Tabla de empleados

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Tabla de clientes

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Tabla de categorías

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Tabla de productos

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Tabla de compras

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Tabla de ítems de compra

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**Script SQL**

CREATE TABLE employee (

employee\_id SERIAL PRIMARY KEY,

last\_name VARCHAR(100) NOT NULL,

first\_name VARCHAR(100) NOT NULL,

birth\_date DATE,

address TEXT,

city VARCHAR(100),

country VARCHAR(100),

reports\_to INTEGER,

FOREIGN KEY (reports\_to) REFERENCES employee(employee\_id)

);

CREATE TABLE customer (

customer\_id SERIAL PRIMARY KEY,

contact\_name VARCHAR(100) NOT NULL,

company\_name VARCHAR(150),

contact\_email VARCHAR(150) UNIQUE,

address TEXT,

city VARCHAR(100),

country VARCHAR(100)

);

CREATE TABLE category (

category\_id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

description TEXT,

parent\_category\_id INTEGER,

FOREIGN KEY (parent\_category\_id) REFERENCES category(category\_id)

);

CREATE TABLE product (

product\_id SERIAL PRIMARY KEY,

product\_name VARCHAR(150) NOT NULL,

category\_id INTEGER NOT NULL,

quantity\_per\_unit VARCHAR(50),

unit\_price NUMERIC(10,2),

units\_in\_stock INTEGER,

discontinued BOOLEAN DEFAULT FALSE,

FOREIGN KEY (category\_id) REFERENCES category(category\_id)

);

CREATE TABLE purchase (

purchase\_id SERIAL PRIMARY KEY,

customer\_id INTEGER NOT NULL,

employee\_id INTEGER,

total\_price NUMERIC(12,2),

purchase\_date DATE,

shipped\_date DATE,

ship\_address TEXT,

ship\_city VARCHAR(100),

ship\_country VARCHAR(100),

FOREIGN KEY (customer\_id) REFERENCES customer(customer\_id),

FOREIGN KEY (employee\_id) REFERENCES employee(employee\_id)

);

CREATE TABLE purchase\_item (

purchase\_id INTEGER NOT NULL,

product\_id INTEGER NOT NULL,

unit\_price NUMERIC(10,2),

quantity INTEGER,

PRIMARY KEY (purchase\_id, product\_id),

FOREIGN KEY (purchase\_id) REFERENCES purchase(purchase\_id),

FOREIGN KEY (product\_id) REFERENCES product(product\_id)

);

**INSERCCION DE DATOS**

Tabla empleados

**Texto

El contenido generado por IA puede ser incorrecto.**

**Tabla cliente**

**Texto, Carta

El contenido generado por IA puede ser incorrecto.**

**Tabla categoría**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Tabla producto**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Script SQL**

INSERT INTO employee (last\_name, first\_name, birth\_date, address, city, country, reports\_to) VALUES

