

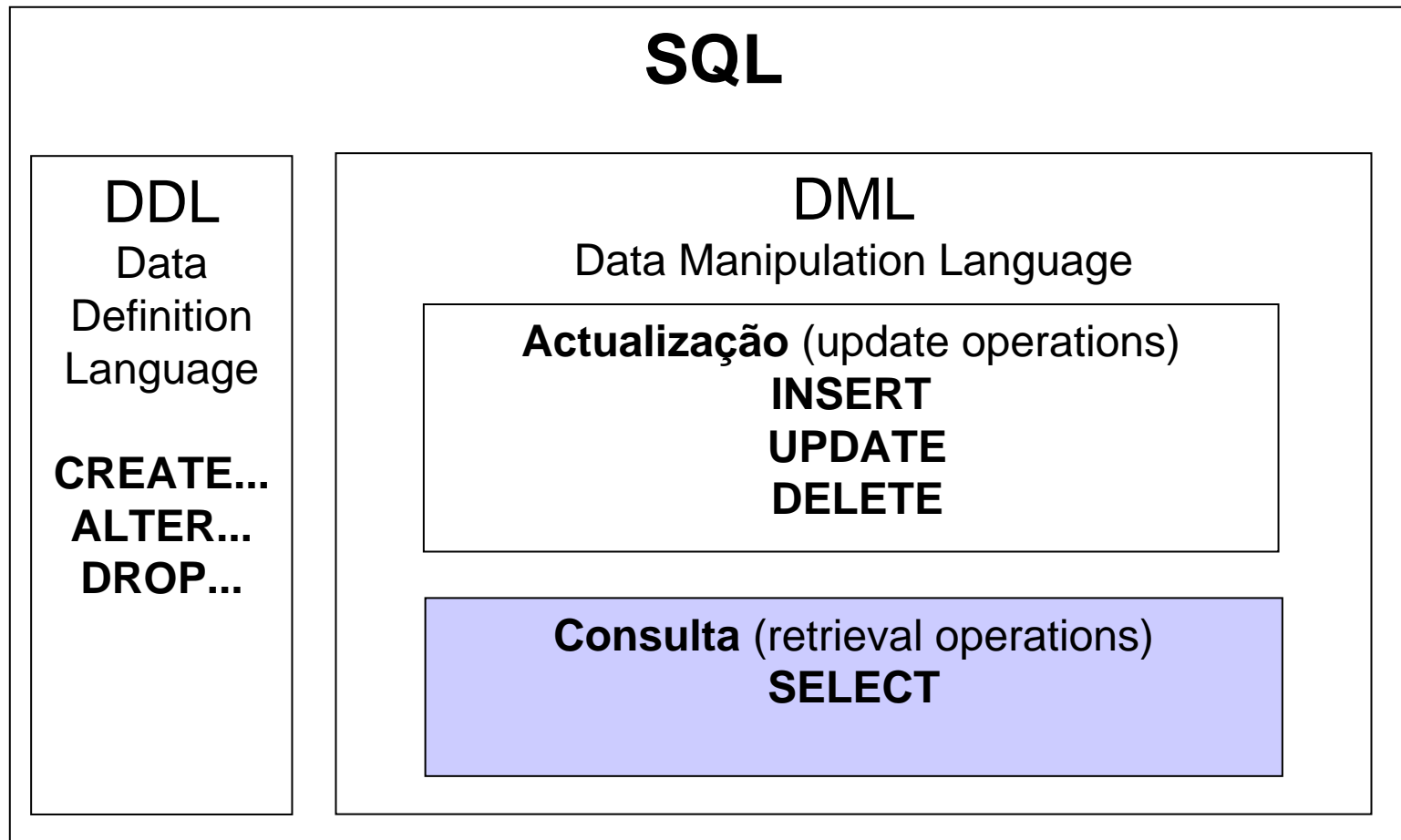
Projecto de Bases de Dados

Linguagem SQL

O que é o SQL?

- **SQL** ("ess-que-el") significa Structured Query Language.
- É uma **linguagem standard** (universal) para comunicação com sistemas de gestão de **bases de dados relacionais**.
- É utilizada para tarefas de **actualização ou recolha** de dados
- É suportada pela maioria de SGBDs, mas todos os sistemas incluem algumas funções específicas.
- Os comandos de SQL standard, tais como **SELECT**, **INSERT**, **UPDATE**, **DELETE** e **CREATE** permitem executar todas as tarefas essenciais.
- Mais informação em: <http://pt.wikipedia.org/wiki/SQL>

Introdução à linguagem SQL

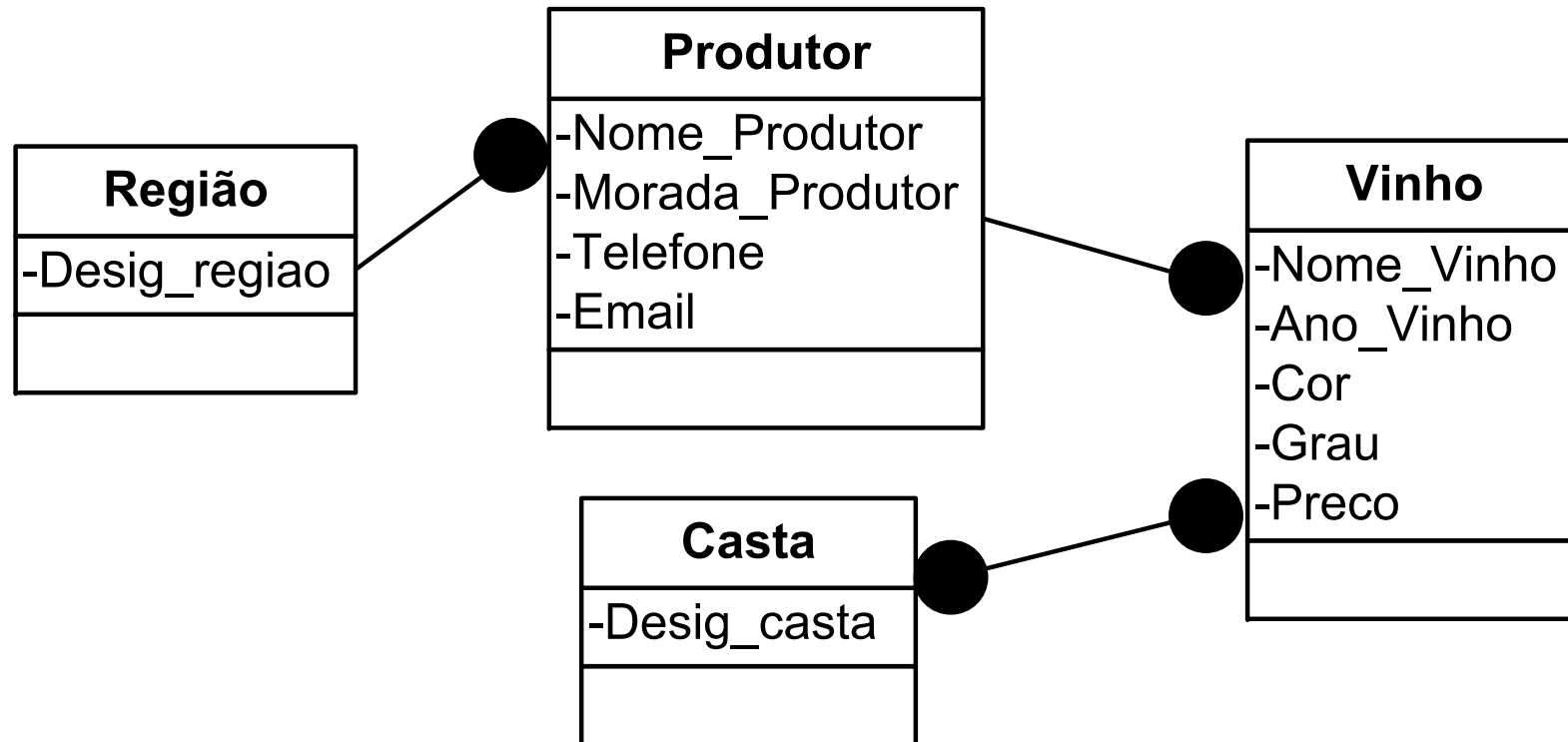


MySQL

- “The world's most popular **open source** database “
- <http://www.mysql.com/>
- MySQL tools:
 - <http://www.mysql.com/products/tools/>
 - MySQL Administrator
 - MySQL Query Browser



EXEMPLO



CREATE TABLE

Syntax:

```
CREATE TABLE
```

```
table_name (  
column_name1 data_type,  
column_name2 data_type,  
..... );
```

Região
-Desig_regiao

```
CREATE TABLE Regiao (  
Regiao_ID INT PRIMARY KEY,  
Desig_regiao VARCHAR(15) );
```

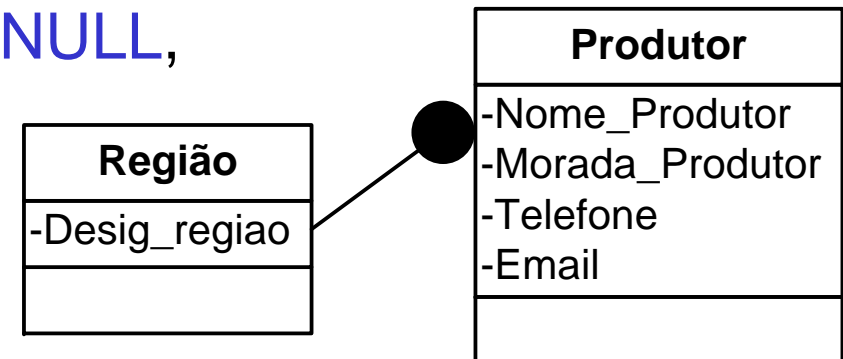
DATA TYPES

Os tipos de dados mais importantes são:

- Boolean
 - True/False values (**BOOLEAN**)
- Character
 - Fixed width (**CHAR**) or variable size (**VARCHAR**)
- Numeric
 - Integer (**INT**), Real (**FLOAT**), Real (**DOUBLE**)
- Temporal
 - Time (**TIME**), Date (**DATE**)

