

## Fundamentos de Bases de datos

**Nombre:** Luis Enrique Pérez Señalín

### --A. CONSULTAS GENERALES

--1. ¿Cuáles son los productos que cuestan más de 2 dólares?

```
SELECT * FROM "productos" WHERE "precio_unit" > 2;
```

--2. ¿Cuáles son los productos que cuestan entre 2 y 5 dólares?

```
SELECT * FROM "productos" WHERE "precio_unit" BETWEEN 2 AND 5;
```

--3. ¿Cuáles son los productos que cuestan entre 2 y 5 dólares

-- y pertenecen al grupo D?

```
SELECT * FROM "productos"
```

```
NATURAL JOIN "tipos"
```

```
WHERE "precio_unit" BETWEEN 2 AND 5
```

```
AND "tipo" LIKE 'D%';
```

--4. ¿Qué clientes han comprado algún producto con "Gum"?

```
SELECT "nombre_cli" FROM "clientes"
```

```
NATURAL JOIN "facturas"
```

```
NATURAL JOIN "items"
```

```
NATURAL JOIN "productos"
```

```
WHERE "nombre_prod" LIKE 'Gum%';
```

--5. ¿Cuáles son los clientes que no compraron Gummy?

```
SELECT "nombre_cli" FROM "clientes"
```

```
EXCEPT
```

```
SELECT "nombre_cli" FROM "clientes"
```

```
NATURAL JOIN "facturas"
```

```
NATURAL JOIN "items"
```

```
NATURAL JOIN "productos"
```

```
WHERE "nombre_prod" = 'Gummy';
```

--6. ¿Qué productos fueron comprados tanto por el cliente x como

-- por el cliente y?

```
SELECT "nombre_prod"
```

```
NATURAL JOIN "items"
NATURAL JOIN "facturas"
NATURAL JOIN "clientes"
WHERE "clientes"."nombre_cli" = 'ClienteX'
INTERSECT
SELECT "nombre_prod"
NATURAL JOIN "items"
NATURAL JOIN "facturas"
NATURAL JOIN "clientes"
WHERE "clientes"."nombre_cli" = 'ClienteY'
```

--7. ¿Qué productos han sido comprados en cantidad superior a 5?

```
SELECT P."nombre_prod"
FROM "productos" P
NATURAL JOIN "items"
GROUP BY "nombre_prod"
HAVING SUM("cant") > 5;
```

--8. ¿Cuál es el producto más vendido?

```
SELECT P."nombre_prod"
FROM "productos" P
NATURAL JOIN "items"
GROUP BY "nombre_prod"
ORDER BY SUM("cant") DESC
LIMIT 1;
```

--9. ¿Cuál es el cliente que ha gastado más dinero en la tienda?

```
SELECT "clientes"."nombre_cli"
FROM "clientes"
NATURAL JOIN "facturas"
GROUP BY "clientes"."nombre_cli"
ORDER BY "total_gastado" DESC
LIMIT 1;
```

--10. ¿Cuántos clientes han comprado los productos x y y simultáneamente?

--B. SUBCONSULTAS

```
SELECT COUNT(DISTINCT "clientes"."id_cliente") as num_clientes
FROM "clientes"
NATURAL JOIN "facturas"
NATURAL JOIN "items"
NATURAL JOIN "productos"
WHERE "productos"."nombre_prod" = 'ProductoX'
AND "clientes"."id_cliente" IN (
    SELECT "clientes"."id_cliente"
    FROM "clientes"
    NATURAL JOIN "facturas"
    NATURAL JOIN "items"
    NATURAL JOIN "productos"
    WHERE "productos"."nombre_prod" = 'ProductoY'
);
```

--11. ¿Cuántos clientes han comprado productos de el proveedor "nombre\_proveedor"?

```
SELECT COUNT(DISTINCT "clientes"."id_cliente") as num_clientes
FROM "clientes"
NATURAL JOIN "facturas"
NATURAL JOIN "items"
NATURAL JOIN "productos"
NATURAL JOIN "proveedores"
WHERE "proveedores"."nombre_prov" = 'nombre_proveedor';
```

--12. ¿Cuáles son los clientes que han comprado el producto más caro

-- de todos?

```
SELECT "clientes"."nombre_cli"
FROM "clientes"
NATURAL JOIN "facturas"
NATURAL JOIN "items"
```

```
NATURAL JOIN "productos"
WHERE "productos"."precio_unit" = (
    SELECT MAX("precio_unit") FROM "productos"
);
```

--13. ¿A qué tipos de productos pertenecen los productos comprados  
-- por "Juan Luna"?

```
SELECT DISTINCT "tipos"."tipo"
FROM "productos"
NATURAL JOIN "items"
NATURAL JOIN "facturas"
NATURAL JOIN "clientes"
NATURAL JOIN "tipos"
WHERE "clientes"."nombre_cli" = 'Juan Luna';
```

