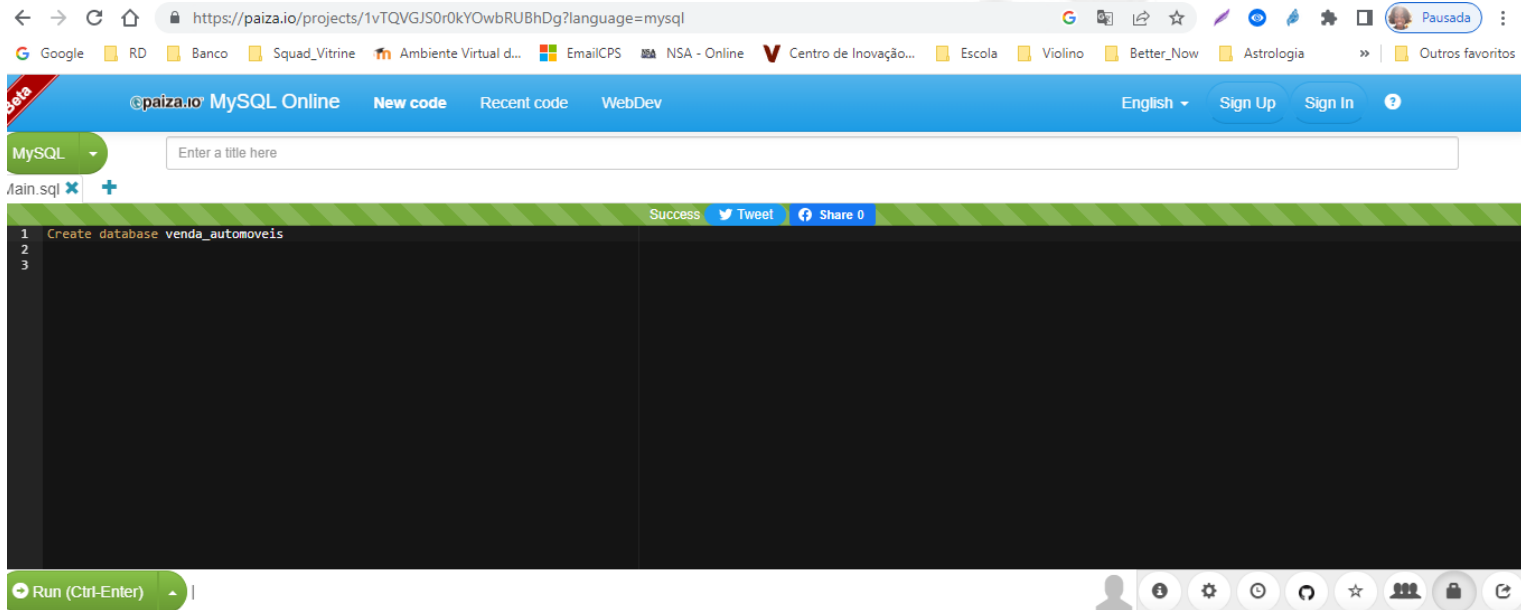


Banco de Dados

Criação de Tabelas

DDL (DATA DEFINITION LANGUAGE)

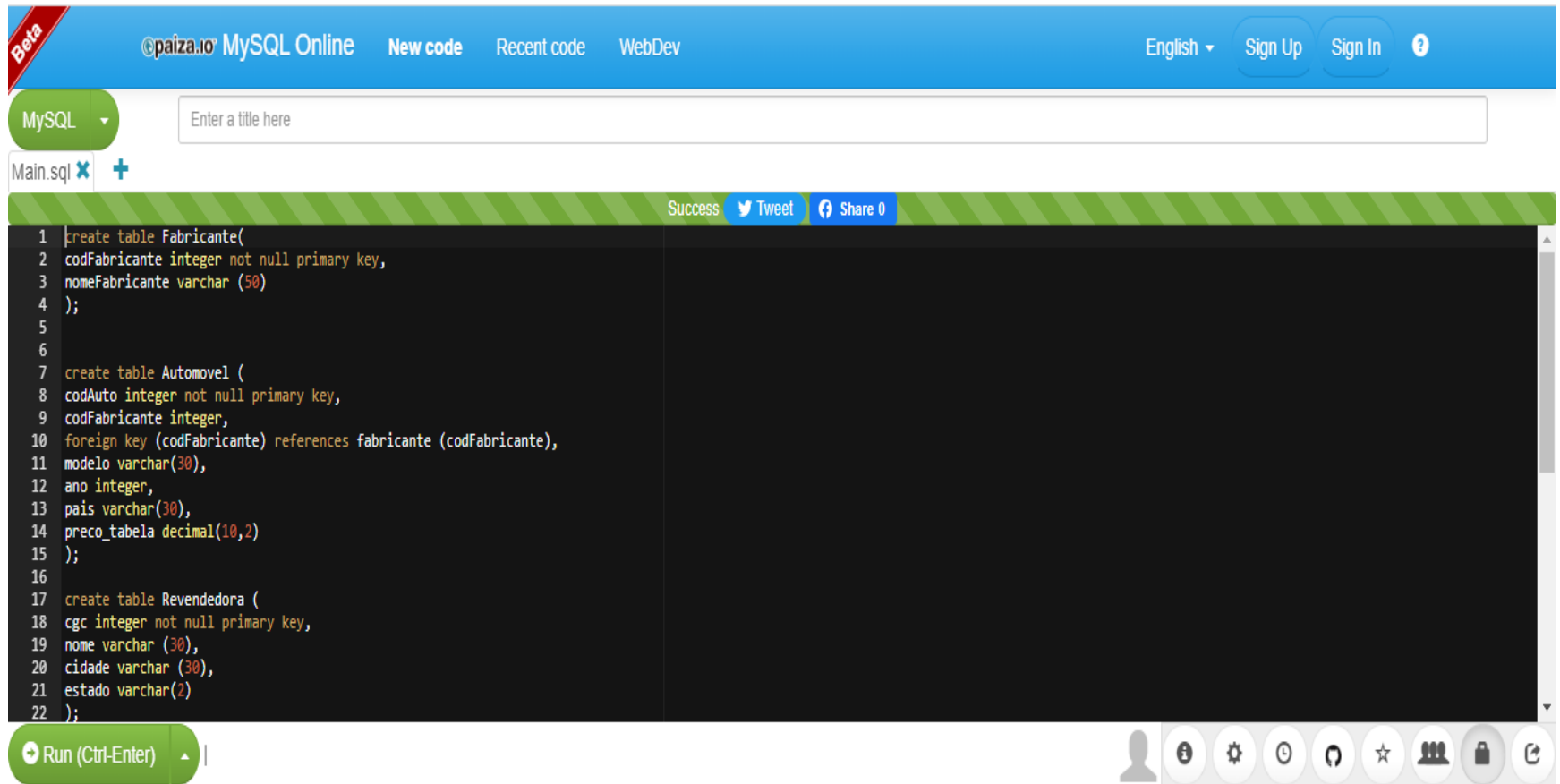
- Criando uma base dados.
- O primeiro passo para iniciar a criação de uma Base ou Banco de dados é criar uma DATABASE, para isso utilizamos o comando:
- **Create database [nome da base de dados]**



- Acima temos a imagem da criação de uma database no paiza io

DDL (DATA DEFINITION LANGUAGE)

- Criação de uma tabela.
- Em seguida podem então ser criadas as tabelas.



