



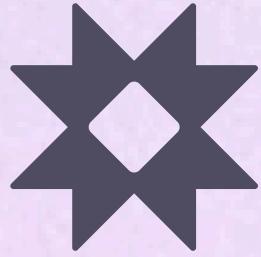
INTRODUÇÃO À BANCO DE DADOS

# Aniverse



Alunos:

Amanda Fernandes Alves  
Carolina Vilazante Portella  
Gabriel Camargos da Silveira



# Tópicos

01. MINI-MUNDO

---

02. PROJETO CONCEITUAL

---

03. PROJETO LÓGICO

---

04. CONSULTAS E RESULTADOS

---

05. CONCLUSÃO

---

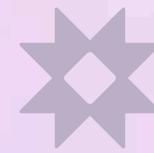




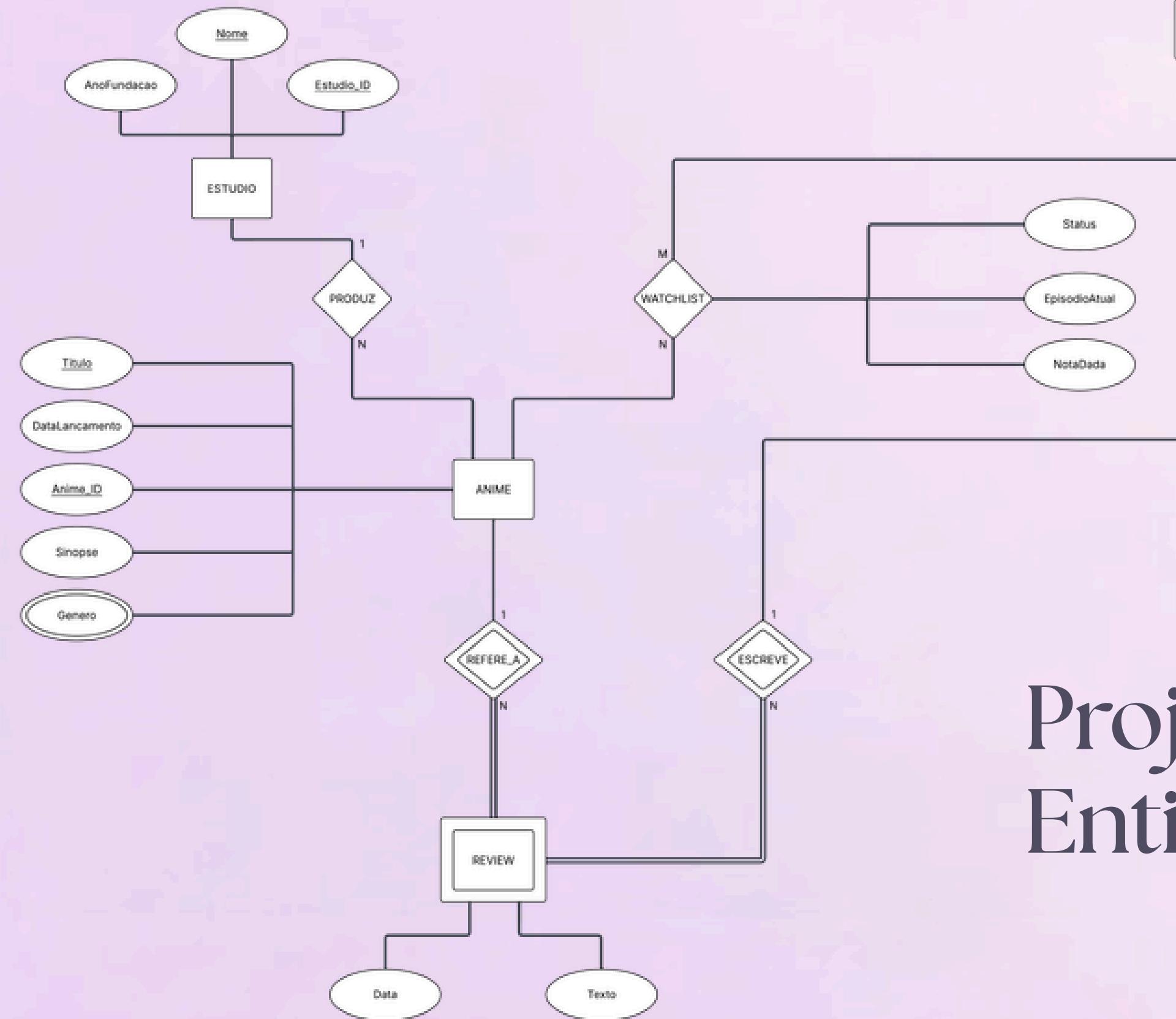
# Mini-Mundo: AniVerse

- **Aplicação:** Uma plataforma onde os usuários podem escrever avaliações, além de ter o controle de animes já vistos e os seus progressos nas listas de episódios.
- **Requisitos atendidos:**
  - 4 entidades (Anime, Estúdio, Usuário e Review);
  - 4 relacionamentos;
  - 1 relacionamento com cardinalidade M:N (Watchlist);
  - Toda entidade possui pelo menos 2 atributos, além de sua chave (identificador);
  - 1 atributo multivvalorado (Gênero).



