



**Instituto Federal do Rio Grande do Sul - IFRS
Campus Rio Grande**

Curso Técnico de Informática

Programa E-Tec Brasil

Disciplina	BANCO DE DADOS
Professores	Jeison Luiz Ferreira Vieira
Tutores	Leonardo Alvarenga Pereira

1. INTRODUÇÃO A BANCO DE DADOS

Bem vindos à disciplina de Bancos de Dados.

Durante esta disciplina, iremos aprender a utilizar o MySQL. Veremos como preparar um banco de dados, como armazenar e recuperar dados de um banco de dados e ainda, como conectar uma aplicação PHP com um banco de dados.

1. Introdução

Antes de mais nada, alguma teoria.

Dado

Dado é tudo aquilo que se precisa saber para tratar um problema. Se queremos controlar os alunos de uma escola, alguns dados relevantes seriam nome, data de nascimento, sexo, endereço, telefone, nome do pai, nome da mãe. Mas cuidado para não confundir dado com informação. Durante esta disciplina, dado significa algo concreto e informação significa aquilo que se pode inferir ou calcular com base nos dados. Para facilitar o entendimento, considere o peso de uma pessoa. Isso é dado ou é informação? É dado. Com base no peso, podemos inferir informações do tipo magro ou gordo. Apenas dados devem ser armazenados. Informações não!

Banco de dados (BD)

Banco de dados é uma coleção de dados armazenados de forma organizada em um ou vários arquivos. É muito importante não confundir bancos de dados com sistemas gerenciadores de bancos de dados.

Sistema gerenciador de bancos de dados (SGBD)

É o programa (serviço) que gerencia bancos de dados. É ele que, via requisições escritas em SQL, armazena, altera, exclui e recupera dados em um banco de dados. É função dele garantir a persistência, a integridade, a consistência e a segurança dos dados armazenados nos bancos de dados.

SQL

SQL é a linguagem estruturada com a qual nos comunicamos com os SGBDs. É com ela que escrevemos requisições para incluir, alterar, excluir ou recuperar dados em um banco de dados.

Antes de irmos mais adiante, vamos entender como um SGBD funciona e como uma aplicação cliente se comunica com ele. Observe a figura 1.

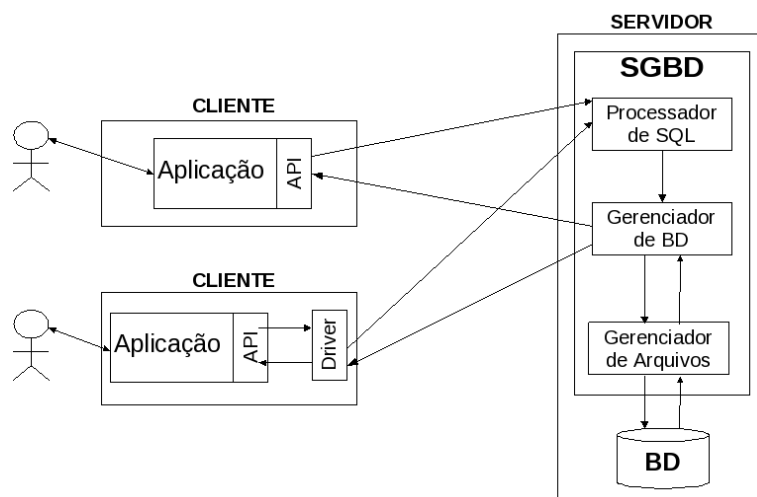


Figura 1 - Funcionamento de um SGBD

O computador no qual o SGBD está executando é chamado servidor. Os computadores que executam as aplicações que se comunicam com o SGBD são chamados clientes. Embora tenham sido representados como entidades distintas, um mesmo computador pode ser servidor e cliente ao mesmo tempo.

Primeiro, vamos entender como funciona o SGBD. O processo começa com o SGBD esperando por requisições SQL dos clientes. Quando uma requisição chega, é analisada pelo processador de SQL e uma lista dos dados necessários é repassada ao gerenciador de BD. O gerenciador de BD repassa ao gerenciador de arquivos a lista dos dados necessários. O gerenciador de arquivos localiza, recupera e devolve os dados ao gerenciador de BD. Por fim, o gerenciador de BD processa e organiza os dados conforme o requisitado pelo processador de SQL e devolve o resultado ao cliente.

Agora, vamos entender como as aplicações clientes se comunicam com os SGBDs. São duas as formas mais comuns: diretamente (via API) ou indiretamente (via API e driver). A comunicação é direta se a linguagem de programação oferece comandos específicos para a comunicação com um determinado SGBD. E é indireta se a linguagem de programação oferece comandos genéricos para comunicação com SGBDs. Neste caso, um driver deve ser utilizado como intermediário para prover a comunicação com um SGBD específico. A linguagem PHP oferece uma API específica para conexão com o MySQL.

