## ✅ Table Creation (DDL)

CREATE TABLE categories (

category\_id SERIAL PRIMARY KEY,

category\_name VARCHAR(50) NOT NULL UNIQUE,

description TEXT

);

CREATE TABLE suppliers (

supplier\_id SERIAL PRIMARY KEY,

supplier\_name VARCHAR(100) NOT NULL,

contact\_email VARCHAR(100),

phone VARCHAR(20)

);

CREATE TABLE products (

product\_id SERIAL PRIMARY KEY,

product\_name VARCHAR(100) NOT NULL,

category\_id INT REFERENCES categories(category\_id),

supplier\_id INT REFERENCES suppliers(supplier\_id),

price NUMERIC(10, 2) CHECK (price >= 0),

stock\_quantity INT CHECK (stock\_quantity >= 0),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**✅ DML Queries**

**(a)** List all products with category and supplier names:

SELECT p.product\_name, c.category\_name, s.supplier\_name

FROM products p

JOIN categories c ON p.category\_id = c.category\_id

JOIN suppliers s ON p.supplier\_id = s.supplier\_id;

**(b)** Products with stock below 5:

SELECT product\_name, stock\_quantity

FROM products

WHERE stock\_quantity < 5;

**(c)** Add new column discount\_percent:

ALTER TABLE products

ADD COLUMN discount\_percent NUMERIC(5,2) DEFAULT 0;

**(d)** Reduce price of Electronics products by 15%:

UPDATE products

SET price = price \* 0.85

WHERE category\_id = (

SELECT category\_id FROM categories WHERE category\_name = 'Electronics'

);

**✅ Aggregates, Filtering, Grouping, Sorting**

**(e)** Total number of products:

SELECT COUNT(\*) AS total\_products FROM products;

**(f)** Average price:

SELECT AVG(price) AS avg\_price FROM products;

**(g)** Max and Min price of Electronics:

SELECT MAX(price) AS max\_price, MIN(price) AS min\_price

FROM products

WHERE category\_id = (SELECT category\_id FROM categories WHERE category\_name = 'Electronics');

**(h)** Categories with count of products:

SELECT c.category\_name, COUNT(p.product\_id) AS product\_count

FROM categories c

LEFT JOIN products p ON c.category\_id = p.category\_id

GROUP BY c.category\_name;

**(i)** Suppliers who supply products priced $50–$200:

SELECT DISTINCT s.supplier\_name

FROM suppliers s

JOIN products p ON s.supplier\_id = p.supplier\_id

WHERE p.price BETWEEN 50 AND 200;

**(j)** Products whose category\_id is in (1, 3):

SELECT product\_name

FROM products

WHERE category\_id IN (1, 3);

**(k)** Total stock per category (only >1 product):

SELECT c.category\_name, SUM(p.stock\_quantity) AS total\_stock

FROM categories c

JOIN products p ON c.category\_id = p.category\_id

GROUP BY c.category\_name

HAVING COUNT(p.product\_id) > 1;

**(l)** Average price per supplier (only > $100):

SELECT s.supplier\_name, AVG(p.price) AS avg\_price

FROM suppliers s

JOIN products p ON s.supplier\_id = p.supplier\_id

GROUP BY s.supplier\_name

HAVING AVG(p.price) > 100;

**(m)** Products sorted by price (desc):

SELECT product\_name, price

FROM products

ORDER BY price DESC;

**(n)** Total stock value per category (sorted high → low):

SELECT c.category\_name, SUM(p.price \* p.stock\_quantity) AS total\_value

FROM categories c

JOIN products p ON c.category\_id = p.category\_id

GROUP BY c.category\_name

ORDER BY total\_value DESC;

**✅ 10 JOIN Queries**

**(a)** Products with category name (INNER JOIN):

SELECT p.product\_name, c.category\_name

FROM products p

INNER JOIN categories c ON p.category\_id = c.category\_id;

**(b)** Products with category name (LEFT JOIN):

SELECT p.product\_name, c.category\_name

FROM products p

LEFT JOIN categories c ON p.category\_id = c.category\_id;

**(c)** Categories with product count (including empty ones):

SELECT c.category\_name, COUNT(p.product\_id) AS product\_count

FROM categories c

LEFT JOIN products p ON c.category\_id = p.category\_id

GROUP BY c.category\_name;

**(d)** Products with supplier names (LEFT JOIN):

SELECT p.product\_name, s.supplier\_name

FROM products p

LEFT JOIN suppliers s ON p.supplier\_id = s.supplier\_id;

**(e)** Suppliers and products they supply (RIGHT JOIN):

SELECT s.supplier\_name, p.product\_name

FROM products p

RIGHT JOIN suppliers s ON p.supplier\_id = s.supplier\_id;

**(f)** Products without supplier:

SELECT product\_name

FROM products

WHERE supplier\_id IS NULL;

**(g)** Products with category and supplier names (multi-join):

SELECT p.product\_name, c.category\_name, s.supplier\_name

FROM products p

JOIN categories c ON p.category\_id = c.category\_id

JOIN suppliers s ON p.supplier\_id = s.supplier\_id;

**(h)** All suppliers and categories (FULL OUTER JOIN):

SELECT s.supplier\_name, c.category\_name

FROM suppliers s

FULL OUTER JOIN products p ON s.supplier\_id = p.supplier\_id

FULL OUTER JOIN categories c ON p.category\_id = c.category\_id;

**(i)** Products where supplier email is not null:

SELECT p.product\_name, s.supplier\_name

FROM products p

JOIN suppliers s ON p.supplier\_id = s.supplier\_id

WHERE s.contact\_email IS NOT NULL;

**(j)** Categories with products supplied by **Global Goods**:

SELECT DISTINCT c.category\_name

FROM categories c

JOIN products p ON c.category\_id = p.category\_id

JOIN suppliers s ON p.supplier\_id = s.supplier\_id

WHERE s.supplier\_name = 'Global Goods';