Universidade Federal de Minas Gerais

Introdução à Bancos de Dados

**Trabalho Prático 2**

Danilo Favato  
Samuel Cunha

Sonia Fraga

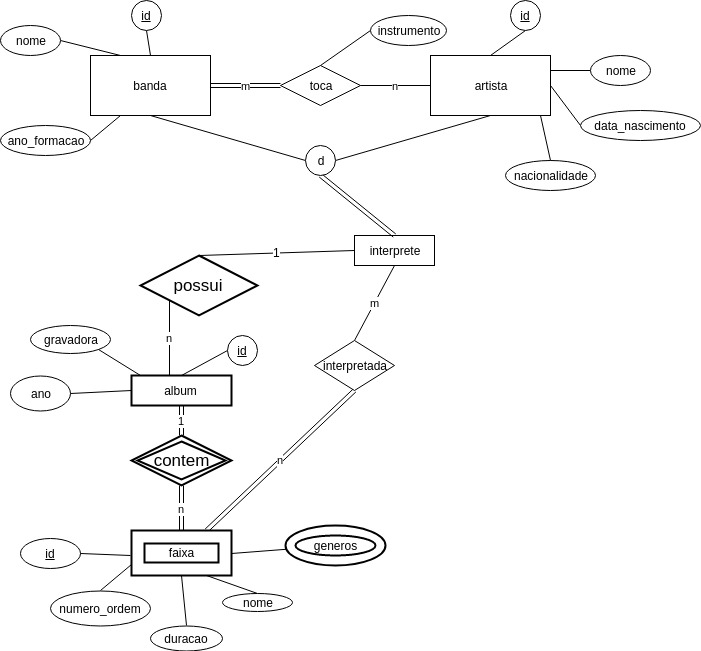
Tatiana Camelo

1. **Introdução**

Bancos de dados são um conjunto de arquivos que relacionam-se entre si com o propósito de ordenar, guardar e manipular informações. A ordenação de dados em empresas, por exemplo, é vital para manter a qualidade de suas operações e ter fácil acesso para informações importantes de forma simples e ordenada.

O banco de dados em questão é constituído por um total de 3,2 MB de dados e foi feito sobre relações de músicas, contando com relações entre banda, artista, intérprete, álbum e faixa, de acordo com as especificidades e informações contidas em cada atributo. Estes dados foram criados por um gerador de dados (acesso em: http://www.freedatagenerator.com). Devido a isso, algumas informações do banco podem ser contra-intuitivas (como faixa 2587 de um álbum ou nomes repetidos) mas isto não altera seu valor para o desenvolvimento do projeto e para a consulta ao banco. O desenho abaixo é o esquema relacional implementado.

As consultas englobam diferentes conceitos de banco de dados e foram implementadas de forma que cada consulta tem pelo menos duas versões possíveis e 2 envolvem apenas as operações de seleção e projeção; 3 envolvam a junção de duas relações; 3 envolvam a junção de três ou mais relações e 2 envolvam funções de agregação sobre o resultado da junção de pelo menos duas relações. Com isso, aplicamos os conhecimentos adquiridos ao longo do semestre.

****

***Figura 1: Esquema relacional***

***Esquema lógico relacional***

genero\_faixa [faixa\_id] ->(p) faixa [id]

faixa [album\_id] ->(p) album [id]

album [interprete\_id] ->(b) interprete [id]

interprete [artista\_id ] ->(p) artista [id]

interprete [banda\_id ] ->(p) banda [id]

interprete\_faixa [interprete\_id] ->(b) interprete [id]

interprete\_faixa [faixa\_id] ->(b) faixa [id]

toca [artista\_id] ->(b) artista [id]

toca [banda\_id] -> (b) banda [id]

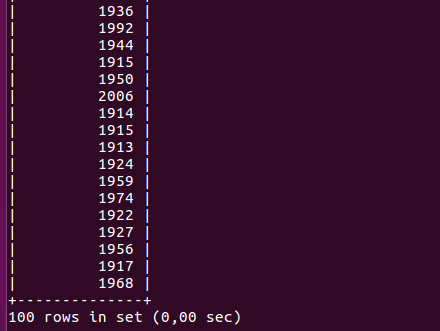
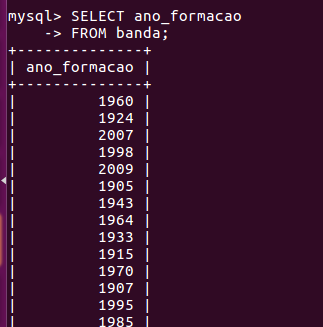
1. **Consultas**

