usach

Presentación: SQL

Diseño de bases de datos - 01/2025 - Grupo 1

Integrantes:
Angel Benavides
Camila Raipán
Nicolás Rojas
Nicolás Valdés

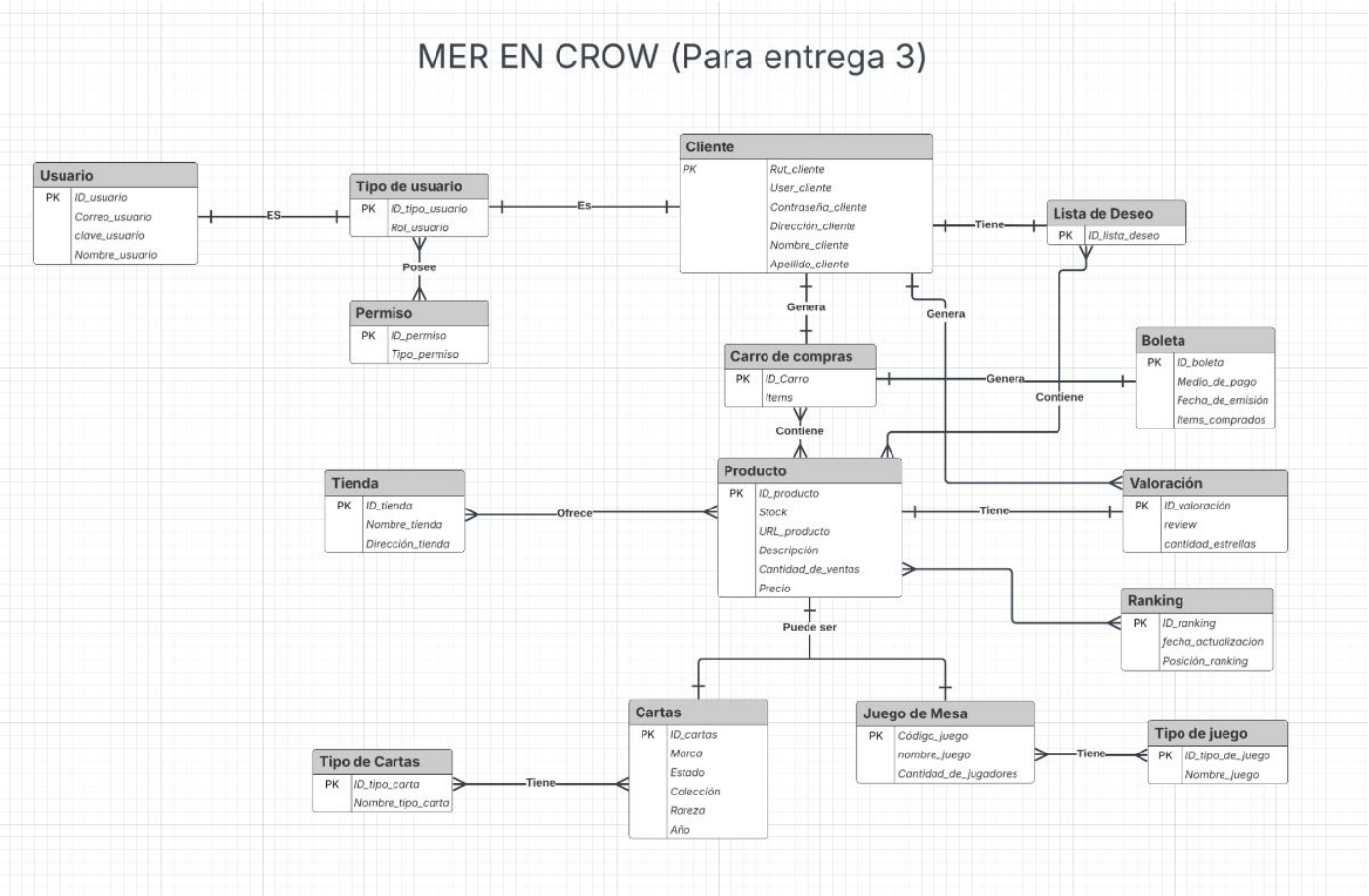
Índice

usach

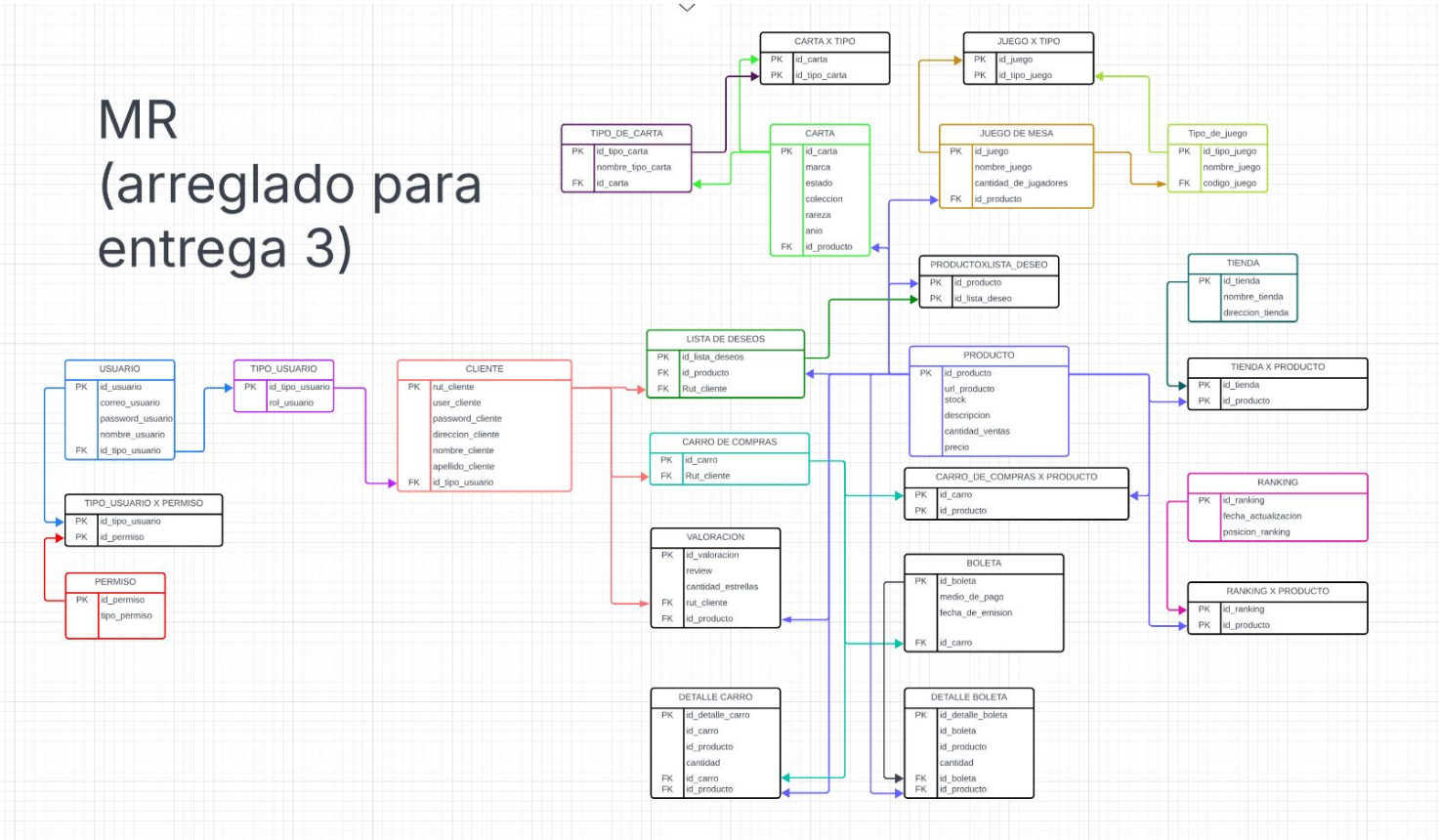
usach

Introducción

MER Corregido



MR Corregido



SQL

Comando CREATE: Creación de las tablas

Creación de las tablas

```sql
CREATE TABLE IF NOT EXISTS cliente(
  rut_cliente VARCHAR(12), -- formato: 12345678-9
  user_cliente VARCHAR(30) UNIQUE NOT NULL,
  password_cliente VARCHAR(60) NOT NULL,
  direccion_cliente VARCHAR(100),
  nombre_cliente VARCHAR(30) NOT NULL,
  apellido_cliente VARCHAR(30) NOT NULL,
  id_tipo_usuario INT,
  PRIMARY KEY (rut_cliente),
  FOREIGN KEY (id_tipo_usuario) REFERENCES tipo_usuario(id_tipo_usuario)
);
```

```sql
CREATE TABLE IF NOT EXISTS producto(
  id_producto SERIAL,
  url_producto VARCHAR(200),
  stock INT NOT NULL CHECK (stock >= 0),
  descripcion TEXT,
  cantidad_ventas INT DEFAULT 0 CHECK (cantidad_ventas >= 0),
  precio INT NOT NULL CHECK (precio >= 0),
  PRIMARY KEY (id_producto)
);
```

SQL

Comando INSERT: Carga de datos

Carga de datos

```
INSERT INTO carta (marca, estado, coleccion, rareza, anio)
VALUES ('Magic: The Gathering', 'Poor', 'Fuego II', 'Común', 2014),
('Yu-Gi-Oh!', 'Good', 'Hechizo', 'Poco común', 2021),
('Pokemon TCG', 'Excellent', 'Agua', 'Rara', 2016),