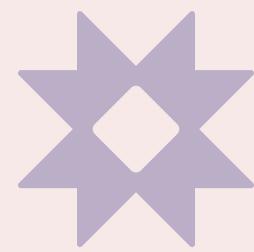


# Projeto Aniverse



Projeto Conceitual Diagrama  
Entidade Relacionamento (ER)

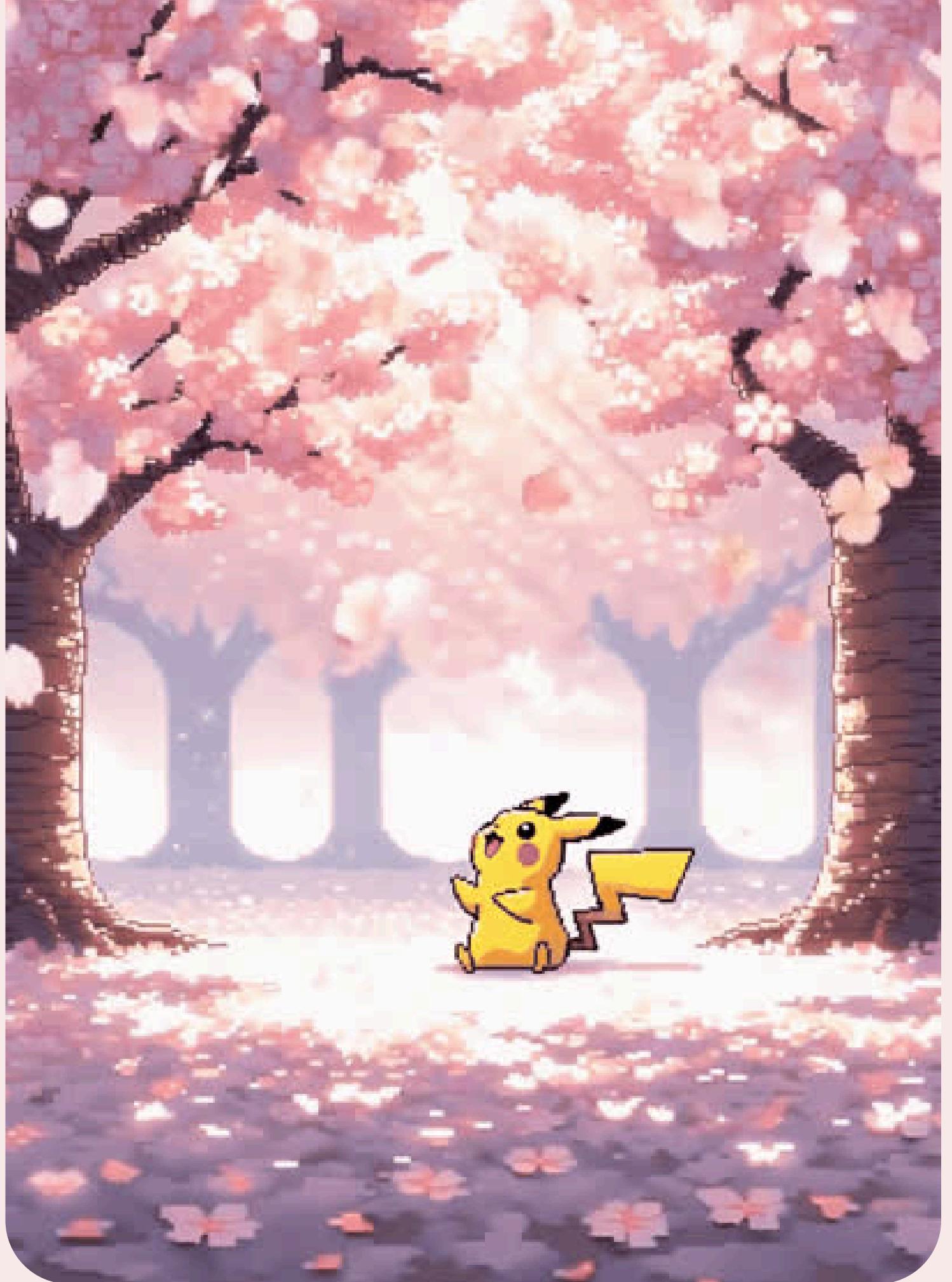




# Projeto Conceitual – Requisitos Textuais e Regras de Negócio

A modelagem satisfaz as restrições semânticas e requisitos de especificação a seguir:

- **Cadastros:** O sistema armazena dados de Usuários e Estúdios;
- **Catálogo:** Cada Anime é produzido por um Estúdio. Porém, a inserção do estúdio não é necessária, uma vez que, para aproximar o modelo da realidade, nem todo anime possui estúdio;
- **Classificação:** Um Anime pode ter múltiplos gêneros. Portanto, Gênero é um atributo multivalorado e, por isso, exige normalização;
- **Watchlist:** Relacionamento M:N entre Usuário e Anime, com atributos: Status, Nota, Episódio Atual;
- **Interação:** Usuários escrevem reviews. Review é uma entidade fraca e depende de um Usuário e de um Anime para existir.



# Projeto Lógico – Esquema Relacional

Tabelas	Atributos
ESTUDIO	<u>EstudioID</u> (nn), <u>Nome</u> (nn), AnoFundacao
USUARIO	<u>UsuarioID</u> (nn), <u>UserName</u> (nn), <u>Email</u> (nn), DataCadastro(nn)
ANIME	<u>AnimelD</u> (nn), <u>Titulo</u> (nn), Sinopse, DataLancamento, EstudioID_FK
ANIME_GENERO	<u>AnimelD_FK</u> (nn), <u>Genero</u> (nn).
WATCHLIST	<u>UsuarioID_FK</u> (nn), <u>AnimelD_FK</u> (nn), Status(nn), NotaDada, EpisodioAtual
REVIEW	<u>UsuarioID_FK</u> (nn), <u>AnimelD_FK</u> (nn), Texto(nn), Data(nn)

- Legenda: as chaves primárias estão sublinhadas e (nn) indica not null.
- Observação: para garantir a regra de negócio de apenas uma avaliação por par Usuário-Obra, a tabela REVIEW utiliza (UsuarioID, AnimelD) como chave primária composta.



# Restrições de Integridade Referencial e Opções de Remoção

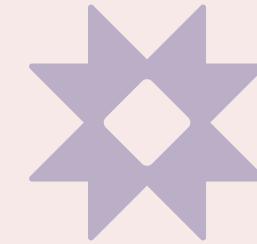
- **SET NULL (n)**

- **ANIME** [EstudioID\_FK] → **ESTUDIO** [EstudioID].
- **Justificativa:** ao apagar um estúdio os animes produzidos por ele não são removidos.