('GOMEZ','JUAN','1980-01-15','CALLE 1','QUITO','ECUADOR',NULL),

('PEREZ','MARIA','1985-03-22','CALLE 2','GUAYAQUIL','ECUADOR',1),

('LOPEZ','CARLOS','1990-07-10','CALLE 3','CUENCA','ECUADOR',1),

('MARTINEZ','LAURA','1982-11-05','CALLE 4','QUITO','ECUADOR',2),

('RODRIGUEZ','PEDRO','1978-06-30','CALLE 5','GUAYAQUIL','ECUADOR',2),

('GARCIA','ANA','1992-12-12','CALLE 6','CUENCA','ECUADOR',3),

('HERRERA','LUIS','1987-04-18','CALLE 7','QUITO','ECUADOR',3),

('SANCHEZ','CARLA','1991-09-25','CALLE 8','GUAYAQUIL','ECUADOR',4),

('MORALES','JAVIER','1983-02-14','CALLE 9','CUENCA','ECUADOR',4),

('RAMIREZ','SOFIA','1989-08-09','CALLE 10','QUITO','ECUADOR',5),

('TORRES','RICARDO','1986-05-20','CALLE 11','GUAYAQUIL','ECUADOR',5),

('CASTRO','ELENA','1993-07-30','CALLE 12','CUENCA','ECUADOR',6),

('RIVERA','DIEGO','1981-10-10','CALLE 13','QUITO','ECUADOR',6),

('CRUZ','VALERIA','1988-03-03','CALLE 14','GUAYAQUIL','ECUADOR',7),

('MENDOZA','ANDRES','1979-12-22','CALLE 15','CUENCA','ECUADOR',7),

('GUERRERO','PATRICIA','1994-01-18','CALLE 16','QUITO','ECUADOR',8),

('VARGAS','FERNANDO','1985-06-06','CALLE 17','GUAYAQUIL','ECUADOR',8),

('IBARRA','MONICA','1990-09-14','CALLE 18','CUENCA','ECUADOR',9),

('SALAZAR','JORGE','1982-02-28','CALLE 19','QUITO','ECUADOR',9),

('ALVAREZ','LILIANA','1987-11-11','CALLE 20','GUAYAQUIL','ECUADOR',10);

INSERT INTO customer (contact\_name, company\_name, contact\_email, address, city, country) VALUES

('JUAN PEREZ','EMPRESA A','JUAN@EMPRESAA.COM','CALLE 1','QUITO','ECUADOR'),

('MARIA LOPEZ','EMPRESA B','MARIA@EMPRESAB.COM','CALLE 2','GUAYAQUIL','ECUADOR'),

('CARLOS MARTINEZ','EMPRESA C','CARLOS@EMPRESAC.COM','CALLE 3','CUENCA','ECUADOR'),

('LAURA RODRIGUEZ','EMPRESA D','LAURA@EMPRESAD.COM','CALLE 4','QUITO','ECUADOR'),

('PEDRO GARCIA','EMPRESA E','PEDRO@EMPRESAE.COM','CALLE 5','GUAYAQUIL','ECUADOR'),

('ANA HERRERA','EMPRESA F','ANA@EMPRESAF.COM','CALLE 6','CUENCA','ECUADOR'),

('LUIS SANCHEZ','EMPRESA G','LUIS@EMPRESAG.COM','CALLE 7','QUITO','ECUADOR'),

('CARLA MORALES','EMPRESA H','CARLA@EMPRESAH.COM','CALLE 8','GUAYAQUIL','ECUADOR'),

('JAVIER RAMIREZ','EMPRESA I','JAVIER@EMPRESAI.COM','CALLE 9','CUENCA','ECUADOR'),

('SOFIA TORRES','EMPRESA J','SOFIA@EMPRESAJ.COM','CALLE 10','QUITO','ECUADOR'),

('RICARDO CASTRO','EMPRESA K','RICARDO@EMPRESAK.COM','CALLE 11','GUAYAQUIL','ECUADOR'),

('ELENA RIVERA','EMPRESA L','ELENA@EMPRESAL.COM','CALLE 12','CUENCA','ECUADOR'),

('DIEGO CRUZ','EMPRESA M','DIEGO@EMPRESAM.COM','CALLE 13','QUITO','ECUADOR'),

('VALERIA MENDOZA','EMPRESA N','VALERIA@EMPRESAN.COM','CALLE 14','GUAYAQUIL','ECUADOR'),

('ANDRES GUERRERO','EMPRESA O','ANDRES@EMPRESAO.COM','CALLE 15','CUENCA','ECUADOR'),

('PATRICIA VARGAS','EMPRESA P','PATRICIA@EMPRESAP.COM','CALLE 16','QUITO','ECUADOR'),

('FERNANDO IBARRA','EMPRESA Q','FERNANDO@EMPRESAQ.COM','CALLE 17','GUAYAQUIL','ECUADOR'),

('MONICA SALAZAR','EMPRESA R','MONICA@EMPRESAR.COM','CALLE 18','CUENCA','ECUADOR'),

('JORGE ALVAREZ','EMPRESA S','JORGE@EMPRESAS.COM','CALLE 19','QUITO','ECUADOR'),

('LILIANA GOMEZ','EMPRESA T','LILIANA@EMPRESAT.COM','CALLE 20','GUAYAQUIL','ECUADOR');