The screenshot displays the Paiza.io MySQL Online environment. The top navigation bar includes a 'Beta' badge, the 'paiza.io MySQL Online' logo, and links for 'New code', 'Recent code', and 'WebDev'. On the right, there are language and user options: 'English', 'Sign Up', 'Sign In', and a help icon. Below the navigation bar, a green tab labeled 'MySQL' is active, with a text input field 'Enter a title here'. A tab for 'Main.sql' is also visible. A green success banner at the top of the code editor area reads 'Success' and includes 'Tweet' and 'Share 0' buttons. The code editor contains the following SQL code:

```
1 create table Fabricante(  
2   codFabricante integer not null primary key,  
3   nomeFabricante varchar (50)  
4 );  
5  
6  
7 create table Automovel (  
8   codAuto integer not null primary key,  
9   codFabricante integer,  
10  foreign key (codFabricante) references fabricante (codFabricante),  
11  modelo varchar(30),  
12  ano integer,  
13  pais varchar(30),  
14  preco_tabela decimal(10,2)  
15 );  
16  
17 create table Revendedora (  
18   cgc integer not null primary key,  
19   nome varchar (30),  
20   cidade varchar (30),  
21   estado varchar(2)  
22 );
```

At the bottom left, there is a 'Run (Ctrl-Enter)' button. The bottom right corner features a series of icons for user profile, information, settings, timer, undo, redo, star, group, lock, and share.

Acima temos a imagem a criação de tabelas no paiza io

DDL (DATA DEFINITION LANGUAGE)

- Para criação de nossas tabelas utilizamos as seguintes informações:

```
create table Fabricante(  
codFabricante integer not null primary key,  
nomeFabricante varchar (50)  
);
```

```
create table Automovel (  
codAuto integer not null primary key,  
codFabricante integer,  
foreign key (codFabricante) references fabricante (codFabricante),  
modelo varchar(30),  
ano integer,  
pais varchar(30),  
preco_tabela decimal(10,2)  
);
```

DDL (DATA DEFINITION LANGUAGE)

- Para criação de nossas tabelas utilizamos as seguintes informações:

```
table Revendedora (  
  cgc integer not null primary key,  
  nome varchar (30),  
  cidade varchar (30),  
  estado varchar(2)  
);
```

```
create table Consumidor (  
  codConsumidor integer not null primary key,  
  nome varchar (50),  
  Sobrenome varchar (50)  
);
```

DDL (DATA DEFINITION LANGUAGE)

- Para criação de nossas tabelas utilizamos as seguintes informações:

```
create table Negocio (  
  codNegocio integer not null primary key,  
  codComprador integer,  
  codRevenda integer,  
  codAuto integer,  
  foreign key (codComprador) references Consumidor(codConsumidor),  
  foreign key (codRevenda) references Revendedora(cgc),  
  foreign key (codAuto) references Automovel(codAuto),  
  data date,  
  Preco decimal (10,2)  
);
```

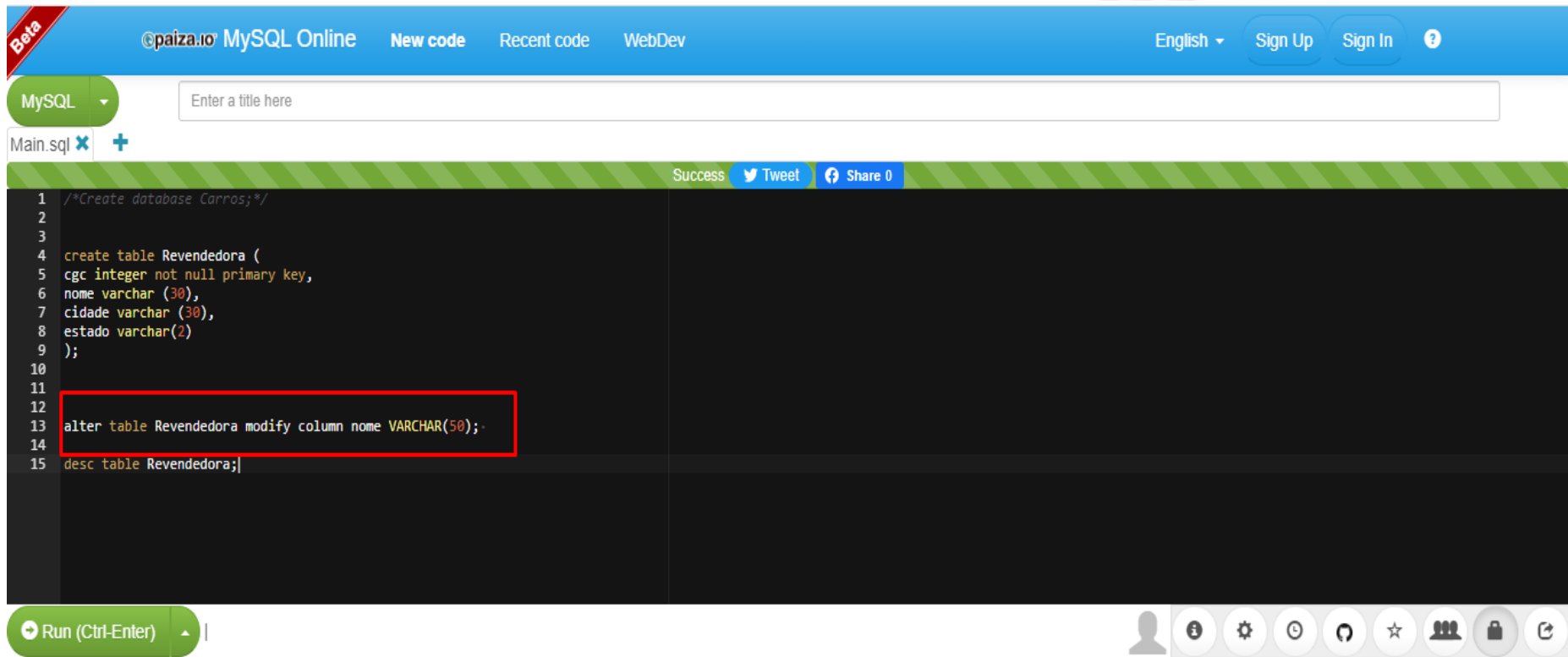
DDL (DATA DEFINITION LANGUAGE)

- **Alter** – altera o objeto do banco de dados
 - Alterando uma tabela adicionando uma nova coluna:

ALTER TABLE [nome da tabela] ADD COLUMN (FAX VARCHAR(10) NOT NULL);
 - Alterando uma tabela modificando uma coluna:
ALTER TABLE [nome da tabela] MODIFY COLUMN TELEFONE VARCHAR(12);
- Apagando os dados de uma tabela retornando-a ao estado de origem:
- TRUNCATE TABLE [nome da tabela];

DDL (DATA DEFINITION LANGUAGE)

- **Alter** – altera o objeto do banco de dados



The screenshot shows the @paiza.io MySQL Online interface. The top navigation bar is blue with the @paiza.io logo, 'MySQL Online' text, and links for 'New code', 'Recent code', and 'WebDev'. On the right, there are links for 'Sign Up', 'Sign In', and a help icon. Below the navigation bar, there's a green 'MySQL' dropdown menu and a search bar with the placeholder 'Enter a title here'. The main editor area has a tab labeled 'Main.sql'. A green success banner at the top of the editor displays 'Success' and social sharing options for 'Tweet' and 'Share 0'. The SQL code in the editor is as follows:

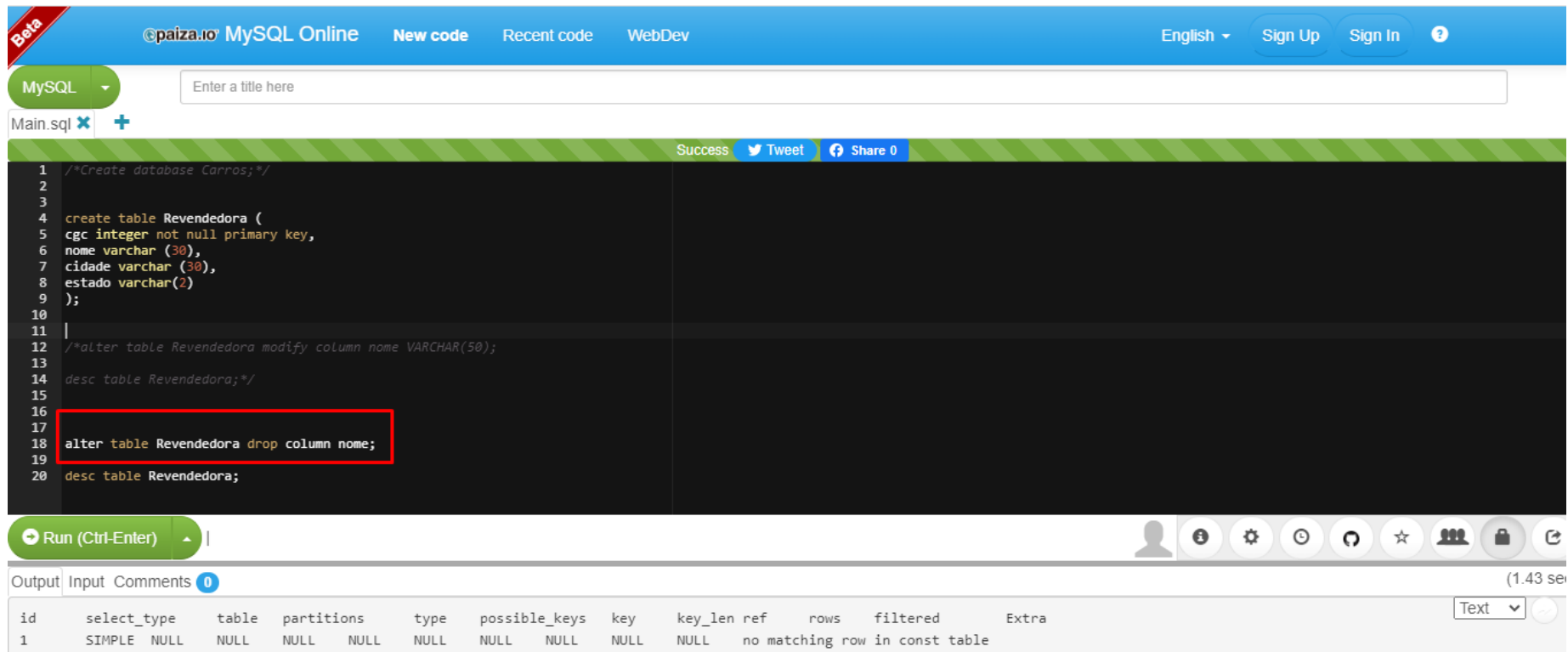
```
1 /*Create database Carros;*/
2
3
4 create table Revendedora (
5   cgc integer not null primary key,
6   nome varchar (30),
7   cidade varchar (30),
8   estado varchar(2)
9 );
10
11
12
13 alter table Revendedora modify column nome VARCHAR(50);-
14
15 desc table Revendedora;
```

The line `alter table Revendedora modify column nome VARCHAR(50);-` is highlighted with a red rectangular box. At the bottom of the interface, there is a green 'Run (Ctrl-Enter)' button and a row of utility icons including a user profile, information, settings, a clock, a refresh icon, a star, a group of people, a lock, and a share icon.

Acima temos a imagem da sintaxe de alteração de um campo da tabela .

DDL (DATA DEFINITION LANGUAGE)

- **Alter** – altera o objeto do banco de dados



The screenshot shows the @paiza.io MySQL Online interface. The code editor contains the following SQL commands:

```
1  /*Create database Carros;*/
2
3
4  create table Revendedora (
5  cgc integer not null primary key,
6  nome varchar (30),
7  cidade varchar (30),
8  estado varchar(2)
9  );
10
11 |
12 /*alter table Revendedora modify column nome VARCHAR(50);
13
14 desc table Revendedora;*/
15
16
17
18 alter table Revendedora drop column nome;
19
20 desc table Revendedora;
```

The command `alter table Revendedora drop column nome;` is highlighted with a red box. The output pane shows the following result:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	NULL	NULL	NULL	NULL	NULL	NULL	no matching row in const table			

Acima temos a imagem da sintaxe para apagar um campo da tabela.

DDL (DATA DEFINITION LANGUAGE)

- **Drop** – apaga o objeto do banco de dados
- **Deletando uma tabela:**
DROP TABLE [nome da tabela];

The screenshot shows the @paiza.io MySQL Online interface. The top navigation bar includes 'New code', 'Recent code', and 'WebDev' buttons. The main editor area displays a SQL script with the following content:

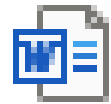
```
1  /*Create database Carros;*/
2
3
4  create table Revendedora (
5  cgc integer not null primary key,
6  nome varchar (30),
7  cidade varchar (30),
8  estado varchar(2)
9  );
10
11
12  drop table Revendedora;
13
14  desc table Revendedora;
```

The line `drop table Revendedora;` is highlighted with a red box. Above the editor, a message indicates a 'Runtime error(Exit status:1)'. At the bottom, the 'Output' tab shows the error message: 'ERROR 1146 (42S02) at line 15: Table 'test.Revendedora' doesn't exist', which is also highlighted with a red box. The interface includes a 'Run (Ctrl-Enter)' button and a status bar at the bottom right showing '(1.41 sec)'.

Acima temos a imagem da sintaxe para apagar um campo da tabela.

Desafio

Atividades – Criando Tabelas.



Imobiliária_Criação tabelas_ATIVIDADE

DML (Data Manipulation Language)

- **Insert** – insere tupla (linha) na tabela existente
 - **insert into** [nome da tabela] **values** (informações)
- **Update** – altera o tupla (linha) da tabela
 - **update** [nome da tabela] **set** [nome do campo e valor]
where [código ou chave primária]
- **Delete** – deleta o tupla (linha) da tabela
 - **delete from** [nome da tabela] **where** [código ou chave primária]

DML (Data Manipulation Language)

- **Insert** – insere tupla (linha) na tabela existente.