CREATE TABLE

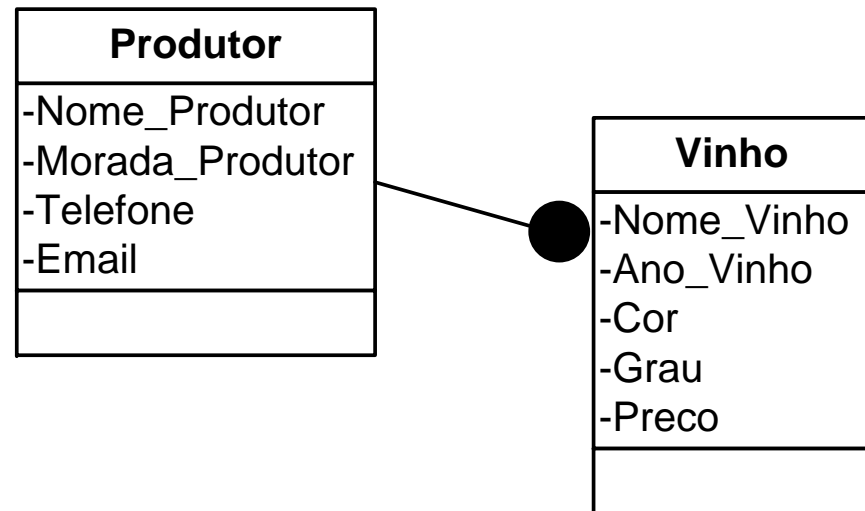
```
CREATE TABLE Produtor (  
  Produtor_ID INT,  
  Nome_Produtor VARCHAR(25) NOT NULL,  
  Morada_Produtor VARCHAR(15),  
  Telefone VARCHAR(15),  
  Email VARCHAR(30),  
  Regiao_ID INT,
```



```
  PRIMARY KEY(Produtor_ID),  
  FOREIGN KEY (Regiao_ID) REFERENCES Regiao(Regiao_ID));
```

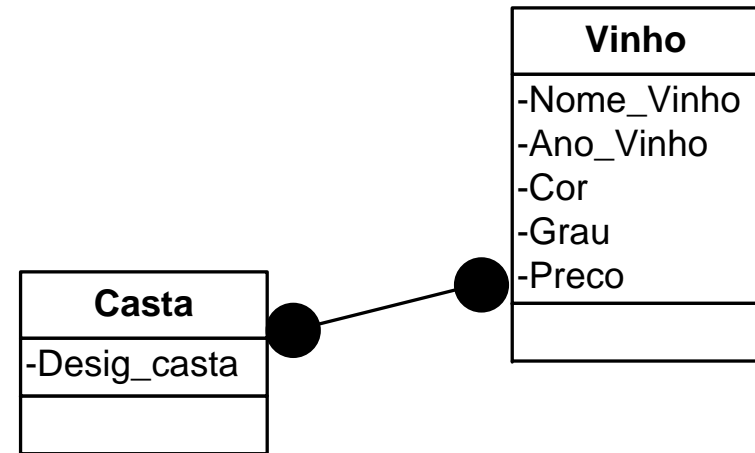
CREATE TABLE

```
CREATE TABLE Vinho (  
Vinho_ID INT PRIMARY KEY,  
Nome_vinho VARCHAR(20),  
Ano_vinho INT,  
Cor VARCHAR(10),  
Grau FLOAT,  
Preco FLOAT,  
Produtor_ID INT,  
FOREIGN KEY (Produtor_ID) REFERENCES  
Produtor(Produtor_ID ));
```



CREATE TABLE

```
CREATE TABLE Casta (  
Casta_ID INT PRIMARY KEY,  
Desig_casta VARCHAR(20) );
```



```
CREATE TABLE Casta_Vinho (  
Casta_ID INT,  
Vinho_ID INT,
```

```
PRIMARY KEY(Casta_ID, Vinho_ID),
```

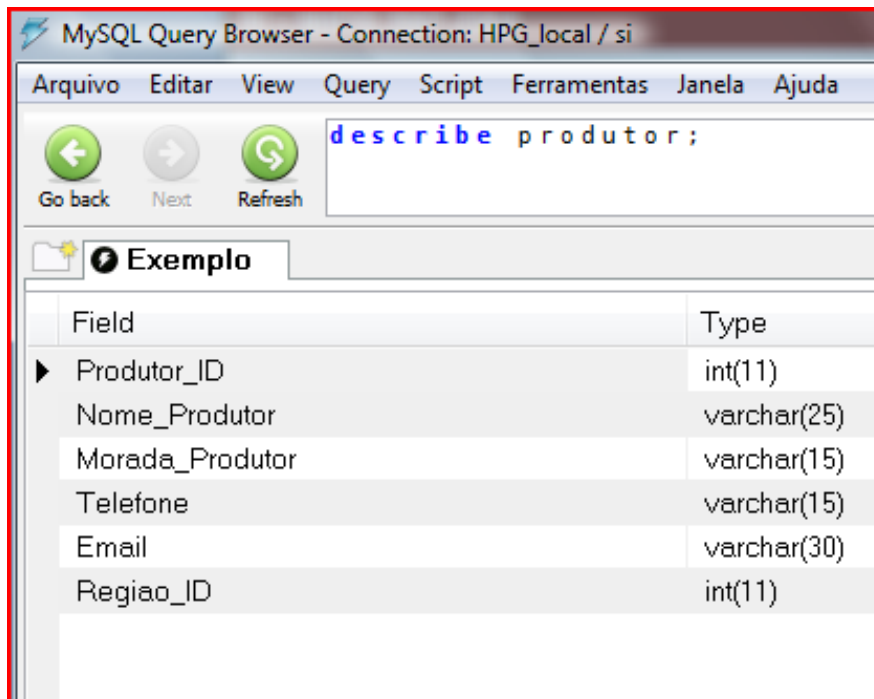
```
FOREIGN KEY (Vinho_ID) REFERENCES Vinho(Vinho_ID),
```

```
FOREIGN KEY (Casta_ID) REFERENCES Casta(Casta_ID));
```

Chaves compostas tem
que ser definidas
assim!

DESCRIBE

DESCRIBE Produtor;



MySQL Query Browser - Connection: HPG_local / si

Arquivo Editar View Query Script Ferramentas Janela Ajuda

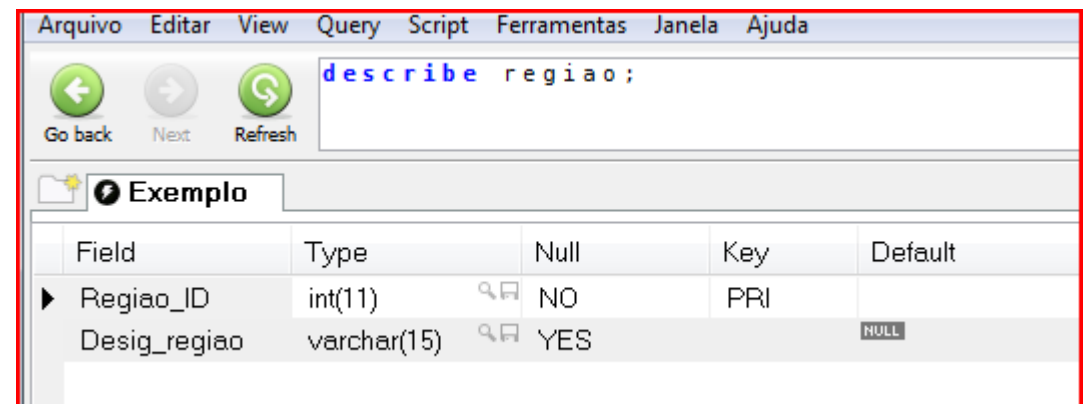
Go back Next Refresh

```
describe produtor;
```

Exemplo

Field	Type
▶ Produtor_ID	int(11)
Nome_Produtor	varchar(25)
Morada_Produtor	varchar(15)
Telefone	varchar(15)
Email	varchar(30)
Regiao_ID	int(11)

DESCRIBE Regiao;



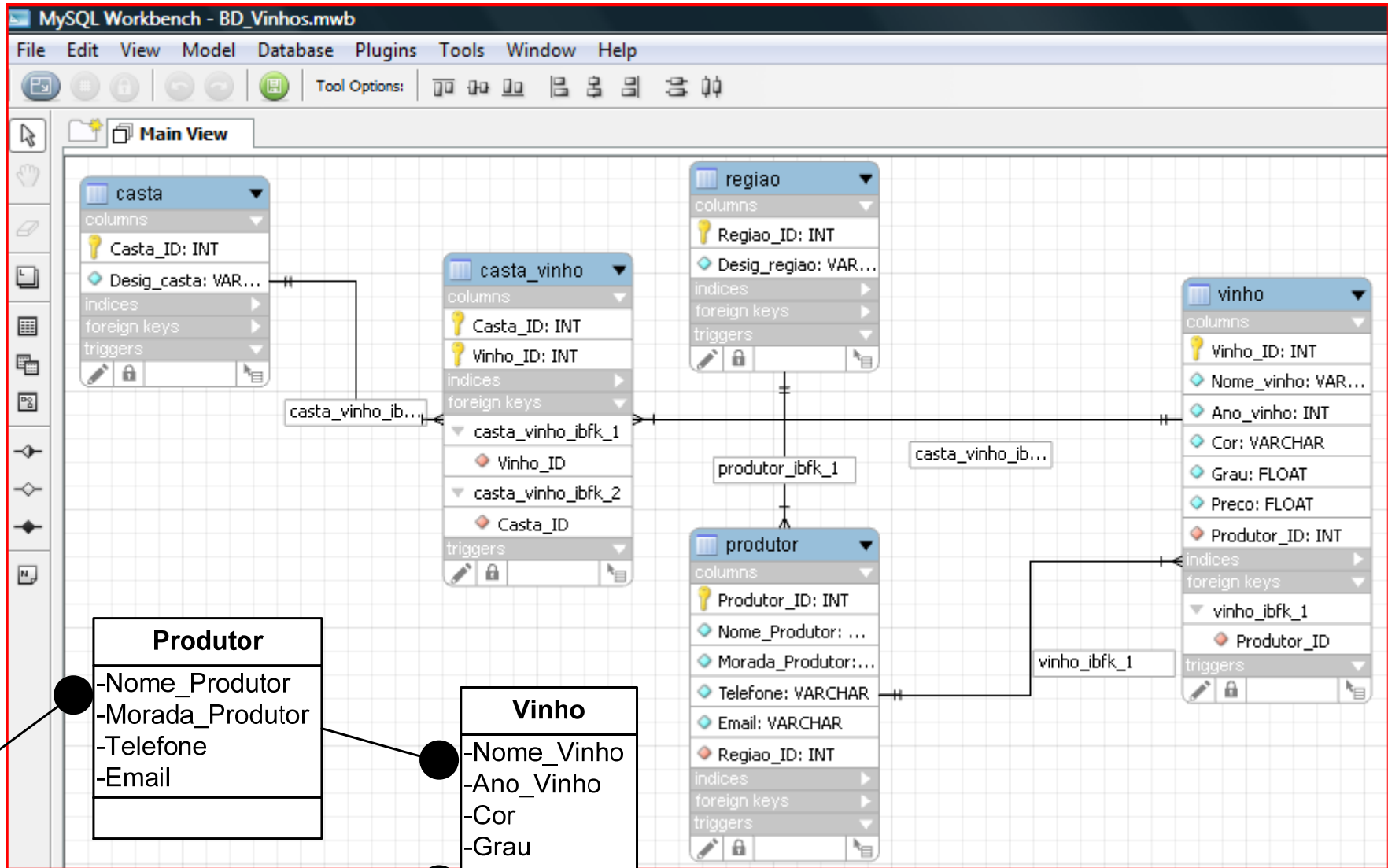
Arquivo Editar View Query Script Ferramentas Janela Ajuda

