



UNIVERSIDADE
LUSÓFONA

Base de Dados

SQL: Select, Insert, Update e Delete

2022/2023

Sumário

Preparação do Ambiente

- Docker
- Cliente

SQL

- Select
- Insert
- Update
- Delete

Preparação Ambiente de Trabalho

1. Instalação do docker

<https://www.docker.com/products/docker-desktop/>

2. Instalação DBeaver

<https://dbeaver.io/>


3. Iniciar Container exemplo

```
docker run --name mysqldb  
            --env MARIADB_ROOT_PASSWORD=password  
            -p 3306:3306  
            -d mariadb:latest
```

4. Efetuar ligação DBeaver a Base de Dados

GitHub

🔑 main ▾ aula03 / README.md

 jpaclcarneiro Update README.md

👤 1 contributor

☰ 98 lines (58 sloc) | 5 KB

aula03

1. Prepare o seu ambiente de trabalho
2. Comando Select
3. Comando Insert
4. Comando Update
5. Comando Delete

- Exercícios publicados no repositório git: <https://github.com/ULHT-BD/aula03>

A blue ribbon graphic with a 3D effect, featuring a dark blue shadow on the left side. The text "Comando SELECT" is centered on the ribbon in white.

Comando SELECT

SQL: SELECT

- O comando SELECT permite selecionar dados de uma tabela

SELECT

<atributo1>, <atributo2>, ...

FROM

<nome-tabela>;

estudante(eid, nome, login, idade)

eid	nome	idade	Login
15912	Mariza	19	Mariza@lem
15920	Rui	31	rui@li
16005	Mafalda	23	Mafalda@lem

- Obter atributos nome e idade da relação estudante

SELECT

nome, idade

FROM

estudante;

nome	idade
Mariza	19
Rui	31
Mafalda	23

SQL: SELECT *

- * (asterisco) significa todas as colunas/atributos
- SELECT * permite selecionar todos os atributos de uma relação

SELECT
*

FROM

<nome-tabela>;

estudante(eid, nome, login, idade)

eid	nome	idade	Login
15912	Mariza	19	Mariza@lem
15920	Rui	31	rui@li
16005	Mafalda	23	Mafalda@lem

- Obter todos os atributos de todos os estudantes

SELECT
*

FROM

estudante;

eid	nome	idade	Login
15912	Mariza	19	Mariza@lem
15920	Rui	31	rui@li
16005	Mafalda	23	Mafalda@lem

*** significa todos os atributos**

SQL: SELECT - WHERE

- Selecionar dados de uma tabela

SELECT

<atributo1>, <atributo2>, ...

FROM

<nome-tabela>

WHERE

<condição>;

A clausula **WHERE** especifica as condições que o resultado deve satisfazer

estudante(eid, nome, login, idade)

eid	nome	idade	Login
15912	Mariza	19	Mariza@lem
15920	Rui	31	rui@li
16005	Mafalda	23	Mafalda@lem

- Obter toda informação sobre estudante cujo nome é 'Mariza'

'Mariza'

SELECT

*

FROM

estudante

WHERE

nome='Mariza';

eid	nome	idade	Login
15912	Mariza	19	Mariza@lem

O resultado é o conjunto de tuplos para os quais a condição **nome = 'Mariza'** é verdadeira

SQL: SELECT - WHERE

- Selecionar dados de uma tabela

SELECT

<atributo1>, <atributo2>, ...

FROM

<nome-tabela>

WHERE

<condição>;

A clausula **WHERE** especifica as condições que o resultado deve satisfazer

estudante(eid, nome, login, idade)

eid	nome	idade	Login
15912	Mariza	19	Mariza@lem
15920	Rui	31	ru@li
16005	Mafalda	23	Mafalda@lem

- Obter toda informação sobre estudante cujo nome é 'Mariza'

'Mariza'

SELECT

nome, idade

FROM

estudante

WHERE

idade > 20;

nome	idade
Rui	31
Mafalda	23

O resultado é o conjunto de tuplos para os quais a condição **idade > 20** é verdadeira

A blue ribbon graphic with a folded end on the left side, containing the text "Comando INSERT" in white.

