

Aula 022

Professores:

Marta Mattoso

Vanessa Braganholo

Conteúdo:

SQL- EXISTS

Organização do Curso

- ➡ Conceitos Gerais
- ➡ SGBDs e Modelo de Dados
- ➡ Modelo ER
- ➡ Arquiteturas
- ➡ Modelo Relacional
- ➡ Álgebra Relacional
- ➡ Mapeamento ER-Relacional
- ➡ **SQL**
- ➡ Normalização
- ➡ Evolução dos Modelos

Parte 2 Modelo Relacional :

Capítulo do livro texto Elmasri/Navathe:

→ **8 - SQL - LMD:**

Linguagem de consulta:

SELECT aninhado, EXISTS, IS NULL;

Consultas aninhadas

- Uma consulta com SELECTs embutidos ou aninhados é chamada de *consulta aninhada*
- Esse tipo de consulta pode ser especificado dentro da cláusula WHERE de uma outra consulta, chamada de *consulta externa*
- Diversas das consultas anteriores podem ser especificadas de modo alternativo usando aninhamento
- Em geral, é possível haver vários níveis de consultas aninhadas

Consultas aninhadas

→ Consulta 1: Obter o nome e endereço de todos os empregados que trabalham no departamento de 'Pesquisa'.

Q1: **SELECT** PNOME, UNOME, ENDERECO
 FROM EMPREGADO
 WHERE DNO **IN** (**SELECT** DNUMERO
 FROM DEPARTAMENTO
 WHERE DNOME='Pesquisa')

Consultas aninhadas

MySQL Query Browser – root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh

```
SELECT PNAME, UNOME, ENDERECO
FROM EMPREGADO
WHERE DNO IN (SELECT DNUMERO FROM DEPARTAMENTO
                WHERE DNAME='Pesquisa')|
```

Transaction Explain Compare SELECT FROM WHERE ORDER GROUP HAVING

	PNAME	UNOME	ENDERECO
▶	John	Smith	731 Fondren, Houston, TX
	Franklin	Wong	638 Voss, Houston, TX
	Joyce	English	5631 Rice, Houston, TX
	Ramesh	Narayan	975 Fire Oak, Humble, TX

Criando as tabelas

→ Clique no botão abaixo para abrir o MySQL Query Browser

Abrir MySQL

→ Clique no botão abaixo para que o banco de dados e as tabelas sejam criados.

Executar

→ Clique no botão abaixo para que os registros exemplo sejam inseridos.

Executar

Q1- passo a passo (interna)

A consulta interna obtém o número do departamento de Pesquisa

The screenshot shows the MySQL Query Browser interface. The query window contains the following SQL code:

```
SELECT DNUMERO
FROM DEPARTAMENTO
WHERE DNAME='Pesquisa'
```

The results pane shows a single row with the value 5 in the DNUMERO column. The schema browser on the right lists tables under the EMP schema: DEPARTAMENTO, DEPENDENTE, DEPT_LOCALIZACOES, and EMPREGADO.

Q1- passo a passo (externa)

Q1: **SELECT PNAME, UNOME, ENDERECO
FROM EMPREGADO
WHERE DNO IN (...)**

A consulta externa seleciona a tupla de EMPREGADO caso o valor de seu DNO pertença ao resultado da consulta aninhada

The screenshot shows the MySQL Query Browser interface. The query window contains:

```
MySQL Query Browser - root@localhost via socket
Edit View Query Script Tools Help
SELECT DNO, PNAME, UNOME
FROM EMPREGADO
```

The results pane displays a table with columns DNO, PNAME, and UNOME. A red circle highlights the first row where DNO is 5, corresponding to the tuple (5, John, Smith) in the EMPREGADO table.

DNO	PNAME	UNOME
5	John	Smith
5	Franklin	Wong
5	Joyce	English
5	Ramesh	Narayan
1	James	Borg
4	Jennifer	Wallace
4	Ahmad	Jabbar
4	Alicia	Zelaya

The schema pane shows the database structure:

- EMP
 - DEPARTAMENTO
 - DEPENDENTE
 - DEPT_LOCALIZACOES
 - EMPREGADO

Q1- passo a passo

```
Q1: SELECT PNOME, UNOME, ENDERECO  
      FROM EMPREGADO  
      WHERE DNO IN
```

DNUMERO
5

)

O operador de comparação **IN** compara o valor **v** com um conjunto de valores **V**, e retorna **VERDADEIRO** caso **v** seja um dos elementos de **V**