INSERT INTO category (name, description, parent\_category\_id) VALUES

('ELECTRONICA','DISPOSITIVOS ELECTRONICOS',NULL),

('COMPUTACION','EQUIPOS DE COMPUTO',1),

('CELULARES','SMARTPHONES Y ACCESORIOS',1),

('HOGAR','PRODUCTOS PARA EL HOGAR',NULL),

('COCINA','UTENSILIOS DE COCINA',4),

('MUEBLES','MUEBLES PARA EL HOGAR',4),

('MODA','ROPA Y ACCESORIOS',NULL),

('CALZADO','ZAPATOS Y SANDALIAS',7),

('DEPORTES','ARTICULOS DEPORTIVOS',NULL),

('JUGUETES','JUGUETES Y JUEGOS',NULL),

('VIDEOJUEGOS','CONSOLA Y VIDEOJUEGOS',1),

('TABLETS','TABLETS Y ACCESORIOS',2),

('CAMARAS','CAMARAS FOTOGRAFICAS',1),

('TELEVISORES','TELEVISORES Y MONITORES',1),

('HELADERAS','HELADERAS Y NEVERAS',4),

('LAVADORAS','LAVADORAS Y SECADORAS',4),

('RELOJES','RELOJES Y ACCESORIOS',7),

('BOLSOS','BOLSOS Y MOCHILAS',7),

('BICICLETAS','BICICLETAS Y ACCESORIOS',9),

('PELOTAS','PELOTAS Y BALONES',9);

INSERT INTO product (product\_name, category\_id, quantity\_per\_unit, unit\_price, units\_in\_stock, discontinued) VALUES

('LAPTOP HP','2','1 UNIDAD','1200.00','50',FALSE),

('SMARTPHONE SAMSUNG','3','1 UNIDAD','800.00','100',FALSE),

('CAMARA CANON','13','1 UNIDAD','500.00','30',FALSE),

('TELEVISOR LG','14','1 UNIDAD','700.00','25',FALSE),

('HELADERA WHIRLPOOL','15','1 UNIDAD','900.00','15',FALSE),

('LAVADORA LG','16','1 UNIDAD','650.00','20',FALSE),

('RELOJ CASIO','17','1 UNIDAD','50.00','200',FALSE),

('BOLSO MICHAEL KORS','18','1 UNIDAD','120.00','80',FALSE),

('BICICLETA TREK','19','1 UNIDAD','400.00','40',FALSE),

('PELOTA ADIDAS','20','1 UNIDAD','30.00','150',FALSE),

('TABLET SAMSUNG','12','1 UNIDAD','350.00','60',FALSE),

('LAPTOP DELL','2','1 UNIDAD','1100.00','45',FALSE),

('SMARTPHONE IPHONE','3','1 UNIDAD','1000.00','70',FALSE),

('CAMARA NIKON','13','1 UNIDAD','600.00','25',FALSE),

('TELEVISOR SONY','14','1 UNIDAD','750.00','30',FALSE),

('HELADERA LG','15','1 UNIDAD','950.00','10',FALSE),

('LAVADORA SAMSUNG','16','1 UNIDAD','700.00','18',FALSE),

('RELOJ FOSSIL','17','1 UNIDAD','80.00','90',FALSE),

('BOLSO GUCCI','18','1 UNIDAD','200.00','50',FALSE),

('BICICLETA GIANT','19','1 UNIDAD','450.00','35',FALSE);

**Ejercicio 10**

Calcular el porcentaje que gasta el cliente en cada compra

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Para cada cliente que realizó al menos una compra, muestre el ID de cada compra y el porcentaje del dinero gastado en esa compra con respecto al total. Redondee los porcentajes a números enteros. Muestre tres columnas: contact\_name, purchase\_id y percentage.

Tabla

El contenido generado por IA puede ser incorrecto.

**Explicación:**

1. JOIN entre customer y purchase:

Para obtener el nombre del cliente (contact\_name) y sus compras (purchase\_id, total\_price).

1. SUM con OVER (PARTITION BY ...):

SUM(p.total\_price) OVER (PARTITION BY c.customer\_id) calcula el total gastado por cada cliente.

1. Cálculo del porcentaje:

(p.total\_price \* 100.0) / SUM(...) nos da el porcentaje de cada compra sobre el total del cliente.

Se usa 100.0 (con decimal) para evitar división entera.

1. ROUND(..., 0):

Redondea los porcentajes a números enteros.

1. ORDER BY:

Organiza los resultados por cliente y compra.