--C. TRANSACCIONES

--14) Realizar una transacción en la cual se realice una venta de productos y que  
-- se decremente el stock de esos productos e insertar los items respectivos.  
-- EXITOSA.

```
BEGIN;
```

```
INSERT INTO "facturas" (num_factura, total, iva, fecha_fact, id_cliente)
VALUES (31, 100.00, 12.00, '2023-06-18', 1);
```

-- Insertar los items correspondientes

```
INSERT INTO "items" (cant, num_factura, id_producto)
VALUES (5, 31, 1), (10, 31, 2);
```

-- Decrementar el stock de los productos

```
UPDATE "productos"
SET "unidad" = "unidad" - 5
WHERE "id_producto" = 1;
```

```
UPDATE "productos"
```

```
SET "unidad" = "unidad" - 10
```

```
WHERE "id_producto" = 2;
```

```
COMMIT;
```

```
--15. Igual a a) pero termina SIN EXITO.
```

```
BEGIN;
```

```
-- Asumiendo que hay una columna "stock" en la tabla "productos"
```

```
-- Insertar una nueva factura
```

```
INSERT INTO "facturas" (num_factura, total, iva, fecha_fact, id_cliente)
```

```
VALUES (32, 200.00, 24.00, '2023-06-18', 2);
```

```
-- Insertar los items correspondientes
```

```
INSERT INTO "items" (cant, num_factura, id_producto)
```

```
VALUES (5, 32, 3), (10, 32, 4);
```

```
-- Decrementar el stock de los productos
```

```
UPDATE "productos"
```

```
SET "unidad" = "unidad" - 5
```

```
WHERE "id_producto" = 3;
```

```
-- Generar un error intencionalmente para abortar la transacción
```

```
UPDATE "productos"
```

```
SET "unidad" = "unidad" - 10
```

```
WHERE "id_producto" = 999;
```

```
ROLLBACK;
```

```
--D. RESTRICCIONES DE INTEGRIDAD
```

```
--16. Crear una tabla que tenga restricciones de integridad de todos los tipos.
```

```
CREATE TABLE "equipos" (
```

```

    "id_equipo" int PRIMARY KEY,
    "nombre_equipo" varchar(50) NOT NULL,
    "lugar" varchar(50) NOT NULL
    "cant_jugadores" int CHECK (cantidad >= 0),
    "fecha_creacion" date DEFAULT CURRENT_DATE,
    "director" varchar(50) NOT NULL,
    "id_categoria" int,
    CONSTRAINT fk_categoria FOREIGN KEY ("id_categoria") REFERENCES
    "categorias"("id_categoria")
);

```

--17. Realizar 5 consultas sobre esta tabla, en las que intervengan las restricciones de  
 -- integridad colocadas..

--E. OTROS OPERACIONES CON SQL

--18. Incremente, en un 10%, el precio de los productos del tipo C (Coloque aquí uno de  
 -- los nombres de la base de datos).

```

UPDATE productos
SET precio_unit = precio_unit * 1.10
WHERE id_tipo IN (SELECT id_tipo FROM tipos WHERE tipo LIKE 'C%');

```

--19. Elimine los productos que nunca han sido comprados

```

DELETE FROM productos
WHERE id_producto NOT IN (SELECT DISTINCT id_producto FROM items);

```

--20. Cuáles son los productos (nombre, precio unitario) que han sido provistos  
 -- por proveedores con más de 200 unidades entregadas?.

```

SELECT p.nombre_prod, p.precio_unit
FROM productos p
NATURAL JOIN proveedores pr
WHERE p.unidad > '200';

```

--21. Liste los proveedores con las cantidades provistas a la empresa.

```

SELECT pr.nombre_prov, SUM(CAST(p.unidad AS INTEGER)) as cantidad_provista

```

```
FROM proveedores pr
NATURAL JOIN productos p
GROUP BY pr.nombre_prov;
```

--22. Listar los nombres y la dirección de los clientes que compraron 'Cebolla'.

```
SELECT c.nombre_cli, c.dir_cli
FROM clientes c
NATURAL JOIN facturas f
NATURAL JOIN items i
NATURAL JOIN productos p
WHERE p.nombre_prod = 'Cebolla';
```

--23. Liste los nombres y los precios de los productos, mostrando también el tipo al que pertenecen.

```
SELECT p.nombre_prod, p.precio_unit, t.tipo
FROM productos p
NATURAL JOIN tipos t;
```

--25. Ingrese un cliente que viva en el barrio 'Los ceibos', que se llame 'Angel Calle'  
-- que no tenga teléfono y que haya nacido en febrero del año 1999.

```
INSERT INTO clientes (nombre_cli, dir_cli, fecha_nac) VALUES ('Angel Calle', 'Los ceibos', '1999-02-01');
```

--26. Cambie la cantidad de los productos que tienen la sílaba 'ca' en su nombre y que no han sido vendidos por SELE

```
UPDATE items i
SET i.cant = i.cant + 10
FROM productos p
NATURAL JOIN proveedores pr ON p.id_proveedor = pr.id_proveedor
WHERE i.id_producto = p.id_producto
AND p.nombre_prod LIKE '%ca%'
AND pr.nombre_prov != 'SELE';
```

--NOTA: Los ejercicios serán realizados usando JOIN.