**1ª) Selecione o ano de formação de todas as bandas – Seleção e projeção**

SELECT ano\_formacao

FROM banda;

* Tempo médio de execução em segundos: 0.00020555



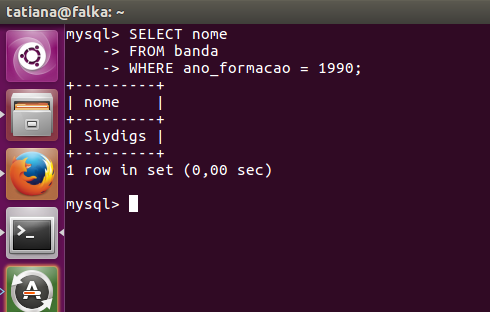
**2ª) Selecione o nome das bandas que se formaram em 1990 – Seleção e projeção**

SELECT nome

FROM banda

WHERE ano\_formacao = 1990;

* Tempo médio de execução em segundos: 0.00023738



**3ª) Selecione o nome dos álbuns e a primeira faixa de cada um. – Junção de duas relações**

SELECT a.nome, f.nome

FROM album AS a INNER JOIN faixa AS f ON a.id = f.album\_id

WHERE f.numero\_ordem = 1;

* Tempo médio de execução em segundos: 0,00376646

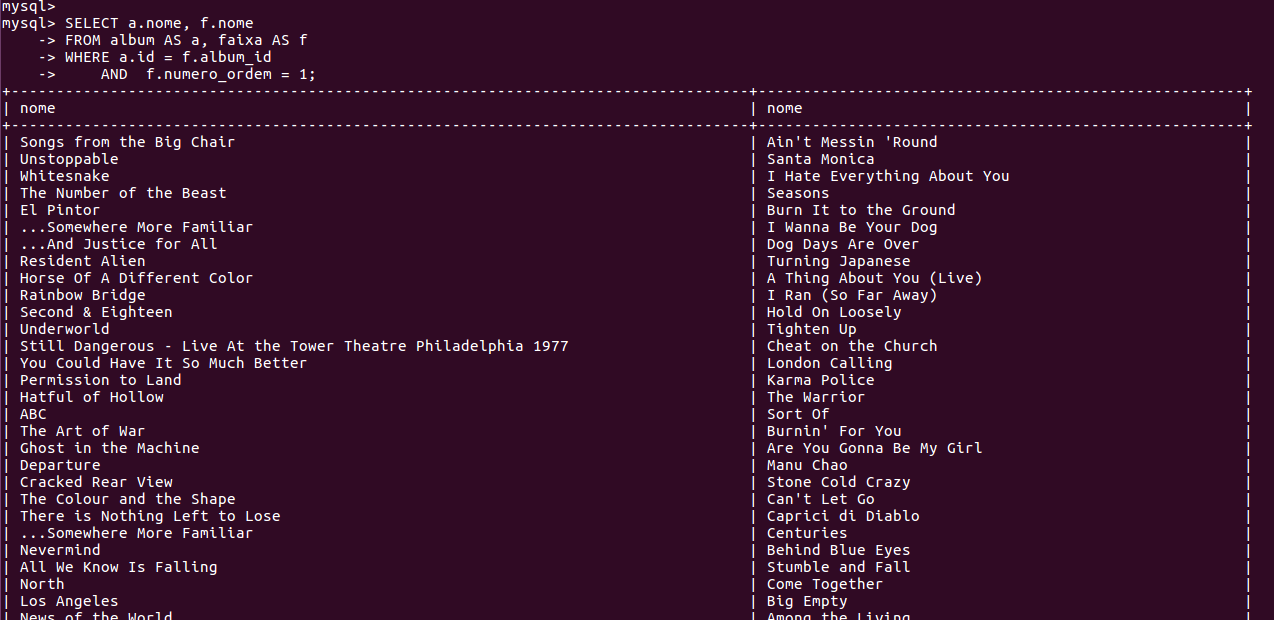
SELECT a.nome, f.nome

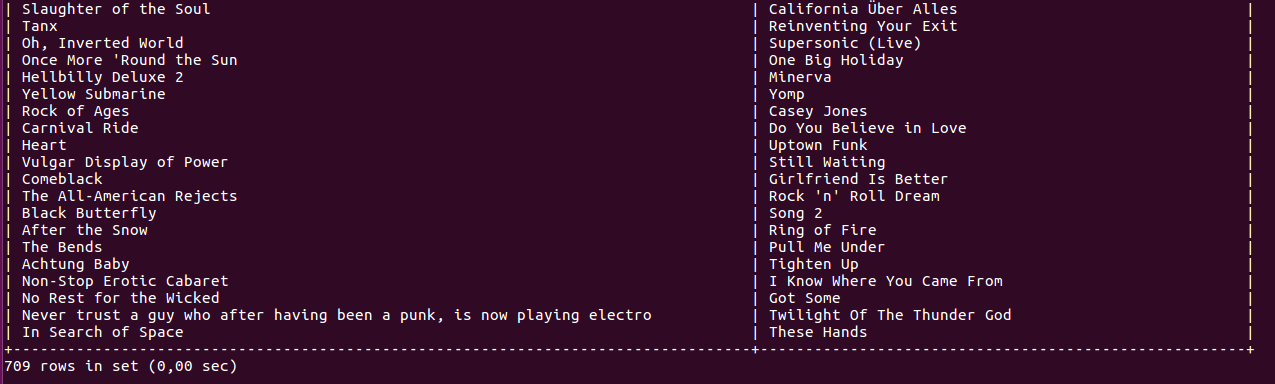
FROM album AS a, faixa AS f

WHERE a.id = f.album\_id

AND f.numero\_ordem = 1;

* Tempo médio de execução em segundos: 0,003229257
* Diferença de tempos: -14%





**4ª) Selecione a nacionalidade dos artistas que tocam bateria. – Junção de duas relações**

SELECT a.nacionalidade

FROM artista AS a, toca AS t

WHERE a.id = t.artista\_id

AND t.instrumento = 'bateria';

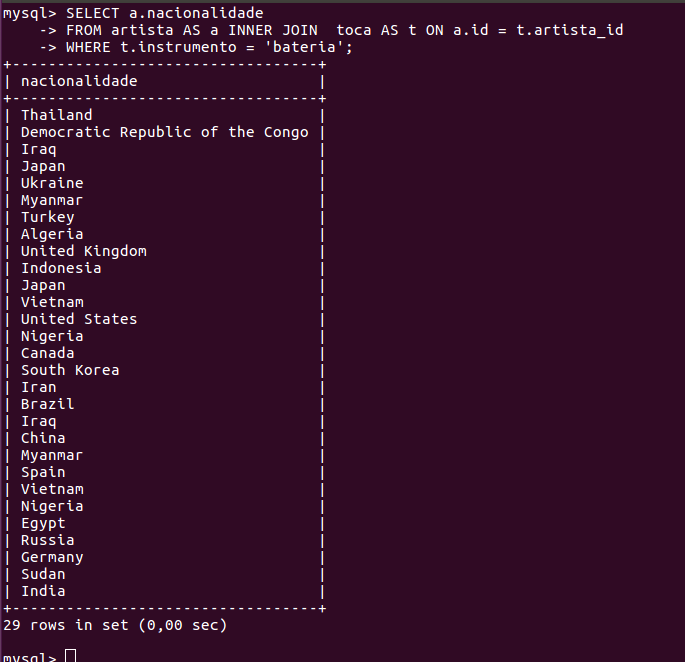
* Tempo médio de execução em segundos: 0,00033850

