Consultando Dados

Aula 13 — SELECT, WHERE e ORDER BY

UniSenac Centro Universitário RS
Senac
Fecomércio
IFEP
Sesc
Sindicatos Empresariais

O mapa de hoje

- **Bloco 1** — `SELECT` e `FROM`: escolher colunas
- **Bloco 2** — `WHERE`: escolher linhas
- **Bloco 3** — `ORDER BY`, `LIMIT`, `DISTINCT` e o `SELECT` pelo Python
- **Bloco 4** — mão na massa: perguntas reais sobre os dados do projeto

> Hoje é a primeira aula em que vocês **tiram** informação do banco. Até agora só colocamos coisas dentro.

Retomando a Aula 12

- Qual é o hábito antes de rodar um `DELETE`?
- Por que `'2026-10-21'` e não `'21/10/2026'`?
- O que acontece se esquecer o `commit()`?

> Abram o pgAdmin no banco do grupo, com dados dentro. Se o banco estiver vazio, me chame agora — hoje é tudo sobre consultar dados.

Bloco 1

SELECT — escolhendo colunas

A pergunta mais comum de todas

```
SELECT titulo, genero
FROM   jogo;
```

GLOSSÁRIO

`SELECT`	selecionar — quais colunas mostrar
`FROM`	de — de qual tabela

Lê-se: *"selecione as colunas titulo e genero da tabela jogo"*.

`SELECT *` — todas as colunas

```
SELECT * FROM jogo;
```

O `*` quer dizer "todas as colunas".

Ótimo para **explorar** uma tabela que você não conhece. Ruim em programa de verdade: se alguém acrescentar uma coluna depois, seu código recebe uma coluna que ele não esperava.

AS — dando nome ao resultado

```
SELECT titulo AS "Nome do jogo",
       genero AS "Categoria"
FROM   jogo;
```

GLOSSÁRIO

AS	como — apelido para a coluna no resultado
alias	apelido

Aqui as **aspas duplas** são corretas: elas marcam nome de coluna — exatamente o que a Aula 12 disse. Por isso o apelido pode ter espaço e acento.

Checagem do Bloco 1

Escreva a consulta:

- Mostrar só o `nome` e o `telefone` de todos os clientes
- Mostrar tudo da tabela `plataforma`
- Mostrar o `titulo` dos jogos, com a coluna chamada "Jogo"



`forms.gle/4i47cdaYRsufVD78A`

Intervalo

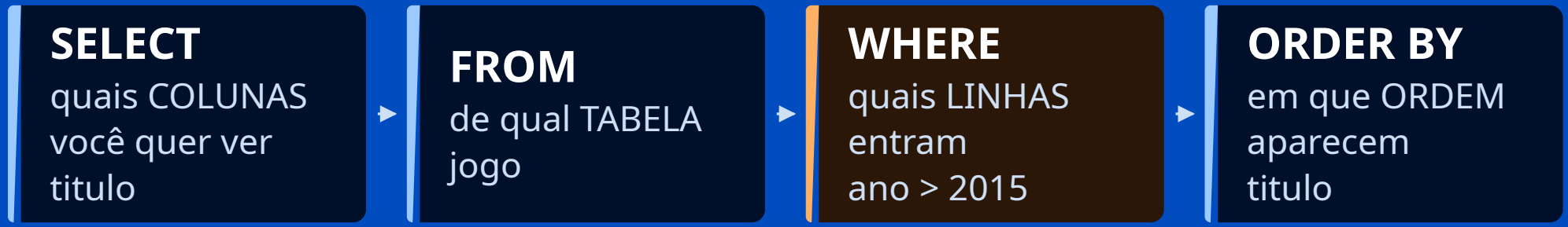
10 minutos

Bloco 2

WHERE — escolhendo linhas

As quatro partes de uma consulta

AS QUATRO PARTES DE UMA CONSULTA



Cada parte responde uma pergunta diferente. O `WHERE` é o que filtra **linhas**.