Uma outra questão importante ainda deve ser abordada, o modelo de dados. Atualmente, o modelo mais utilizado é o relacional. Neste modelo, os dados são organizados em tabelas, linhas e colunas. O MySQL é um SGBD relacional.

Tabela

Pode-se entender uma tabela como uma coleção ordenada de registros sobre uma, e somente uma, das entidades do problema sendo tratado. À cada tabela deve ser associado a um nome único no banco de dados.

Registro ou linha ou tupla

Cada registro, ou linha, de uma tabela armazena os campos de uma, e somente uma, instância desta entidade. A cada registro deve estar associado uma chave primária.

Campo ou coluna

Cada campo, ou coluna, de uma linha armazena um, e somente um, dos dados de uma instância desta entidade. A cada campo deve ser associado um nome único e um tipo de dados.

Tipo de dados

Cada campo pode armazenar apenas determinado tipo de dado. Os tipos de dados mais comuns são números inteiros, números reais, caracteres e sequência de caracteres, data, hora e timestamp (data e hora juntos).

Chave Primária

Chave primária é uma coluna ou uma combinação de colunas cujos valores identificam um, e somente um, registro em uma tabela.

Para melhor entender o modelo, considere o exemplo:

MÉDICO

Código	Nome	Especialidade
M1	João da Silva	Pediatria
M2	José da Silva	Clínica geral
M3	Maria da Silva	Ginecologia

PACIENTE

Código	Nome	Sexo	Data de nascimento
P1	Pedro Oliveira	M	15/10/1975
P2	João Pereira	M	20/03/2000
P3	José Macieira	M	05/08/1980
P4	Maria Laranjeira	F	30/06/1995

CONSULTA

Médico	Paciente	Dia	Horário
M1	P2	20/03/2005	14:00
M3	P4	20/03/2005	14:00
M2	P1	20/03/2005	14:00
M2	P3	20/03/2005	14:45

Neste banco de dados, três entidades estão envolvidas: médicos, pacientes e consultas. À cada uma está associada uma tabela.

Na tabela MÉDICO, são mostradas três linhas, uma para cada médico da clínica. Em cada linha existem três colunas, uma para cada dado necessário sobre um médico: código, nome e especialidade. Além disso, à cada coluna ainda está associado um tipo de dados. Código, nome e especialidade são sequências de caracteres. Como chave primária, foi utilizado o código do médico.

Na tabela PACIENTE, são mostradas quatro linhas, uma para cada paciente da clínica. Em cada linha existem quatro colunas, uma para cada dado necessário sobre um paciente: código, nome, sexo e data de nascimento. Além disso, à cada coluna ainda está associado um tipo de dados. Código e nome são

seqüências de caracteres, sexo é um caracter e data de nascimento é uma data. Como chave primária, foi utilizado o código do paciente.

Na tabela CONSULTA, são mostradas quatro linhas, uma para cada consulta realizada na clínica. Em cada linha existem quatro colunas, uma para cada dado necessário sobre uma consulta: código do médico, código do paciente, dia e hora. Além disso, à cada coluna ainda está associado um tipo de dados. Código do médico e do paciente são seqüências de caracteres, dia é uma data e horário é uma hora. Como chave primária, foi utilizada a composição código do médico, código do paciente, dia e horário. Observe que foram utilizados os campos código do médico e código do paciente para identificar o médico e o paciente de cada consulta.

Mais explicações sobre aspectos dos bancos de dados relacionais, como a escolha da chave primária de cada tabela, serão abordados mais adiante, no conteúdo modelagem de dados.

Com base na tabela CONSULTA, pode-se observar que, no dia 20/03/2005, o pediatra João da Silva realizou uma consulta: João Pereira, às 14:00; a ginecologista Maria da Silva realizou uma consulta: Maria Laranjeira, às 14:00 e o clínico geral José da Silva realizou duas consultas: Pedro Oliveira, às 14:00, e José Macieira, às 14:45.

Para finalizar a parte teórica, vamos definir os papéis humanos em bancos de dados.

Usuário da aplicação

O usuário da aplicação não sabe (e nem precisa saber) sobre a existência do SGBD. Ele interage com o SGBD somente através da aplicação.

Programador da aplicação

Entende os conceitos envolvidos e o funcionamento de um SGBD. Sabe como se comunicar com eles via requisições SQL para armazenar, alterar, excluir e recuperar os dados necessários à sua aplicação.

Administrador de banco de dados

Domina os conceitos envolvidos e o funcionamento de um SGBD. Sabe como se comunicar com eles via requisições SQL para armazenar, alterar, excluir e recuperar os dados das aplicações e ainda dados sobre tabelas, integridade, usuários, segurança e desempenho.

Neste curso, vocês estarão sendo preparados para desempenhar o papel de programador de aplicação, embora algumas vezes, executem algumas tarefas de administrador de bancos de dados.