SELECT a.nacionalidade

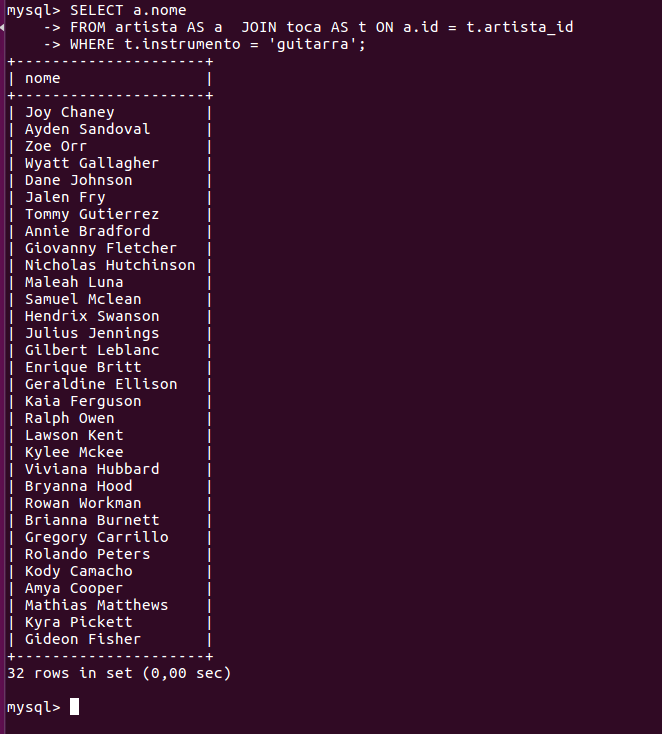
FROM artista AS a INNER JOIN toca AS t ON a.id = t.artista\_id

WHERE t.instrumento = 'bateria';

* Tempo médio de execução em segundos: 0,004098236
* Diferença de tempos: +1110%



**5ª) Selecione o nome dos artistas que tocam guitarra. – Junção de duas relações**



SELECT a.nome

FROM artista AS a JOIN toca AS t ON a.id = t.artista\_id

WHERE t.instrumento = 'guitarra';

* Tempo médio de execução em segundos: 0,00030971

SELECT a.nome

FROM artista AS a, toca AS t

WHERE a.id = t.artista\_id

AND t.instrumento = 'guitarra';

* Tempo médio de execução em segundos: 0,000583265
* Diferença de tempos: +88,33%

**6ª) Selecione o nome dos álbuns que possuem faixas do gênero rock. – Junção de três ou mais relações**

SELECT DISTINCT a.nome

FROM album AS a

INNER JOIN faixa AS f ON a.id = f.album\_id

INNER JOIN genero\_faixa AS gf ON f.id = gf.faixa\_id

WHERE gf.genero = ‘rock’;

* Tempo médio de execução em segundos: 0,00418917

SELECT DISTINCT a.nome

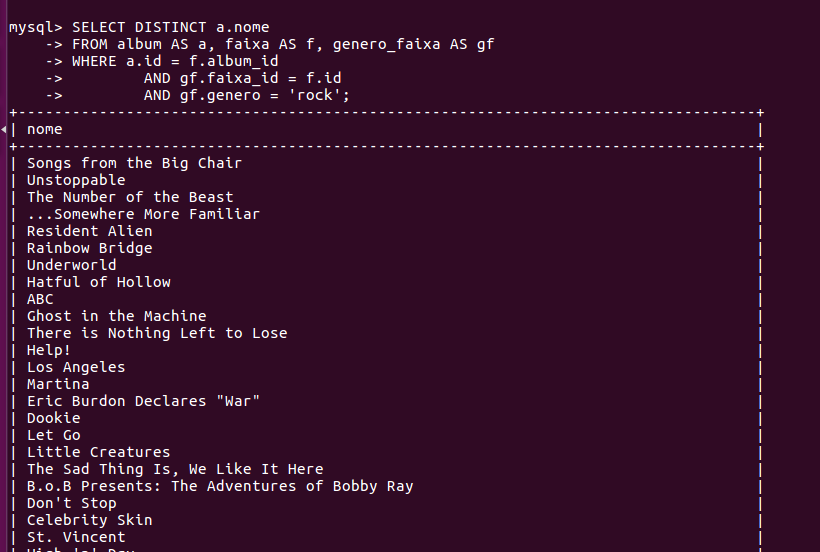
FROM album AS a, faixa AS f, genero\_faixa AS gf

