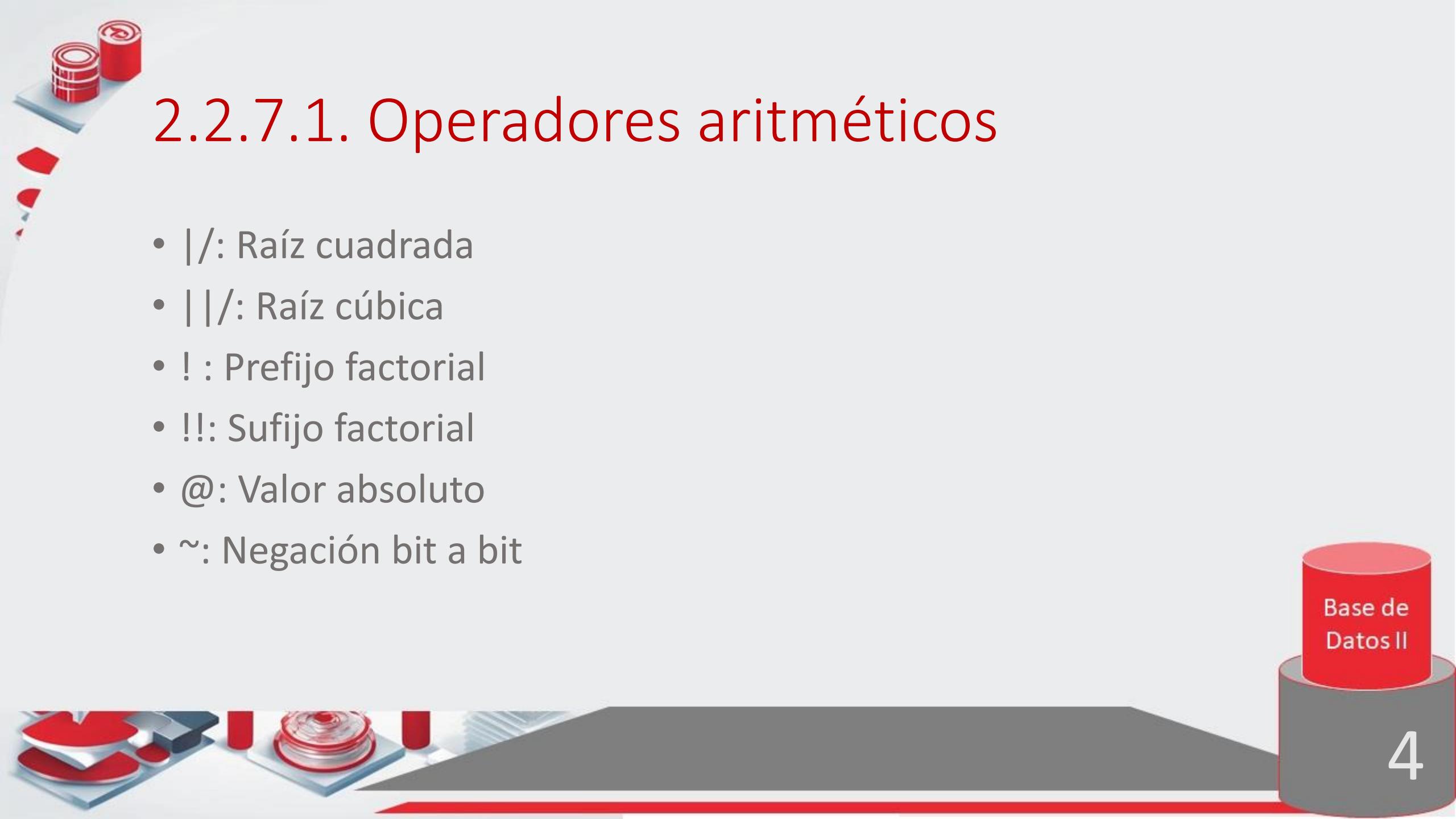
2



## 2.2.7.1. Operadores aritméticos

- +: Adición
- -: Sustracción
- \*: Multiplicación
- /: División
- %: Módulo (resto de la división entera)
- ^: Exponenciación





## 2.2.7.1. Operadores aritméticos

- |/: Raíz cuadrada
- ||/: Raíz cúbica
- ! : Prefijo factorial
- !!: Sufijo factorial
- @: Valor absoluto
- ~: Negación bit a bit