1.1. O banco de dados loja

Este banco, com cinco tabelas, controla vendedores, estoque e vendas de uma loja de produtos de informática.

tabela **comissao**: comissões dos produtos da loja.

codigo: inteiro, código da comissão

comissao: real, valor da comissão, de 0 a 100.

tabela **vendedor**: vendedores da loja.

codigo: inteiro, código do vendedor.

nome: seqüência de caracteres, nome do vendedor.

salario: real, salário fixo mensal do vendedor.

tabela **produto**: produtos da loja.

codigo: inteiro, código do produto.

descricao: sequência de caracteres, descrição do produto.

preco: real, preço do produto.

comissao: inteiro, código da comissão do produto, referencia o campo codigo da tabela comissao.

tabela **venda**: vendas da loja.

codigo: inteiro, código da venda.

vendedor: inteiro, código do vendedor que efetuou a venda, referencia o campo codigo da tabela vendedor.

dia: data, data da venda.

tabela **item**: itens das vendas da loja.

venda: inteiro, código da venda do item, referencia o campo codigo da tabela venda.

produto: inteiro, código do produto do item, referencia o campo codigo da tabela produto.

quantidade: inteiro, quantidade vendida do produto do item.

Figura 2 - Modelagem do banco de dados loja

Sobre cada vendedor são armazenados um código único que o identifica, seu nome e seu salário fixo mensal. Sobre cada comissão são armazenados um código único que a identifica e um valor de comissão entre 0 e 100. Sobre cada produto são armazenados um código único que o identifica, sua descrição, seu preço unitário para venda e o código da sua comissão, sendo que um produto pode ou não pagar comissão. Sobre cada venda são armazenados um código único que a identifica, o código do vendedor que a efetuou e a data em que foi efetuada. Sobre cada item são armazenados o código da venda a que pertence, o código do produto que foi vendido e a quantidade que foi vendida deste produto.

Para exemplificar, observe o registro de uma venda.

```
=====
Venda      00100                               15/01/2000
           Vendedor 001  Fulano
-----
005  Mouse 2 botões USB  R$  8,75 x 2 un  R$ 17,50
                                Comissao 10%  R$  1,75
045  Teclado ABNT2 USB  R$ 23,50 x 1 un  R$ 23,50
                                Comissao  5%  R$  1,18
-----
                                TOTAL R$ 41,00
                                COMISSÃO TOTAL R$  2,93
=====
```

Figura 3 - Exemplo de uma nota de venda

O vendedor Fulano (código 001) efetuou uma venda (código 00100) em 15/01/2000 com dois itens: 2 unidades de Mouse 2 botões USB (código 005), a R\$ 8,75 cada e pagando 10% de comissão e 1 unidade de Teclado ABNT2 USB (código 045), a R\$ 23,40 cada e pagando 5% de comissão. O valor total da venda foi de R\$ 41,00 e o vendedor recebeu R\$ 2,93 de comissão por ela.

2. SQL

Todos os exemplos discutidos neste material são baseados no banco de dados `loja`, que deve ter sido criado e importado anteriormente do arquivo `loja.sql`.

Para todas as requisições SQL, foi utilizado o programa `mysql`, disponível com o SGBD MySQL.

2.1. SELECT

É a instrução de recuperação de dados. Sua sintaxe mais usual é:

```
SELECT
  { * | [ DISTINCT ] { expressão | nome_coluna } [ AS nome_coluna_saída ] } [, ...]
  [ FROM origem [, ...] ]
  [ WHERE condição ]
  [ GROUP BY nome_coluna [, ...] ]
  [ HAVING condição ]
  [ { UNION | INTERSECT | EXCEPT } select ]
  [ ORDER BY { expressão | nome_coluna } [ { ASC | DESC } ] [, ...] ]
  [ LIMIT quantidade_linhas ] [ OFFSET início ]
origem pode ser:
  nome_tabela [ [ AS ] apelido_tabela ]
  ( select ) [ AS ] apelido_tabela
  origem [ NATURAL ] tipo_junção JOIN origem [ ON condição ]
tipo_junção pode ser:
  { CROSS | [ INNER ] }
  { LEFT | RIGHT | FULL } [ OUTER ]
legenda:
  [ ]   opcional
  { | } seleção
  ...   repetição
```

Figura 4 - Sintaxe do `select`

Vamos a alguns exemplos simples.