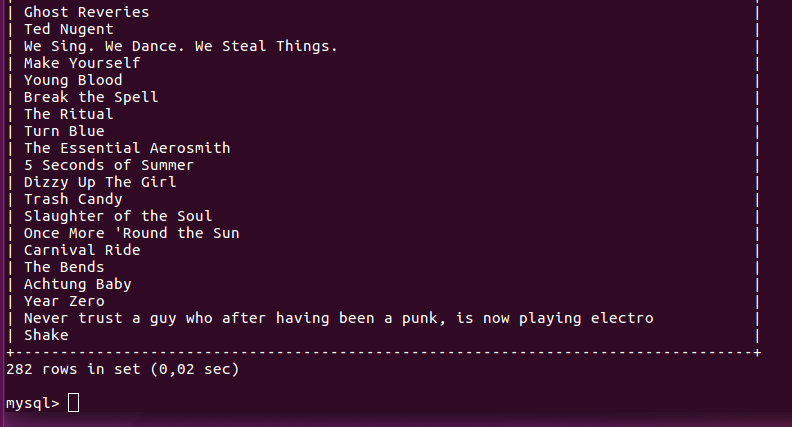
WHERE a.id = f.album\_id

AND gf.faixa\_id = f.id

AND gf.genero = 'rock';

* Tempo médio de execução em segundos: 0,014734159
* Diferença dos tempos: +251,72%





**7ª) Selecione o nome das faixas interpretadas por uma banda. – Junção de três ou mais relações**

SELECT f.nome

FROM faixa AS f

INNER JOIN interprete\_faixa AS inf ON f.id = inf.faixa\_id

INNER JOIN interprete AS i ON inf.interprete\_id = i.id

WHERE i.artista\_id IS NULL AND i.banda\_id IS NOT NULL;

* Tempo médio de execução em segundos: 0,00761200

SELECT f.nome

FROM faixa AS f, interprete\_faixa AS inf, interprete AS i

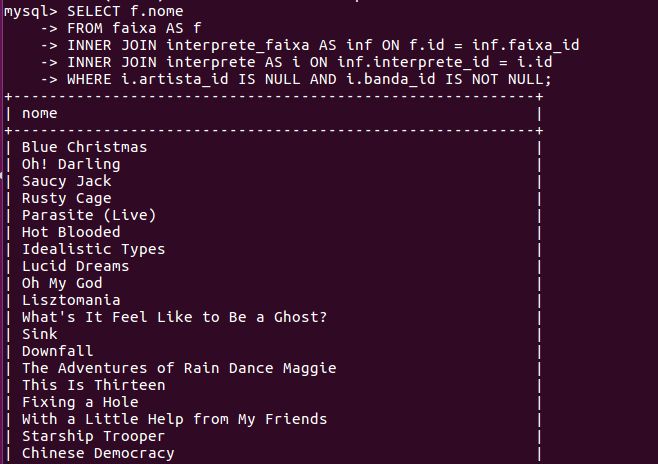
WHERE f.id = inf.faixa\_id

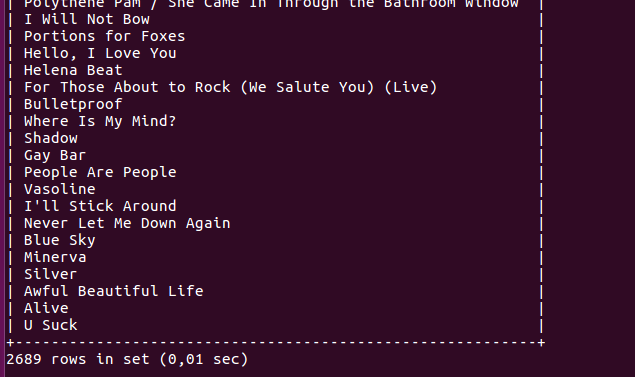
AND inf.interprete\_id = i.id

AND i.artista\_id IS NULL

AND i.banda\_id IS NOT NULL;

* Tempo médio de execução em segundos: 0,00414325
* Diferença de tempos: +83,72%





**8ª) Selecione os nomes das faixas que estão nos álbuns do artista Greyson Boone – Junção de três ou mais relações**

SELECT f.nome

FROM faixa AS f

INNER JOIN album AS a ON f.album\_id = a.id

INNER JOIN interprete AS i ON a.interprete\_id = i.id

INNER JOIN artista AS ar ON i.artista\_id = ar.id

WHERE ar.nome = 'Greyson Boone';