The screenshot shows the @paiza.io MySQL Online interface. The top navigation bar is blue and contains the text "@paiza.io MySQL Online", "New code", "Recent code", "WebDev", "English", "Sign Up", "Sign In", and a help icon. Below the navigation bar is a search bar with the text "Enter a title here". The main code editor has a dark background and shows the following SQL code:

```
43 Preco decimal (10,2)
44 );
45
46 insert into Fabricante values (1,'Volkswagen');
47 insert into Fabricante values (2,'Peugeot');
48 insert into Fabricante values (3,'Chevrolet');
49 insert into Fabricante values (4,'Caoa');
50 insert into Fabricante values (5,'Ford');
51 insert into Fabricante values (6,'Fiat');
52
53
54 insert into Automovel values (1,6,'Toro',2000,'Brasil',25000.00);
55 insert into Automovel values (2,5,'EcoSport',2005,'Argentina',39000.00);
56 insert into Automovel values (3,2,'P2008',1990,'Brasil',15000.00);
57 insert into Automovel values (4,3,'Corsa Sedã',1995,'Brasil',12500.00);
58 insert into Automovel values (5,4,'Tiggo',2003,'Argentina',20000.00);
59 insert into Automovel values (6,1,'Gol',1995,'Argentina',10000.00);
60
61
62 insert into Revendedora values (121215,'Garagem Auto-Estrada','Palmas','TO');
63 insert into Revendedora values (154956,'Siga Bem','Paraiso','TO');
64 insert into Revendedora values (147896,'Vá adiante','Goiânia','GO');
```

A red box highlights the first six insert statements for the 'Fabricante' table. The bottom output panel shows the text "Output Input Comments 0" and a "Text" dropdown menu. The execution time is displayed as "(1.38 sec)".

DML (Data Manipulation Language)

- **Update** – altera o tupla da tabela

Beta @paiza.io MySQL Online New code Recent code WebDev English Sign Up Sign In ?

MySQL Enter a title here

Main.sql +

Success Tweet Share 0

```
60
61
62 insert into Revendedora values (121215,'Garagem Auto-Estrada','Palmas','TO');
63 insert into Revendedora values (154956,'Siga Bem','Paraíso','TO');
64 insert into Revendedora values (147896,'Vá adiante','Goiânia','GO');
65
66
67 insert into Consumidor values (159753,'José','Da Conceição');
68 insert into Consumidor values (158741,'Paulo','Da Silva Cunha');
69 insert into Consumidor values (789564,'Maria','José dPaula');
70 insert into Consumidor values (498534,'Joana','Maranhão da Silvéria');
71
72
73 insert into Negocio values (1,158741,121215,3,'2000/05/15',17500.00);
74 insert into Negocio values (2,498534,154956,1,'2022/08/22',28000.00);
75 insert into Negocio values (3,159753,147896,4,'2021/10/12',15500.00);
76 insert into Negocio values (4,498534,147896,2,'2023/01/28',42000.00);
77
78
79 update Automovel set preco_tabela=130000 where codAuto =5;
80 select * from Automovel where codAuto =5;
81
```

Run (Ctrl-Enter)

Output Input Comments 0 (1.39 sec)

codAuto	codFabricante	modelo	ano	pais	preco_tabela
5	4	Tiggo	2003	Argentina	130000.00

Help

DML (Data Manipulation Language)

- **Delete** – deleta o tupla (linha) da tabela

The screenshot shows the Paiza Online MySQL interface. The top bar is blue with the Paiza logo, 'MySQL Online', and navigation links like 'New code', 'Recent code', and 'WebDev'. There are also links for 'Sign Up' and 'Sign In'. Below the bar, there's a green bar with 'MySQL' and a search input. The main area is a dark editor with SQL code. A red box highlights the 'delete from Automovel where codAuto =5;' statement. Below the editor, there's a 'Run (Ctrl-Enter)' button. The output section shows a table with 6 columns: codAuto, codFabricante, modelo, ano, pais, and preco_tabela. The table contains 6 rows of data. A red box highlights the first four rows of the output table. The bottom right corner has a 'PaizaCloud' logo and a 'Help' button.

```
60
61
62 insert into Revendedora values (121215,'Garagem Auto-Estrada','Palmas','TO');
63 insert into Revendedora values (154956,'Siga Bem','Paraíso','TO');
64 insert into Revendedora values (147896,'Vá adiante','Goiânia','GO');
65
66
67 insert into Consumidor values (159753,'José','Da Conceição');
68 insert into Consumidor values (158741,'Paulo','Da Silva Cunha');
69 insert into Consumidor values (789564,'Maria','José dPaula');
70 insert into Consumidor values (498534,'Joana','Maranhão da Silvéria');
71
72
73 insert into Negocio values (1,158741,121215,3,'2000/05/15',17500.00);
74 insert into Negocio values (2,498534,154956,1,'2022/08/22',28000.00);
75 insert into Negocio values (3,159753,147896,4,'2021/10/12',15500.00);
76 insert into Negocio values (4,498534,147896,2,'2023/01/28',42000.00);
77
78
79 delete from Automovel where codAuto =5;
80 rollback * from Automovel;
81
```

Run (Ctrl-Enter)

codAuto	codFabricante	modelo	ano	pais	preco_tabela
1	6	Toro	2000	Brasil	25000.00
2	5	EcoSport	2005	Argentina	39000.00
3	2	P2008	1990	Brasil	15000.00
4	3	Corsa Sedã	1995	Brasil	12500.00
https://paiza.io	Gol	1995	Argentina		10000.00

PaizaCloud

Desafio

Atividades – Inserindo Dados

 Imobiliária_Criação Inserts_ATIVIDADE

DQL (DATA QUERY LANGUAGE)

- O DQL (Data Query Language) é o conjunto SQL que define o comando mais popular da linguagem, o SELECT. Esse comando é essencial para que possamos **consultar os dados que armazenamos em nosso banco**.
- select – consulta os registros do banco de dados com uma descrição do resultado desejado
 - **select** [* = todos os campos, ou campos específicos] **from** [nome da tabela]
 - **select** cpf, nome [campos específicos] **from** [nome da tabela] **where** [condição]

DQL (DATA QUERY LANGUAGE) - Selects

Beta @paiza.io MySQL Online New code Recent code WebDev English Sign Up Sign In ?

MySQL Enter a title here

Main.sql x +

Success Tweet Share 0

```
63
64 insert into Revendedora values (121215,'Garagem Auto-Estrada','Palmas','TO');
65 insert into Revendedora values (154956,'Siga Bem','Paraíso','TO');
66 insert into Revendedora values (147896,'Vá adiante','Goiânia','GO');
67
68
69 insert into Consumidor values (159753,'José','Da Conceição');
70 insert into Consumidor values (158741,'Paulo','Da Silva Cunha');
71 insert into Consumidor values (789564,'Maria','José dPaula');
72 insert into Consumidor values (498534,'Joana','Maranhão da Silvéria');
73
74
75 insert into Negocio values (1,158741,121215,3,'2000/05/15',17500.00);
76 insert into Negocio values (2,498534,154956,1,'2022/08/22',28000.00);
77 insert into Negocio values (3,159753,147896,4,'2021/10/12',15500.00);
78 insert into Negocio values (4,498534,147896,2,'2023/01/28',42000.00);
79
80
81 select * from Negocio;
82
83
84
```

Run (Ctrl-Enter)

Output Input Comments 0 (1.50 sec)

codNegocio	codComprador	codRevenda	codAuto	data	Preco
1	158741	121215	3	2000-05-15	17500.00
2	498534	154956	1	2022-08-22	28000.00
3	159753	147896	4	2021-10-12	15500.00
4	498534	147896	2	2023-01-28	42000.00

PaizaCloud ? Help

Desafio

Atividades – Selecionando dados.