Q1- passo a passo

The screenshot shows the Oracle SQL Developer interface. On the left is a toolbar with 'File', 'Edit', 'View', 'Query', 'Script', 'Go Back', 'Go Next', 'Refresh', and 'Transaction' buttons. Below the toolbar is a results grid displaying data from the 'EMPREGADO' table. The columns are DNO, PNOME, and UNOME. The data rows are:

	DNO	PNOME	UNOME
→	5	John	Smith
5	Franklin	Wong	
5	Joyce	English	
5	Ramesh	Narayan	
1	James	Borg	
4	Jennifer	Wallace	
4	Ahmad	Jabbar	
→	4	Alicia	Zelaya

To the right of the results grid is a query editor window containing the following SQL code:

```
SELECT PNOME, UNOME, ENDERECO  
FROM EMPREGADO  
WHERE DNO IN ( DNUMERO )
```

The 'DNUMERO' placeholder in the WHERE clause is highlighted and has a dropdown menu open, showing the value '5' selected.

On the far right, there is a schema browser pane showing the structure of the 'EMP' table:

- EMP
 - DEPARTAMENTO
 - DEPENDENTE
 - DEPT_LOCALIZACOES
 - EMPREGADO

At the bottom of the interface are tabs for 'Syntax', 'Functions', 'Params', and 'Trx'. The 'Syntax' tab is currently active.

Q1- passo a passo

**SELECT PNOME, UNOME, ENDERECO
FROM EMPREGADO
WHERE DNO IN (**

DNO	PNOME	UNOME
5	John	Smith
5	Franklin	Wong
5	Joyce	English
5	Ramesh	Narayan
1	James	Borg
4	Jennifer	Wallace
4	Ahmad	Jabbar
4	Alicia	Zelaya

)

The screenshot shows a database interface with a query window containing the SQL code. A red oval highlights the value '5' in the 'DNO' column of the table, indicating it is the value being selected. The interface includes a toolbar with 'Go Back', 'Go Next', and 'Refresh' buttons, and a transaction control section with buttons for play, stop, and refresh.

Syntax Functions Params Trx

Q1 - resultado

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh Execute

```
SELECT PNAME, UNOME, ENDERECO
FROM EMPREGADO
WHERE DNO IN (SELECT DNUMERO
                FROM DEPARTAMENTO
                WHERE DNAME='Pesquisa')
```

Transaction Explain Compare

	PNAME	UNOME	ENDERECO
▶	John	Smith	731 Fondren, Houston, TX
	Franklin	Wong	638 Voss, Houston, TX
	Joyce	English	5631 Rice, Houston, TX
	Ramesh	Narayan	975 Fire Oak, Humble, TX

Schemata Bookmarks History

- EMP
 - DEPARTAMENTO
 - DEPENDENTE
 - DEPT_LOCALIZACOES
 - EMPREGADO
 - EQUIPAMENTO

Consultas aninhadas correlacionadas

- Caso a condição da cláusula WHERE da *consulta interna* referencia um atributo de uma relação declarada na *consulta externa*, as duas consultas são ditas **correlacionadas**
- O resultado de uma consulta aninhada correlacionada é *diferente para cada tupla (ou combinação de tuplas) da relação(ões) da consulta externa*

Consultas aninhadas correlacionadas

- Caso a condição da cláusula WHERE da consulta interna referencie um atributo de uma relação decida por uma consulta externa, as duas consultas são ditas **correlacionadas**
- O resultado de uma consulta aninhada correlacionada é *diferente para cada tupla (ou combinação de tuplas) da relação(ões) da consulta externa*

No ex. anterior o resultado da consulta interna era fixo. Independende das tuplas da relação da consulta externa

Consultas aninhadas correlacionadas

→ Consulta Qc: Obter o nome de cada empregado que possui um dependente com o mesmo sexo do empregado (o responsável).

```
Qc: SELECT E.PNOME, E.UNOME  
      FROM EMPREGADO AS E  
      WHERE E.SSN IN (SELECT ESSN  
                      FROM DEPENDENTE  
                     WHERE ESSN=E.SSN  
                       AND E.SEXO=SEXO )
```

Aqui no SELECT interno há referência a uma tabela que só aparece no SELECT externo

Consultas correlacionadas

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh

```
SELECT PNAME, UNOME, SEXO
FROM EMPREGADO AS E
WHERE SSN IN (SELECT ESSN
               FROM DEPENDENTE
               WHERE ESSN=E.SSN AND E.SEXO=SEXO)
```