2.1.1. Usando o `select` para fazer contas:

Quando enviamos ao SGBD a requisição abaixo:

```
select 1+2*3;
```

Obtemos como resposta:

```
+-----+
| 1+2*3 |
+-----+
|      7 |
+-----+
```

A tabela 1 mostra os operadores matemáticos que podem ser utilizados:

Operador	Descrição
*	Multiplicação
/	Divisão
%	Resto
+	Adição
-	Subtração
-	Troca de sinal

Tabela 1 - Operadores matemáticos

2.1.2. Alterando o nome das colunas na resposta com `as`:

```
select 1+2*3 as resultado;
```

```
+-----+
| resultado |
+-----+
|          7 |
+-----+
```

2.1.3. Obtendo a hora do sistema:

```
select current_time;
```

```
+-----+
| current_time |
+-----+
| 13:00:00     |
+-----+
```

2.1.4. Obtendo a data do sistema:

```
select current_date;
```

```
+-----+
| current_date |
+-----+
| 2010-01-01   |
+-----+
```

2.1.5. Obtendo a data e a hora do sistema juntas:

```
select current_timestamp;
```

```
+-----+
| current_timestamp |
+-----+
| 2010-01-01 13:00:00 |
+-----+
```

Agora, vamos a alguns exemplos utilizando uma tabela.

2.1.6. Recuperando colunas individuais:

```
select r.nome from vendedor r;
```

```
+-----+
| nome |
+-----+
| Mauricio Costa Quaresma |
| Paulo Lopes Nunes |
| Patricia Menezes Silva |
| Cristiano Lopes Bezerra |
| Michele Menezes Santos |
| Cristiane Nunes Brandao |
| Pedro Rosa Loureiro |
| Jose Souza Martins |
| Gabriel Albuquerque Menezes |
| Mauricio Bezerra Lopes |
+-----+
```

Observe que o apelido `r` foi dado à tabela `vendedor` e utilizado para indicar a origem da coluna `nome`. Embora `r` tenha sido utilizado, qualquer outro apelido poderia ter sido escolhido para a tabela `vendedor`. É muito útil e comum a utilização de apelidos curtos para os nomes das tabelas, a fim de compactar a escrita de algumas instruções `select` mais complexas. Por este motivo, adotaremos a utilização de apelidos em todas as consultas envolvendo tabelas.

2.1.7. Recuperando múltiplas colunas:

```
select r.nome,r.salario from vendedor r;
```

```
+-----+-----+
| nome | salario |
+-----+-----+
| Mauricio Costa Quaresma | 500 |
| Paulo Lopes Nunes | 500 |
| Patricia Menezes Silva | 500 |
| Cristiano Lopes Bezerra | 550 |
| Michele Menezes Santos | 550 |
| Cristiane Nunes Brandao | 500 |
| Pedro Rosa Loureiro | 850 |
| Jose Souza Martins | 550 |
| Gabriel Albuquerque Menezes | 500 |
| Mauricio Bezerra Lopes | 550 |
+-----+-----+
```

2.1.8. Recuperando todas colunas:

```
select * from vendedor r;
```

```
+-----+-----+-----+
| codigo | nome | salario |
+-----+-----+-----+
| 1 | Mauricio Costa Quaresma | 500 |
| 2 | Paulo Lopes Nunes | 500 |
| 3 | Patricia Menezes Silva | 500 |
| 4 | Cristiano Lopes Bezerra | 550 |
| 5 | Michele Menezes Santos | 550 |
| 6 | Cristiane Nunes Brandao | 500 |
| 7 | Pedro Rosa Loureiro | 850 |
| 8 | Jose Souza Martins | 550 |
| 9 | Gabriel Albuquerque Menezes | 500 |
| 10 | Mauricio Bezerra Lopes | 550 |
+-----+-----+-----+
```

2.1.9. Ordenando a resposta por uma coluna em ordem crescente:

```
select * from vendedor r order by r.nome asc;
```

codigo	nome	salario
6	Cristiane Nunes Brandao	500
4	Cristiano Lopes Bezerra	550
9	Gabriel Albuquerque Menezes	500
8	Jose Souza Martins	550
10	Mauricio Bezerra Lopes	550
1	Mauricio Costa Quaresma	500
5	Michele Menezes Santos	550
3	Patricia Menezes Silva	500
2	Paulo Lopes Nunes	500
7	Pedro Rosa Loureiro	850

Neste exemplo, a resposta foi ordenada por nome.

2.1.10. Ordenando a resposta por uma coluna em ordem decrescente:

```
select * from vendedor r order by r.nome desc;
```

codigo	nome	salario
7	Pedro Rosa Loureiro	850
2	Paulo Lopes Nunes	500
3	Patricia Menezes Silva	500
5	Michele Menezes Santos	550
1	Mauricio Costa Quaresma	500
10	Mauricio Bezerra Lopes	550
8	Jose Souza Martins	550
9	Gabriel Albuquerque Menezes	500
4	Cristiano Lopes Bezerra	550
6	Cristiane Nunes Brandao	500

Neste exemplo, a resposta também foi ordenada por nome.