* Tempo médio de execução em segundos: 0,00761200

SELECT f.nome

FROM faixa AS f, album AS a, interprete AS i , artista AS ar

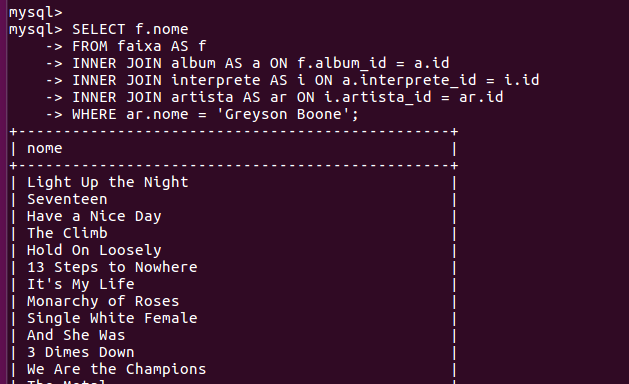
WHERE f.album\_id = a.id

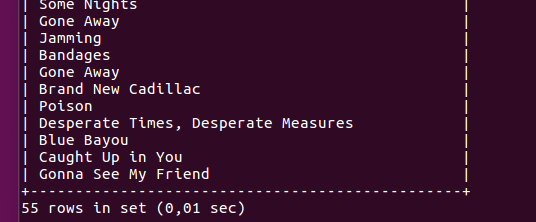
AND a.interprete\_id = i.id

AND i.artista\_id = ar.id

AND ar.nome = 'Greyson Boone';

* Tempo médio de execução em segundos:
* Diferença de tempos: -68,49%





**9ª)** **Selecione o nome de cada álbum e a quantidade de faixas nele presentes ordenado pela quantidade crescentemente. – Funções de agregação**

SELECT a.nome,

COUNT(\*) AS qtde

FROM faixa AS f

INNER JOIN album a ON f.album\_id = a.id

GROUP BY f.album\_id

ORDER BY qtde ASC;