Transaction Explain Compare

SELECT FROM WHERE ORDER GROUP

	PNAME	UNOME	SEXO
▶	John	Smith	M
	Franklin	Wong	M

Schemata Bookmarks History

EMP

- DEPARTAMENTO
- DEPENDENTE

A screenshot of the MySQL Query Browser application. The title bar reads "MySQL Query Browser - root@localhost via socket". The menu bar includes File, Edit, View, Query, Script, Tools, and Help. Below the menu is a toolbar with icons for Go Back, Go Next, Refresh, Transaction, Explain, Compare, and various SQL keywords like SELECT, FROM, WHERE, ORDER, and GROUP. The main area contains a query editor with the following SQL code:

```
SELECT PNAME, UNOME, SEXO
FROM EMPREGADO AS E
WHERE SSN IN (SELECT ESSN
               FROM DEPENDENTE
               WHERE ESSN=E.SSN AND E.SEXO=SEXO)
```

Below the query editor is a table with columns PNAME, UNOME, and SEXO. It displays two rows of data: one for John Smith (M) and one for Franklin Wong (M). To the right of the table is a sidebar titled "Schemata" which shows the database structure with tables EMP, DEPARTAMENTO, and DEPENDENTE.

Consulta Qc: Obter o nome e sexo de cada empregado que possui um dependente com o mesmo sexo do empregado (o responsável).

Consultas correlacionadas

EMPREGADO

PNAME	SSN	SEXO
John	123456789	M
Franklin	333445555	M
Joyce	453453453	F
Ramesh	666884444	M
James	888665555	M
Jennifer	987654321	F
Ahmad	987987987	M
Alicia	999887777	F

DEPENDENTE

ESSN	SEXO
123456789	F
123456789	F
123456789	M
333445555	F
333445555	F
333445555	M
987654321	M

Consultas correlacionadas, como Qc, geram um resultado diferente no select interno para cada tupla da relação externa.

Qc - passo 1

EMPREGADO



PNAME	SSN	SEXO
John	123456789	M
Franklin	333445555	M
Joyce	453453453	F
Ramesh	666884444	M
James	888665555	M
Jennifer	987654321	F
Ahmad	987987987	M
Alicia	999887777	F

DEPENDENTE

ESSN	SEXO
123456789	F
123456789	F
123456789	M
333445555	F
333445555	F
333445555	M
987654321	M

Tuplas da relação interna que satisfazem a condição de junção:
 $ESSN=E.SSN$
 Para
 $E.SSN=123456789$

Consultas correlacionadas, como Qc, geram um resultado diferente no select interno para cada tupla da relação externa.

Qc - passo 1

EMPREGADO



PNAME	SSN	SEXO
John	123456789	M
Franklin	333445555	M
Joyce	453453453	F
Ramesh	666884444	M
James	888665555	M
Jennifer	987654321	F
Ahmad	987987987	M
Alicia	999887777	F

DEPENDENTE

ESSN	SEXO
123456789	F
123456789	F
123456789	M
333445555	F
333445555	M
987654321	M

V

Consultas correlacionadas, como Qc, geram um resultado diferente no select interno para cada tupla da relação externa.

Qc - passo 1

EMPREGADO

PNAME	SSN	SEXO
John	123456789	M
Franklin	333445555	M
Joyce	453453453	F
Ramesh	666884444	M
James	888665555	M
Jennifer	987654321	F
Ahmad	987987987	M
Alicia	999887777	F



DEPENDENTE

ESSN	SEXO
123456789	F
123456789	F
123456789	M
333445555	F
333445555	F
333445555	M
987654321	M

V

Consultas correlacionadas, como Qc, geram um resultado diferente no select interno para cada tupla da relação externa.

Qc - passo 1

EMPREGADO

PNAME	SSN	SEXO
John	123456789	M
Franklin	333445555	M
Joyce	453453453	F
Ramesh	666884444	M
James	888665555	M
Jennifer	987654321	F
Ahmad	987987987	M
Alicia	999887777	F



DEPENDENTE

ESSN	SEXO
123456789	F
123456789	F
123456789	M
333445555	F
333445555	F
333445555	M
987654321	M



F

Consultas correlacionadas, como Qc, geram um resultado diferente no select interno para cada tupla da relação externa.

Consultas correlacionadas

- Consulta Qc: Obter o nome de cada empregado que possui um dependente com o mesmo sexo do empregado .
- Uma consulta escrita com o bloco `SELECT... FROM... WHERE...` aninhado e que usa as operações de comparação = ou **IN** pode ser **sempre** expressa como um único bloco de consulta. Por exemplo, Qc pode ser escrita como Qca:

Qca:

```
SELECT E.PNOME, E.UNOME, E.SEXO,
       NOME_DEPENDENTE
  FROM EMPREGADO AS E,
DEPENDENTE AS D
 WHERE E.SSN=D.ESSN
   AND E.SEXO=D.SEXO
```

Consultas correlacionadas

- Consulta Qc: Obter o nome de cada empregado que possui um dependente com o mesmo sexo do empregado .
- Uma consulta escrita com o bloco `SELECT... FROM... WHERE...` aninhado e que usa as operações de comparação = ou **IN** pode ser **sempre** expressa como um único bloco de consulta. Por exemplo, Qc pode ser escrita como Qca:

Nesse caso, atributos da "relação interna" podem fazer parte do resultado.

