
MySQL - SQL

Banco de Dados

Criando Banco de Dados

Cria um banco de dados com nome <nome>:

- `create database if not exists <Nome>;`

Especifica o banco de dados a ser utilizado:

- `use <Nome>`

Apaga o banco de dados:

- `drop database if exists <Nome>`

Criando tabelas

```
create table <Nome> (  
    atributo1 Tipo Constraints,  
    atributo2 Tipo Constraints,  
    atributo3 Tipo Constraints,  
    ...  
);
```

```
create table if not exists Usuarios  
(  
    Nome VARCHAR(10) NOT NULL,  
    Data_nasc DATE,  
    cpf CHAR(12) NOT NULL,  
    PRIMARY KEY(cpf)  
);
```

Apagar tabelas

Apagar tabela:

- `drop table if exists <Nome>;`

Alterar tabelas - Adicionar coluna

Adicionar coluna:

- **alter table** <NomeTabela> **add column** <Nome> <Tipo>;

Adicionar em posição específica:

- **alter table** <NomeTabela> **add column** <Nome> <Tipo> **first**;
- **alter table** <NomeTabela> **add column** <Nome> <Tipo>
after <OutraColuna>;

Alterar tabelas - Remover/alterar coluna

Remover coluna:

- **alter table** <NomeTabela> **drop column** <Nome>;

Alterar coluna:

- **alter table** <NomeTabela> **modify column** <Nome> <Tipo> <Constraints>;

Alterar tabelas - Renomear coluna/tabela

Renomear coluna:

- **alter table** <NomeTabela> **change column**
 <Nome> <NovoNome> <NovoTipo> <NovasConstraints> ;

Renomear tabela:

- **alter table** <NomeTabela> **rename to** <NovoNome>;

Chaves estrangeiras - Criação da tabela

```
create table <Nome> (  
    atributo1 Tipo Constraints,  
    atributoFk Tipo Constraints,  
    FOREIGN KEY(atributoFk)  
        REFERENCES  
        <NomeTabela>(<NomePk>)  
);
```

```
create table if not exists Usuarios (  
    Nome VARCHAR(10) NOT NULL,  
    Fk int,  
    cpf CHAR(12) NOT NULL,  
    PRIMARY KEY(cpf),  
    FOREIGN KEY(Fk)  
        REFERENCES  
        <NomeTabela>(<NomePk>)  
);
```