2.1.11. Ordenando a resposta por várias colunas:

```
select * from vendedor r order by r.salario desc,r.nome asc;
```

codigo	nome	salario
7	Pedro Rosa Loureiro	850
4	Cristiano Lopes Bezerra	550
8	Jose Souza Martins	550
10	Mauricio Bezerra Lopes	550
5	Michele Menezes Santos	550
6	Cristiane Nunes Brandao	500
9	Gabriel Albuquerque Menezes	500
1	Mauricio Costa Quaresma	500
3	Patricia Menezes Silva	500
2	Paulo Lopes Nunes	500

Neste exemplo, a resposta foi ordenada primeiro pelo salário em ordem decrescente. Após, as linhas que apresentam um mesmo valor de salário foram ordenadas pelo nome em ordem crescente. O mesmo raciocínio se aplica para casos que utilizam mais critérios.

2.1.12. Alterando a linha inicial e limitando a quantidade de linhas da resposta:

```
select * from produto p limit 5 offset 0;
```

codigo	descricao	preco	comissao
1	acessorio kit p/ vga low profile	4.2	NULL
2	adaptador dvi-vga misc xfx	2.53	NULL
3	adaptador p/ carro de dc/ac aukson inversor 100watts 220 volts	20.5	NULL
4	adaptador p/ipod klipx icc-050 car charger	7.6	NULL
5	adaptador satellite usb 2. 0 al-09	1.7	NULL

```
select * from produto p limit 5 offset 5;
```

codigo	descricao	preco	comissao
6	adaptador satellite usb 2. 0 al-10	1.8	NULL
7	adaptador data switch non power 2 in 1 port paralelo ap201n	15.5	NULL
8	adaptador p/ impressao tel100-plu trendnet usb em uso de rede	63.25	NULL
9	adaptador p/ipod yamaha yds10 dock	107	NULL
10	adaptador satellite usb 2. 0 al-08	1.6	NULL

A utilização de `limit` altera a quantidade de linhas na resposta. Sem `limit`, a resposta teria 1873 linhas e sua visualização seria impossível. Já `offset` altera a linha inicial a partir da qual a resposta é montada. O primeiro exemplo mostra as 5 primeiras linhas da resposta, enquanto que o segundo mostra as 5 linhas seguintes. Isto é chamado paginação. Sempre se deve utilizar paginação para mostrar a resposta de um `select`!

Muito cuidado com a ordem das várias partes de uma instrução `select`. Como pode ser visto na figura 4, `limit` e `offset` devem ser as últimas partes de um `select`.

2.1.13. Obtendo dados de um vendedor:

```
select * from vendedor r where (r.codigo = 1);
```

codigo	nome	salario
1	Mauricio Costa Quaresma	500

O `where` é uma das partes mais importantes do `select`. É através do `where` que especificamos condições de pesquisa, ou filtros, e obtemos um subconjunto de interesse a partir dos dados originais.

Neste exemplo, o objetivo é mostrar os dados do vendedor cujo código é 1. Assim como para um vendedor, este exemplo pode ser adaptado para obter os dados de um produto, de uma venda ou de qualquer outra entidade do problema sendo tratado.

2.1.14. Filtrando vendedores pelo salário:

```
select * from vendedor r where (r.salario = 500.00);
```

codigo	nome	salario
1	Mauricio Costa Quaresma	500
2	Paulo Lopes Nunes	500
3	Patricia Menezes Silva	500

```

|      6 | Cristiane Nunes Brandao |      500 |
|      9 | Gabriel Albuquerque Menezes |      500 |
+-----+-----+-----+

```

```
select * from vendedor r where (r.salario > 600.00);
```

```

+-----+-----+-----+
| codigo | nome           | salario |
+-----+-----+-----+
|      7 | Pedro Rosa Loureiro |      850 |
+-----+-----+-----+

```

No primeiro exemplo, foram requisitados todos os dados dos vendedores que recebem \$ 500,00 fixo por mês. Observe que a resposta contém 5 dos 10 vendedores. No segundo, foram requisitados os dados dos vendedores que recebem mais de \$ 600,00 fixo por mês. Observe que a resposta contém apenas 1 vendedor. É importante ressaltar que a quantidade de linhas da resposta de um `select` com `where` depende dos dados, e não do operador utilizado no filtro.

A tabela 2 mostra os operadores relacionais permitidos em um filtro:

Operador	Descrição
=	igual a
!=	diferente de
<	menor que
<=	menor ou igual a
>	maior que
>=	maior ou igual a

Tabela 2 - Operadores relacionais

2.1.15. Filtrando vendas pela data:

```
select * from venda v where (v.dia = '2000-01-10');
```

```

+-----+-----+-----+
| codigo | vendedor | dia      |
+-----+-----+-----+
|      96 |      8 | 2000-01-10 |
|      97 |      7 | 2000-01-10 |
|      98 |      7 | 2000-01-10 |
|      99 |      7 | 2000-01-10 |
|     100 |     10 | 2000-01-10 |
|     101 |      3 | 2000-01-10 |
|     102 |      1 | 2000-01-10 |
|     103 |      6 | 2000-01-10 |
|     104 |      9 | 2000-01-10 |
|     105 |      7 | 2000-01-10 |
|     106 |      2 | 2000-01-10 |
|     107 |      5 | 2000-01-10 |
|     108 |      1 | 2000-01-10 |
|     109 |      2 | 2000-01-10 |
|     110 |     10 | 2000-01-10 |
|     111 |      7 | 2000-01-10 |
|     112 |      4 | 2000-01-10 |
+-----+-----+-----+

```

Datas também podem ser utilizadas para filtros. Entretanto, um cuidado especial deve ser tomado. Nos SGBDs, as datas são sempre representadas como seqüências de caracteres no formato 'ano-mês-dia', e não

no formato 'dia/mês/ano'. No exemplo, foram requisitadas todas as vendas efetuadas em 10 de janeiro de 2000.