Qca: **SELECT** E.PNOME, E.UNOM,
 NOME_DEPENDENTE
 FROM EMPREGADO AS E,
 DEPENDENTE AS D
 WHERE E.SSN=D.ESSN
 AND E.SEXO=D.SEXO

Consultas aninhadas correlacionadas

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh

```
SELECT E.PNOME, E.UNOME, E.SEXO, NOME_DEPENDENTE
FROM EMPREGADO AS E,
DEPENDENTE AS D
WHERE E.SSN=D.ESSN
AND E.SEXO=D.SEXO
```

Transaction Explain Compare

SELECT FROM WHERE ORDER GROUP HAVING

	PNOME	UNOME	SEXO	NOME_DEPENDENTE
▶	John	Smith	M	Michael
	Franklin	Wong	M	Theodore

Schemata Bookmarks History

EMP

- DEPARTAMENTO
- DEPENDENTE
- DEPT_LOCALIZACOES

The screenshot shows the MySQL Query Browser interface. In the top-left, there's a toolbar with icons for back, forward, and refresh. The menu bar includes File, Edit, View, Query, Script, Tools, and Help. Below the menu is a query editor window containing a SELECT statement. The statement joins two tables: 'EMPREGADO' (aliased as E) and 'DEPENDENTE' (aliased as D). It filters rows where E.SSN equals D.ESSN and E.SEXO equals D.SEXO. To the right of the query editor is a results grid displaying four columns: PNOME, UNOME, SEXO, and NOME_DEPENDENTE. The first row shows 'John Smith M Michael'. The second row shows 'Franklin Wong M Theodore'. On the left side of the interface, there's a transaction control panel with buttons for play, commit, and rollback. Below that is a toolbar with icons for Explain, Compare, and various SQL keywords like SELECT, FROM, WHERE, ORDER, GROUP, and HAVING. At the bottom, there's a schema browser showing the 'EMP' database with its tables: DEPARTAMENTO, DEPENDENTE, and DEPT_LOCALIZACOES.

Consulta Qca equivalente a Qc, incluindo o nome do dependente no resultado.

Consultas aninhadas correlacionadas

→ Consulta 12: Obter o nome de cada empregado que possui um dependente com o mesmo primeiro nome do empregado (o responsável).

```
Q12: SELECT E.PNOME, E.UNOME  
      FROM EMPREGADO  
      WHERE SSN IN (SELECT ESSN  
                    FROM DEPENDENTE  
                    WHERE ESSN=SSN  
                    AND PNOME=NOME_DEPENDENTE)
```

MySQL - Q12

O resultado dessa consulta é vazio, pois não há coincidência entre os primeiros nomes

```
SELECT PNAME, UNOME  
FROM EMPREGADO  
WHERE DNO IN (SELECT ESSN  
               FROM DEPENDENTE  
              WHERE SSN=ESSN  
            AND PNAME=NOME DEPENDENTE)
```

Transaction Explain Compare SELECT FROM WHERE ORDER GR

PNAME	UNOME
-------	-------

Schemata Bookmarks F

- EMP
 - DEPARTAMENTO
 - DEPENDENTE

Consultas aninhadas correlacionadas

→ Consulta 12: escrita de forma equivalente, sem o SELECT aninhado.

Q12A:

```
SELECT E.PNOME, E.UNOME
      FROM EMPREGADO AS E,
           DEPENDENTE AS D
     WHERE E.SSN=D.ESSN
       AND E.PNOME=NOME_DEPENDENTE
```

A função EXISTS

- EXISTS é usada para verificar se o resultado de uma consulta aninhada correlacionada é vazio (não contém nenhuma tupla) ou não
- A consulta Q12 pode ser formulada de modo alternativo usando EXISTS

Q12A: **SELECT** E.PNOME, E.UNOME
FROM EMPREGADO
WHERE **EXISTS** (**SELECT** *
 FROM DEPENDENTE
 WHERE ESSN=SSN
 AND PNOME=NOME_DEPENDENTE)

A função EXISTS

MySQL Query Browser – root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh

```
SELECT PNAME, UNOME
FROM EMPREGADO
WHERE EXISTS (SELECT *
               FROM DEPENDENTE
               WHERE SSN=ESSN
               AND PNAME=NOME_DEPENDENTE)
```