Go back Next Refresh

```
describe regiao;
```

Exemplo

Field	Type	Null	Key	Default
▶ Regiao_ID	int(11)	NO	PRI	
Desig_regiao	varchar(15)	YES		NULL



DML (Data Manipulation Language)

Operações para actualização de dados:

INSERT

INSERT [INTO] *table_or_view* [(*column_list*)] *data_values*

UPDATE

UPDATE *table_name*

SET { *column_name* = { *expression* | DEFAULT | NULL }

DELETE

DELETE FROM *table_name*

WHERE (*condition*)

INSERT - Regiao

Syntax:

INSERT

[**INTO**] *table_or_view*

[(*column_list*)]

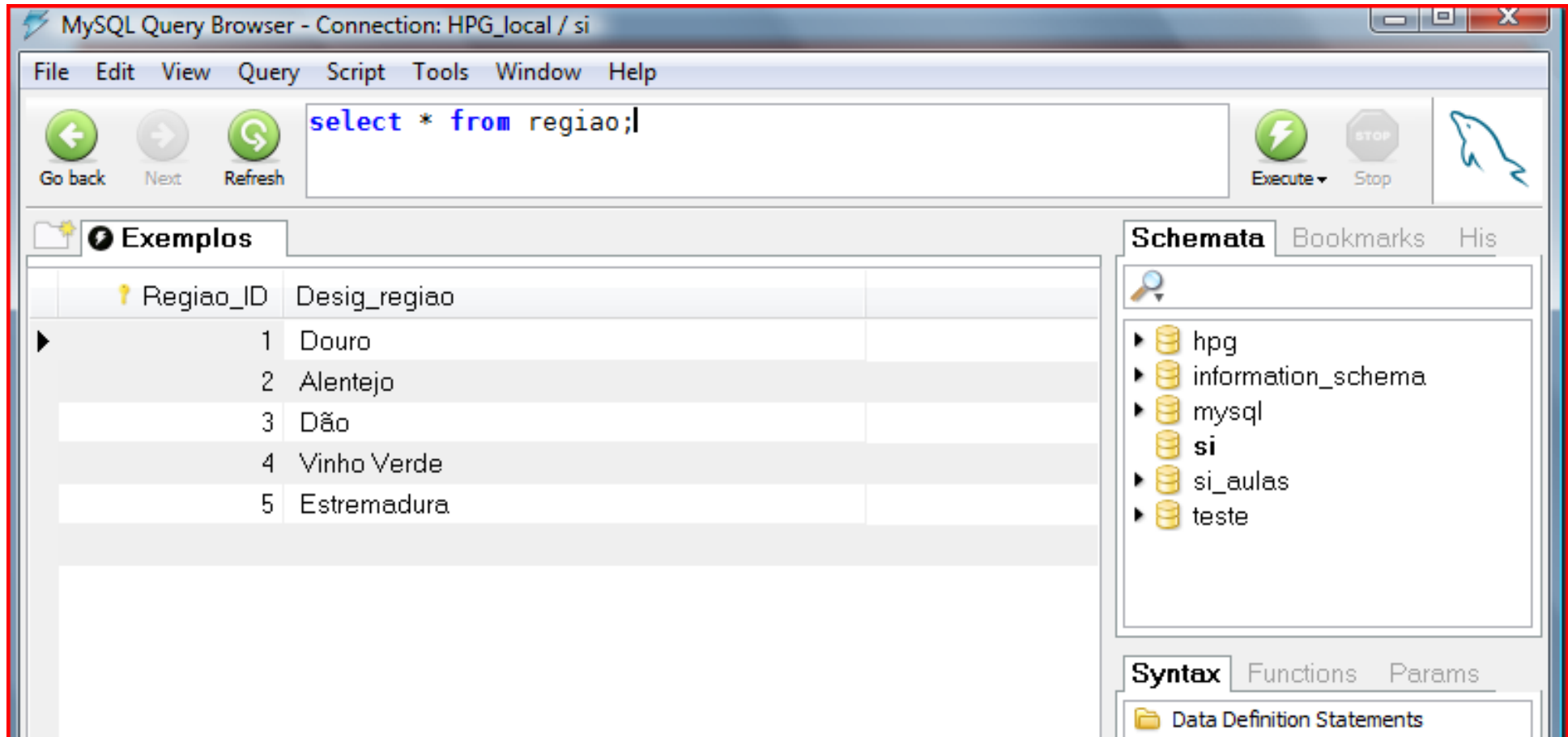
VALUES (*data_values*);

Região
-Desig_regiao

```
INSERT INTO regiao (regiao_id,desig_regiao) VALUES (1,'Douro');
```

```
INSERT INTO regiao VALUES (2,'Alentejo');
```

```
INSERT INTO regiao VALUES (3,'Dão');
```



The screenshot shows the MySQL Query Browser interface. The title bar indicates the connection is to 'HPG_local / si'. The menu bar includes File, Edit, View, Query, Script, Tools, Window, and Help. The query editor contains the SQL statement: `select * from regioao;`. Below the query editor are navigation buttons: Go back, Next, Refresh, Execute, and Stop. The main area displays the results of the query in a table format. The table has two columns: 'Regiao_ID' and 'Desig_regiao'. The results are as follows:

Regiao_ID	Desig_regiao
1	Douro
2	Alentejo
3	Dão
4	Vinho Verde
5	Estremadura

On the right side, the 'Schemata' panel shows a tree view of databases: hpg, information_schema, mysql, si, si_aulas, and teste. Below it, the 'Syntax' panel is open to 'Data Definition Statements'.

INSERT - Produtor

```
INSERT INTO Produtor  
VALUES (1,'Qt. Vallado','Régua','254323147','vallado@mail.telepac.pt',1);
```

```
INSERT INTO Produtor  
VALUES (3,'Finagra','Reguengos','266509270','esporao@esporao.com',2);
```

```
INSERT INTO Produtor (Produtor_ID, Nome_Produtor, Regiao_ID)  
VALUES (4,'Dão Sul',3);
```

```
INSERT INTO Produtor (Produtor_ID, Nome_Produtor, Regiao_ID)  
VALUES (6,'Kolheita Ideias',1);
```

INSERT - Casta

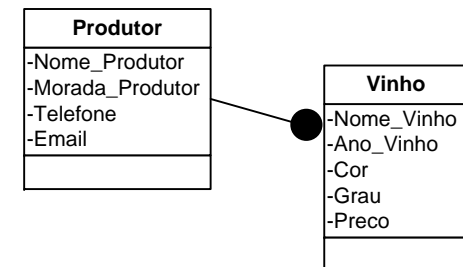
Casta
-Desig_casta

INSERT INTO Casta (Casta_ID,Desig_casta) VALUES (1,'Touriga Nacional');

INSERT INTO Casta VALUES (2,'Tinta Roriz');

INSERT INTO Casta (Casta_ID,Desig_casta) VALUES (3,'Tinta Barroca');

INSERT - Vinho



INSERT INTO Vinho VALUES (1,'Esporão Reserva',2004, 'Tinto',14.5,18.5,3);

INSERT INTO Vinho VALUES (2,'Quinta do Vallado',2004,'Tinto',14.0,6.5,1);

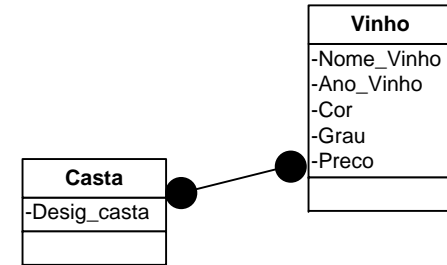
INSERT INTO Vinho VALUES (3,'Muros Antigos',2006,'Branco',13.0,7.5,7);

Insert - Casta_Vinho

INSERT INTO Casta_Vinho VALUES (6,1);

INSERT INTO Casta_Vinho VALUES (7,1);

INSERT INTO Casta_Vinho VALUES (10,1);



Casta_ID	Desig_casta
1	Touriga Nacional
2	Tinta Roriz
3	Tinta Barroca
4	Touriga Franca
5	Sousão
6	Trincadeira
7	Aragonês
8	Castelão
9	Moreto
10	Cabernet Sauvignon

Vinho_ID	Nome_vinho	Ano_vinho	Co
1	Esporão Reserva	2004	Tir
2	Quinta do Vallado	2004	Tir
3	Muros Antigos	2006	Bra
4	EA	2006	Tir
5	K	2002	Tir
6	Kolheita	2003	Tir
7	Esboço	2001	Tir
8	Duas Quintas	2003	Tir
9	Pera Manca	1995	Tir
10	Quinta de Cabriz	2005	Tir
11	Quinta de Cabriz Reserva	2004	Tir
12	Casa de Santar	2006	Bra
13	Casa de Santar	2004	Tir
14	Casa de Santar Reserva	2004	Tir

UPDATE

Syntax:

UPDATE *table_name*

SET { *column_name* = {
expression | DEFAULT |
NULL };

UPDATE Produtor

SET Telefone= '259351397'

WHERE Produtor_ID = 6;

UPDATE Produtor

SET Telefone= **CONCAT**('+351',Telefone)

WHERE Telefone <>";

Antes

Nome_Produtor	Telefone
Qt. Vallado	254323147
Qt. Pacheca	254313228
Finagra	266509270
Dão Sul	232960140
Fund. Eug. Almeida	NULL
Kolheita Ideias	NULL
Anselmo Mendes	NULL
Adriano Ramos Pinto	223707000
Domingos Alves Sousa	254822111
Provam	251534207