As horas e datas e horas juntas também são representados como seqüências de caracteres. O formato para horas é 'hora:minuto:segundo' e para data e hora juntas é 'ano-mês-dia hora:minuto:segundo'.

Todos os operadores relacionais da tabela 1 podem ser utilizados com datas.

```
select * from venda v where (v.dia > '2000-01-10');
```

codigo	vendedor	dia
113	3	2000-01-11
114	9	2000-01-11
115	4	2000-01-11
116	9	2000-01-11
117	8	2000-01-11
118	2	2000-01-11
119	1	2000-01-11
120	5	2000-01-11
121	10	2000-01-11
122	5	2000-01-11
123	4	2000-01-11
124	10	2000-01-11
...		
5673	7	2001-04-20
5674	9	2001-04-20
5675	10	2001-04-20
5676	1	2001-04-20
5677	3	2001-04-20
5678	7	2001-04-20
5679	7	2001-04-20
5680	2	2001-04-20
5681	4	2001-04-20
5682	1	2001-04-20
5683	8	2001-04-20
5684	1	2001-04-20
5685	6	2001-04-20
5686	3	2001-04-20

Observe que no exemplo acima, a resposta contém 5574 linhas. Nestes casos, sempre se deve utilizar paginação com `limit` e `offset`.

As reticências (...) na resposta indicam que linhas foram suprimidas para melhorar a diagramação do texto e a visualização do exemplo.

2.1.16. Filtrando vendedores por intervalo de salário:

```
select * from vendedor r where (r.salario >= 500.00) and (r.salario <= 600.00);
```

codigo	nome	salario
1	Mauricio Costa Quaresma	500
2	Paulo Lopes Nunes	500
3	Patricia Menezes Silva	500
4	Cristiano Lopes Bezerra	550
5	Michele Menezes Santos	550
6	Cristiane Nunes Brandao	500
8	Jose Souza Martins	550
9	Gabriel Albuquerque Menezes	500
10	Mauricio Bezerra Lopes	550

Para estipular um intervalo, precisamos de uma condição de pesquisa um pouco mais complexa. Necessitamos de operadores lógicos. A tabela 3 mostra os operadores lógicos permitidos em um filtro:

Operador	Descrição
and	E
or	Ou
not	Não

Tabela 3 - Operadores lógicos

No exemplo, foram requisitados todos os vendedores que ganham entre \$ 500,00 e \$ 600,00.

2.1.17. Outra forma de filtrar vendedores por intervalo de salário:

```
select * from vendedor r where (r.salario between 500.00 and 600.00);
```

	codigo	nome	salario
	1	Mauricio Costa Quaresma	500
	2	Paulo Lopes Nunes	500
	3	Patricia Menezes Silva	500
	4	Cristiano Lopes Bezerra	550
	5	Michele Menezes Santos	550
	6	Cristiane Nunes Brandao	500
	8	Jose Souza Martins	550
	9	Gabriel Albuquerque Menezes	500
	10	Mauricio Bezerra Lopes	550

Para filtros com intervalos, também pode ser utilizado o operador `between`. É importante ressaltar que em cada `between` apenas dois valores devem ser informados, o início e o fim do intervalo. Observe que a requisição SQL com `between` é equivalente a do exemplo 2.1.16 e produz exatamente a mesma resposta.

2.1.18. Filtrando vendas por intervalo de datas:

```
select * from venda v where (v.dia >= '2000-01-05') and (v.dia <= '2000-01-10');
select * from venda v where (v.dia between '2000-01-05' and '2000-01-10');
```

	codigo	vendedor	dia
	47	5	2000-01-05
	48	7	2000-01-05
	49	1	2000-01-05
	50	8	2000-01-05
	51	1	2000-01-05
	52	1	2000-01-05
	53	4	2000-01-05
	...		
	96	8	2000-01-10
	97	7	2000-01-10
	98	7	2000-01-10
	99	7	2000-01-10
	100	10	2000-01-10
	101	3	2000-01-10
	102	1	2000-01-10
	103	6	2000-01-10
	104	9	2000-01-10
	105	7	2000-01-10
	106	2	2000-01-10
	107	5	2000-01-10
	108	1	2000-01-10
	109	2	2000-01-10

110	10	2000-01-10
111	7	2000-01-10
112	4	2000-01-10

O `between` pode ser utilizado tanto com intervalos inteiros e reais quanto com intervalos de datas e horas.

