





### Revisão Prática 1 — Banco de Dados (BDR)

**Tema:** Rede de Lojas de Jogos e E-Sports — rede\_games

#### Cenário

A empresa **rede\_games** precisa de um sistema para gerenciar informações sobre suas **lojas**, os **jogos** vendidos, os **clientes** cadastrados e as **compras realizadas**.

Você foi contratado para modelar e implementar a base de dados no PostgreSQL.

As tabelas necessárias são:

- loja
  - id\_loja (PK)
  - o nome
  - o cidade
- jogo
  - id\_jogo (PK)
  - o titulo
  - o ano\_lancamento
  - o genero
- cliente
  - id\_cliente (PK)
  - o nome

- email (UNIQUE)
- o cidade
- compra
  - o id\_compra (PK)
  - data\_compra
  - id\_cliente (FK → cliente.id\_cliente)
  - id\_loja (FK → loja.id\_loja)
- compra\_jogo (tabela associativa)
  - id\_compra (FK → compra.id\_compra)
  - o id\_jogo (FK → jogo.id\_jogo)
  - o quantidade

### Questão 1 — Criação do banco

Crie o banco de dados chamado rede\_games.

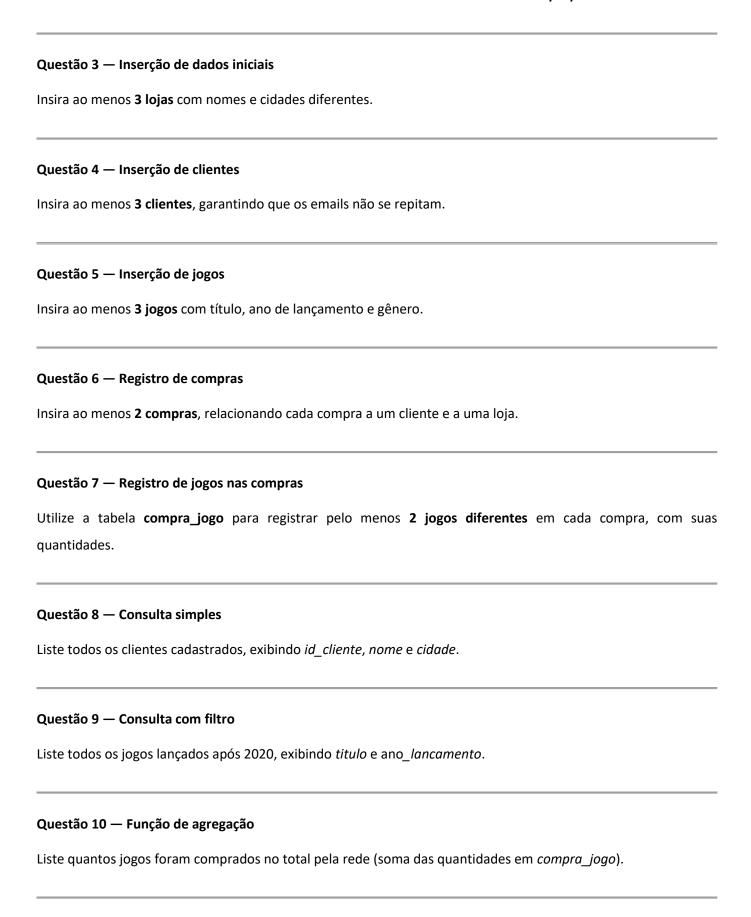
### Questão 2 — Criação das tabelas

Implemente as tabelas conforme o modelo acima, incluindo PK, FK e a restrição UNIQUE em cliente.email.















### Observações

- Todas as questões são incrementais (o resultado de uma será usado na próxima).
- O esquema envolve PK, FK, UNIQUE, N:M (via tabela associativa) e consultas com agregações.
- A prova prática cobre exatamente o que aprenderam nas aulas.

### Forma de Entrega — Revisão Prática 1 BDR

## 1. Local de entrega:

- o Cada aluno deve ter um repositório individual no GitHub (já utilizado nas aulas).
- Criar uma pasta chamada:
- o /revisao\_pratica1\_bdr/

#### 2. Arquivos a serem entregues:

- o schema.sql → contendo os comandos de criação do banco e das tabelas (Questões 1 e 2).
- o **inserts.sql** → contendo os comandos de inserção de dados (Questões 3 a 7).
- o **consultas.sql** → contendo os comandos das consultas (Questões 8 a 10).

## 3. Organização sugerida dentro da pasta:

- 4. /revisao\_pratica1\_bdr/
- 5. ├— schema.sql
- 6. ├— inserts.sql
- 7. └─ consultas.sql

#### 8. Execução dos scripts:

- o Os scripts devem rodar no **PostgreSQL (pgAdmin ou psql)** sem erros.
- A ordem de execução deve ser:
  - 1. schema.sql
  - 2. inserts.sql
  - 3. consultas.sql

#### 9. Entrega final:







- o Fazer commit no GitHub até o prazo definido.
- o Compartilhar o **link do repositório** na plataforma institucional (ou via e-mail para o professor).

# 10. Critérios de avaliação:

- Organização dos scripts.
- o Correção dos comandos SQL.
- o Funcionamento das consultas.
- o Boa prática no uso de nomes de tabelas/atributos.
- o Cumprimento do prazo.