- **CASCADE (p)**

- **ANIME\_GENERO** [AnimeID\_FK] → **ANIME** [AnimeID];
- **WATCHLIST** [UsuarioID\_FK] → **USUARIO** [UsuarioID];
- **WATCHLIST** [AnimeID\_FK] → **ANIME** [AnimeID];
- **REVIEW** [UsuarioID\_FK] → **USUARIO** [UsuarioID];
- **REVIEW** [AnimeID\_FK] → **ANIME** [AnimeID].
- **Justificativa:** Dependencia existencial, um não existe sem o outro.





# Implementação e Análise de Desempenho – Metodologia

- O bando de dados foi populado com dados sintéticos, gerados por uma ferramenta externa chamada [Tonic.ia](#), eles são:
  - 1.000 Usuários;
  - 2.000 Animes;
  - 20.000 Registros na Watchlist;
  - 10.000 Reviews.
- Cada consulta foi realizada 5 vezes e analisamos a média dos tempos;
- Executamos as consultas em um ambiente local (MySQL Server 8.0 rodado em localhost) para anular a latência de rede e obter tempos de CPU puros.



# Consulta 1 – Seleção e Projeção (Filtragem por Data)



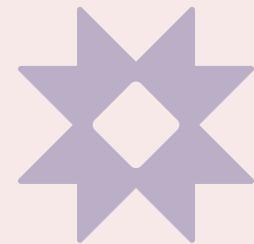
Filtra todos os usuários cadastrados no dia 15 de maio de 2024

Observação: foi criado um índice temporário pro atributo *DataCadastro*

```
-- QUERY "RUIM": Aplica função na coluna
SELECT Username, Email
FROM USUARIO
WHERE DATE(DataCadastro) = '2024-05-15';
```

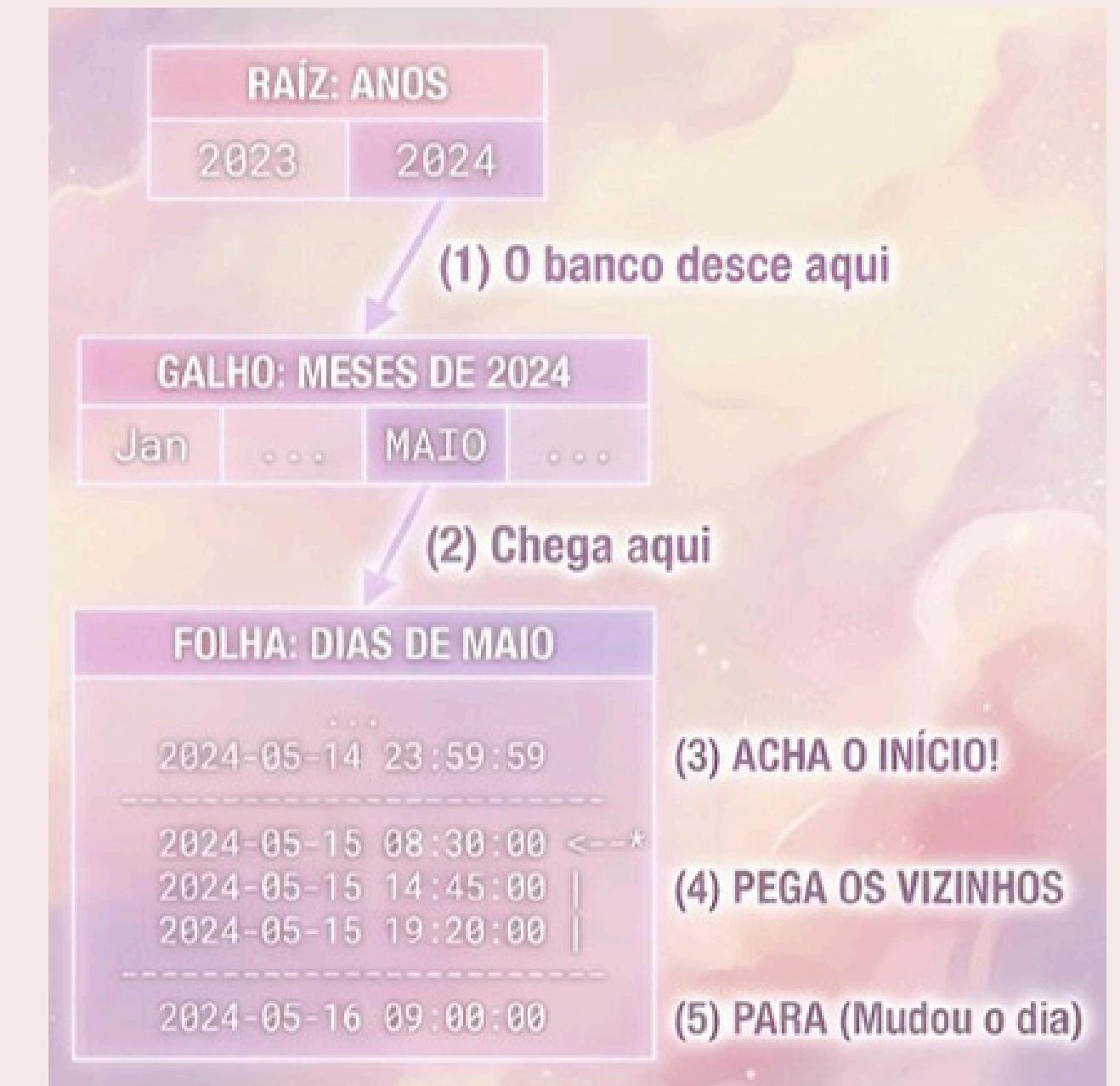
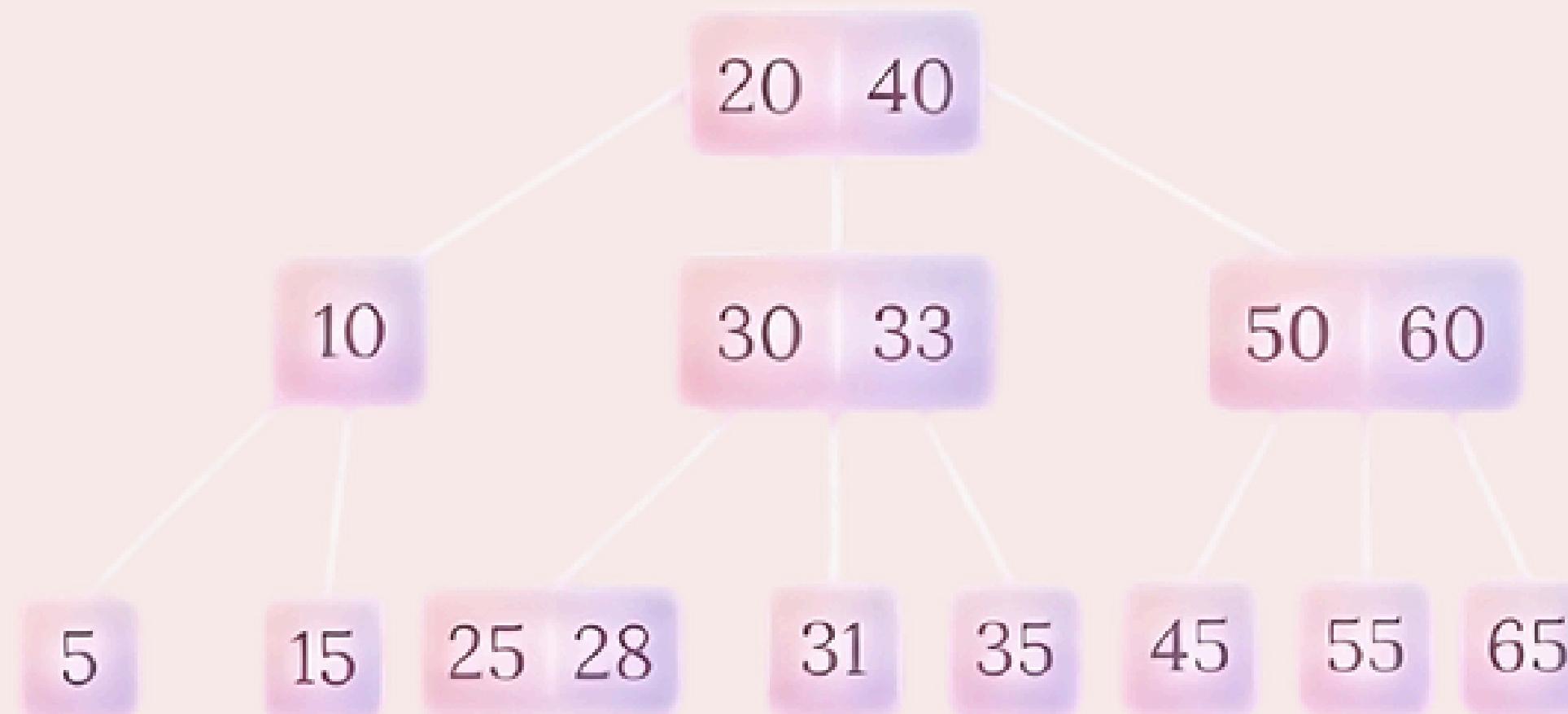
```
-- QUERY "BOA": Compara range puro
SELECT Username, Email
FROM USUARIO
WHERE DataCadastro >= '2024-05-15 00:00:00'
    AND DataCadastro <= '2024-05-15 23:59:59';
```