Comando INSERT

SQL: INSERT/UPDATE/DELETE

- Inserir um tuplo numa tabela
INSERT INTO <nome-tabela> **VALUES**

```
(  
    <atributo1>,  
    <atributo2>,  
    ...,  
    <atributoN>  
);
```

- Inserir na tabela estudante
INSERT INTO estudante VALUES (

```
    15912,  
    'Mariza',  
    19,  
    'Mariza@lem'  
);
```

estudante(eid, nome, login, idade)

eid	nome	idade	Login
15912	Mariza	19	Mariza@lem

Os valores respeitam a ordem de definição dos atributos no esquema:

```
CREATE TABLE estudante (  
    eid int,  
    nome varchar(50),  
    idade int,  
    login varchar(50),  
    primary key(eid));
```

SQL: INSERT/UPDATE/DELETE

- Inserir um tuplo numa tabela

INSERT INTO <nome-tabela>

(
 <nome-atributo1>,
 <nome-atributo2>,
 ...,
 <nome-atributoN>
)

VALUES

(
 <valor-atributo1>,
 <valor-atributo2>,
 ...,
 <valor-atributoN>
);

estudante(eid, nome, login, idade)

eid	nome	idade	Login
15912	Mariza	NULL	NULL

Os atributos a inserir são explicitamente indicados na operação de inserção:

- Inserir na tabela estudante

```
INSERT INTO estudante  
    (eid, nome)  
VALUES (  
    15912,  
    'Mariza'  
);
```

```
CREATE TABLE estudante (  
    eid int,  
    nome varchar(50),  
    idade int,  
    login varchar(50),  
    primary key(eid));
```

SQL: INSERT/UPDATE/DELETE

- Inserir vários tuplos numa tabela

INSERT INTO <nome-tabela> **VALUES**

(<tuplo1-a1>,<tuplo1-a2>, ..., <tuplo1-aN>),
(<tuplo2-a1>,<tuplo2-a2>, ..., <tuplo2-aN>),
...
(<tuploM-a1>,<tuploM-a2>, ..., <tuploM-aN>);

- Inserir na tabela estudante

```
INSERT INTO estudante VALUES  
(15912, 'Mariza', 19, 'Mariza@lem'),  
(15920, 'Rui', 31, 'Rui@li'),  
(16005, 'Mafalda', 23, 'Mafalda@lem');
```

Podemos inserir vários tuplos de uma só vez

estudante(eid, nome, login, idade)

eid	nome	idade	Login
15912	Mariza	19	Mariza@lem
15920	Rui	31	rui@li
16005	Mafalda	23	Mafalda@lem

Os valores respeitam a ordem de definição dos atributos no esquema:

```
CREATE TABLE estudante (  
    eid int,  
    nome varchar(50),  
    idade int,  
    login varchar(50),  
    primary key(eid));
```

SQL: INSERT/UPDATE/DELETE

- Inserir vários tuplos numa tabela

INSERT INTO <nome-tabela>

(<nome-atributo1>, <nome-atributo2>,
..., <nome-atributoN>)

VALUES

(<tuplo1-a1>,<tuplo1-a2>, ..., <tuplo1-aN>),
(<tuplo2-a1>,<tuplo2-a2>, ..., <tuplo2-aN>),
...
(<tuploM-a1>,<tuploM-a2>, ..., <tuploM-aN>);

- Inserir na tabela estudante

INSERT INTO estudante

(eid, nome)

VALUES

(15912, 'Mariza'),

(15920, 'Rui', 31),

(16005, 'Mafalda');

Podemos inserir vários tuplos de uma só vez

estudante(eid, nome, login, idade)

eid	nome	idade	Login
15912	Mariza	NULL	NULL
15920	Rui	NULL	NULL
16005	Mafalda	NULL	NULL

Os atributos a inserir são explicitamente indicados na operação de inserção:

CREATE TABLE estudante (
 eid int,
 nome varchar(50),
 idade int,
 login varchar(50),
 primary key(eid));

A blue ribbon graphic with a folded end on the left side, containing the text "Comando UPDATE" in white.

Comando UPDATE

SQL: INSERT/UPDATE/DELETE

- Alterar dados numa tabela

UPDATE <nome-tabela>

SET

<nome-atributo1> = <valor-atributo1>,

<nome-atributo2> = <valor-atributo2>,

...,

<nome-atributoN> = <valor-atributoN>

WHERE <condição>;

- Alterar idade da Mariza

UPDATE estudante SET idade=43

WHERE nome='Mariza';

- **Todos os estudantes “Mariza”
alterados!**

- UPDATE estudante SET idade=43
WHERE eid=15912;

estudante(eid, nome, login, idade)

eid	nome	idade	Login
15912	Mariza	43	Mariza@lem
15920	Rui	31	rui@li
16005	Mafalda	23	Mafalda@lem



Comando DELETE

SQL: INSERT/UPDATE/DELETE

- Apagar dados numa tabela

DELETE FROM <nome-tabela>
WHERE <condição>;

- Apagar a estudante Mariza
DELETE FROM estudante WHERE
nome='Mariza';

- DELETE FROM estudante WHERE
eid=15912;

estudante(eid, nome, login, idade)

eid	nome	idade	Login
15920	Rui	31	rui@li
16005	Mafalda	23	Mafalda@lem

DELETE FROM <nome-tabela>;

Apaga todos os tuplos!
(todos os tuplos respeitam condição)



Obrigado.