* Tempo médio de execução em segundos: 0,00599888

SELECT a.nome,

COUNT(\*) AS qtde

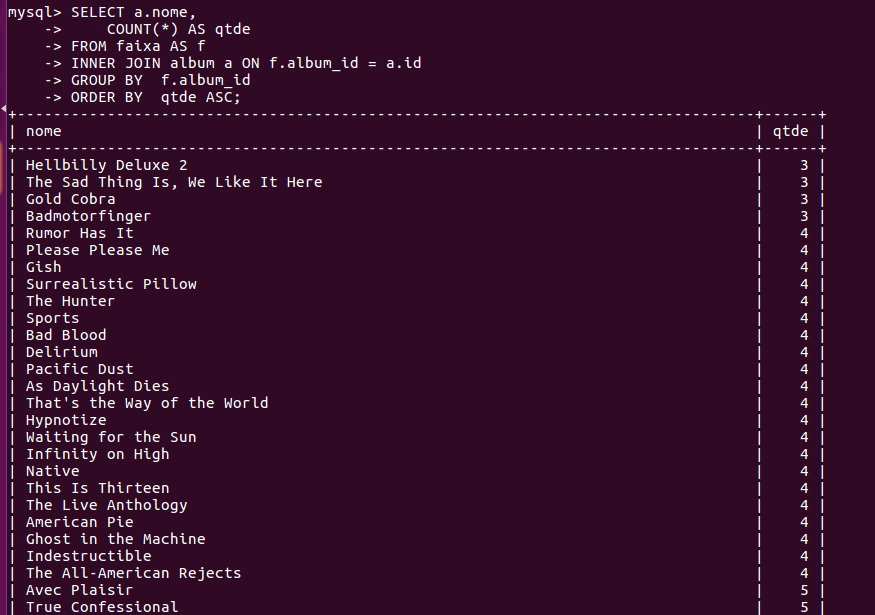
FROM faixa AS f, album AS a

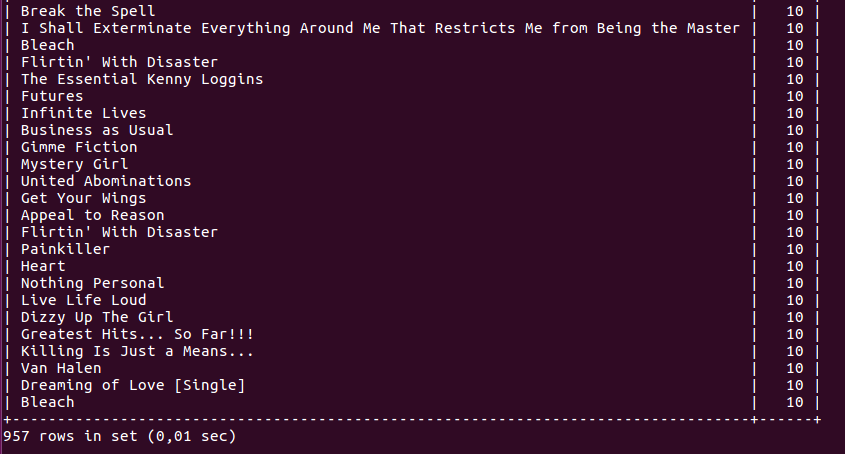
WHERE f.album\_id = a.id

GROUP BY f.album\_id

ORDER BY qtde ASC;

* Tempo médio de execução em segundos: 0,008774465
* Diferença de tempos: +46,27%





**10ª) Selecione todos os álbuns que possuem a maior quantidade de faixas. – Funções de agregação**

SELECT a.nome,

COUNT(\*) AS qtde

FROM faixa AS f

INNER JOIN album AS a ON f.album\_id = a.id

GROUP BY f.album\_id

HAVING qtde =

(SELECT MAX(qtde) FROM

(SELECT COUNT(\*) AS qtde

FROM faixa AS f

INNER JOIN album AS a ON f.album\_id = a.id

GROUP BY f.album\_id) AS qtdes); ]