('Dragon Ball Super Card Game', 'Near Mint', 'Saiyajin Blue', 'Muy rara', 2022),
('Digimon', 'Mint', 'Fuego VI', 'Legendaria', 2010);
```

SQL

Sentencias SQL: Consultas

1. Agregar un juego de mesa / carta al carrito de compras

```
-- SENTENCIA 1. Agregar un juego de mesa / carta al carrito de compras.

-- supuestos:
-- existe un carro de compras con id_carro = 1 en la tabla carro_de_compras
-- el producto a agregar ya esta en la tabla producto (en esta caso tiene id_producto = 5)
-- y en este caso tiene id_producto = 5

-- Ejemplo para verificar uso:

--hacemos:
select * from detalle_carro
-- veremos que en carro con id = 1 tenemos objetos 3 objetos: 1 con id 2, 2 con id 4

-- ejecutamos
-- SENTENCIA 1
INSERT INTO detalle_carro (id_carro, id_producto, cantidad)
VALUES (1, 5, 1)
ON CONFLICT (id_carro, id_producto)
DO UPDATE SET cantidad = detalle_carro.cantidad + 1;


-- verificamos el cambio con
select * from detalle_carro
--(se recomienda ejecutar 3 veces este cambio para
-- verificar el uso de la siguiente sentencia)
```

2. Eliminar un juego de mesa / carta del carrito de compras

```sql
-- SENTENCIA 2. Eliminar un juego de mesa / carta del carrito de compras.

-- supuestos:
-- los valores de ejemplo, son los del producto que agregamos en la sentencia de arriba
-- existe un carro de compras con id_carro = 1 en la tabla carro_de_compras