2.1.19. Filtrando as vendas fora de um intervalo de datas:

```
select * from venda v where (v.dia < '2000-01-05') or (v.dia > '2000-01-10');
select * from venda v where (v.dia not between '2000-01-05' and '2000-01-10');
```

codigo	vendedor	dia
1	3	2000-01-01
2	8	2000-01-01
3	8	2000-01-01
4	6	2000-01-01
5	1	2000-01-01
6	4	2000-01-01
7	4	2000-01-01
8	9	2000-01-01
9	9	2000-01-01
10	4	2000-01-01
11	8	2000-01-01
12	1	2000-01-01
13	3	2000-01-01
...		
42	1	2000-01-04
43	4	2000-01-04
44	1	2000-01-04
45	9	2000-01-04
46	4	2000-01-04
113	3	2000-01-11
114	9	2000-01-11
115	4	2000-01-11
116	9	2000-01-11
117	8	2000-01-11
118	2	2000-01-11
119	1	2000-01-11
120	5	2000-01-11
121	10	2000-01-11
122	5	2000-01-11
123	4	2000-01-11
124	10	2000-01-11
...		
5673	7	2001-04-20
5674	9	2001-04-20
5675	10	2001-04-20
5676	1	2001-04-20
5677	3	2001-04-20
5678	7	2001-04-20
5679	7	2001-04-20
5680	2	2001-04-20
5681	4	2001-04-20
5682	1	2001-04-20
5683	8	2001-04-20
5684	1	2001-04-20
5685	6	2001-04-20
5686	3	2001-04-20

As duas requisições, `sem between` e `com between` são equivalentes e produzem exatamente a mesma resposta.

Repare que há uma descontinuidade na sequência de valores de `dia` entre 4 e 11 de janeiro de 2000. Os dados das vendas entre 5 e 10 de janeiro de 2000 não foram incluídos na resposta.

Observe ainda que, apesar da condição sem `between` ser escrita com `or`, a condição com `between` ainda é escrita com `and`. Apenas foi acrescentado o operador lógico `not` antes de `between`.

2.1.20. Listando os dados de memórias:

Para este tipo de situação, existe o operador `like`. O `like` pode ser utilizado apenas e somente em seqüências de caracteres. Mas cuidado, quando presente em um `select`, é o principal culpado pela demora da resposta. Utilize `like` com sabedoria.

```
select * from produto p where (p.descricao like 'mem');
```

codigo	descricao	preco	comissao
--------	-----------	-------	----------

A requisição acima é equivalente a:

```
select * from produto p where (p.descricao = 'mem');
```

Observe que a resposta não contém linhas, indicando que não existem produtos com descrição igual a 'mem'.

Para permitir maior flexibilidade no filtro, podemos utilizar o caracter coringa '%', que substitui qualquer seqüência de caracteres de comprimento zero ou maior.

```
select * from produto p where (p.descricao like 'mem%');
```

codigo	descricao	preco	comissao
799	mem card micro sd 4gb kingston c/ 2 adaptador	13.2	3
800	mem card micro sd 16gb markvision c/ adaptador	48.2	3
801	mem card sd 4gb kingston	12.75	3
802	mem card sd 8gb kingston high capacity class 4	20.85	3
803	mem card sd 16gb kingston high capacity class 4	42	3
804	mem card sd 32gb kingston high capacity class 4	105	3
805	mem card stick pro duo 4gb p/sony	18.6	3
806	mem card stick pro duo 8gb sandisk	28.2	3
807	mem card stick pro duo 8gb sony s/ adaptador	28.5	3
808	mem card stick pro duo 16gb markvision c/adaptador	53.75	3
809	mem ddr 512mb 333mhz corsair	21.75	3
810	mem ddr 512mb 400mhz corsair	21.75	3
811	mem ddr 512mb 400mhz kingston	20.1	3
812	mem ddr 512mb 400mhz markvision	18.6	3
813	mem ddr 1gb 400mhz corsair	38.75	3
814	mem ddr 1gb 400mhz kingston	36.5	3
815	mem ddr 1gb 400mhz markvision	34.5	3
816	mem ddr2 1gb 667mhz corsair	27.75	3
817	mem ddr2 1gb 667mhz kingston	27.6	3
818	mem ddr2 1gb 667mhz markvision	25.5	3
819	mem ddr2 1gb 800mhz corsair	28.5	3
820	mem ddr2 1gb 800mhz kingston n6	28	3
821	mem ddr2 1gb 800mhz markvision	25.5	3
822	mem ddr2 1gb 800mhz ocz	19.3	3
823	mem ddr2 1gb 1066mhz corsair dominator	36.75	3
824	mem ddr2 1gb 667mhz kingston ecc	29	3
825	mem ddr2 2gb 1066mhz corsair	61.5	3
826	mem ddr2 2gb 667mhz corsair	52.2	3
827	mem ddr2 2gb 667mhz kingston cl15	48.2	3
828	mem ddr2 2gb 667mhz markvision	47.5	3
829	mem ddr2 2gb 800mhz corsair	50.5	3
830	mem ddr2 2gb 800mhz kingston	49.5	3
831	mem ddr2 2gb 800mhz markvision	47.5	3
832	mem ddr2 4gb 1066mhz corsair dominator kit c/ 2pcs 2gb	152	3