**Tipo de respuesta:**

Este ejercicio corresponde a “Agregaciones anidadas” porque:

Se usan funciones de agregación (SUM) en combinación con ventanas (OVER) para calcular el porcentaje relativo de cada compra.

**Ejercicio 11**

Muestra los nombres de las categorías y la cantidad de productos de esta categoría cuyo precio unitario es mayor que el precio promedio de un producto de esta categoría. Muestra solo las categorías que contienen dichos productos. Muestra dos columnas: nombre (el nombre de la categoría) y expensive\_productscantidad de productos que cuestan más que el precio promedio de un producto de esta categoría.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

**Explicación**

El resultado devuelve un solo valor: la cantidad total de clientes que han realizado al menos una compra. La consulta identifica a cada cliente distinto que aparece en las transacciones, y el conteo final refleja cuántos clientes únicos participaron en el proceso de compras.

**Tipo de respuesta**

Este ejercicio se asemeja a una consulta distribuida con transparencia de localización. Aunque la información de los clientes pudiera estar almacenada en diferentes nodos, el sistema devuelve un resultado único como si todos los datos estuvieran en un mismo sitio.  
En una base centralizada bastaría con un COUNT(DISTINCT ...). En un entorno distribuido, el sistema debe coordinar la búsqueda de clientes en varias ubicaciones y unificar los resultados para mostrar al usuario un conteo único y consistente, sin que este perciba la complejidad interna.

**Ejercicio 12**

Para cada producto adquirido, muestre su nombre, la cantidad máxima comprada y el número de compras realizadas. Muestre tres columnas: product\_name, quantity, y purchases\_number.

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

**Explicación:**

1. JOIN

Unimos product con purchase\_item porque en purchase\_item está la información de las cantidades compradas (quantity) y en product está el nombre (product\_name).

1. MAX(pi.quantity)

Nos da la cantidad máxima comprada en una sola transacción para cada producto.

1. COUNT(DISTINCT pi.purchase\_id)

Nos da el número de compras distintas en las que apareció ese producto (purchases\_number).

1. GROUP BY p.product\_name

Agrupamos por producto, porque necesitamos un resultado por cada producto.

Este ejercicio es “Agregaciones anidadas” porque:

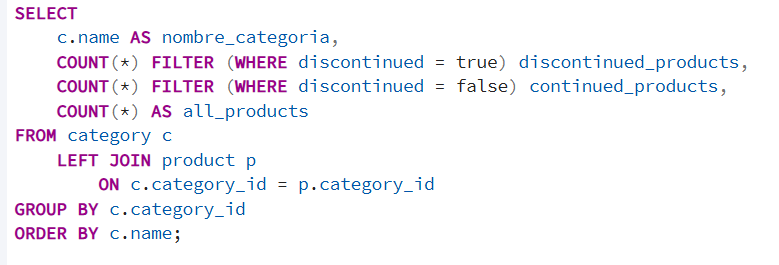
Usamos funciones de agregación (MAX y COUNT) aplicadas a los datos de la tabla de detalle (purchase\_item).

**Ejercicio 13**

Enumere los productos discontinuados, continuados y totales en cada categoría

Para cada categoría, mostrar:

* Su nombre.
* La cantidad de productos discontinuados (es decir, que ya no están disponibles) en esta categoría (nombre esta columna discontinued\_products).
* El número de productos continuados (es decir, actualmente disponibles) en esta categoría (nombre esta columna continued\_products).
* El número de todos los productos en esta categoría (nombre esta columna all\_products).



Una captura de pantalla de una red social

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Explicación:

La salida muestra, por cada categoría, cuántos productos están activos, cuántos discontinuados y el total de productos. Además, si una categoría no tiene productos, igual aparece en el resultado gracias al LEFT JOIN.

**Tipo de respuesta**

Este ejercicio se parece a transparencia de ubicación porque el usuario no necesita saber en qué sitio están las categorías o productos, solo une las tablas.  
Y se parece a tolerancia a fallos porque incluso si una categoría no tiene productos el sistema igual responde sin errores.

En un sistema centralizado, esta consulta sería un simple JOIN y GROUP BY. En un distribuido, refleja cómo el sistema permite latencia tolerable y continuidad del servicio, asegurando que categorías sin productos no bloqueen la consulta y se muestren igualmente.