Nome_Produtor	Telefone
Qt. Vallado	+351254323147
Qt. Pacheca	+351254313228
Finagra	+351266509270
Dão Sul	+351232960140
Fund. Eug. Almeida	NULL
Kolheita Ideias	+351259351397
Anselmo Mendes	NULL
Adriano Ramos Pinto	+351223707000
Domingos Alves Sousa	+351254822111
Provam	+351251534207

Depois

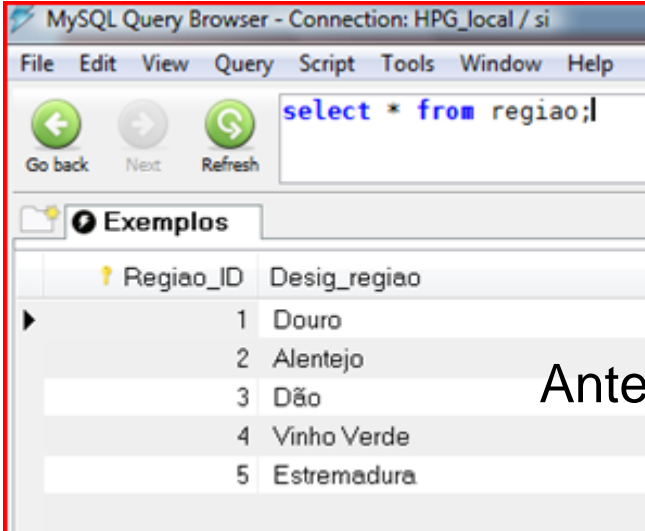
DELETE

Syntax:

`DELETE FROM`
table_name

`WHERE condition;`

`DELETE FROM regiao`
`WHERE Regiao_ID = 5;`



MySQL Query Browser - Connection: HPG_local / si

File Edit View Query Script Tools Window Help


Go back Next Refresh

`select * from regiao;`

Exemplos

Regiao_ID	Desig_regiao
1	Douro
2	Alentejo
3	Dão
4	Vinho Verde
5	Estremadura

Antes



Arquivo Editar View Query Script Ferramentas Janela Ajuda

Go back Next Refresh

`SELECT * FROM regiao r;`

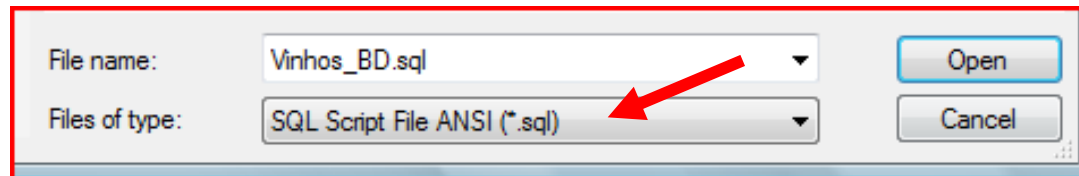
Exemplo

Regiao_ID	Desig_regiao
1	Douro
2	Alentejo
3	Dão
4	Vinho Verde

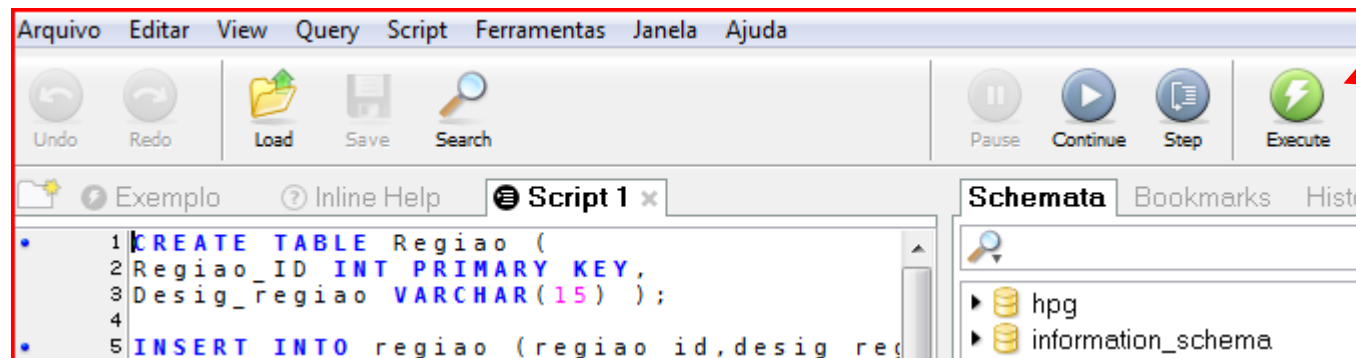
Depois

MySQL

- Para criar a BD do exemplo fazer o download do ficheiro Vinhos_BD.sql
- No menu ARQUIVO escolher OPEN SCRIPT...
- Abrir ficheiro com opção
 - file type: SQL Script File ANSI (*.sql)



- Executar script



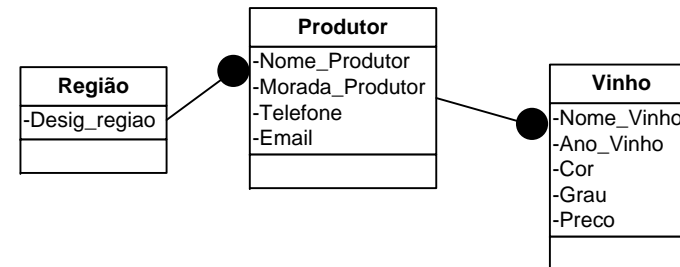
DML (Data Manipulation Language)

O comando SELECT para operações de consulta

```
SELECT select_list  
[ INTO new_table ]  
FROM table_source  
[ WHERE search_condition ]  
[ GROUP BY group_by_expression ]  
[ HAVING search_condition ]  
[ ORDER BY order_expression [ ASC | DESC ] ] ;
```

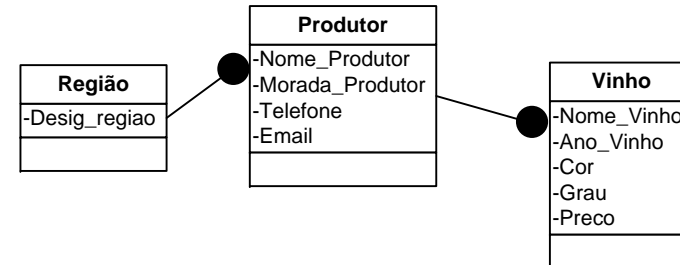
SELECT- uma tabela

SELECT * FROM Produtor;



Produtor_ID	Nome_Produtor	Morada_Produtor	Telefone	Email	Regiao_ID
1	Qt. Vallado	Régua	+351254323147	vallado@mail.telepac.pt	1
2	Qt. Pacheca	Lamego	+351254313228	pacheca@mail.telepac.pt	1
3	Finagra	Reguengos	+351266509270	esporao@esporao.com	2
4	Dão Sul	Carregal do Sal	+351232960140	daosul@daosul.com	3
5	Fund. Eug. Almeida	NULL	NULL	NULL	2
6	Kolheita Ideias	NULL	+351259351397	NULL	1
7	Anselmo Mendes	NULL	NULL	NULL	4
8	Adriano Ramos Pinto	Gaia	+351223707000	ramospinto@ramospinto.pt	1
9	Domingos Alves Sousa	Cumieira	+351254822111		1
10	Provam	Barbeita	+351251534207	pq@provam.com	2
11	Sogrape	NULL	NULL	NULL	NULL

SELECT- uma tabela

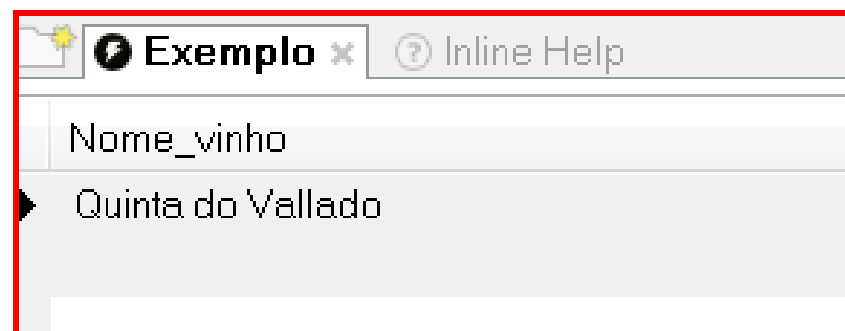
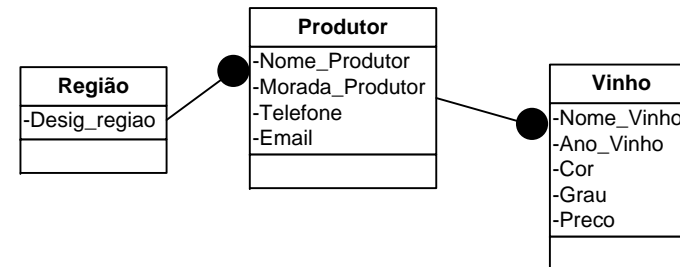


```
SELECT Nome_Produtor
FROM Produtor
WHERE Regiao_ID = 1;
```

Nome_Produtor
Qt. Vallado
Qt. Pacheca
Kolheita Ideias
Adriano Ramos Pinto
Domingos Alves Sousa

SELECT- uma tabela

```
SELECT Nome_vinho  
FROM Vinho  
WHERE Ano_vinho = 2004  
AND Grau > 13.5  
AND Preco < 10;
```



SELECT- uma tabela

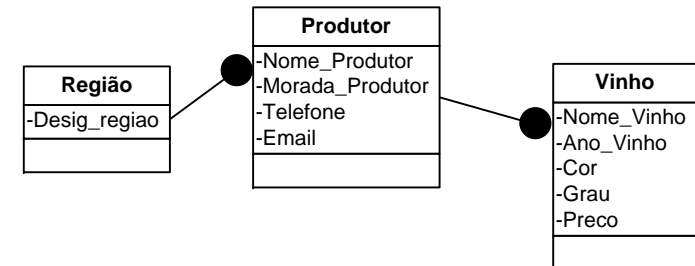
SELECT Nome_vinho, Ano_vinho, Preço

FROM Vinho

WHERE Ano_vinho = 2004

AND Preço > 12

OR Preço < 10 AND Ano_vinho <> 2004 ORDER BY Preço, Ano_vinho DESC;