## 2.2.7.1. Operadores aritméticos

- &: AND bit a bit
- |: OR bit a bit
- #: XOR bit a bit
- <<: Desplazamiento a la izquierda bit a bit
- >>: Desplazamiento a la derecha bit a bit

Base de  
Datos II



## 2.2.7.2. Operadores de comparación

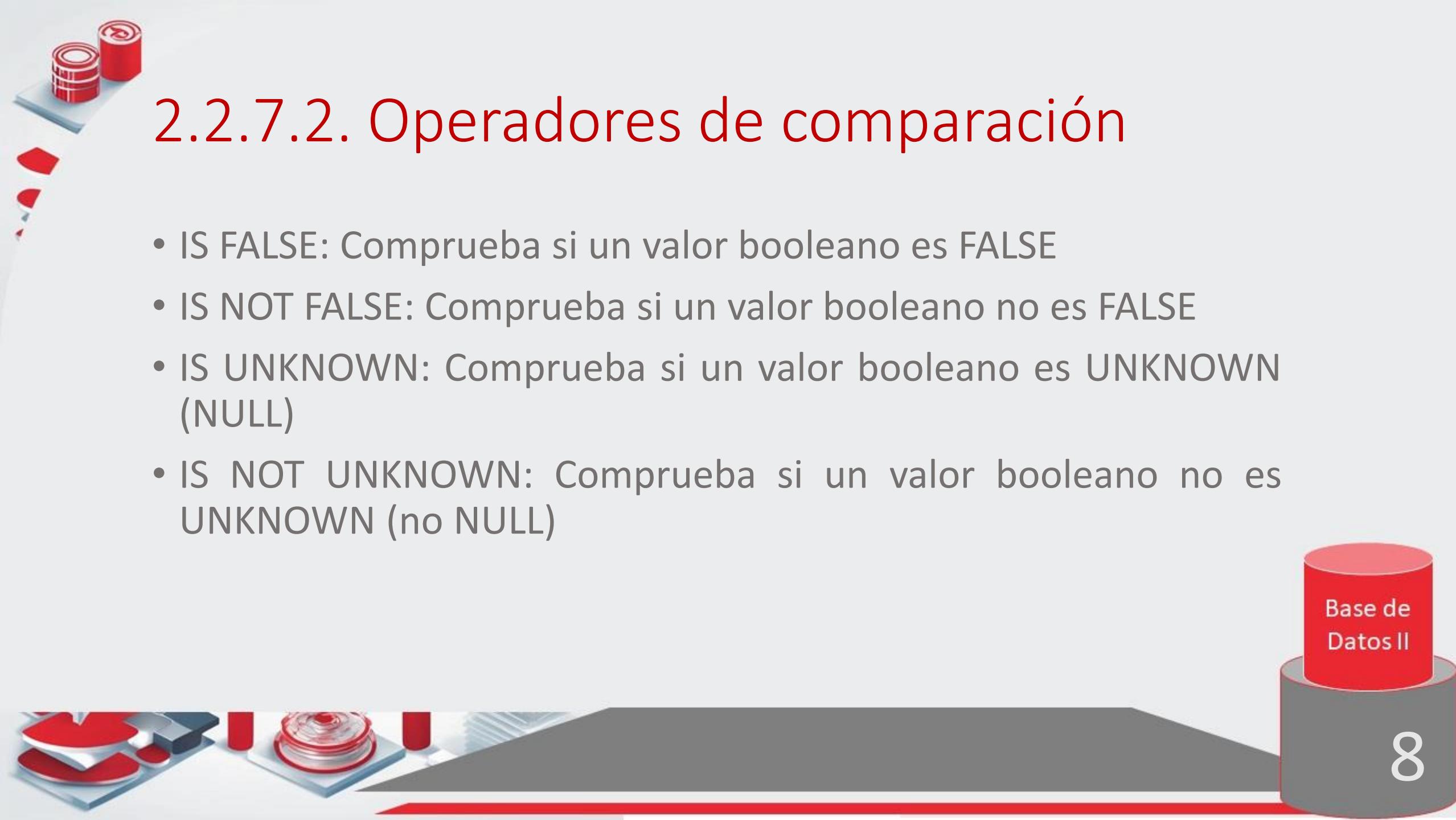
- $=$ : Igual a
- $>$ : Mayor que
- $<$ : Menor que
- $\geq$ : Mayor o igual que
- $\leq$ : Menor o igual que