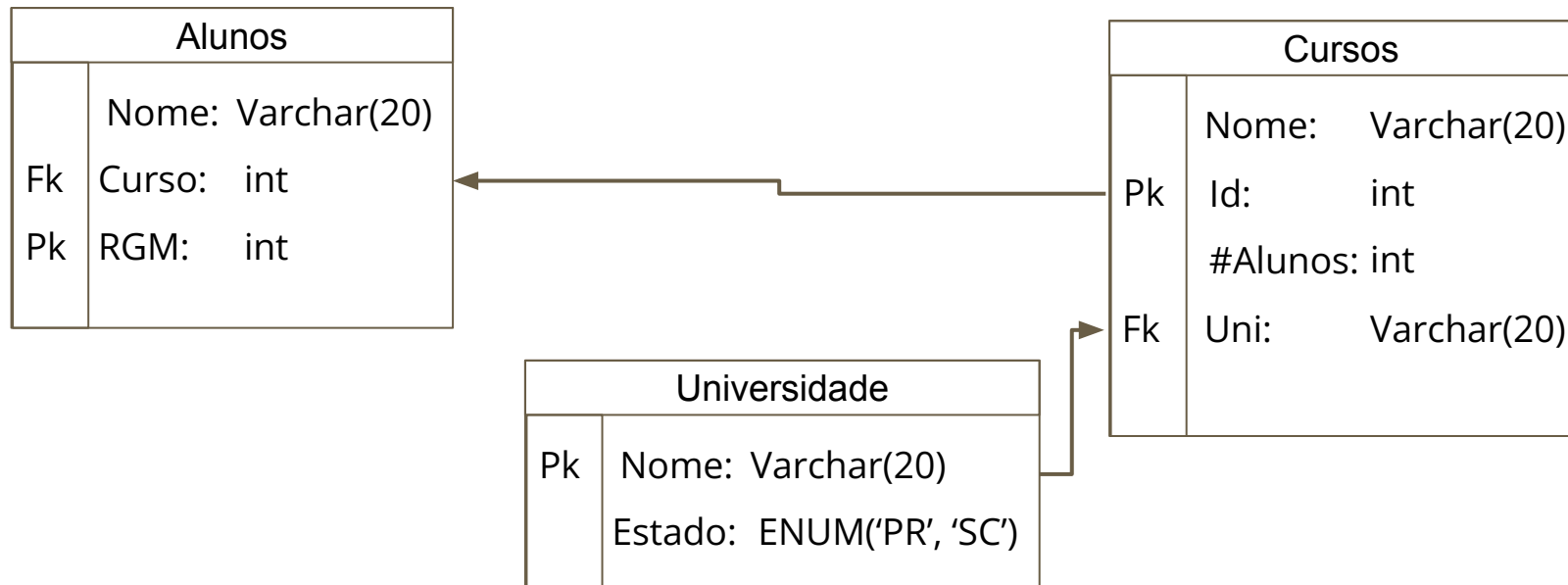

Chaves estrangeiras - Edição da tabela

Adicionar em tabela já criada:

- **alter table** <NomeTabela> **add column** <NomeFk> <TipoPk>;
- **alter table** <NomeTabela> **add foreign key** (<NomeFk>) **references** <NomeOutraTabela>(<NomePk>;

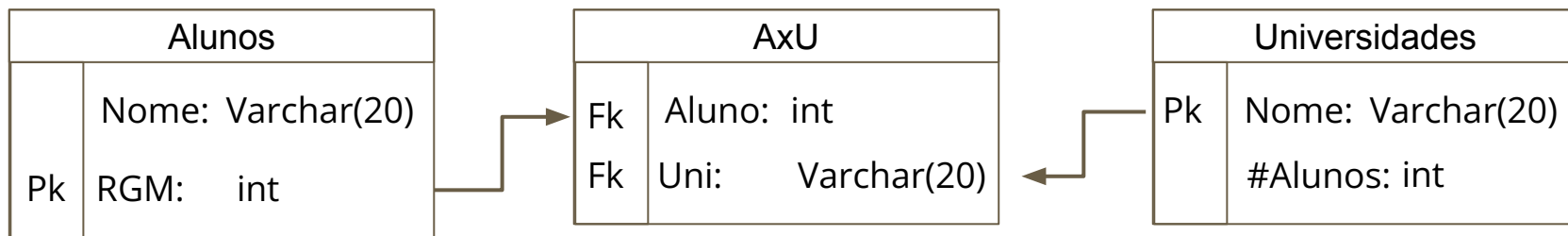
Exemplo

Converta o seguinte modelo lógico para seu modelo físico correspondente:



Exercício

Converta o seguinte modelo lógico para seu modelo físico correspondente:



Inserir dados

Inserir tuplas

Inserir tupla:

- **insert into** Usuarios (Nome, Data_nasc, cpf)
values ('<Nome>', '<Data_nasc>', '<cpf>');

Ex:

- **insert into** Usuarios (Nome, Data_nasc, cpf)
values ('Marcos Vitor', '2003-02-27', '099.546.849-7');

Variações na inserção (1)

Podemos não definir campos que não são NOT NULL:

- **insert into Usuarios (Nome, cpf)**
values ('<Nome>', '<cpf>');

Podemos deixar o valor DEFAULT:

- **insert into Usuarios (Nome, Data_nasc, cpf)**
values ('<Nome>', DEFAULT, '<cpf>');

Variações na inserção (2)

Se estivermos passando todos os atributos EM ORDEM:

- **insert into Usuarios values ('<Nome>', '<Data_nasc>', '<cpf>');**

Podemos adicionar várias tuplas de uma só vez:

- **insert into Usuarios (Nome, Data_nasc, cpf)**
values ('<Nome>', DEFAULT, '<cpf>'),
('<Nome 2>', '<Data_nasc 2>', '<cpf 2>'),
('<Nome 3>', '<Data_nasc 3>', '<cpf 3>');