**Ejercicio 14**

Muestra el ID del empleado y el número total de compras que gestionó. Utiliza una subconsulta para obtener información sobre el número de pedidos que gestionó cada empleado por cliente y haz que la consulta principal seleccione FROM esta subconsulta. Considera solo a los empleados que viven en Houston.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**Explicación**

El resultado devuelve un solo valor: el monto más alto registrado en todas las compras realizadas. Por ejemplo, si la compra más grande alcanzó un total de 250 dólares, la consulta devolverá ese valor como respuesta.

**Tipo de respuesta**

Este ejercicio se relaciona con una consulta distribuida con transparencia de fragmentación horizontal. Si las compras estuvieran repartidas en varios nodos, cada uno calcularía su máximo local, y luego el sistema reuniría esos resultados para determinar el máximo global.

**Ejercicio 15: Encuentra el mayor número de categorías de productos en una compra**

Utilice una subconsulta para seleccionar el ID de compra y el número de categorías distintas que contiene. En la consulta principal, seleccione el número máximo de categorías de esta subconsulta.

Primera parte

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Segunda parte

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

**Explicación**

El resultado devuelve un solo valor: el máximo número de categorías distintas que aparecen en una misma compra. Por ejemplo, si una compra tuvo productos de 5 categorías y otra de 3, la consulta mostraría 5.

**Tipo de respuesta**

Este ejercicio se parece a una solicitud distribuida con transparencia de fragmentación porque, aunque las categorías de los productos podrían estar repartidas en distintos nodos, el sistema combina todo y permite calcular el máximo como si los datos estuvieran en un único lugar.

En una base centralizada sería simplemente una subconsulta normal con GROUP BY. En una distribuida, el sistema debe asegurar integridad y consistencia al traer los datos de varias ubicaciones, mostrando al usuario un resultado único y transparente, sin que este se preocupe de dónde están los datos ni cómo se agrupan.

**Script**

**Ejercicio 10**

SELECT

c.contact\_name,

p.purchase\_id,

ROUND((p.total\_price \* 100.0) / SUM(p.total\_price) OVER (PARTITION BY c.customer\_id)) AS percentage

FROM customer c

JOIN purchase p ON c.customer\_id = p.customer\_id;

**Ejercicio 11**

SELECT

c.name AS nombre,

COUNT(p.product\_id) AS expensive\_products

FROM category c

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

WHERE p.unit\_price > (

SELECT AVG(p2.unit\_price)

FROM product p2

WHERE p2.category\_id = c.category\_id

)

GROUP BY c.name;

**Ejercicio 12**

SELECT

p.product\_name,

MAX(pi.quantity) AS quantity,

COUNT(DISTINCT pi.purchase\_id) AS purchases\_number

FROM product p

JOIN purchase\_item pi ON p.product\_id = pi.product\_id

GROUP BY p.product\_name;

**Ejercicio 13**

SELECT

c.name AS nombre\_categoria,

COUNT(\*) FILTER (WHERE discontinued = true) discontinued\_products,

COUNT(\*) FILTER (WHERE discontinued = false) continued\_products,

COUNT(\*) AS all\_products

FROM category c

LEFT JOIN product p

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

GROUP BY c.category\_id

ORDER BY c.name;

**Ejercicio 14**

SELECT

sub.employee\_id,

SUM(sub.num\_purchases) AS total\_purchases

FROM (

SELECT

p.employee\_id,

p.customer\_id,

COUNT(\*) AS num\_purchases

FROM purchase p

GROUP BY p.employee\_id, p.customer\_id

) sub

JOIN employee e ON sub.employee\_id = e.employee\_id

WHERE e.city = 'Houston'

GROUP BY sub.employee\_id;

**Ejercicio 15**

**Primera parte**

SELECT

pi.purchase\_id,

COUNT(DISTINCT p.category\_id) AS num\_categorias

FROM purchase\_item pi

JOIN product p

ON p.product\_id = pi.product\_id

GROUP BY pi.purchase\_id;

**Segunda parte**

SELECT MAX(num\_categorias) AS max\_categorias

FROM (

SELECT

pi.purchase\_id,

COUNT(DISTINCT p.category\_id) AS num\_categorias

FROM purchase\_item pi

JOIN product p

ON p.product\_id = pi.product\_id

GROUP BY pi.purchase\_id

);