Arquivo Editar View Query Script Ferramentas Janela Ajuda

Go back Next Refresh

```

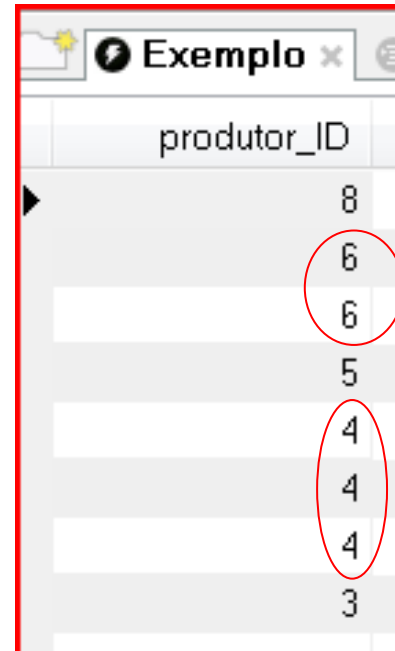
SELECT Nome_vinho, Ano_vinho, Preço
FROM Vinho
WHERE Ano_vinho = 2004
AND Preço > 12 OR Preço < 10 AND Ano_vinho <> 2004
ORDER BY Preço, Ano_vinho DESC;
  
```

Exemplo x Inline Help

Nome_vinho	Ano_vinho	Preço
▶ Casa de Santar	2006	5
EA	2006	7
Muros Antigos	2006	7.5
Esboço	2001	9.5
Esporão Reserva	2004	18.5
Quinta de Cabriz Reserva	2004	30

SELECT DISTINCT

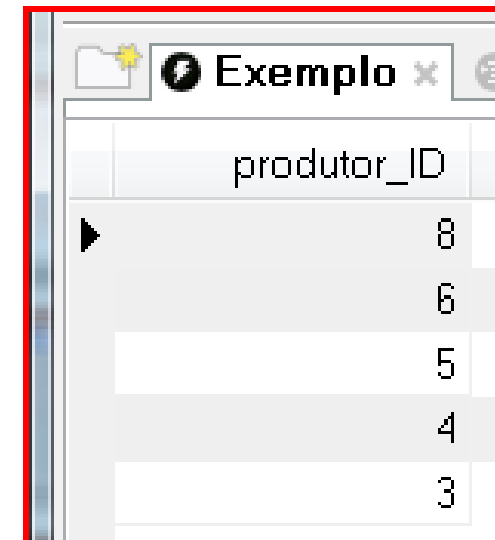
```
SELECT produtor_ID FROM Vinho  
WHERE preco > 10  
ORDER BY produtor_ID DESC;
```



A screenshot of a database query result window titled "Exemplo x". The window displays a table with the column "produtor_ID". The data is sorted in descending order. The values are 8, 6, 6, 5, 4, 4, 4, and 3. The two '6's and the three '4's are circled in red, indicating that the query returned duplicate values.

produtor_ID
8
6
6
5
4
4
4
3

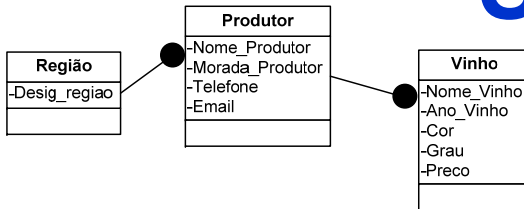
```
SELECT DISTINCT produtor_ID FROM Vinho  
WHERE preco > 10  
ORDER BY produtor_ID DESC;
```



A screenshot of a database query result window titled "Exemplo x". The window displays a table with the column "produtor_ID". The data is sorted in descending order and contains only distinct values: 8, 6, 5, 4, and 3. This result is the output of the query using the DISTINCT keyword.

produtor_ID
8
6
5
4
3

SELECT - duas tabelas



```

SELECT Nome_Produtor, Desig_regiao
FROM Produtor, Regiao
WHERE Produtor.Regiao_ID = Regiao.Regiao_ID
ORDER BY Nome_Produtor;
  
```

Exemplo x Script 7

Nome_Produtor	Desig_regiao
Adriano Ramos Pinto	Douro
Anselmo Mendes	Vinho Verde
Dão Sul	Dão
Domingos Alves Sousa	Douro
Finagra	Alentejo
Fund. Eug. Almeida	Alentejo
Kolheita Ideias	Douro
Provam	Alentejo
Qt. Pacheca	Douro
Qt. Vallado	Douro

```

SELECT Nome_Produtor, Desig_regiao
FROM Produtor, Regiao
WHERE Produtor.Regiao_ID = Regiao.Regiao_ID
ORDER BY Nome_Produtor;
  
```

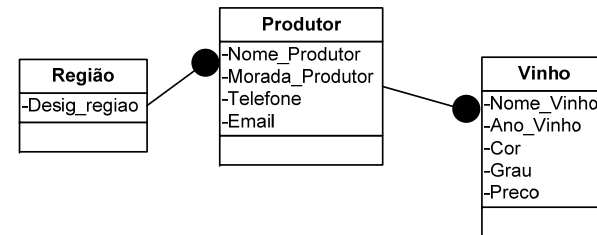
Regiao_ID	Desig_regiao
1	Douro
2	Alentejo
3	Dão
4	Vinho Verde
5	Estremadura
6	Bairrada

Resultset 1

Nome_Produtor	Desig_regiao
Adriano Ramos Pinto	Douro
Adriano Ramos Pinto	Bairrada
Adriano Ramos Pinto	Estremadura
Adriano Ramos Pinto	Vinho Verde
Adriano Ramos Pinto	Dão
Adriano Ramos Pinto	Alentejo
Anselmo Mendes	Bairrada
Anselmo Mendes	Estremadura
Anselmo Mendes	Vinho Verde
Anselmo Mendes	Dão

SELECT - duas tabelas

```
SELECT Nome_Produtor, Desig_regiao
FROM Produtor INNER JOIN Regiao
ON Produtor.Regiao_ID = Regiao.Regiao_ID
ORDER BY Nome_Produtor;
```



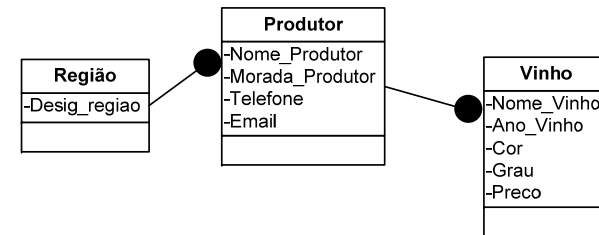
Combina linhas das duas tabelas que satisfazem a igualdade.

Equivalente ao slide anterior!

Nome_Produtor	Desig_regiao
Adriano Ramos Pinto	Douro
Anselmo Mendes	Vinho Verde
Dão Sul	Dão
Domingos Alves Sousa	Douro
Finagra	Alentejo
Fund. Eug. Almeida	Alentejo
Kolheita Ideias	Douro
Provam	Alentejo
Qt. Pacheca	Douro
Qt. Vallado	Douro

SELECT - duas tabelas

```
SELECT Nome_Produtor, Desig_regiao  
FROM Produtor LEFT JOIN Regiao  
ON Produtor.Regiao_ID = Regiao.Regiao_ID  
ORDER BY Nome_Produtor;
```

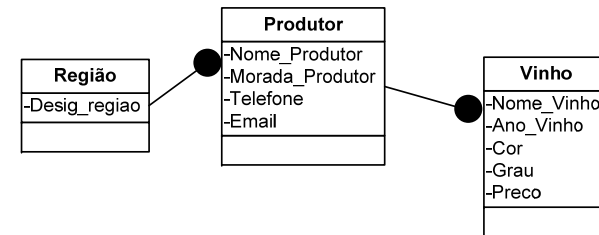


Retorna todas as linhas da tabela da esquerda e, caso existam, os valores da tabela da direita que satisfazem a igualdade.

Nome_Produtor	Desig_regiao
Adriano Ramos Pinto	Douro
Anselmo Mendes	Vinho Verde
Dão Sul	Dão
Domingos Alves Sousa	Douro
Finagra	Alentejo
Fund. Eug. Almeida	Alentejo
Kolheita Ideias	Douro
Provam	Alentejo
Qt. Pacheca	Douro
Qt. Vallado	Douro
Sogrape	NULL

SELECT - duas tabelas

```
SELECT Nome_Produtor, Desig_regiao
FROM Produtor RIGHT JOIN Regiao
ON Produtor.Regiao_ID = Regiao.Regiao_ID
ORDER BY Nome_Produtor;
```

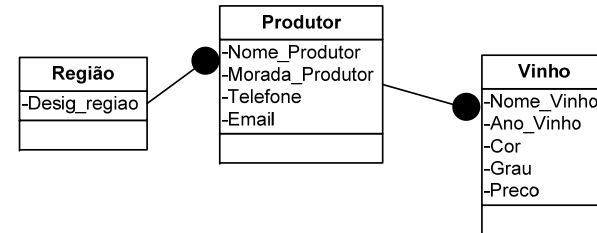


Nome_Produtor	Desig_regiao
HULL	Bairrada
HULL	Estremadura
Adriano Ramos Pinto	Douro
Anselmo Mendes	Vinho Verde
Dão Sul	Dão
Domingos Alves Sousa	Douro
Finagra	Alentejo
Fund. Eug. Almeida	Alentejo
Kolheita Ideias	Douro
Provam	Alentejo
Qt. Pacheca	Douro
Qt. Vallado	Douro

Retorna todas as linhas da tabela da direita e, caso existam, os valores da tabela da esquerda que satisfazem a igualdade.

SELECT - duas tabelas

```
SELECT Nome_Produtor, Desig_regiao
FROM Produtor FULL JOIN Regiao
ON Produtor.Regiao_ID = Regiao.Regiao_ID
ORDER BY Nome_Produtor;
```



Retorna todas as linhas da tabela da direita e todas as linhas da tabela da esquerda, combinado as linha para os casos que satisfazem a igualdade.

Nome_Produtor	Desig_regiao
NULL	Estremadura
NULL	Bairrada
Adriano Ramos Pinto	Douro
Anselmo Mendes	Vinho Verde
Dão Sul	Dão
Domingos Alves Sousa	Douro
Finagra	Alentejo
Fund. Eug. Almeida	Alentejo
Kolheita Ideias	Douro
Provam	Alentejo
Qt. Pacheca	Douro
Qt. Vallado	Douro
Sogrape	NULL

MySQL e ACCESS não suportam FULL JOIN !!!