Transaction Explain Compare

SELECT FROM WHERE ORDER GROUP HAVING

PNAME	UNOME
-------	-------

Schemata Bookmarks History

EMP

- DEPARTAMENTO
- DEPENDENTE
- DEPT_LOCALIZACOES
- EMPREGADO

A função EXISTS

Modificando a consulta interna, para saber apenas quais funcionários possuem dependentes

SQL Query Browser - root@localhost via socket

File Tools Help

```
SELECT PNAME, UNOME  
FROM EMPREGADO  
WHERE EXISTS (SELECT *  
               FROM DEPENDENTE  
              WHERE SSN=ESSN)
```

Execute

Transaction Explain Compare

SELECT FROM WHERE ORDER GROUP HAVING

PNAME	UNOME
John	Smith
Franklin	Wong
Jennifer	Wallace

Schemata Books

- EMP
 - DEP
 - DEPT

A função EXISTS

Com a função EXISTS negada, o resultado exibe apenas os funcionários que NÃO possuem dependentes

MySQL Query Browser – root@localhost via socket

Tools Help

```
SELECT PNAME, UNOME
FROM EMPREGADO
WHERE NOT EXISTS (SELECT *
                   FROM DEPENDENTE
                   WHERE SSN=ESSN )
```

Transaction



Explain

Compare



	PNAME	UNOME
▶	Joyce	English
	Ramesh	Narayan
	James	Borg
	Ahmad	Jabbar
	Alicia	Zelaya

Schemata Bookmarks History



EMP

- ▶ DEPARTAMENTO
- ▶ DEPENDENTE
- ▶ DEPT_LOCALIZACOES
- ▶ EMPREGADO

A função EXISTS

- EXISTS é usada para verificar se o resultado de uma consulta aninhada correlacionada é vazio (não contém nenhuma tupla) ou não
- A consulta Q6 obtém o nome dos empregados que não tem dependentes com NOT EXISTS

```
Q6: SELECT E.PNOME, E.UNOME
      FROM EMPREGADO
      WHERE NOT EXISTS (SELECT *
                          FROM DEPENDENTE
                          WHERE ESSN=SSN )
```

A função EXISTS

- EXISTS é usada para verificar se o resultado de uma consulta aninhada correlacionada é vazio (não contém nenhuma tupla) ou não
- A consulta Q6 obtém o nome dos empregados dependentes com NOT EXISTS

Q6: `SELECT E.PNOME, E.UNOM
 FROM EMPREGADO
 WHERE NOT EXISTS (SELECT *
 FROM DEPENDENTE
 WHERE ESSN=SSN)`

Em Q6, a consulta aninhada correlacionada obtém todas as tuplas de DEPENDENTE relacionadas à tupla corrente de EMPREGADO. Se *não existe nenhuma*, a tupla de EMPREGADO é selecionada.

A função EXISTS é fundamental para o poder de expressão da SQL

Conjuntos explícitos

- É também possível usar um **conjunto de valores explícito (enumerado)** na cláusula WHERE, ao invés de uma consulta aninhada
- Consulta 13: Obter o SSN de todos os empregados que trabalham em projetos de números 1, 2, ou 3.

Q13: **SELECT** **DISTINCT** ESSN
 FROM TRABALHA_EM
 WHERE PNO **IN** (1, 2, 3)

Conjuntos explícitos

MySQL Query Browser - root@localhost

File Edit View Query Script Tools Help

Go Back Go Next Refresh Explain Transaction

```
SELECT DISTINCT ESSN, PNO
FROM TRABALHA_EM
```

	ESSN	PNO
▶	123456789	1
	123456789	2
	333445555	2
	333445555	3
	333445555	10
	333445555	20
	453453453	1
	453453453	2
	666884444	3
	888665555	20
	987654321	20
	987654321	30
	987987987	10
	987987987	30
	999887777	10
	999887777	30

MySQL Query Browser - root@localhost

File Edit View Query Script Tools Help

Go Back Go Next Refresh Explain Transaction

```
SELECT DISTINCT ESSN
FROM TRABALHA_EM
WHERE PNO IN (1, 2, 3)
```

	ESSN
▶	123456789
	453453453
	333445555
	666884444

Uso de NULL em SQL

- SQL permite que a consulta verifique se o valor de um atributo é NULL (ausente ou indefinido ou não se aplica)
- SQL usa **IS** ou **IS NOT** para comparar NULLs pois considera que cada valor NULL é distinto de outros valores NULL, assim, comparação via igualdade não é apropriado.