-- el producto a disminuir existe en la tabla producto y tiene id_producto = 5
-- ya existe en detalle_carro una fila con (id_carro = 1, id_producto = 999)

-- Ejemplo para verificar uso:

-- para este ejemplo, se asume que se ejecuto la sentencia de arriba 3 veces,
-- por lo que id_carro = 1 debería tener 3 unidades de id_producto = 5


--SENTENCIA 2 (se deben ejecutar las dos partes al mismo tiempo)
-- si solo hay una unidad, se elimina completamente
DELETE FROM detalle_carro
WHERE id_carro = 1 AND id_producto = 5 AND cantidad = 1;
-- si hay más de una unidad, se resta a la cantidad
UPDATE detalle_carro
SET cantidad = cantidad - 1
WHERE id_carro = 1 AND id_producto = 5 AND cantidad > 1;


-- para ir verificando los cambios se hace:
SELECT * from detalle_carro

--  (se recomienda eliminar todos los productos con
--  id_producto = 5 para que el ejemplo dado en la SENTENCIA 4
--  funcione como se ejemplicó)




select * FROM detalle_carro
```

3. Mostrar los juegos de mesa / cartas del carrito de compras

```sql
-- SENTENCIA 3: Mostrar los juegos de mesa / cartas del carrito de compras

-- Queríamos implementar esta sentencia con join, pero se nos complico la cosa,
-- por lo que solo mostraremos el id de los productos que tiene en el carro
SELECT id_producto, cantidad
FROM detalle_carro
WHERE id_carro = 1;
```