Operadores de comparação

```sql
SELECT * FROM jogo WHERE ano_lancamento > 2015;
SELECT * FROM jogo WHERE genero = 'RPG';
SELECT * FROM jogo WHERE genero <> 'Esporte';
```

Operador	EN → PT
`=`	igual
`<>` ou `!=`	diferente
`>` `<` `>=` `<=`	maior, menor, maior ou igual, menor ou igual

Texto vai entre **aspas simples**; número, não. É a mesma regra do `INSERT` da Aula 12.

`AND` e `OR` — combinando condições

```sql
SELECT * FROM jogo
WHERE  genero = 'RPG' AND ano_lancamento > 2015;

SELECT * FROM jogo
WHERE  genero = 'RPG' OR genero = 'Aventura';
```

Cuidado ao misturar: use **parênteses** para deixar claro o que pertence a quem.

```sql
WHERE (genero = 'RPG' OR genero = 'Aventura') AND ano_lancamento > 2015
```

`LIKE` — buscar por pedaço de texto

```sql
SELECT * FROM jogo WHERE titulo LIKE 'God%';    -- começa com God
SELECT * FROM jogo WHERE titulo LIKE '%War';    -- termina com War
SELECT * FROM jogo WHERE titulo LIKE '%of%';    -- contém "of"
SELECT * FROM jogo WHERE titulo ILIKE '%god%';  -- ignora maiúscula/minúscula
```

GLOSSÁRIO

`LIKE`	parecido com
`%`	curinga — qualquer coisa, de qualquer tamanho
`ILIKE`	*insensitive like* — ignora maiúscula e minúscula (só no PostgreSQL)

`IN`, `BETWEEN` e `IS NULL`

```sql
SELECT * FROM jogo WHERE genero IN ('RPG', 'Aventura', 'Ação');
SELECT * FROM jogo WHERE ano_lancamento BETWEEN 2015 AND 2020;
SELECT * FROM cliente WHERE email IS NULL;
SELECT * FROM cliente WHERE email IS NOT NULL;
```

Para nulo, **não** funciona `= NULL`. Nulo é "não sei", e "não sei" não é igual a nada — nem a outro "não sei". Use `IS NULL`.

Contas e textos dentro do SELECT

```
SELECT titulo,
       preco_diaria * 3        AS "Preço de 3 dias",
       2026 - ano_lancamento   AS "Idade do jogo"
FROM   jogo;
```

Lembram do atributo **derivado** da Aula 04 (a `idade` que a gente decidiu **não** guardar)? É aqui que ele aparece — calculado na hora da consulta, sempre certo, sem ninguém precisar atualizar nada.

⚠ Erro comum — consultar sem saber o que se quer

> Escrever `SELECT` antes de saber a **pergunta em português** é a receita para consulta que roda, devolve linhas e não significa nada.

O método, sempre:

1. Escreva a pergunta em **português**
2. Identifique **qual tabela** tem essa informação
3. Identifique **quais linhas** interessam (o `WHERE`)
4. Identifique **quais colunas** mostrar (o `SELECT`)
5. Só então escreva o SQL

Checagem do Bloco 2

Escreva o `WHERE`:

- Jogos lançados entre 2018 e 2022
- Clientes que **não** informaram e-mail
- Jogos de RPG **ou** Aventura, lançados depois de 2015
- Jogos cujo título contém a palavra "Zelda"

forms.gle/PhCNLgDhS25n5F3i9

Intervalo

10 minutos

Bloco 3

Ordenar, limitar e consultar pelo Python

`ORDER BY` — a ordem do resultado

```sql
SELECT * FROM jogo ORDER BY titulo;              -- A a Z
SELECT * FROM jogo ORDER BY ano_lancamento DESC; -- mais novo primeiro
SELECT * FROM jogo ORDER BY genero, titulo;      -- por gênero, e dentro dele por título
```

GLOSSÁRIO

`ORDER BY`	ordenar por
`ASC`	ascending — crescente (é o padrão)
`DESC`	descending — decrescente