Uso de NULL em SQL

→ Consulta 14: Obter os nomes de todos os empregados que não possuem supervisores.

```
Q14: SELECT      PNOME, UNOME  
       FROM        EMPREGADO  
       WHERE       SUPERSSN IS NULL
```

Nota: Quando há atributos, numa condição de junção, que possuem valor NULL, as tuplas desses valores não são incluídas no resultado da junção

Variações em junções

- A relação operando da junção pode ser especificada na cláusula FROM
- Parece com qualquer outra relação, mas é o resultado de uma junção
- Permite a especificação de junções de tipos diferentes (por ex., JUNÇÃO NATURAL)

Variações em junções

A condição de junção implícita é DNUMERO=DNUMERO

The screenshot shows the MySQL Query Browser interface. A tooltip box highlights the implicit join condition "DNUMERO=DNUMERO". The query window displays the following SQL code:

```
MySQL Query Browser - root@localhost via socket
File Tools Help
SELECT *
FROM DEPT_LOCALIZACOES
NATURAL JOIN DEPARTAMENTO
```

The results pane shows the following data:

	DNUMERO	DLOCALIZACAO	DNOME	GERSSN	GERDATAINICIO
▶	1	Houston	Sede administrativa	888665555	1981-06-19
	4	Staffor	Administração	987654321	1995-01-01
	5	Bellaire	Pesquisa	333445555	1988-05-22
	5	Sugarland	Pesquisa	333445555	1988-05-22

Variações em junções

MySQL Query Browser – root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh

```
SELECT DNAME, DLOCALIZACAO, DEPT_LOCALIZACOES.DNUMERO  
FROM DEPT_LOCALIZACOES NATURAL JOIN DEPARTAMENTO
```

Transaction Explain Compare

SELECT FROM WHERE ORDER GROUP HAVING

	DNAME	DLOCALIZACAO	DNUMERO
▶	Sede administrativa	Houston	1
	Administração	Staffor	4
	Pesquisa	Bellaire	5
	Pesquisa	Sugarland	5

Variações em junções

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh

```
SELECT DNAME, DLOCALIZACAO, DEPT_LOCALIZACOES.DNUMERO
FROM DEPT_LOCALIZACOES NATURAL JOIN DEPARTAMENTO
WHERE DNAME='Pesquisa'
```

Transaction Explain Compare

SELECT FROM WHERE ORDER GROUP HAVING

	DNAME	DLOCALIZACAO	DNUMERO
▶	Pesquisa	Bellaire	5
	Pesquisa	Sugarland	5

Variações em junções

→ A consulta Q2 pode ser re-escrita como:

```
Q2:  SELECT  PNUMERO, DNUM, UNOME,  
          DATANASC, ENDERECO  
FROM PROJETO JOIN DEPARTAMENTO  
          ON DNUM=DNUMERO  
          JOIN EMPREGADO ON GERSSN=SSN  
WHERE PLOCALIZACAO='Stafford'
```

Variações em junções

→ A consulta Q2 pode ser re-escrita como:

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh

```
SELECT PNUMERO, DNUM, UNOME, DATANASC, ENDERECO
FROM PROJETO JOIN DEPARTAMENTO ON DNUM=DNUMERO
JOIN EMPREGADO ON GERSSN=SSN
WHERE PLOCALIZACAO='Stafford'
```

Transaction Explain Compare SELECT FROM WHERE ORDER GROUP

	PNUMERO	DNUM	UNOME	DATANASC	ENDERECO
▶	10	4	Wallace	1941-06-20	291 Berry, Bellaire, TX
	30	4	Wallace	1941-06-20	291 Berry, Bellaire, TX

Variações em junções

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh

```
SELECT PNUMERO, DNUM, UNOME, DATANASC, ENDERECO
FROM PROJETO JOIN DEPARTAMENTO ON DNUM=DNUMERO
JOIN EMPREGADO ON DNUMERO=DNO
WHERE ENDERECO LIKE '%Houston, TX'
```

Transaction Explain Compare SELECT FROM WHERE ORDER GROUP

	PNUMERO	DNUM	UNOME	DATANASC	ENDERECO
▶	20	1	Borg	1937-11-10	450 Stone, Houston, TX
	10	4	Jabbar	1969-03-29	980 Dallas, Houston, TX
	30	4	Jabbar	1969-03-29	980 Dallas, Houston, TX
	1	5	Smith	1965-01-09	731 Fondren, Houston, TX
	1	5	Wong	1955-12-08	638 Voss, Houston, TX
	1	5	English	1972-07-31	5631 Rice, Houston, TX
	2	5	Smith	1965-01-09	731 Fondren, Houston, TX
	2	5	Wong	1955-12-08	638 Voss, Houston, TX
	2	5	English	1972-07-31	5631 Rice, Houston, TX
	3	5	Smith	1965-01-09	731 Fondren, Houston, TX
	3	5	Wong	1955-12-08	638 Voss, Houston, TX
	3	5	English	1972-07-31	5631 Rice, Houston, TX