SELECT – GROUP BY

[GROUP BY *group_by_expression*]

O resultado da cláusula GROUP BY é uma tabela em que as linhas com valores iguais para a expressão *group_by_expression* são agrupadas.

Todas as colunas da cláusula SELECT têm de estar incluídas na *group_by_expression*.

SELECT *select_list*

[INTO *new_table*]

FROM *table_source*

[WHERE *search_condition*]

[GROUP BY *group_by_expression*]

[HAVING *search_condition*]

[ORDER BY *order_expression* [ASC | DESC]] ;

SELECT – GROUP BY

```
SELECT Produtor_ID  
FROM Vinho  
ORDER BY Produtor_ID;
```

Produtor_ID
NULL
NULL
1
3
4
4
4
4
4
5
5
6
6
6
7
8

```
SELECT Produtor_ID, count(*)  
FROM Vinho  
GROUP BY Produtor_ID  
ORDER BY Produtor_ID;
```

Produtor_ID	count(*)
NULL	2
1	1
3	1
4	5
5	2
6	3
7	1
8	1



SELECT – GROUP BY

```
SELECT Produtor_ID, count(*)
FROM Vinho
GROUP BY Produtor_ID
ORDER BY Produtor_ID;
```

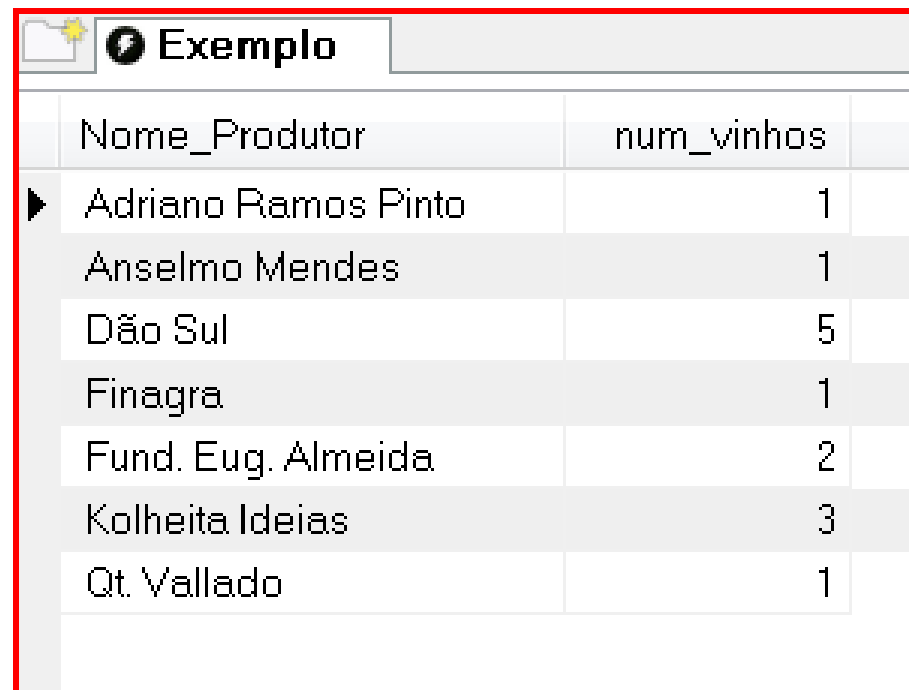
Produtor_ID	count(*)
NULL	2
1	1
3	1
4	5
5	2
6	3
7	1
8	1

```
SELECT Produtor_ID, count(Produtor_id)
AS num_vinhos
FROM Vinho
GROUP BY Produtor_ID
ORDER BY Produtor_ID;
```

Produtor_ID	num_vinhos
NULL	0
1	1
3	1
4	5
5	2
6	3
7	1
8	1

SELECT – GROUP BY

```
SELECT Nome_Produtor, COUNT(Vinho.Produutor_ID) AS num_vinhos
FROM Vinho, Produtor
WHERE Vinho.Produutor_ID = Produtor.Produutor_ID
GROUP BY Vinho.Produutor_ID, Nome_Produtor
ORDER BY Nome_Produtor;
```



Nome_Produtor	num_vinhos
Adriano Ramos Pinto	1
Anselmo Mendes	1
Dão Sul	5
Finagra	1
Fund. Eug. Almeida	2
Kolheita Ideias	3
Qt. Vallado	1

SELECT – GROUP BY

Funcões

agregadas

principais:

COUNT(),

AVG(),

MIN(),

MAX(),

STDEV(),

VARIANCE()

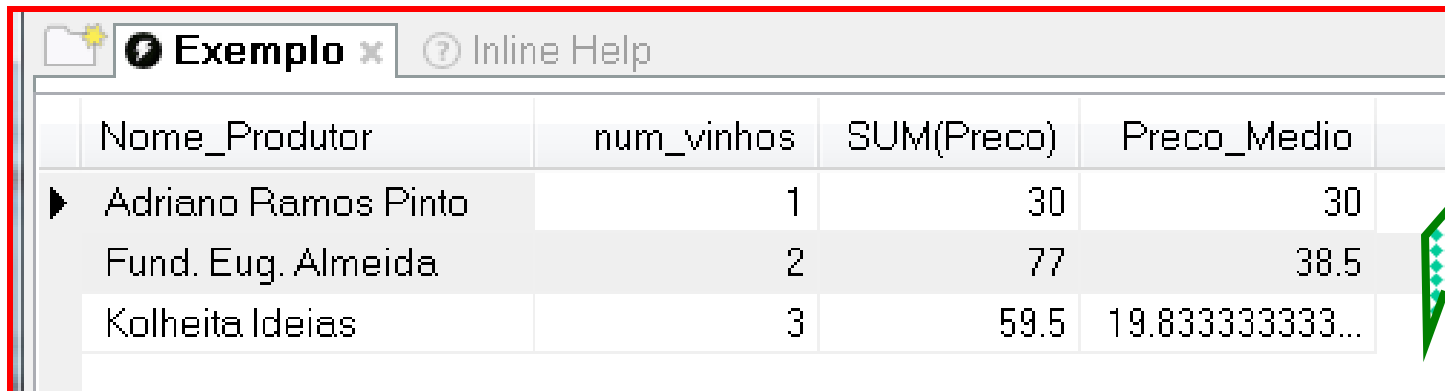
```
SELECT Nome_Produtor, COUNT(Vinho.Produtor_id) AS
num_vinhos, SUM(Preco), AVG(Preco)
FROM Vinho, Produtor
WHERE Vinho.Produtor_ID = Produtor.Produtor_ID
GROUP BY Vinho.Produtor_ID, Nome_Produtor
ORDER BY Nome_Produtor;
```

Nome_Produtor	num_vinhos	SUM(Preco)	AVG(Preco)
Adriano Ramos Pinto	1	30	30
Anselmo Mendes	1	7.5	7.5
Dão Sul	5	82	16.4
Finagra	1	18.5	18.5
Fund. Eug. Almeida	2	77	38.5
Kolheita Ideias	3	59.5	19.833333...
Qt. Vallado	1	6.5	6.5

SELECT - GROUP BY - HAVING

[*HAVING search_condition*] Elimina do resultado os grupos que não satisfazem a condição *search_condition*

```
SELECT Nome_Produtor, COUNT(Vinho.Produutor_id) AS num_vinhos,  
SUM(Preco), AVG(Preco) AS Preco_Medio  
FROM Vinho, Produtor  
WHERE Vinho.Produutor_ID = Produtor.Produutor_ID  
GROUP BY Vinho.Produutor_ID, Nome_Produtor  
HAVING Preco_Medio > 19  
ORDER BY Nome_Produtor;
```



Nome_Produtor	num_vinhos	SUM(Preco)	Preco_Medio
Adriano Ramos Pinto	1	30	30
Fund. Eug. Almeida	2	77	38.5
Kolheita Ideias	3	59.5	19.8333333333...

Comparação de strings

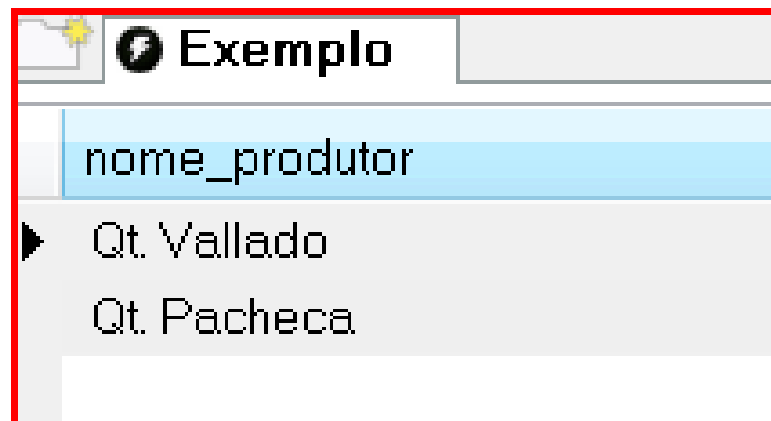
Comparação com uma string usando **LIKE**:

- '%' vale por qualquer sequência de 0 ou mais caracteres:

- Nome_Produtor **LIKE** 'Qt.%' (MySQL)

- Nome_Produtor **LIKE** 'Qt.*' (Access)

- verdadeira para



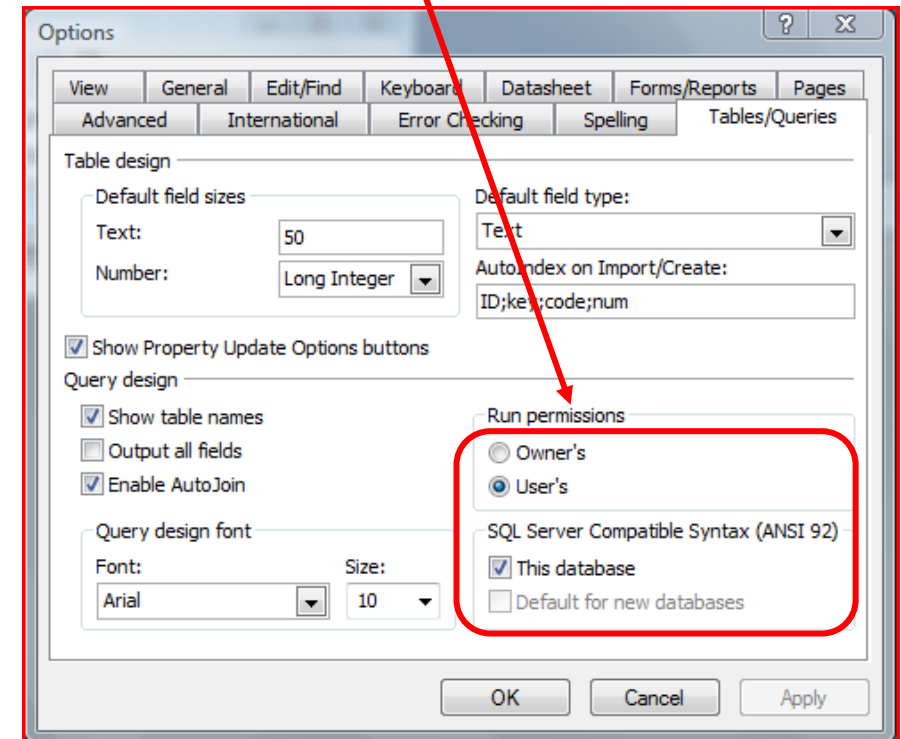
Filtro - LIKE

```
SELECT nome_produto FROM produtor  
WHERE nome_produto LIKE 'Qt%';
```