Exercício

Altere a tabela alunos do exercício anterior e insira os dados dos seguintes usuários:

Alunos	
Pk	Nome: Varchar(20) RGM: int Data_nasc: Date Filme_pref: varchar(50)

Dados para inserir:

(João Pedro, 2, 04/05/2007, Rocky)

(Marco Antonio, 8, 22/07/1985, Matrix)

(Luiz Antonio, 1, 04/10/1900, As Branquelas)

(Ana Clara Stefan, 7, 14/12/1995, As Branquelas)

Para verificar: `Select * from Alunos;`

SELECT

Select → Projeção π

Selecionar todas as colunas:

- **Select** * **from** Alunos;

Selecionar as colunas Nome e Idade:

- **Select** Nome, Data_nasc **from** Alunos;

Nome	RGM	Data_nasc	Filme_pref
João Pedro	2	2007-05-04	Rocky
Marco Antonio	8	1985-07-22	Matrix
Luiz Antonio	1	1900-10-04	As Branquelas
Ana Clara Stefan	7	1995-12-14	As Branquelas

Select → Projeção π

Selecionar todas as colunas:

- **Select** * **from** Alunos;

Selecionar as colunas Nome e Idade:

- **Select** Nome, Data_nasc **from** Alunos;

π Nome, Data_nasc

Where → Seleção

σ (Tabela)
Data_nasc > '1950-12-04'

Nome	RGM	Data_nasc	Filme_pref
João Pedro	2	2007-05-04	Rocky
Marco Antonio	8	1985-07-22	Matrix
Luiz Antonio	1	1900-10-04	As Branquelas
Ana Clara Stefan	7	1995-12-14	As Branquelas

```
SELECT * FROM Alunos WHERE Data_nasc > '1950-12-04';
```

Where → Seleção

σ

(Tabela)

Data_nasc > '1950-12-04'

SELECT * FROM Alunos WHERE Data_nasc > '1950-12-04';

Nome	RGM	Data_nasc	Filme_pref
João Pedro	2	2007-05-04	Rocky
Marco Antonio	8	1985-07-22	Matrix
Luiz Antonio	1	1900-10-04	As Branquelas
Ana Clara Stefan	7	1995-12-14	As Branquelas

Select em apenas algumas linhas $\rightarrow \pi (\sigma ())$

Linhas com determinada condição:

- **Select** * **from** Alunos **WHERE** <condição>;

Ex:

- **Select** * **from** Alunos **WHERE** Nome = 'João Pedro';
- **Select** Nome **from** Alunos **WHERE** Data_nasc > '1950-12-04';

Operadores de comparação

WHERE

Operadores Aritméticos

=	Igualdade
!=	Diferente
<	Menor
<=	Menor ou igual
>	Maior
>=	Maior ou igual

Operadores Lógicos

AND
OR
NOT

Select em apenas algumas linhas

Ex:

- **Select** * **from** Alunos
WHERE Nome = 'João Pedro' **and** Data_nasc > '1950-12-04';
- **Select** * **from** Alunos
WHERE (Nome = 'João Pedro' **and** Data_nasc > '1950-12-04')
or Filme_pref = 'Rocky';

Definindo caracteres suportados

Padrão do BD:

- `create database <Nome>`
 `default character set utf8mb4`
 `default collate utf8mb4_general_ci;`
- `create database <Nome>`
 `default character set latin1`
 `default collate latin1_general_ci;`

Definindo caracteres suportados

Padrão de uma **tabela**:

```
create table if not exists Usuarios  
(  
    Nome VARCHAR(10) NOT NULL,  
    Data_nasc DATE,  
    cpf CHAR(12) NOT NULL,  
    PRIMARY KEY(cpf)  
) default charset = utf8mb4;
```

Padrão de um **atributo**:

```
create table if not exists Usuarios (  
    Nome VARCHAR(10)  
    charset utf8mb4 NOT NULL,  
    Data_nasc DATE,  
    cpf CHAR(12) NOT NULL,  
    PRIMARY KEY(cpf)  
);
```

Cópias de segurança

Para gerar um Backup

No Workbench:

Server → Data Export

The screenshot shows the 'MySQL Data Export' window. At the top, there's a 'MySQL Data Export' title bar with an 'Advanced Options...' button. Below the title bar, there are two tabs: 'Object Selection' (active) and 'Export Progress'. The 'Object Selection' tab contains several sections: 'Tables to Export' with a list of schemas (cadastro, cadastro_latim, cadastro_utf8, sys) and a 'Refresh' button; 'Objects to Export' with checkboxes for 'Dump Stored Procedures and Functions', 'Dump Events', and 'Dump Triggers'; and 'Export Options' with radio buttons for 'Export to Dump Project Folder' (selected) and 'Export to Self-Contained File', each with a file path input field. There are also checkboxes for 'Create Dump in a Single Transaction (self-contained file only)' and 'Include Create Schema'. At the bottom right, there is a 'Start Export' button. The bottom of the window has an 'Action Output' dropdown menu.

MySQL Data Export

Advanced Options...

Object Selection Export Progress

Tables to Export

Expo Schema

- ☐ cadastro
- ☐ cadastro_latim
- ☐ cadastro_utf8
- ☐ sys

Refresh

Expo Schema Objects

Dump Structure and Data ▾ Select Views Select Tables Unselect All

Objects to Export

☐ Dump Stored Procedures and Functions ☐ Dump Events ☐ Dump Triggers

Export Options

☒ Export to Dump Project Folder /home/war/snap/mysql-workbench-community/10/dumps/Dump20220817 ...

Each table will be exported into a separate file. This allows a selective restore, but may be slower.

☐ Export to Self-Contained File /home/war/snap/mysql-workbench-community/10/dumps/Dump20220817.sql ...

All selected database objects will be exported into a single, self-contained file.

☐ Create Dump in a Single Transaction (self-contained file only) ☐ Include Create Schema

Press [Start Export] to start...