Variações com LIKE

Aqui é possível fixar a posição do caractere sendo procurado. Nesse exemplo são selecionados os que nasceram em janeiro (formato americano: AAAA-MM-DD)

The screenshot shows the MySQL Query Browser interface. The query window displays the following SQL code:

```
MySQL Query Browser - root@localhost via socket
Tools Help
SELECT PNUMERO, DNUM, UNOME, DATANASC, ENDERECO
FROM PROJETO JOIN DEPARTAMENTO ON DNUM=DNUMERO
JOIN EMPREGADO ON DNUMERO=DNO
WHERE DATANASC LIKE '____-01-'
```

The result set shows employees born in January:

	PNUMERO	DNUM	UNOME	DATANASC	ENDERECO
▶	10	4	Zelaya	1968-01-19	3321 Castle, Spring, TX
	30	4	Zelaya	1968-01-19	3321 Castle, Spring, TX
	1	5	Smith	1965-01-09	731 Fondren, Houston, TX
	2	5	Smith	1965-01-09	731 Fondren, Houston, TX
	3	5	Smith	1965-01-09	731 Fondren, Houston, TX

Operações Aritméticas

→ Consulta 27: Mostre o efeito de dar a todos os empregados que trabalham no projeto 'ProdutoX' um aumento de 10%.

Q27:

```
SELECT PNOME, UNOME, 1.1*SALARIO  
FROM EMPREGADO, TRABALHA_EM, PROJETO  
WHERE SSN=ESSN  
      AND PNO=PNUMBER  
      AND PNOME='ProdutoX'
```

Operações Aritméticas

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh Execute

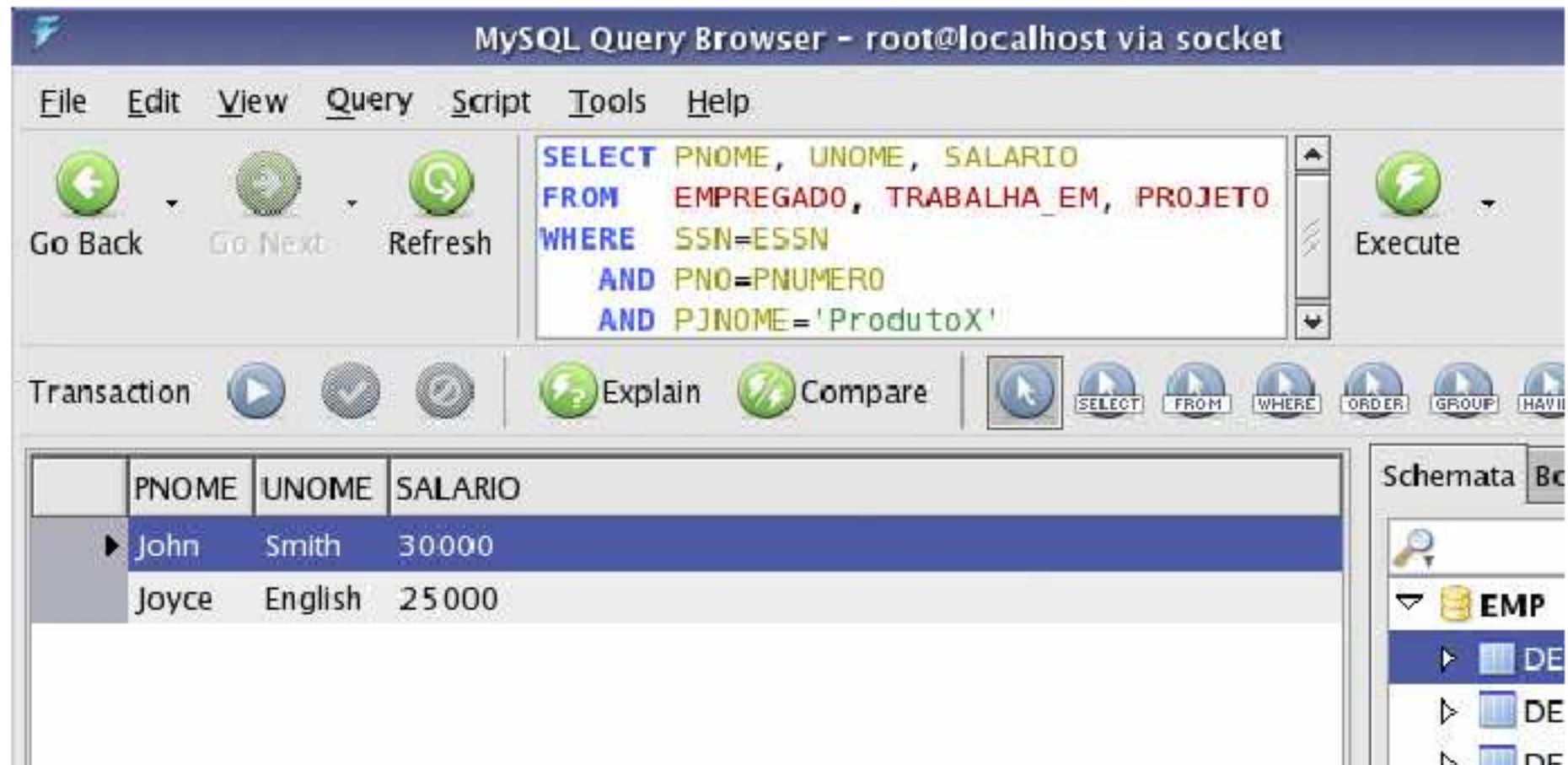
```
SELECT PNOME, UNOME, SALARIO  
FROM EMPREGADO, TRABALHA_EM, PROJETO  
WHERE SSN=ESSN  
AND PNO=PNUMERO  
AND PJNOME='ProdutoX'
```