LIMIT e DISTINCT

```sql
SELECT * FROM jogo ORDER BY ano_lancamento DESC LIMIT 5;  -- só os 5 mais novos

SELECT DISTINCT genero FROM jogo;  -- cada gênero aparece uma vez só
```

GLOSSÁRIO

`LIMIT n`	limitar a n linhas
`DISTINCT`	distinto — remove as repetições do resultado

A ordem em que o banco realmente executa

```
SELECT titulo[3] FROM jogo[1] WHERE ano > 2015[2] ORDER BY titulo[4]
```

Você escreve nesta ordem. O banco executa na de baixo.



Você **escreve** SELECT primeiro, mas o banco **executa** o FROM e o WHERE antes. Por isso o apelido criado no `SELECT ... AS` não pode ser usado no `WHERE` — naquele momento ele ainda não existe.

SELECT pelo Python — lendo o resultado

```python
cursor.execute("SELECT titulo, genero FROM jogo ORDER BY titulo;")
linhas = cursor.fetchall()

for linha in linhas:
    print(linha[0], "-", linha[1])
```

GLOSSÁRIO

`fetchall`	buscar todas — traz todas as linhas do resultado
`fetchone`	buscar uma — traz só a próxima linha

`SELECT` **não** precisa de `commit()` — ele só lê, não muda nada.

Checagem do Bloco 3

- Como mostrar os 3 jogos mais antigos?
- Como listar os gêneros sem repetir?
- Por que `SELECT` não precisa de `commit()`?
- Por que este comando dá erro?

```
SELECT titulo AS nome FROM jogo WHERE nome = 'FIFA 25';
```

forms.gle/6sC6XncQpodjevWu6

Intervalo

10 minutos

MÃO NA MASSA

Perguntas reais sobre o projeto

Escrevam consultas que respondam perguntas **do negócio de vocês**.

1. Escrevam **5 perguntas em português** que o dono do negócio faria
2. Traduzam cada uma para `SELECT`, no pgAdmin
3. Usem pelo menos um `WHERE` com `AND` ou `OR`, um `LIKE` e um `ORDER BY`
4. Repitam **duas** dessas consultas no notebook, com `fetchall()`
5. Confiram que o resultado é o **mesmo** nos dois caminhos

MÃO NA MASSA

O combinado desta entrega

GLOSSÁRIO

`grupoNN-aulaNN-artefato.<ext>` `grupo03-aula13-consultas.sql`

O que eu confiro: Cada consulta tem uma **pergunta em português** em cima · A consulta **responde** mesmo aquela pergunta · Usaram `AND` / `OR` com **parênteses** onde precisava · Alguém tentou `= NULL` em vez de `IS NULL` · As duas consultas do notebook deram o **mesmo** resultado do pgAdmin

Terminou? Tentem uma pergunta que precisa de **duas tabelas** — "qual o nome do cliente de cada empréstimo?". Vão travar, e é de propósito: é o `JOIN` de amanhã.

PARA CASA · OPCIONAL

Antes da próxima aula

Testar em casa: escreva 2 consultas novas sobre o projeto, diferentes das de aula. Comece pela pergunta em português.

Tarefa opcional: tente responder "qual o nome do cliente de cada empréstimo?" com o que você sabe hoje. Anote por que não dá — amanhã a gente resolve.

Para treinar: `locadora-exercicios-13-consultas.sql` — nove perguntas do balcão, com a conferência no fim do arquivo. Como fica a entrega do grupo: `locadora-exemplo-13-consultas.sql`.

Para ver acontecer: demonstrações da Aula 13 — montar a consulta peça por peça, e descobrir o que o LIKE realmente casa — e oito perguntas com resposta na hora. No site da UC, na página da aula.

Para a próxima aula

Todas as consultas de hoje olharam **uma tabela só**. Como responder uma pergunta que mora em **duas**?

UniSenac Centro Universitário RS
Senac
Fecomércio
IFEP
Sesc
Sindicatos Empresariais