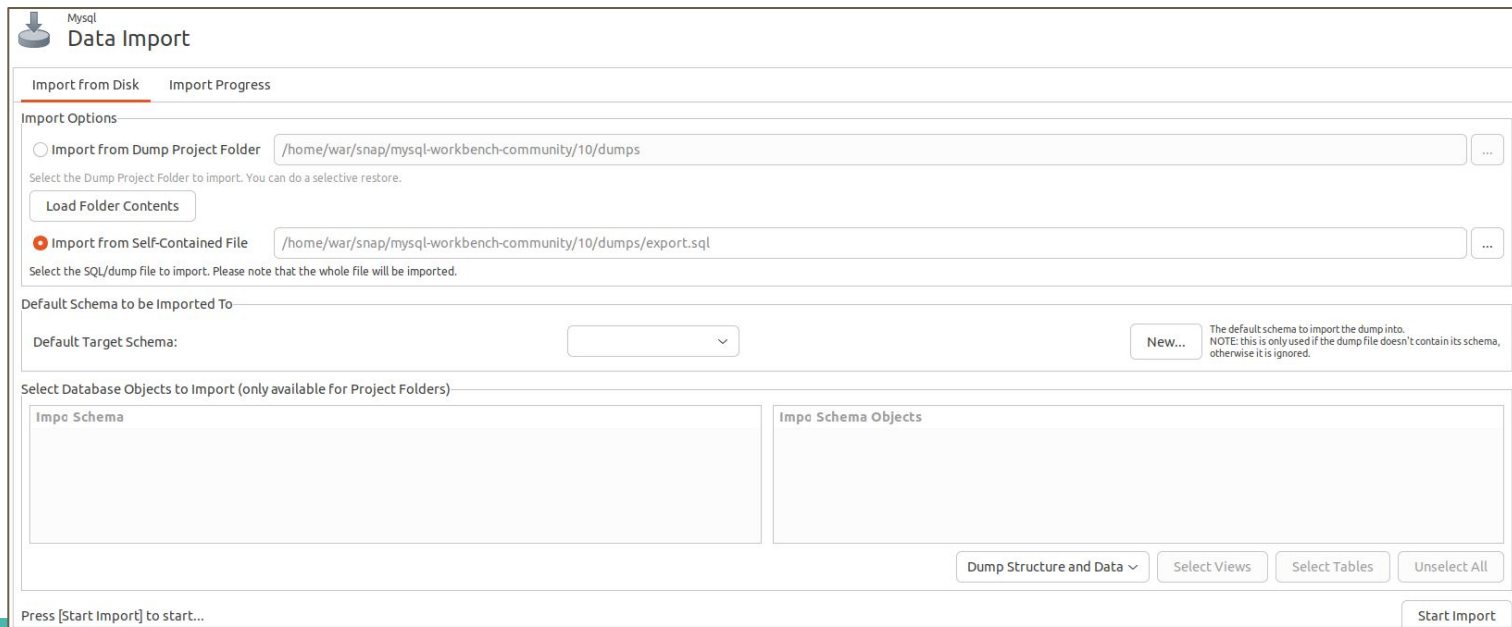
Start Export

Action Output ▾

Para recuperar com um Backup

No Workbench:

Server → Data Import



The screenshot shows the 'Data Import' dialog box in MySQL Workbench. The 'Import from Disk' tab is active. Under 'Import Options', the 'Import from Self-Contained File' radio button is selected, with the file path '/home/war/snap/mysql-workbench-community/10/dumps/export.sql' entered. Below this, the 'Default Schema to be Imported To' section shows a dropdown menu for 'Default Target Schema' and a 'New...' button. A note explains that the default schema is used if the dump file doesn't contain its schema. The 'Select Database Objects to Import' section is currently empty. At the bottom, there are buttons for 'Dump Structure and Data', 'Select Views', 'Select Tables', 'Unselect All', and a 'Start Import' button.

MySQL Data Import

Import from Disk Import Progress

Import Options

☐ Import from Dump Project Folder /home/war/snap/mysql-workbench-community/10/dumps ...

Select the Dump Project Folder to import. You can do a selective restore.

Load Folder Contents

☒ Import from Self-Contained File /home/war/snap/mysql-workbench-community/10/dumps/export.sql ...

Select the SQL/dump file to import. Please note that the whole file will be imported.

Default Schema to be Imported To

Default Target Schema: ...

New... The default schema to import the dump into.
NOTE: this is only used if the dump file doesn't contain its schema, otherwise it is ignored.

Select Database Objects to Import (only available for Project Folders)

Import Schema

Import Schema Objects

Dump Structure and Data ▾ Select Views Select Tables Unselect All

Press [Start Import] to start... Start Import

Exemplos de exercícios

1. Carregue o banco de dados “World.sql”
2. Liste todas as tabelas do banco de dados
3. Use o comando SELECT para listar todos os países
4. Liste apenas os nomes de todas as línguas
5. Use o comando SELECT com WHERE para listar apenas os nomes das línguas oficiais
 - Obtenha um modelo lógico do banco de dados “World.sql”

Exercícios

1. Carregue o banco de dados "World.sql"
2. Liste todas as tabelas do banco de dados
3. Use o comando SELECT para listar todos os países
4. Liste apenas os nomes de todas as cidades
5. Use o comando SELECT com WHERE para listar apenas os nomes das cidades com mais de 1M de habitantes
6. Liste apenas os nomes de todas as línguas oficiais, faladas por mais de com mais de 30 % dos habitantes e menos de 80 % deles

Alterar dados

Alterar tuplas

Alterar registro específico:

- **update** <Tabela> **set** <Atributo> = <Valor> **where** <Condição>;
- **update** <Tabela>
 set <Atributo 1> = <Valor 1>, <Atributo 2> = <Valor 2>
 where <Condição>;

Alterar vários registros:

- **LIMIT**
- Desabilitar safe update

Alterar tuplas

Alterar registro específico:

- **update** <Tabela> **set** <Atributo> = <Valor> **where** <Condição>;
- **update** <Tabela>
 set <Atributo 1> = <Valor 1>, <Atributo 2> = <Valor 2>
 where <Condição> **LIMIT** 10;

Alterar vários registros:

- **LIMIT**
- Desabilitar safe update

Apagar tuplas

Para apagar um registro:

```
delete from <Tabela> where <Coluna> = <Valor>;
```

```
delete from <Tabela> where <Coluna> = <Valor> LIMIT 1;
```