Transaction Explain Compare SELECT FROM WHERE ORDER GROUP HAVING

	PNOME	UNOME	SALARIO
▶	John	Smith	30000
	Joyce	English	25000

Schemata

- EMP
 - DE
 - DE
 - DF



Operações Aritméticas

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh Execute

```
SELECT PNOME, UNOME, 1.1*SALARIO
FROM EMPREGADO, TRABALHA_EM, PROJETO
WHERE SSN=ESSN
AND PNO=PNUMERO
AND PJNAME='ProdutoX'
```

Transaction Explain Compare SELECT FROM WHERE ORDER GROUP HAVING

	PNOME	UNOME	1.1*SALARIO
▶	John	Smith	33000
	Joyce	English	27500

Schemata B
EMP
DE
DE

The screenshot shows the MySQL Query Browser interface. The title bar indicates the connection is to 'root@localhost via socket'. The menu bar includes File, Edit, View, Query, Script, Tools, and Help. Below the menu is a toolbar with icons for Go Back, Go Next, Refresh, and Execute. The main area contains a SQL query window with the following code:

```
SELECT PNOME, UNOME, 1.1*SALARIO
FROM EMPREGADO, TRABALHA_EM, PROJETO
WHERE SSN=ESSN
AND PNO=PNUMERO
AND PJNAME='ProdutoX'
```

Below the query window are transaction control buttons (Play, Commit, Rollback) and analysis tools (Explain, Compare). To the right of the results table are buttons for SELECT, FROM, WHERE, ORDER, GROUP, and HAVING. The results table displays three rows of data:

	PNOME	UNOME	1.1*SALARIO
▶	John	Smith	33000
	Joyce	English	27500

The schema browser on the right shows tables EMP, DE, and another DE table. The bottom of the interface has a large empty space.

Operações combinadas

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh

```
SELECT DISTINCT PNOME, UNOME, 1.1*SALARIO
FROM EMPREGADO, TRABALHA_EM, PROJETO
WHERE SSN=ESSN
    AND PNO=PNUMERO
    AND PJNOME LIKE 'Produto%'
ORDER BY SALARIO
```

Transaction ▶ Explain Compare SELECT FROM WHERE ORDER

	PNOME	UNOME	1.1*SALARIO
▶	Joyce	English	27500
	John	Smith	33000
	Ramesh	Narayan	41800
	Franklin	Wong	44000

Operações combinadas

MySQL Query Browser - root@localhost via socket

File Edit View Query Script Tools Help

Go Back Go Next Refresh

```
SELECT DISTINCT PJNOME, PNOME, UNOME, 1.1*SALARIO
FROM EMPREGADO, TRABALHA_EM, PROJETO
WHERE SSN=ESSN
AND PNO=PNUMERO
AND PJNOME LIKE 'Produto%'
ORDER BY PJNOME, SALARIO
```

Transaction Explain Compare

SELECT FROM WHERE ORDER GROUP HAVING

	PJNOME	PNOME	UNOME	1.1*SALARIO
▶	ProdutoX	Joyce	English	27500
	ProdutoX	John	Smith	33000
	ProdutoY	Joyce	English	27500
	ProdutoY	John	Smith	33000
	ProdutoY	Franklin	Wong	44000
	ProdutoZ	Ramesh	Narayan	41800
	ProdutoZ	Franklin	Wong	44000