MySQL
ou
ACCESS

```
SELECT nome_produto FROM produtor  
WHERE nome_produto LIKE 'Qt*';
```

ACCESS
(default)



Comparação de strings

- Utilização de *Wildcards* com o filtro **LIKE**
- O caracter '_' vale por qualquer letra (uma e uma só);
- ('?' no Access)
 - Nome_Produtor **LIKE** 'D_o%'
 - Nome_Produtor **LIKE** 'D%o%'
- Usando = faz-se a igualdade literal:
 - Nome_Produtor = 'A_r%'
 - só é verdade se nome for 'A_r%'

Wildcard - '_'

nome_produto
Adriano Ramos Pinto
Anselmo Mendes
Dão Sul
Domingos Alves Sousa
Finagra
Fund. Eug. Almeida
Kolheita Ideias
Provam
Qt. Pacheca
Qt. Vallado
Sogrape

```
SELECT nome_produto FROM produtor
WHERE nome_produto LIKE 'D%o%';
```

nome_produto
Dão Sul
Domingos Alves Sousa

```
SELECT nome_produto FROM produtor
WHERE nome_produto LIKE 'D_o%';
```

nome_produto
Dão Sul

```
SELECT nome_produto FROM produtor
WHERE nome_produto LIKE '_o%';
```

nome_produto
Kolheita Ideias
Domingos Alves Sousa
Sogrape

Operador – IS NULL

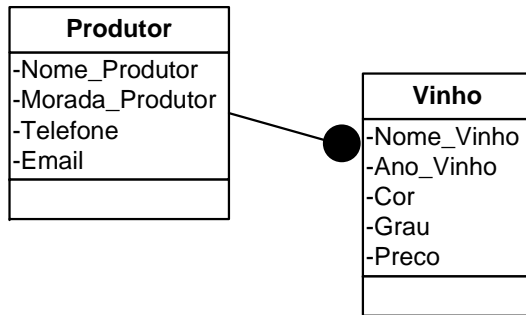
```
SELECT * FROM Produtor
WHERE Morada_Produtor IS NULL;
```

Exemplo						
Produtor_ID	Nome_Produtor	Morada_Produtor	Telefone	Email	Regiao_ID	
5	Fund. Eug. Almeida	NULL	NULL	NULL	2	
6	Kolheita Ideias	NULL	+351259351397	NULL	1	
7	Anselmo Mendes	NULL	NULL	NULL	4	
11	Sogrape	NULL	NULL	NULL	NULL	

```
SELECT * FROM Produtor
WHERE Morada_Produtor IS NOT NULL;
```

Exemplo						
Produtor_ID	Nome_Produtor	Morada_Produtor	Telefone	Email	Regiao_ID	
1	Qt. Vallado	Régua	+351254323147	vallado@mail.telepac.pt	1	
2	Qt. Pacheca	Lamego	+351254313228	pacheca@mail.telepac.pt	1	
3	Finagra	Reguengos	+351266509270	esporao@esporao.com	2	
4	Dão Sul	Carregal do Sal	+351232960140	daosul@daosul.com	3	
8	Adriano Ramos Pinto	Gaia	+351223707000	ramospinto@ramospinto.pt	1	
9	Domingos Alves Sousa	Cumieira	+351254822111		1	
10	Provam	Barbeita	+351251534207	pq@provam.com	2	

Sub-Perguntas (I)



Qual o produtor que vende o vinho mais caro?

Nome_Produtor	MAX(Preco)
Qt. Vallado	6.5
Finagra	18.5
Dão Sul	30
Fund. Eug. Almeida	70
Kolheita Ideias	30
Anselmo Mendes	7.5
Adriano Ramos Pinto	30

SELECT Nome_Produtor , MAX(Preco)

FROM Produtor,Vinho

WHERE Produtor.Produutor_ID = Vinho.Produutor_ID

GROUP BY Produtor.Produutor_ID, Produtor.Nome_Produtor;

SELECT MAX(Preco) **FROM** Vinho;

MAX(Preco)
70

Sub-Perguntas (I)

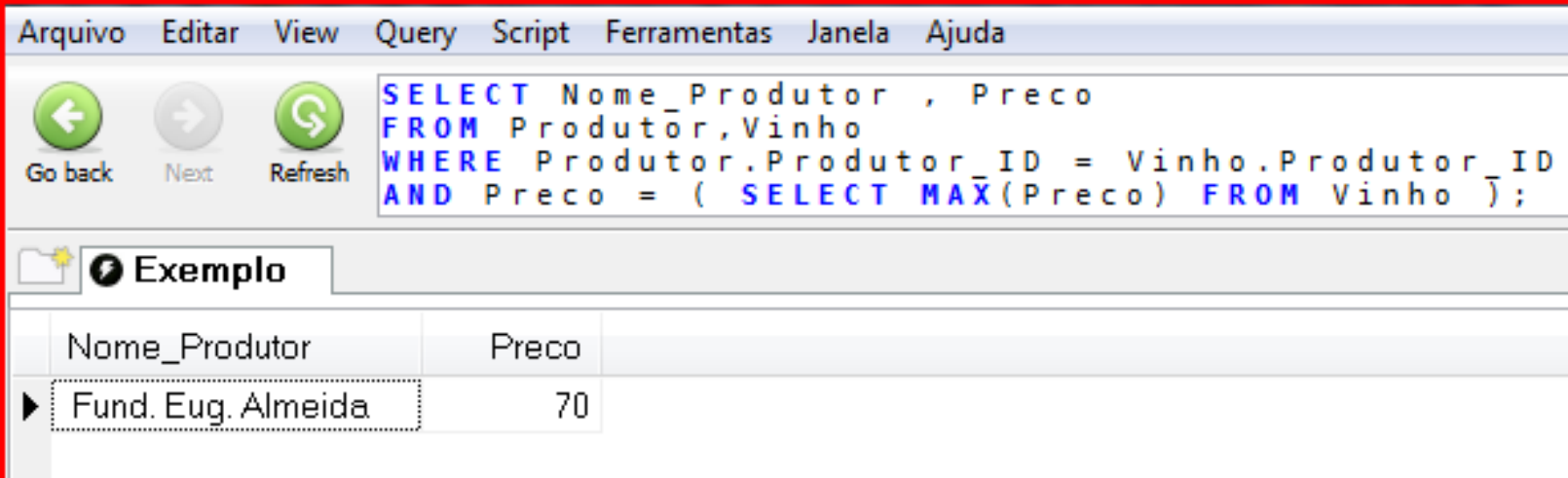
Qual o produtor que vende o vinho mais caro?

```
SELECT Nome_Produtor , Preço
```

```
FROM Produtor,Vinho
```

```
WHERE Produtor.Produtor_ID = Vinho.Produtor_ID
```

```
AND Preço = ( SELECT MAX(Preço) FROM Vinho );
```



The screenshot shows a SQL query editor window with a menu bar (Arquivo, Editar, View, Query, Script, Ferramentas, Janela, Ajuda) and navigation buttons (Go back, Next, Refresh). The query text is: `SELECT Nome_Produtor , Preço FROM Produtor,Vinho WHERE Produtor.Produtor_ID = Vinho.Produtor_ID AND Preço = (SELECT MAX(Preço) FROM Vinho);`. Below the query, a table with the title 'Exemplo' displays the result: a single row with 'Nome_Produtor' as 'Fund. Eug. Almeida' and 'Preço' as 70.

Nome_Produtor	Preço
Fund. Eug. Almeida	70

Sub-Perguntas (II)

Qual o produtor que vende o segundo vinho mais caro?

```
SELECT Nome_Produtor , Preço
```

```
FROM Produtor,Vinho
```

```
WHERE Produtor.Produtor_ID = Vinho.Produtor_ID
```

```
AND Preço =
```

```
( SELECT MAX(Preço) FROM Vinho WHERE Preço <
```

```
( SELECT MAX(Preço) FROM Vinho WHERE Preço ) );
```

Deveria usar DISTINCT!