Formulação	Tempo Médio (s)	Análise
A: Função DATE	0.000686	Mais lenta, pois a função torna a consulta não-SARGable, com isso o cálculo é feito linha a linha.
B: Range (Dia)	0.000184	~3.7x mais rápida. Possibilita o uso de Index Seek na árvore B-Tree.



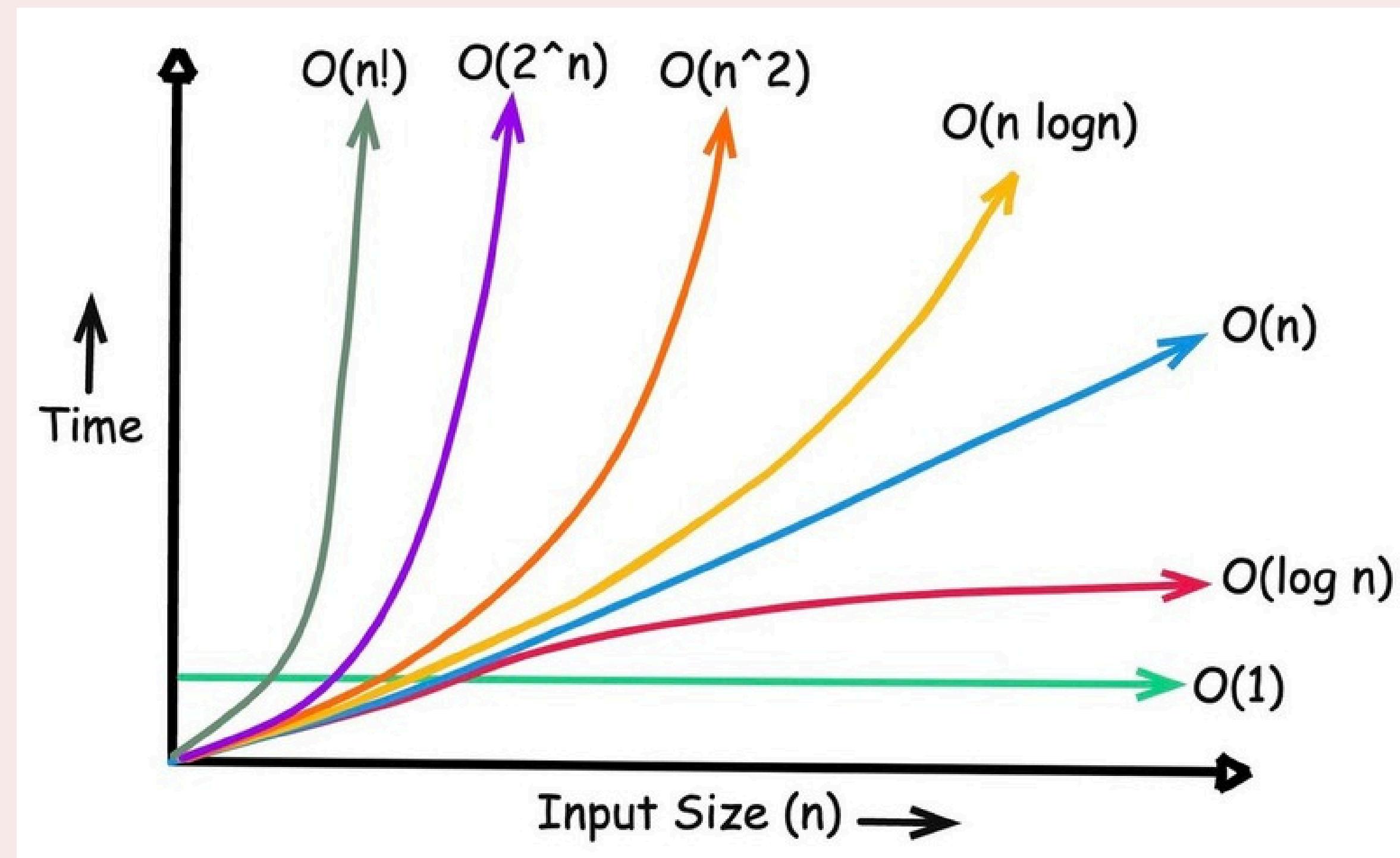
# O que é uma B-Tree?