- **Limit** → Indica o número máximo de registros onde aplicar a ação

Exemplo

Carregue a base de dados world.sql

1. Liste todos os países existentes
2. Altere o nome do primeiro país para 'Alterado'
3. Altere o nome dos dois primeiros países da Antartica para 'NovoNome'
4. Se um país da 'Asia' tiver expectativa de vida menor que 80, altere a expectativa para 30 anos

Exercícios

Carregue a base de dados world.sql

1. Liste todos os países existentes
2. Altere o nome de qualquer país, com área de superfície maior que 500, para 'Grande país'
3. Se um país da Europa tiver expectativa de vida maior 75 anos, altere a expectativa para 85 anos
4. Apague o registro dos dois primeiros países do Caribe

Tabelas temporárias

Criar tabela temporária

Para criar uma tabela temporária:

- `create temporary table ...`

Ao desconectar-se do servidor, as tabelas temporárias são automaticamente descartadas.

Podem ser criadas com o mesmo nome de uma outra *tabela permanente* e preexistente

Criar tabela temporária

Útil para criar e testar queries, usando o nome de uma tabela existente, sem afetá-la

Vários clientes podem abrir tabelas temporárias com o mesmo nome! Cada qual na sua sessão

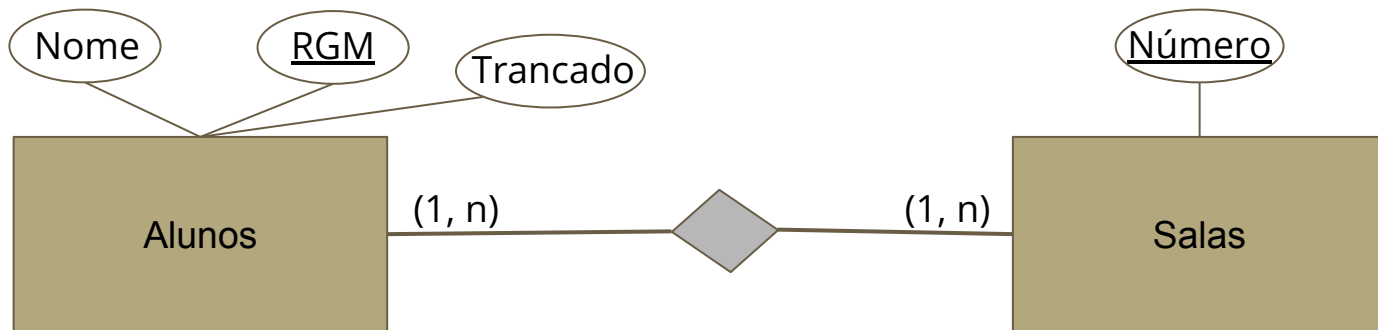
Exemplo

1. Crie um **backup** do seu Banco de Dados
2. Crie uma **tabela temporária** Usuários com os dados da tabela permanente Usuários
3. **Apague** os dados da tabela temporária
4. Imprima o conteúdo da tabela temporária
5. Apague a tabela temporária
6. Imprima novamente os dados dos Usuários

Acessando
relacionamentos

Join

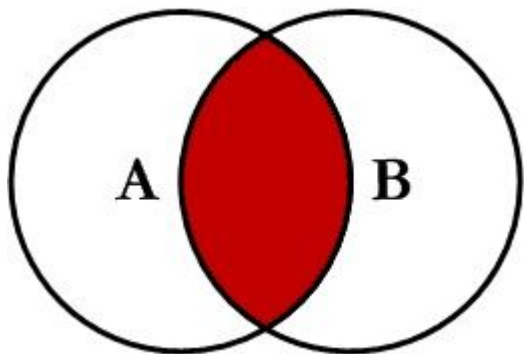
Podemos acessar dados relacionados pelas chaves estrangeiras utilizando
JOINS