- Selecione todas as informações da tabela Proprietário.
- Selecione apenas Nome, endereço e telefone da tabela Proprietário
- Selecione todas as informações da tabela Inquilino
- Selecione apenas os nomes da tabela Inquilino
- Selecione todas as informações da tabela Imóvel
- Selecione endereço, cidade e locado da tabela Imóvel
- Selecione todas as informações da tabela Locação
- Selecione valor aluguel, taxa administrativa início e término da tabela Locação

DQL (DATA QUERY LANGUAGE) - Selects

- Selecionando informações com condição.
- Selecionando informações utilizando campo como condição igualdade

Select nome, endereco from inq where cidade= 'São Carlos';

- Selecionando informações utilizando campo como condição de comparação

Select nome, endereco from inq where codprop <> 3;

- Selecionando informações utilizando campo como condição de comparação

Select nome, endereco from inq where codprop > 3;

DQL (DATA QUERY LANGUAGE) - Selects

- Selecionando informações com condição.

Selecionando informações utilizando campo como condição o comando Like utilizado para campos tipo varchar

- Nomes que comecem com R

select nome from inq where nome like 'R%';

- Nomes que terminem com S

select nome from inq where nome like '%s';

- Nomes que tenham A em qualquer parte da palavra

select nome from inq where nome like '%a%';

DQL (DATA QUERY LANGUAGE) - Selects

- Selecionando informações com condição.

Selecionando informações de acordo com condição e ordenando de A à Z e de Z para A.

- Selecionando e exibindo em ordem crescente ou de A à Z

select nome from inq order by nome asc;

select valor aluguel from loc where locado= 'S' order by valoraluguel asc;

- Selecionando e exibindo em ordem decrescente ou de Z à A

select nome from inq order by nome desc

select valor aluguel from loc where locado= 'S' order by valoraluguel desc;

DQL (DATA QUERY LANGUAGE) - Selects

Operador	Descrição
IS NULL	Verifica se um campo está vazio.
BETWEEN	Verifica se um valor está dentro de um intervalo.
IN	Verifica se um valor está em um conjunto de valores.

- **select codAuto, modelo from Automovel where preco_tabela IS NULL;**
- **select codAuto, modelo FROM Automovel where preco_tabela IS NOT NULL;**
- **select * from Negocio where preco between 5000 and 20000;**
- **select * from Negocio where preco not between 20000 and 50000;**
- **select codAuto, codFabricante, modelo from Automovel where codFabricante in('2','3');**
- **select codAuto, codFabricante, modelo from Automovel where codFabricante not in ('2', '3');**

DQL (DATA QUERY LANGUAGE) - Selects

Função	Retorna...
AVG()	Média aritmética
COUNT()	A quantidade de registros
MAX()	O maior valor
MIN()	O menor valor
SUM()	Soma dos valores

- **select avg(preco_tabela) from Automovel;**
- **select count(*) from Automovel where ano = '1995';**
- **select count(*) from Automovel where preco_tabela > 2000;**
- **select max(preco) from Negocio;**
- **select min(preco) from Negocio;**
- **select sum(preco) from Negocio;**
- **select sum(preco_tabela) from Automovel where ano = '1995';**

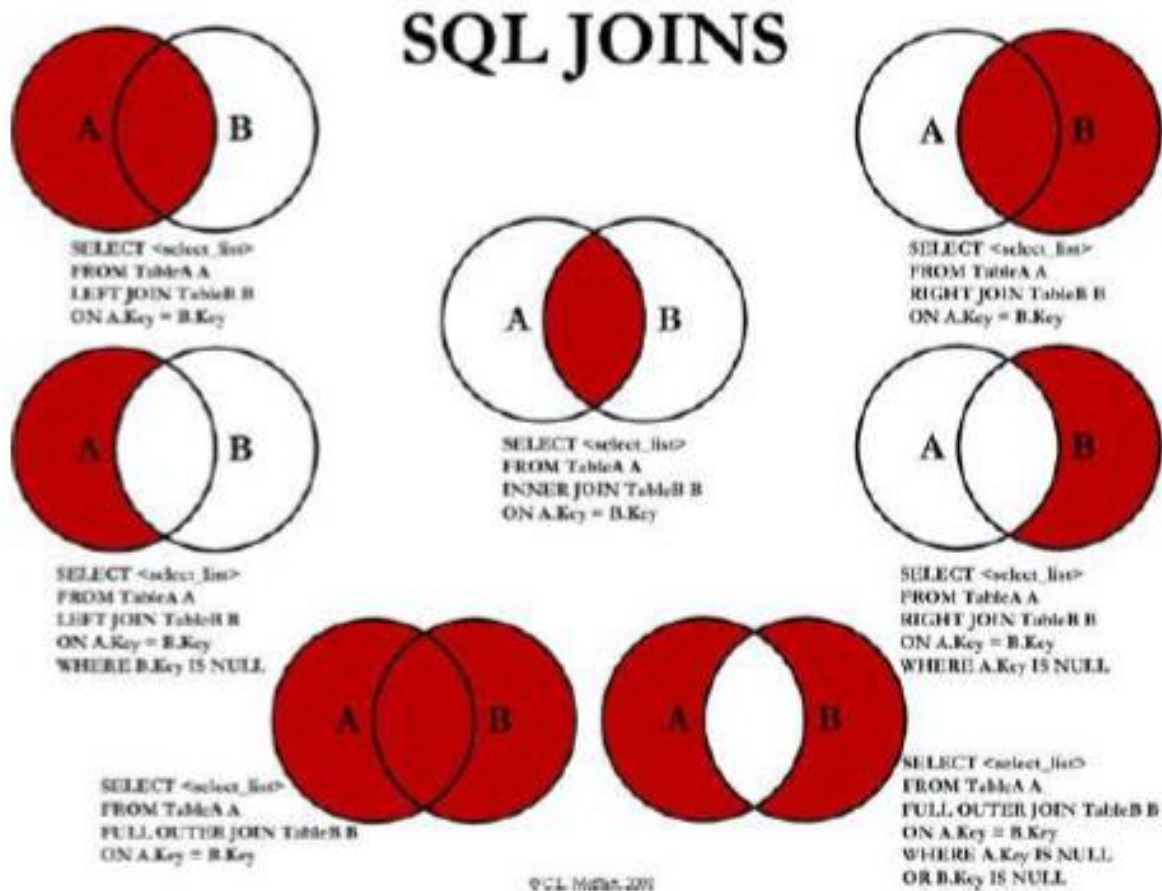
Desafio

Atividades Selects

 Atividades_BD_Compras

DQL (DATA QUERY LANGUAGE) - JOINS

- A palavra Join, significa JUNÇÃO.

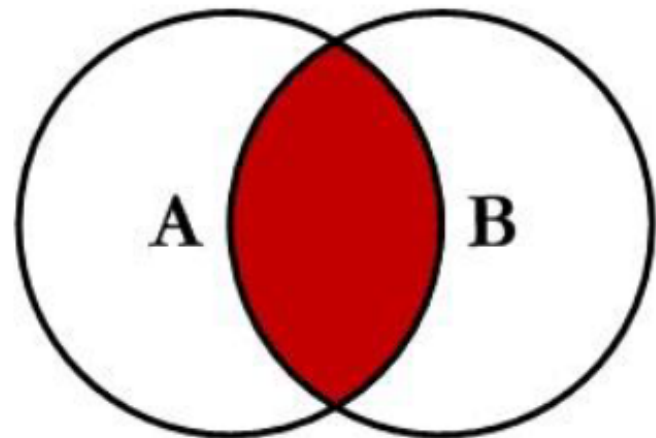


DQL (DATA QUERY LANGUAGE) - JOINS

- Para demonstrar o funcionamento dos métodos de junção, precisamos de tabelas entre as quais deve haver algum relacionamento para que possamos "cruzar" os dados.