4. Mostrar el precio total a pagar por el carrito de compras

```
-- SENTENCIA 4. Mostrar el precio total a pagar por el carrito de compras.

-- Supuestos:
-- cada producto tiene un campo precio en la tabla producto
-- el carrito tiene id_carro = 1
-- la tabla detalle_carro relaciona id_carro con id_producto y cantidad

-- Ejemplo para verificar uso:

-- hacemos:
select * from carro_de_comprasxproducto
-- veremos que en el carro con id_carro = 1, hay productos 1 y 3
-- hacemos
select * from detalle_carro
-- veremos que en carro con id_carro = 1 hay:
-- 1 productos con id_producto 2, 2 productos con id_producto 4
-- hacemos:
select * from producto
-- veremos que: producto 2 cuesta 3000
--              producto 4 cuesta 4000
-- asi: 3000 + (2 * 4000) = 11000
-- que es lo que retorna la consulta

--SENTENCIA 4:
SELECT SUM(producto.precio * Dcarro.cantidad) AS total_a_pagar
FROM detalle_carro Dcarro
JOIN producto ON Dcarro.id_producto = producto.id_producto
WHERE Dcarro.id_carro = 1;
```

5. Mostrar todos los juegos de mesa y cartas que se vendan en una ubicación geográfica específica

```sql
-- SENTENCIA 5. Mostrar todos los juegos de mesa y cartas que se
--              vendan en una ubicacion geografica especifica.

-- juegos de mesa
SELECT Jmesa.nombre_juego AS nombre, tienda.direccion_tienda AS ubicacion, 'Juego de mesa' AS tipo
FROM juego_de_mesa Jmesa
JOIN producto ON Jmesa.id_producto = producto.id_producto
JOIN tienda ON producto.id_tienda = tienda.id_tienda
WHERE tienda.direccion_tienda ILIKE '%Avenida Balatro 2033%'
UNION
-- cartas
SELECT carta.marca AS nombre, tienda.direccion_tienda AS ubicacion, 'Carta' AS tipo
FROM carta carta
JOIN producto producto ON carta.id_producto = producto.id_producto
JOIN tienda tienda ON producto.id_tienda = tienda.id_tienda
WHERE tienda.direccion_tienda ILIKE '%Santiago%'
ORDER BY tipo;
```

6. Mostrar ranking de los productos con más ventas

```sql
-- 6. Mostrar ranking de los productos con mas ventas.

-- juegos de mesa
SELECT JMesa.nombre_juego AS nombre, producto.id_producto, producto.cantidad_ventas, 'Juego de mesa' AS tipo
FROM producto
JOIN juego_de_mesa JMesa ON producto.id_producto = JMesa.id_producto
UNION
-- cartas
SELECT carta.marca AS nombre, producto.id_producto, producto.cantidad_ventas, 'Carta' AS tipo
FROM producto
JOIN carta ON producto.id_producto = carta.id_producto
ORDER BY cantidad_ventas DESC;
```

7. Mostrar Lista de deseos de un usuario

```sql
-- SENTENCIA 7. Mostrar lista de deseos de un usuario.

-- SENTENCIA 7
-- solo muestra el id_producto, tratamos de mostrar algo mas descriptivo,
-- pero no logramos hacerlo por errores con el join
SELECT id_producto
FROM lista_deseos
WHERE rut_cliente = '14520364-1';


/*
INTENTO FALLIDO DE MOSTRAR EL NOMBRE
-- Juegos de mesa
SELECT jm.nombre_juego AS nombre, p.precio, 'Juego de mesa' AS tipo
FROM lista_deseos ld
JOIN producto p ON ld.id_producto = p.id_producto
JOIN juego_de_mesa jm ON p.id_producto = jm.id_producto
WHERE ld.rut_cliente = '14520364-1'
UNION
-- Cartas
SELECT c.marca AS nombre, p.precio, 'Carta' AS tipo
FROM lista_deseos ld
JOIN producto p ON ld.id_producto = p.id_producto
JOIN carta c ON p.id_producto = c.id_producto
WHERE ld.rut_cliente = '14520364-1';
*/
```

Conclusión

usach

Muchas gracias

Diseño de bases de datos - 01/2025 - Grupo 1