- Complexidade logarítmica  $O(\log N)$
- Organização hierárquica: Raiz → Nós Internos → Folhas
- Navegação na árvore via decisões de "maior ou menor" (Index Seek)
- Leitura de disco (I/O) é o principal gargalo em bancos de dados
- Estrutura B-Tree reduz páginas de dados na memória RAM





# Gráfico Big O Notation

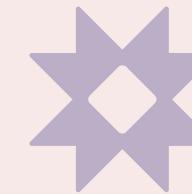




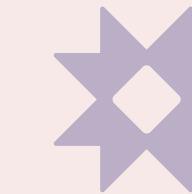
# Amostra de Resultados da Consulta 1

Username	Email
roberto.araujo	vitorteixeira912@outlook.com





# Consulta 2 – Seleção e Projeção (Busca Textual / Autocomplete)



Simula uma funcionalidade de autocomplete, visando recuperar os dados (Username e Email) de usuários cujo nome de usuário inicia com o prefixo textual "ana".

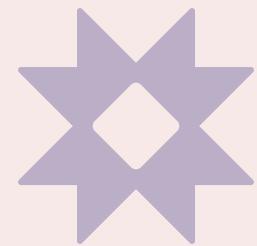
-- QUERY A (Lenta - LEFT)

```
SELECT Username, Email  
FROM USUARIO  
WHERE LEFT.Username, 3) = 'ana';
```

-- QUERY B (Rápida - LIKE)

```
SELECT Username, Email  
FROM USUARIO  
WHERE Username LIKE 'ana%';
```

Formulação	Tempo Médio (s)	Análise
A: Função LEFT	0.000587	Full Index Scan, ou seja, o banco precisa processar a string de cada registro.
B: Operador LIKE	0.000174	~3.3x mais rápida, já que o otimizador usa o índice para ir direto ao prefixo 'ana'.



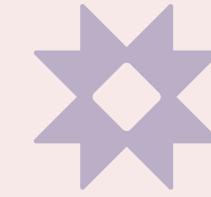
# Amostra de Resultados da Consulta 2

Username	Email
ana_cardoso	fabio.carvalho@terra.com.br
ana_Castro747	marcelomoraism697@bol.com.br
ana_freitas	patriciafernandes385@hotmail.com





# Consulta 3 – Junção de Duas Relações (Anime e Estúdio)

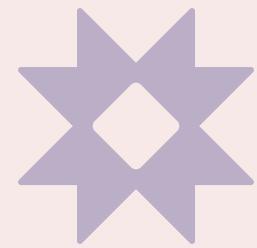


Lista o título de cada anime (ANIME.Titulo) juntamente com o nome do estúdio responsável por sua produção (ESTUDIO.Nome).

```
-- QUERY A (JOIN Padrão)
SELECT A.Titulo, E.Nome
FROM ANIME A
INNER JOIN ESTUDIO E ON A.EstudiodID_FK = E.EstudiodID;
```

