**Fundamentos de Bases de datos**

**Nombre:** Luis Enrique Pérez Señalin

--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.