Base de  
Datos II



## 2.2.7.2. Operadores de comparación

- <> o !=: No igual a
- IS NULL: Comprueba si un valor es NULL
- IS NOT NULL: Comprueba si un valor no es NULL
- IS TRUE: Comprueba si un valor booleano es TRUE
- IS NOT TRUE: Comprueba si un valor booleano no es TRUE



## 2.2.7.2. Operadores de comparación

- IS FALSE: Comprueba si un valor booleano es FALSE
- IS NOT FALSE: Comprueba si un valor booleano no es FALSE
- IS UNKNOWN: Comprueba si un valor booleano es UNKNOWN (NULL)
- IS NOT UNKNOWN: Comprueba si un valor booleano no es UNKNOWN (no NULL)

Base de  
Datos II



## 2.2.7.3. Operadores lógicos

- AND: Y lógico
- OR: O lógico
- NOT: NO lógico

Base de  
Datos II



## 2.2.7.4. Operadores de cadenas de texto

- ||: Concatenación de cadenas
- LIKE: Coincidencia de patrones (con comodines % y \_)
- NOT LIKE: No coincidencia de patrones
- ~: Coincidencia de expresiones regulares (sensible a mayúsculas)



Base de  
Datos II

10



## 2.2.7.4. Operadores de cadenas de texto

- $\sim^*$ : Coincidencia de expresiones regulares (insensible a mayúsculas)
- $!\sim$ : No coincidencia de expresiones regulares (sensible a mayúsculas)
- $!\sim^*$ : No coincidencia de expresiones regulares (insensible a mayúsculas)



Base de  
Datos II



## 2.2.7.5. Operadores a tipos de datos específicos

- Operadores de Fechas y Horas: + (adición de intervalos), - (sustracción de intervalos o fechas), <, >, etc.
- Operadores de Arreglos: || (concatenación), @> (contiene), <@ (está contenido por), =, <>, etc.
- Operadores JSON/JSONB: ->, ->>, #>, #>>, @>, <, ?, ?|, ?&, etc.



Base de  
Datos II



## 2.2.7.5. Operadores a tipos de datos específicos

- Operadores de Rangos: @>, <@, && (se superponen), -|- (adyacente), <, >, <=, >=, =, <>, etc.
- Operadores de Redes (inet, cidr, macaddr): & (AND bit a bit), | (OR bit a bit), << (está contenido por), >> (contiene), =, <>, etc.
- Operadores Geométricos (PostGIS): && (intersección de bounding boxes), @ (contiene), ~ (contenido por), \* (interseca), etc. (si la extensión PostGIS está instalada).
- ::: Operador de conversión de tipo (CAST)



Base de  
Datos II



## 2.2.7.6. Ejemplos

- Verifica si la edad corresponde a mayor de edad

```
CREATE OR REPLACE FUNCTION mayor_de_edad(edad INTEGER) RETURNS BOOLEAN AS
$$
BEGIN
    RETURN edad >= 18;
END;
$$
LANGUAGE plpgsql;

select mayor_de_edad(9)
```





## 2.2.7.6. Ejemplos

- Compare valores booleanos

```
CREATE OR REPLACE FUNCTION verificar_cond(x BOOLEAN, y BOOLEAN)
RETURNS BOOLEAN AS
$$
BEGIN
    RETURN x AND NOT y;
END;
$$
LANGUAGE plpgsql;
select verificar_cond(true,false);
```

Base de  
Datos II



## 2.2.7.6. Ejemplos

- Concatene valores

```
CREATE OR REPLACE FUNCTION concatena(x TEXT)
RETURNS TEXT AS
$$
BEGIN
    RETURN 'bienvenido ' || x || '....';
END;
$$
LANGUAGE plpgsql;
select concatena('juan');
```

Base de  
Datos II



## 2.2.7.6. Ejemplos

- Concatene valores

```
CREATE OR REPLACE FUNCTION concatena(x TEXT)
RETURNS TEXT AS
$$
BEGIN
    RETURN 'bienvenido ' || x || '....';
END;
$$
LANGUAGE plpgsql;
select concatena('juan');
```

Base de  
Datos II