Joins

Inner Join

Mantém apenas tuplas que tem relação com a outra tabela



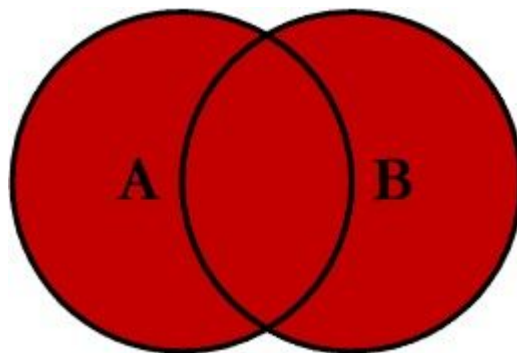
Nome	RGM	Trancado	Número
Carlos	2	Não	30
Marta	4	Não	30
Carlos	2	Não	40

Joins

Nome	RGM	Trancado	Número
Carlos	2	Não	30
Marta	4	Não	30
Carlos	2	Não	40
Joana	7	Sim	NULL
NULL	NULL	NULL	10

Outer Join

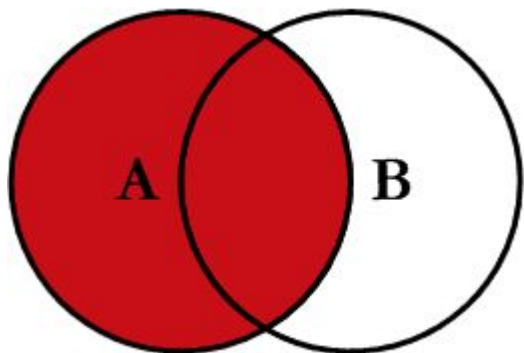
Apresenta mesmo as tuplas sem relação



Joins

Left Outer Join

Mantém todas as tuplas da esquerda junto às da direita com relação a elas



Nome	RGM	Trancado	Número
Carlos	2	Não	30
Marta	4	Não	30
Carlos	2	Não	40
Joana	7	Sim	NULL
NULL	NULL	NULL	10



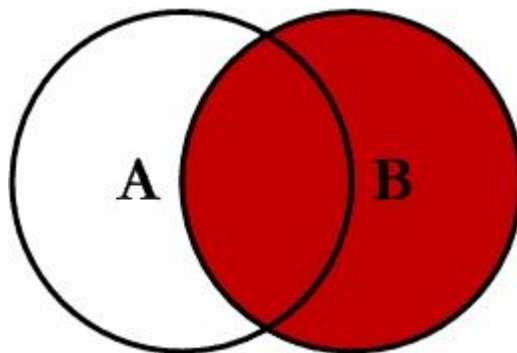
Joins

Nome	RGM	Trancado	Número
Carlos	2	Não	30
Marta	4	Não	30
Carlos	2	Não	40
Joana	7	Sim	NULL
NULL	NULL	NULL	10



Right Outer Join

Mantém todas as tuplas da direita junto às da esquerda com relação a elas



Inner Join

- `Select * from <Tabela> JOIN <Tabela2> ON <Relação entre as tabelas>;`

Ex:

```
Select * from country JOIN city ON city.CountryCode = country.Code;
```

```
Select city.name, country.name
```

```
from country JOIN city
```

```
ON city.CountryCode = country.Code;
```


Outer Join

- `Select * from <Tabela> OUTER JOIN <Tabela2> ON <Relação entre as tabelas>;`

Ex:

```
Select * from country outer join city  
ON city.CountryCode = country.Code;
```

```
Select * from country outer join city  
ON city.CountryCode = country.Code;
```

Left/Right Outer Join

- `Select * from <Tabela> Left/Right OUTER JOIN <Tabela2> ON <Relação entre as tabelas>;`

Ex:

```
Select * from country left outer join city  
ON city.CountryCode = country.Code;
```

```
Select * from country right outer join city  
ON city.CountryCode = country.Code;
```

Exemplos

1. Apresente o Nome de cada cidade, o país dela e a expectativa de vida nesse país
2. Apresente o Nome de cada país juntamente à língua falada nele
3. Apresente o Nome de cada cidade, a língua falada e o nome do país, apenas dos países com uma expectativa de vida maior que 60 anos

Exercícios

1. Apresente o Nome de cada cidade, o continente dela e o ano de independência nesse país
2. Apresente uma lista de línguas juntamente a seus países de uso, se a língua apresentar mais de 50 % do total de falantes do país
3. Apresente o distrito de cada cidade, sua população e a população total de seu país