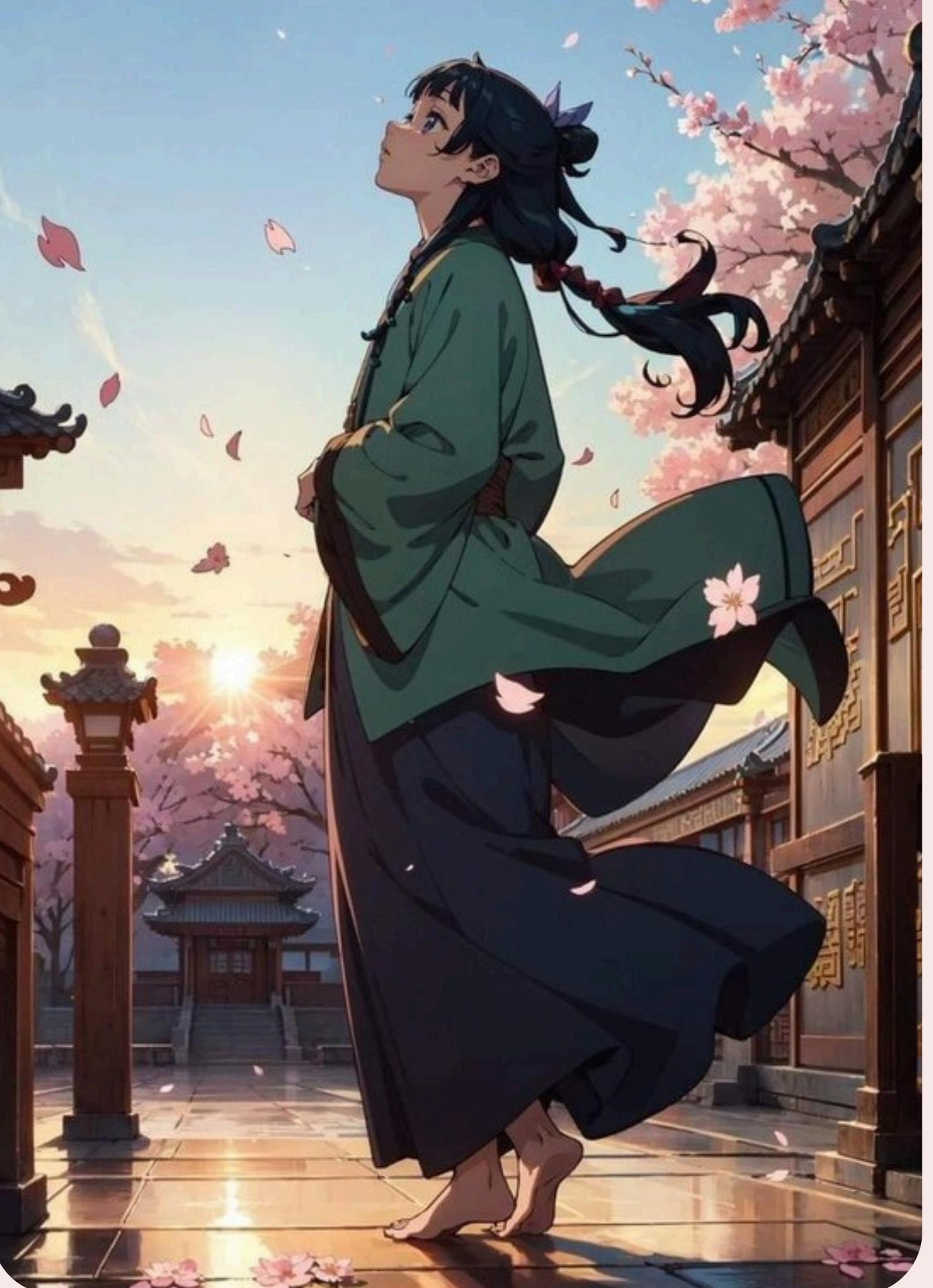
```
-- QUERY B (Subconsulta Correlacionada)
SELECT
    A.Titulo,
    (SELECT E.Nome
     FROM ESTUDIO E
     WHERE E.EstudiodID = A.EstudiodID_FK) AS Nome_Estudio
FROM ANIME A;
```

Formulação	Tempo Médio (s)	Análise
A: INNER JOIN (ANSI-92)	0.006490	Uma vez que, o otimizador do MySQL converte ambas as sintaxes para o mesmo plano de execução há um empate técnico. Em cenários de produção a versão A seria a melhor opção graças a sua clareza, manutenibilidade e menor risco de erros.
B: WHERE (ANSI-89)	0.004458	



# Amostra de Resultados da Consulta 3

Anime (Título)	Estúdio (Nome)
Dream Fate	8bit
The Tale of Legend	8bit
Peace Super	8bit



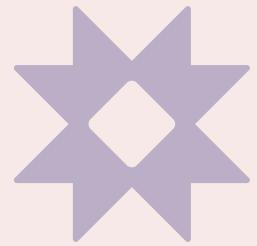
# Consulta 4 – Junção de Duas Relações (Reviews e Autores)

Recupera o conteúdo textual das avaliações (REVIEW.Texto) juntamente com a identificação do autor (USUARIO.Username).

```
-- QUERY A (JOIN Padrão)
SELECT R.Texto, U.Username
FROM REVIEW R
INNER JOIN USUARIO U ON R.UsuarioID_FK = U.UsuarioID;
```

```
-- QUERY B (Subconsulta Correlacionada)
SELECT R.Texto,
       (SELECT U.Username
        FROM USUARIO U
        WHERE U.UsuarioID = R.UsuarioID_FK) AS Autor
FROM REVIEW R;
```

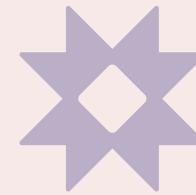
Formulação	Tempo Médio (s)	Análise
A: INNER JOIN (Operação em Conjunto - Otimizada)	0.034246	Empate técnico, mas JOIN tem uma leve vantagem. Ainda que, subconsultas correlacionadas costumem ser lentas, a busca é feita pela chave primária do usuário (Index Seek), o que diminui o custo de execução repetida.
B: Subconsulta (Operação Iterativa)	0.039128	



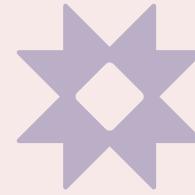
# Amostra de Resultados da Consulta 4

Review (Texto)	Autor (Username)
Vinland Saga has incredible storytelling, complex characters...	adriana_barros
The God of High School is visually spectacular with incredible...	claudia_araujo
The fan service feels gratuitous and detracts from what could...	leonardo_carvalho936





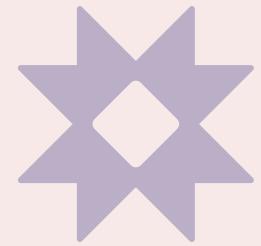
# Consulta 5 – Junção de Duas Relações (Anime e Anime\_Gênero)



Filtra e lista os títulos dos animes classificados sob um gênero específico (neste caso, 'Shounen').

```
-- QUERY A (JOIN)
SELECT A.Titulo
FROM ANIME A
INNER JOIN ANIME_GENERO G ON A.AnimeID = G.AnimeID_FK
WHERE G.Genero = 'Shounen';
```

Formulação	Tempo Médio (segundos)	Análise
A: JOIN	0.002939	O otimizador reescreve a Cláusula IN como uma semi-junção (Materialization), com isso o desempenho da Cláusula é igualado ao JOIN explícito.
B: Cláusula IN	0.002592	