Inner Join

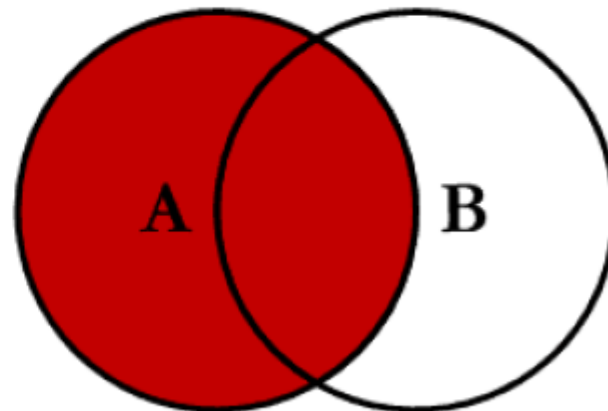
- Método de junção mais conhecido;
- Retorna os registros que são comuns às duas tabelas.



DQL (DATA QUERY LANGUAGE) - JOINS

Left Join

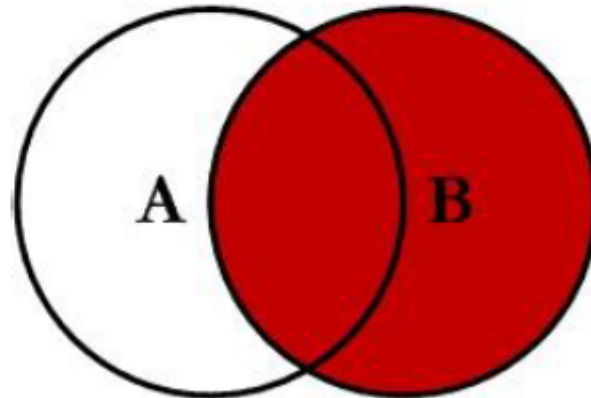
- Tem como resultado todos os registros que estão na tabela A (mesmo que não estejam na tabela B) e os registros da tabela B que são comuns à tabela A.



DQL (DATA QUERY LANGUAGE) - JOINS

Right Join

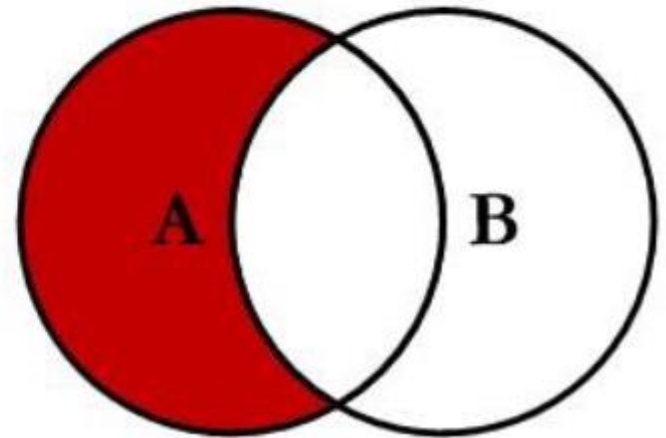
- Teremos como resultado todos os registros que estão na tabela B (mesmo que não estejam na tabela A) e os registros da tabela A que são comuns à tabela B.



DQL (DATA QUERY LANGUAGE) - JOINS

Left Excluding Join

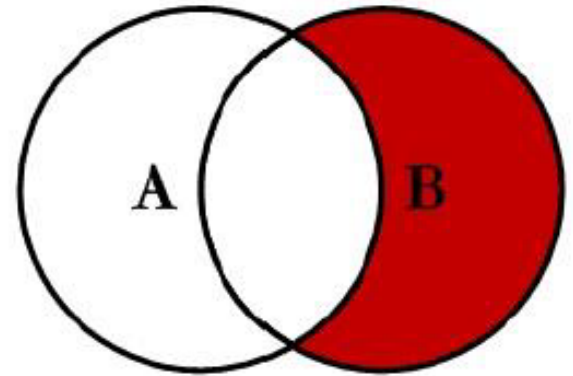
- Retorna como resultado todos os registros que estão na tabela A e que não estejam na tabela B.



DQL (DATA QUERY LANGUAGE) - JOINS

Right Excluding Join

- Retorna como resultado todos os registros que estão na tabela B e que não estejam na tabela A.



DQL (DATA QUERY LANGUAGE) - JOINS

Beta

paiza.io MySQL Online

New code

MySQL

Enter a title here

Main.sql

```
1 create database juncao;
2 use juncao;
3
4
5 CREATE TABLE TabelaA(
6   cod_a integer not null primary key,
7   Nome varchar(50) NULL
8 );
9
10
11 CREATE TABLE TabelaB(
12   cod_a integer not null primary key,
13   Nome varchar(50) NULL
14 );
15
16
17 CREATE TABLE Aula(
18   cod_aula integer not null primary key,
19   Nome varchar(50) NULL,
20   codprof integer,
21   codaluno integer,
22   foreign key (codprof) references TabelaA (cod_a),
23   foreign key (codaluno) references TabelaB (cod_b)
24 );
25
```

Run (Ctrl-Enter)

Output Input Comments 0

Beta

paiza.io MySQL Online

New code

Recent code

WebDev

MySQL

Enter a title here

Main.sql

```
18 cod_aula integer not null primary key,
19 Nome varchar(50) NULL,
20 codprof integer,
21 codaluno integer,
22 foreign key (codprof) references TabelaA (cod_a),
23 foreign key (codaluno) references TabelaB (cod_b)
24 );
25
26 INSERT INTO TabelaA VALUES(1,'Fernanda');
27 INSERT INTO TabelaA VALUES(2,'Josefa');
28 INSERT INTO TabelaA VALUES(3,'Luiz');
29 INSERT INTO TabelaA VALUES(4,'Fernando');
30
31 INSERT INTO TabelaB VALUES(1,'Carlos');
32 INSERT INTO TabelaB VALUES(2,'Fernando');
33 INSERT INTO TabelaB VALUES(3,'Manoel');
34 INSERT INTO TabelaB VALUES(4,'Luiz');
35
36
37 insert into Aula values (1,'Inglês',1,3);
38 insert into Aula values (2,'Geografia',2,4);
39 insert into Aula values (3,'Matemática',3,2);
40 insert into Aula values (4,'História',4,1);
41
42
43 SELECT * FROM Aula;
```