* Tempo médio de execução em segundos: 0,01014000

SELECT a.nome,

COUNT(\*) AS qtde

FROM faixa AS f

INNER JOIN album AS a ON f.album\_id = a.id

GROUP BY f.album\_id

HAVING qtde =

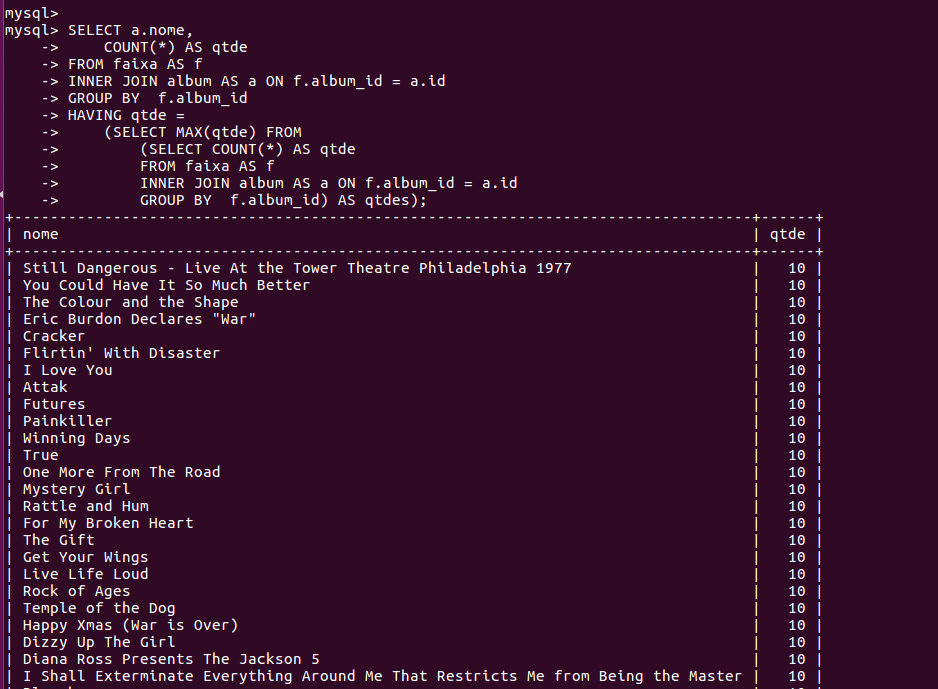
(SELECT COUNT(\*) AS qtde

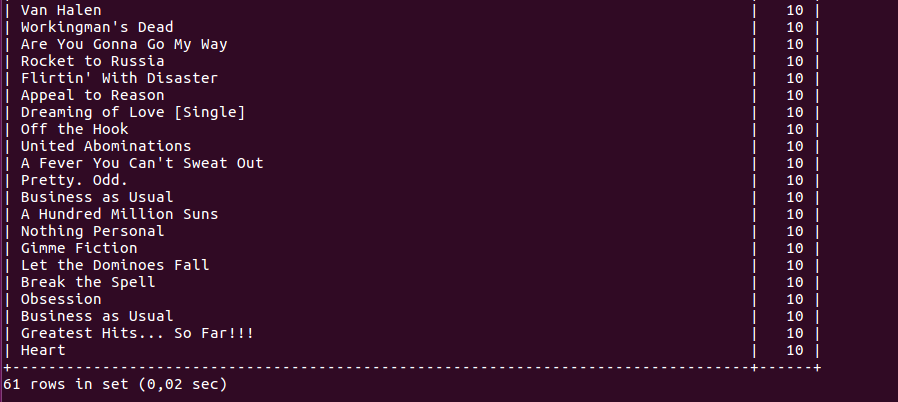
FROM faixa AS f

INNER JOIN album AS a ON f.album\_id = a.id

GROUP BY f.album\_id ORDER BY qtde DESC LIMIT 1);

* Tempo médio de execução em segundos: 0,012522348
* Diferença de tempos: +23,49%





1. **Análise**

Durante a realização dos testes e a apuração dos resultados, podemos notar que o funcionamento do banco de dados está otimizado, levando-se em conta que na maior consulta possível do banco foram computados 12 mil resultados, indicando assim que o banco possui um volume considerável de dados para que possam ser realizadas melhorias em sua estrutura para agilizar as consultas.

Inferimos que o planejamento da implementação dos índices e a preocupação com as relações lógicas, incluindo também os checks para poder evitar problemas estruturais. Com relação às assertions, não foi possível a implementação destas uma vez que o MySQL não possui um suporte para tal.

Podemos notar também uma diferença relevante de tempo de execução entre as consultas realizadas com os operadores JOIN e as cláusulas no WHERE. Em média as consultas com JOIN demoraram cerca de 80% menos tempo para retornarem o resultado do que as que utilizamos o WHERE e produto cartesiano das tabelas.

Conforme visto em sala de aula o operador WHERE realiza o produto cartesiano entre as tabelas selecionadas, o que pode influenciar muito no tempo de execução, uma vez que o JOIN seleciona as condições dadas diretamente na tabela selecionada. Por isto, acreditamos que a opção de utilizar o JOIN, além de retornar a consulta com maior velocidade, também utiliza um espaço de memória menor no processador.

Por fim, também podemos analisar com os resultados dos tempos obtidos que as consultas que envolviam muitas tabelas demoraram mais tempo para serem processadas. Isto ocorre pois a cada operação feita com duas colunas ocupa um espaço de tempo. Se houverem muitas tabelas, consequentemente haverá maior número de operações a serem realizadas, aumentando assim o tempo de consulta das mesmas.

1. **Conclusão**

Com a aplicação prática do processo de construção, consulta e análise em um banco de dados, entendemos melhor a importância e também os desafios da implementação correta e normalizada do mesmo. Houveram algumas dificuldades como por exemplo a de conseguirmos os dados. Ao escolher o tema, cremos que haveria um banco de fácil acesso com valores reais mas tendo em vista a dificuldade de achar um com entradas significativas, optou-se por um gerador.

As tarefas foram sendo feitas a medida da necessidade que encontramos para desenvolver de forma otimizada o nosso banco. Danilo foi responsável pela criação de um script para inserção de dados; Sonia**,** Tatiana e Samuel fizeram as consultas no banco criado e tiraram as imagens de verificação; Danilo, Sonia, Samuel e Tatiana implementaram os comandos de consultas ao banco; Tatiana,Sonia, Samuel, Danilo fizeram as análises de tempo em base de pelo menos 5 consultas. Todos participaram na implementação e verificação de qualidade desde projeto, tal como do relatório.