Nome_Produtor	Preço
Dão Sul	30
Dão Sul	30
Kolheita Ideias	30
Adriano Ramos Pinto	30

Operadores de comparação

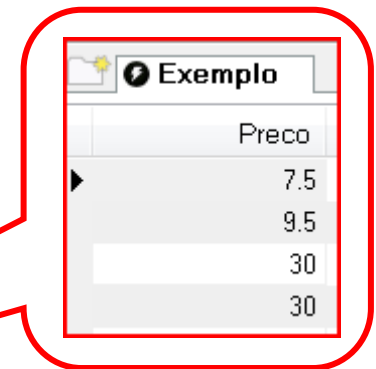
=, <>, <, >, <=, >=	igual, diferente, menor, maior, menor ou igual, maior ou igual
[NOT] BETWEEN X AND Y	$x \leq \text{valor} \leq y$
[NOT] IN	pertença a conjunto
[NOT] EXISTS	Sub-pergunta com pelo menos uma linha no resultado
ANY/SOME	
ALL	
X [NOT] LIKE Y	compara com padrão
IS [NOT] NULL	é valor nulo

Sub-Perguntas (III)

Quais os vinhos cujo preço é menor que todos os preços dos vinhos com 13 graus?

```
SELECT Nome_Produtor , Nome_Vinho, Cor, grau, Preço
FROM Produtor,Vinho
WHERE Produtor.Produtor_ID = Vinho.Produtor_ID
AND Preço < ALL
```

```
( SELECT Preço FROM Vinho WHERE Grau = 13 );
```



Preço
7.5
9.5
30
30



Nome_Produtor	Nome_Vinho	cor	grau	Preço
Qt. Vallado	Quinta do Vallado	Tinto	14	6.5
Dão Sul	Casa de Santar	Branco	12.5	5
Dão Sul	Casa de Santar	Tinto	13.5	5
Fund. Eug. Almeida	EA	Tinto	14	7

Sub-Perguntas (IV)

Quais os vinho cujo preço é menor que o preço de um vinho com 13 graus?

```
SELECT Nome_Produtor , Nome_Vinho, Cor, grau, Preço
```

```
FROM Produtor,Vinho
```

```
WHERE Produtor.Produtor_ID = Vinho.Produtor_ID
```

```
AND Preço < ANY
```

```
( SELECT Preço FROM Vinho WHERE Grau = 13 );
```

Exemplo
Preço
7.5
9.5
30
30

Nome_Produtor	Nome_Vinho	cor	grau	Preço
Qt. Vallado	Quinta do Vallado	Tinto	14	6.5
Finagra	Esporão Reserva	Tinto	14.5	18.5
Dão Sul	Casa de Santar	Branco	12.5	5
Dão Sul	Casa de Santar	Tinto	13.5	5
Dão Sul	Casa de Santar Reserva	Tinto	14	12
Fund. Eug. Almeida	EA	Tinto	14	7
Kolheita Ideias	K	Tinto	14	20
Kolheita Ideias	Esboço	Tinto	13	9.5
Anselmo Mendes	Muros Antigos	Branco	13	7.5

Sub-Perguntas (V)

Quais os vinho que usam castas que os vinhos mais caros usam?

```
SELECT DISTINCT Nome_Produtor , Nome_Vinho, Cor, grau, Preço
FROM Produtor, Vinho, Casta_Vinho
WHERE Produtor.Produtor_ID = Vinho.Produtor_ID
AND Vinho.Vinho_ID = Casta_Vinho.Vinho_ID
AND Casta_ID IN
( SELECT Casta_ID FROM Vinho, Casta_Vinho
  WHERE Vinho.Vinho_ID = Casta_Vinho.Vinho_ID
  AND Preço = (SELECT MAX(Preço) FROM Vinho) );
```



Casta_ID
6
7



Nome_Produtor	Nome_Vinho	Cor	grau	Preço
Finagra	Esporão Reserva	Tinto	14.5	18.5
Fund. Eug. Almeida	Pera Manca	Tinto	14.5	70

Sub-Perguntas (VI)

Quais os vinho que tem as castas Tinta Roriz e Touriga Nacional?

```
SELECT Vinho.Vinho_ID
```

```
FROM Vinho, Casta_Vinho
```

```
WHERE Vinho.Vinho_ID = Casta_Vinho.Vinho_ID
```

```
AND Casta_ID IN
```

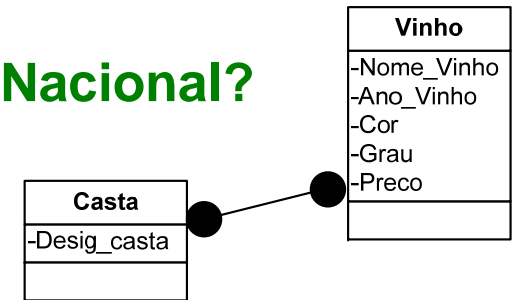
```
( SELECT Casta_ID FROM Casta
```

```
WHERE Desig_Casta = 'Tinta Roriz'
```

```
OR Desig_Casta = 'Touriga Nacional' )
```

```
Group By Vinho_ID
```

```
Having count(*) = 2;
```



Agregação de agregação

Qual o Produtor cujos vinhos tem maior graduação média?

```
SELECT Vinho.Produtor_ID, Nome_Produtor, AVG(Grau)
FROM Vinho,Produtor
WHERE Vinho.Produtor_ID = Produtor.Produtor_ID
GROUP BY Produtor_ID, Nome_Produtor
HAVING AVG(Grau) = (
    SELECT MAX(Media) FROM
        (SELECT AVG(Grau) AS Media FROM Vinho
         GROUP BY Produtor_ID) AS Tabela_Medias )
```

Restrição pela junção

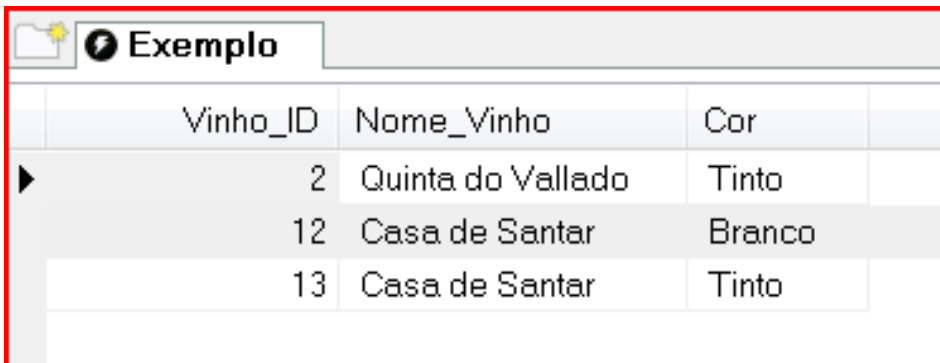
Quais os vinhos cujas castas são conhecidas e que custam menos de 7€?

```
SELECT DISTINCT Vinho.Vinho_ID , Nome_Vinho, Cor
```

```
FROM Vinho, Casta_Vinho
```

```
WHERE Vinho.Vinho_ID = Casta_Vinho.Vinho_ID
```

```
AND Preço < 7 ;
```



Vinho_ID	Nome_Vinho	Cor
2	Quinta do Vallado	Tinto
12	Casa de Santar	Branco
13	Casa de Santar	Tinto

Limita aos vinhos com as casta discriminadas.

Autojunção

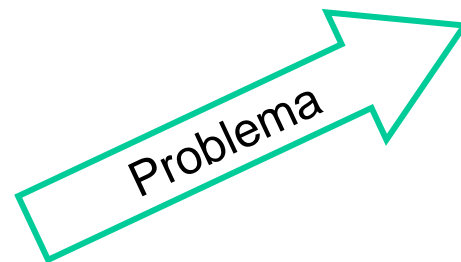
Quais os Produtores com sede na mesma região?

```
SELECT P1.Nome_Produtor, P2.Nome_Produtor
```

```
FROM Produtor AS P1, Produtor AS P2
```

```
WHERE P1.Produtor_ID <> P2.Produtor_ID
```

```
AND P1.Regiao_ID = P2.Regiao_ID;
```



Exemplo	
Nome_Produtor	Nome_Produtor
Qt. Vallado	Qt. Pacheca
Qt. Vallado	Kolheita Ideias
Qt. Vallado	Adriano Ramos Pinto
Qt. Vallado	Domingos Alves Sousa
Qt. Pacheca	Qt. Vallado
Qt. Pacheca	Kolheita Ideias
Qt. Pacheca	Adriano Ramos Pinto
Qt. Pacheca	Domingos Alves Sousa
Finagra	Fund. Eug. Almeida
Finagra	Provam
Fund. Eug. Almeida	Finagra
Fund. Eug. Almeida	Provam

Subtracção de conjuntos

Quais os Produtores que não usam a casta Touriga Nacional?

```
SELECT Nome_Produtor FROM Produtor
WHERE Produtor_ID NOT IN(
    SELECT Produtor_ID FROM Vinho, Casta_Vinho, Casta
    WHERE Vinho.Vinho_ID = Casta_Vinho.Vinho_ID
    AND Casta_Vinho.Casta_ID = Casta.Casta_ID
    AND Desig_Casta = 'Touriga Nacional' );
```

```
SELECT Produtor_ID FROM Produtor MINUS(
    SELECT Produtor_ID FROM Vinho, Casta_Vinho, Casta
    WHERE Vinho.Vinho_ID = Casta_Vinho.Vinho_ID
    AND Casta_Vinho.Casta_ID = Casta.Casta_ID
    AND Desig_Casta = 'Touriga Nacional' );
```

Nome_Produtor
Qt. Pacheca
Finagra
Fund. Eug. Almeida
Kolheita Ideias
Anselmo Mendes
Domingos Alves Sousa
Provam
Sogrape

**MySQL não
suport o
MINUS**

Reunião de conjuntos

Quais os Produtores que ou são do Alentejo ou tem vinhos com mais de 13.5 graus?

```
SELECT Produtor_ID FROM Produtor, Regiao  
WHERE Produtor.Regiao_ID = Regiao.Regiao_ID  
AND Desig_Regiao = 'Alentejo'
```

UNION

```
SELECT DISTINCT Produtor_ID  
FROM Vinho  
WHERE Grau > 13.5;
```

Exemplo	
Produtor_ID	
3	
5	
10	

Exemplo	
Produtor_ID	
1	
3	
4	
5	
6	

Exemplo	
Produtor_ID	
3	
5	
10	
1	
4	
6	

Intersecção de conjuntos

Quais os Produtores que são do Alentejo e tem vinhos com mais de 13.5 graus?

```
SELECT Produtor_ID FROM Produtor, Regiao  
WHERE Produtor.Regiao_ID = Regiao.Regiao_ID  
AND Desig_Regiao = 'Alentejo'
```

INTERSECT

```
SELECT DISTINCT Produtor_ID  
FROM Vinho  
WHERE Grau > 13.5;
```

Produtor_ID
1
3
4
5
6

Produtor_ID
3
5
10

Produtor_ID
3
5

**MySQL não
suport o
INTERSECT**

Similar INTERSECT

```
SELECT Produtor_ID FROM Produtor
WHERE Produtor_ID IN(
    SELECT Produtor_ID FROM Produtor, Regiao
    WHERE Produtor.Regiao_ID = Regiao.Regiao_ID
    AND Desig_Regiao = 'Alentejo')
AND Produtor_ID IN(
    SELECT DISTINCT Produtor_ID
    FROM Vinho
    WHERE Grau > 13.5);
```

Sumário

- SQL é uma linguagem standard para manipulação de dados em bases de dados relacionais
- Operações de: CREATE, INSERT, UPDATE, DELETE e SELECT
- SQL é uma linguagem muito poderosa e fácil de aprender o essencial.
- No entanto, para ser especialista é necessário um grande esforço.