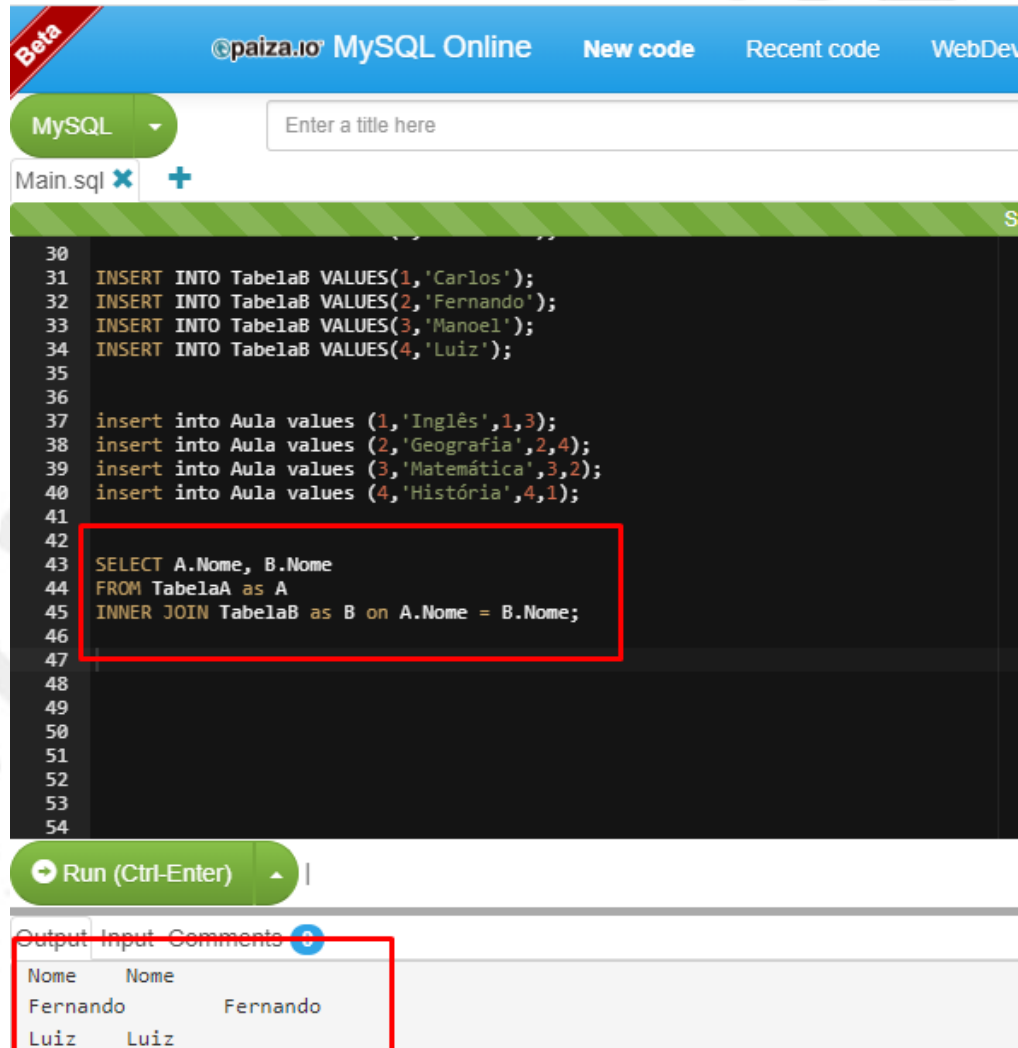
Run (Ctrl-Enter)

Output Input Comments 0

Success

DQL (DATA QUERY LANGUAGE) - JOINS

INNER JOIN



The screenshot shows a web-based MySQL IDE interface. The top navigation bar includes a 'Beta' badge, the 'paiza.io' logo, and links for 'MySQL Online', 'New code', 'Recent code', and 'WebDev'. Below the navigation bar, there's a dropdown menu set to 'MySQL' and a text input field labeled 'Enter a title here'. The main editor area displays a SQL script with line numbers 30 through 54. The script contains four INSERT statements for 'TabelaB' and four INSERT statements for 'Aula'. An INNER JOIN query is highlighted with a red box, showing a join between 'TabelaA' (aliased as A) and 'TabelaB' (aliased as B) on the condition 'A.Nome = B.Nome'. Below the editor, there's a 'Run (Ctrl-Enter)' button. At the bottom, the 'Output' tab is active, showing the results of the query in a table format, which is also highlighted with a red box. The table has two columns, 'Nome', and displays the names 'Fernando' and 'Luiz'.

```
30
31 INSERT INTO TabelaB VALUES(1,'Carlos');
32 INSERT INTO TabelaB VALUES(2,'Fernando');
33 INSERT INTO TabelaB VALUES(3,'Manoel');
34 INSERT INTO TabelaB VALUES(4,'Luiz');
35
36
37 insert into Aula values (1,'Inglês',1,3);
38 insert into Aula values (2,'Geografia',2,4);
39 insert into Aula values (3,'Matemática',3,2);
40 insert into Aula values (4,'História',4,1);
41
42
43 SELECT A.Nome, B.Nome
44 FROM TabelaA as A
45 INNER JOIN TabelaB as B on A.Nome = B.Nome;
46
47
48
49
50
51
52
53
54
```

Run (Ctrl-Enter)

Nome	Nome
Fernando	Fernando
Luiz	Luiz

DQL (DATA QUERY LANGUAGE) - JOINS

LEFT JOIN

Beta paiza.io MySQL Online New code Recent code WebDev

MySQL Enter a title here

Main.sql +

```
30
31 INSERT INTO TabelaB VALUES(1,'Carlos');
32 INSERT INTO TabelaB VALUES(2,'Fernando');
33 INSERT INTO TabelaB VALUES(3,'Manoel');
34 INSERT INTO TabelaB VALUES(4,'Luiz');
35
36
37 insert into Aula values (1,'Inglês',1,3);
38 insert into Aula values (2,'Geografia',2,4);
39 insert into Aula values (3,'Matemática',3,2);
40 insert into Aula values (4,'História',4,1);
41
42
43 SELECT A.Nome, B.Nome
44 FROM TabelaA as A
45 LEFT JOIN TabelaB as B on A.Nome = B.Nome;
46
47
48
49
50
51
```

Run (Ctrl-Enter)

Output Input Comments 0

Nome	Nome
Fernanda	NULL
Josefa	NULL
Luiz	Luiz
Fernando	Fernando

DQL (DATA QUERY LANGUAGE) - JOINS

RIGHT JOIN

Beta paiza.io MySQL Online New code Recent code Web

MySQL Enter a title here

Main.sql

```
30
31 INSERT INTO TabelaB VALUES(1,'Carlos');
32 INSERT INTO TabelaB VALUES(2,'Fernando');
33 INSERT INTO TabelaB VALUES(3,'Manoel');
34 INSERT INTO TabelaB VALUES(4,'Luiz');
35
36
37 insert into Aula values (1,'Inglês',1,3);
38 insert into Aula values (2,'Geografia',2,4);
39 insert into Aula values (3,'Matemática',3,2);
40 insert into Aula values (4,'História',4,1);
41
42
43 SELECT A.Nome, B.Nome
44 FROM TabelaA as A RIGHT JOIN TabelaB as B on A.Nome = B.Nome;
45
46
47
48
49
50
51
```