# Amostra de Resultados da Consulta 5

## Título do Anime

The Magic of Shadow

Ice Princess

Code Geass 4

Dimension Saga

Hunter: Concert





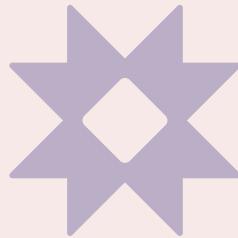
# Consulta 6 – Junção de 3+ Relações (Usuário → Watchlist → Anime → Estúdio)



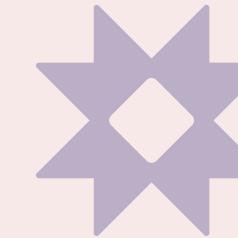
Identifica usuários com alta afinidade por um estúdio específico. O cenário busca listar os Usernames dos usuários que atribuíram a nota máxima (10) a qualquer obra produzida pelo estúdio 'MAPPA'.

```
-- QUERY A (JOIN PLANO)
SELECT DISTINCT U.Username
FROM USUARIO U
INNER JOIN WATCHLIST W ON U.UsuarioID = W.UsuarioID_FK
INNER JOIN ANIME A ON W.AnimeID_FK = A.AnimeID
INNER JOIN ESTUDIO E ON A.EstudioID_FK = E.EstudioID
WHERE E.Nome = 'MAPPA' AND W.NotaDada = 10;
```

```
-- QUERY B (Aninhamento Profundo com IN)
SELECT Username
FROM USUARIO
WHERE UsuarioID IN (
    SELECT UsuarioID_FK
    FROM WATCHLIST
    WHERE NotaDada = 10 AND AnimeID_FK IN (
        SELECT AnimeID
        FROM ANIME
        WHERE EstudioID_FK IN (
            SELECT EstudioID
            FROM ESTUDIO
            WHERE Nome = 'MAPPA'
        )
    )
);
```

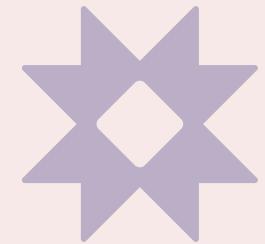


# Consulta 6 – Junção de 3+ Relações (Fãs do Estúdio)



Identifica usuários com alta afinidade por um estúdio específico. O cenário busca listar os Usernames dos usuários que atribuíram a nota máxima (10) a qualquer obra produzida pelo estúdio 'MAPPA'.

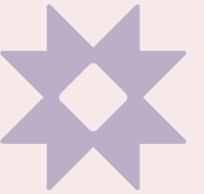
Formulação	Tempo Médio (s)	Análise
A: JOINs Múltiplos	0.000561	Houve um empate técnico.
B: Subconsultas Aninhadas Profundas	0.000791	Demonstra a capacidade do Query Rewriting do SGBD em “desenrolar” (flatten) subconsultas complexas em operações de junção otimizadas.



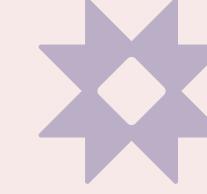
# Amostra de Resultados da Consulta 6

Username
matheus_pinto
sergio_machado
carlos_nunes
raquel_fernandes
elaine.carvalho





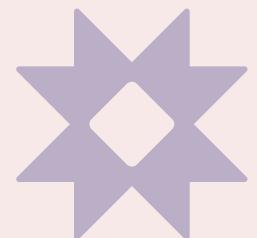
# Consulta 7 – Junção de 3+ Relações (Estúdio, Anime e Review)



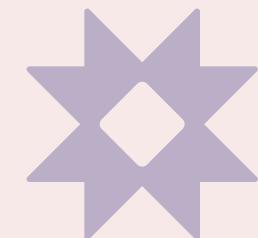
Gera um relatório detalhado contendo o texto das avaliações (REVIEW.Texto), o título da obra (ANIME.Titulo) e o nome do estúdio (ESTUDIO.Nome), filtrando apenas pelos animes produzidos pelo estúdio 'Wit Studio'.

```
-- QUERY A (Ordem Otimizada Manualmente)
SELECT R.Texto, A.Titulo, E.Nome
FROM ESTUDIO E
INNER JOIN ANIME A ON E.EstudioID = A.EstudioID_FK
INNER JOIN REVIEW R ON A.AnimeID = R.AnimeID_FK
WHERE E.Nome = 'Wit Studio';
```

```
-- QUERY B (Ordem Invertida)
SELECT R.Texto, A.Titulo, E.Nome
FROM REVIEW R
INNER JOIN ANIME A ON R.AnimeID_FK = A.AnimeID
INNER JOIN ESTUDIO E ON A.EstudioID_FK = E.EstudioID
WHERE E.Nome = 'Wit Studio';
```

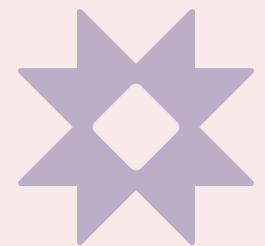


# Consulta 7 – Junção de 3+ Relações (Reviews do Estúdio)



Gera um relatório detalhado contendo o texto das avaliações (REVIEW.Texto), o título da obra (ANIME.Titulo) e o nome do estúdio (ESTUDIO.Nome), filtrando apenas pelos animes produzidos pelo estúdio 'Wit Studio'.

Formulação	Tempo Médio (s)	Análise
A: FROM (Top-Down)	0.000363	Os resultados podem ser considerados identicos. Com isso, comprovamos que o MySQL utiliza um Otimizador Baseado em Custo (CBO), que reordena as junções internamento (Join Reordering) independentemente da escrita.
B: FROM (Bottom-Up)	0.000349	



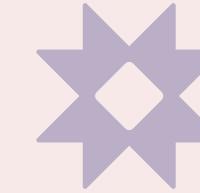
# Amostra de Resultados da Consulta 7

Review (Trecho)	Anime	Estúdio
The animation in Ergo Proxy is artsy but often feels pretentious...	My Dusk Chaos	Wit Studio
Attack on Titan is an absolute masterpiece that keeps you...	My Dusk Chaos	Wit Studio
March Comes in Like a Lion has beautiful animation and...	False Tournament	Wit Studio
Golden Kamuy is unique historical action-adventure...	False Tournament	Wit Studio
Evangelion: 3.0+1.0 is visually stunning and emotionally...	Secret Team	Wit Studio