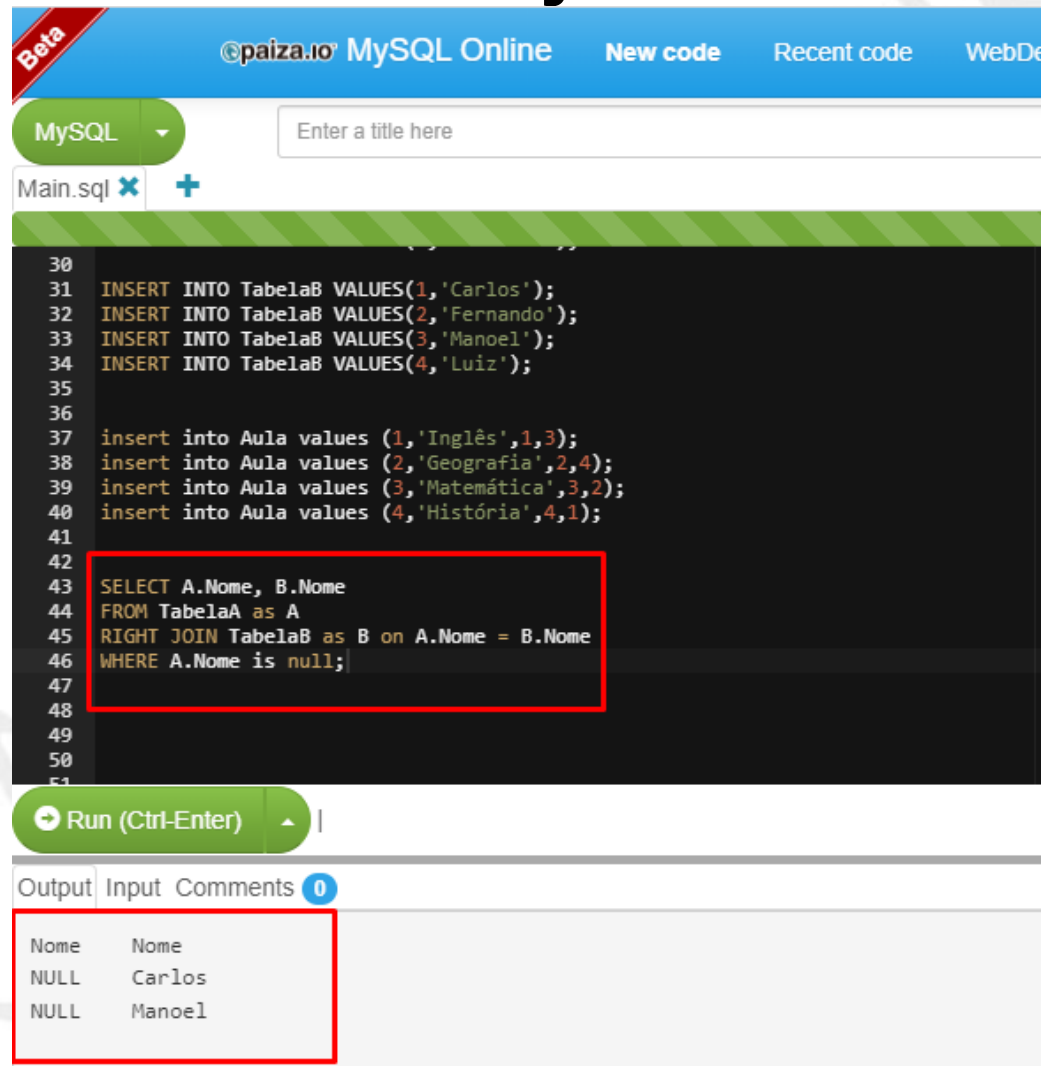
Run (Ctrl-Enter)

Output Input Comments 0

Nome	Nome
NULL	Carlos
Fernando	Fernando
NULL	Manoel
Luiz	Luiz

DQL (DATA QUERY LANGUAGE) - JOINS

RIGHT JOIN com condição.



The screenshot shows the @paiza.io MySQL Online interface. The code editor contains the following SQL queries:

```
30
31 INSERT INTO TabelaB VALUES(1,'Carlos');
32 INSERT INTO TabelaB VALUES(2,'Fernando');
33 INSERT INTO TabelaB VALUES(3,'Manoel');
34 INSERT INTO TabelaB VALUES(4,'Luiz');
35
36
37 insert into Aula values (1,'Inglês',1,3);
38 insert into Aula values (2,'Geografia',2,4);
39 insert into Aula values (3,'Matemática',3,2);
40 insert into Aula values (4,'História',4,1);
41
42
43 SELECT A.Nome, B.Nome
44 FROM TabelaA as A
45 RIGHT JOIN TabelaB as B on A.Nome = B.Nome
46 WHERE A.Nome is null;
47
48
49
50
51
```

The output of the query is displayed in the Output tab, showing the results of the RIGHT JOIN operation:

Nome	Nome
NULL	Carlos
NULL	Manoel

DQL (DATA QUERY LANGUAGE) - JOINS

LEFT JOIN com condição.

The screenshot shows the @paiza.io MySQL Online interface. The SQL editor contains the following code:

```
30
31 INSERT INTO TabelaB VALUES(1,'Carlos');
32 INSERT INTO TabelaB VALUES(2,'Fernando');
33 INSERT INTO TabelaB VALUES(3,'Manoel');
34 INSERT INTO TabelaB VALUES(4,'Luiz');
35
36
37 insert into Aula values (1,'Inglês',1,3);
38 insert into Aula values (2,'Geografia',2,4);
39 insert into Aula values (3,'Matemática',3,2);
40 insert into Aula values (4,'História',4,1);
41
42
43 SELECT A.Nome, B.Nome
44 FROM TabelaA as A LEFT JOIN TabelaB as B on A.Nome = B.Nome
45 WHERE B.Nome is null;
46
47
48
49
50
51
```

The query is highlighted with a red box. Below the editor, the "Run (Ctrl-Enter)" button is visible. The output section shows the results of the query:

Nome	Nome
Fernanda	NULL
Josefa	NULL

The output table is also highlighted with a red box.

Desafio

Atividades Joins



Atividades_BD_Estacionamento

DCL (Data Control Language)

- O DCL (Data Control Language) é o subconjunto no qual encontramos **comandos para controlar o acesso aos dados da nossa base**. Utilizando esse conjunto, conseguimos estabelecer restrições e permissões para quem acessa o banco por meio dos comandos **GRANT** e **REVOKE**.
- É possível utilizar a instrução **SQL GRANT** para conceder **SQL SELECT, UPDATE, INSERT, DELETE**, além de outros privilégios nas utilização de tabelas ou visualizações.
- Exemplo:

GRANT UPDATE ON ORDER_BACKLOG TO JONES WITH GRANT OPTION

(concede ao usuário JONES o privilégio de UPDATE na tabela ORDER_BACKLOG)

DCL (Data Control Language)

REVOKE.

- A instrução **REVOKE** é utilizada para retirar uma permissão concedida pelo **GRANT**.

REVOKE SELECT ON TABLE SMITH.TABLEA FROM BAKER

(Retira o privilégio de **SELECT** concedido ao usuário **BAKER** na tabela **SMITH** e **TABELAA**.)

Referências

- **SQL: O que é e como usar os principais comandos básicos SQL.** Disponível em <<https://blog.betrybe.com/sql/>>

-MACHADO, F. N.; ABREU, M. **Projeto de Banco de Dados: uma visão prática.** 2ª ed. São Paulo. Érica. 1996

Conheça alguns comandos do SQL Server. Disponível em <<https://www.devmedia.com.br/conheca-alguns-comandos-do-sql-server/1334>>

-Alterar Tabelas e Colunas no MySQL com ALTER TABLE. Disponível em <http://www.bosontreinamentos.com.br/mysql/mysql-alterar-tabelas-alter-table-e-visualizar-relacionamentos-10/>

-**A instrução SQL GRANT.** Disponível em <<https://www.ibm.com/docs/pt-br/qmf/11.2?topic=privileges-sql-grant-statement>>