# Consulta 8 – Junção de 3+ Relações (Usuário, Watchlist e Anime\_Gênero)

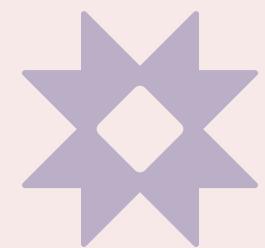


Identifica os usuários que completaram pelo menos um anime do gênero 'Shounen', sem repetições de nomes.

```
-- QUERY A (Gera duplicatas e limpa com DISTINCT)
SELECT DISTINCT U.Username
FROM USUARIO U
INNER JOIN WATCHLIST W ON U.UsuarioID = W.UsuarioID_FK
INNER JOIN ANIME_GENERO G ON W.AnimeID_FK = G.AnimeID_FK
WHERE W.Status = 'Completo' AND G.Genero = 'Shounen';
```

```
-- QUERY B (Verificação de Existência)
SELECT U.Username
FROM USUARIO U
WHERE EXISTS (
    SELECT 1
    FROM WATCHLIST W
    INNER JOIN ANIME_GENERO G ON W.AnimeID_FK = G.AnimeID_FK
    WHERE W.UsuarioID_FK = U.UsuarioID
    AND W.Status = 'Completo'
    AND G.Genero = 'Shounen'
);
```

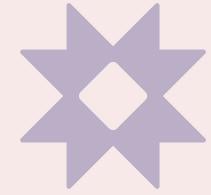
Formulação	Tempo Médio (s)	Análise
A: JOIN + DISTINCT	0.010128	Mais lento, pois exige ordenação e deduplicação custosa de todo o conjunto de dados.
B: EXISTS (Semi-Join)	0.008008	~21% mais rápido, já que realiza Short-Circuit (para encontrar o primeiro registro).



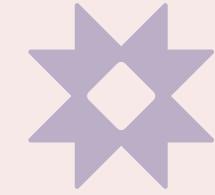
# Amostra de Resultados da Consulta 8

Username
diego.gomes
mauricio.teixeira
tatiana.silva
katia_pereira269
carlos_nunes





# Consulta 9 – Agregação com Junção (Média de Notas por Estúdio)

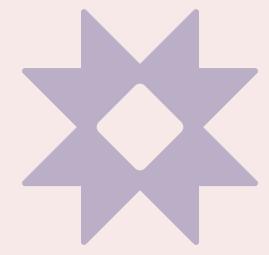


Calcula a média aritmética (AVG) das notas atribuídas pelos usuários na WATCHLIST para todas as obras produzidas por cada ESTUDIO. A operação envolve a agregação de dados através de três tabelas: ESTUDIO → ANIME → WATCHLIST.

Formulação	Tempo Médio (s)	Análise
A: JOIN + GROUP BY	0.035789	Empate técnico/leve vantagem Sunquery. O custo de ordenação do GROUP BY para o volume de dados atual superou ligeramente o custo das buscas indexadas repetidas da subconsulta
B: Subconsultas Correlacionada	0.030704	

```
-- QUERY A (GROUP BY)
SELECT E.Nome, AVG(W.NotaDada) as Media
FROM ESTUDIO E
INNER JOIN ANIME A ON E.EstudioID = A.EstudioID_FK
INNER JOIN WATCHLIST W ON A.AnimeID = W.AnimeID_FK
GROUP BY E.Nome;
```

```
-- QUERY B (Subconsulta por Linha)
SELECT E.Nome,
       (SELECT AVG(W.NotaDada)
        FROM WATCHLIST W
        INNER JOIN ANIME A ON W.AnimeID_FK = A.AnimeID
        WHERE A.EstudioID_FK = E.EstudioID) as Media
FROM ESTUDIO E;
```



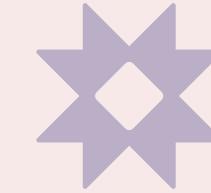
# Amostra de Resultados da Consulta 9

Estúdio	Média de Notas
8bit	4.8667
A-1 Pictures	5.4000
Actas	5.1429
Adventure Lab	5.6522
Adventure Network	5.5333





# Consulta 10 – Agregação com Junção (Popularidade por Gênero)

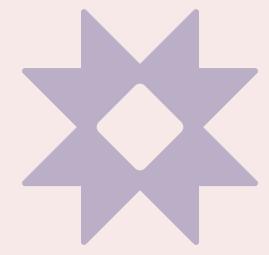


Gera estatísticas de consumo na plataforma, contabilizando quantos animes de cada Gênero foram marcados com o status 'Completo' nas listas dos usuários.

```
-- QUERY A (GROUP BY)
SELECT G.Genero, COUNT(*) as TotalConcluidos
FROM ANIME_GENERO G
INNER JOIN WATCHLIST W ON G.AnimeID_FK = W.AnimeID_FK
WHERE W.Status = 'Completo'
GROUP BY G.Genero
ORDER BY TotalConcluidos DESC;
```

```
-- QUERY B (N+1 Queries)
SELECT DISTINCT G.Genero,
(SELECT COUNT(*)
 FROM WATCHLIST W
 INNER JOIN ANIME_GENERO G2 ON W.AnimeID_FK = G2.AnimeID_FK
 WHERE G2.Genero = G.Genero AND W.Status = 'Completo')
 as TotalConcluidos
FROM ANIME_GENERO G
ORDER BY TotalConcluidos DESC;
```

Formulação	Tempo Médio (s)	Análise
A: JOIN + GROUP BY	0.029690	Mais rápido.
B: Subconsulta no SELECT	18.354369	618 vezes mais lento, essa versão ilustra o problema “N+1 Queries”, onde uma agregação pesada é executada repetidamente para cada gênero, degradando a performance exponencialmente.



# Amostra de Resultados da Consulta 10

Gênero	Total Concluídos
Isekai	829
Adventure	816
Mecha	797
Shoujo	795
Shounen	795



— OBRIGADA! —