	833		mem ddr3 2gb 1333mhz corsair		68.5		3	
	834		mem ddr3 2gb 1600mhz corsair		71.5		3	
	835		mem dimm 512 mb pc133 original brand		13.5		3	
	836		mem dimm 512 mb pc133 samsung		14.5		3	
	837		memoria para notebook 1gb pc2700 ddr kingston		46.5		3	
	838		memory card xd 2gb p/olympus		17.75		3	
+-----+-----+-----+-----+								

Utilizar o '%' no fim, significa que inicia com. No caso, 'mem%' quer dizer que inicia com 'mem'.

```
select * from produto p where (p.descricao like 'mem%');

+-----+-----+-----+-----+
| codigo | descricao | preco | comissao |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Utilizar o '%' no início, significa que termina com. No caso, '%mem' quer dizer que termina com 'mem'.

```
select * from produto p where (p.descricao like '%mem%');

+-----+-----+-----+-----+
| codigo | descricao | preco | comissao |
+-----+-----+-----+-----+
| 132 | camera foto sony dsc-s930 10.1 silver + mem 1gb | 112.5 | 3 |
| 799 | mem card micro sd 4gb kingston c/ 2 adaptador | 13.2 | 3 |
| 800 | mem card micro sd 16gb markvision c/ adaptador | 48.2 | 3 |
| 801 | mem card sd 4gb kingston | 12.75 | 3 |
| 802 | mem card sd 8gb kingston high capacity class 4 | 20.85 | 3 |
| 803 | mem card sd 16gb kingston high capacity class 4 | 42 | 3 |
| 804 | mem card sd 32gb kingston high capacity class 4 | 105 | 3 |
| 805 | mem card stick pro duo 4gb p/sony | 18.6 | 3 |
| 806 | mem card stick pro duo 8gb sandisk | 28.2 | 3 |
| 807 | mem card stick pro duo 8gb sony s/ adaptador | 28.5 | 3 |
| 808 | mem card stick pro duo 16gb markvision c/adaptador | 53.75 | 3 |
| 809 | mem ddr 512mb 333mhz corsair | 21.75 | 3 |
| 810 | mem ddr 512mb 400mhz corsair | 21.75 | 3 |
| 811 | mem ddr 512mb 400mhz kingston | 20.1 | 3 |
| 812 | mem ddr 512mb 400mhz markvision | 18.6 | 3 |
| 813 | mem ddr 1gb 400mhz corsair | 38.75 | 3 |
| 814 | mem ddr 1gb 400mhz kingston | 36.5 | 3 |
| 815 | mem ddr 1gb 400mhz markvision | 34.5 | 3 |
| 816 | mem ddr2 1gb 667mhz corsair | 27.75 | 3 |
| 817 | mem ddr2 1gb 667mhz kingston | 27.6 | 3 |
| 818 | mem ddr2 1gb 667mhz markvision | 25.5 | 3 |
| 819 | mem ddr2 1gb 800mhz corsair | 28.5 | 3 |
| 820 | mem ddr2 1gb 800mhz kingston n6 | 28 | 3 |
| 821 | mem ddr2 1gb 800mhz markvision | 25.5 | 3 |
| 822 | mem ddr2 1gb 800mhz ocz | 19.3 | 3 |
| 823 | mem ddr2 1gb 1066mhz corsair dominator | 36.75 | 3 |
| 824 | mem ddr2 1gb 667mhz kingston ecc | 29 | 3 |
| 825 | mem ddr2 2gb 1066mhz corsair | 61.5 | 3 |
| 826 | mem ddr2 2gb 667mhz corsair | 52.2 | 3 |
| 827 | mem ddr2 2gb 667mhz kingston cl5 | 48.2 | 3 |
| 828 | mem ddr2 2gb 667mhz markvision | 47.5 | 3 |
| 829 | mem ddr2 2gb 800mhz corsair | 50.5 | 3 |
| 830 | mem ddr2 2gb 800mhz kingston | 49.5 | 3 |
| 831 | mem ddr2 2gb 800mhz markvision | 47.5 | 3 |
| 832 | mem ddr2 4gb 1066mhz corsair dominator kit c/ 2pcs 2gb | 152 | 3 |
| 833 | mem ddr3 2gb 1333mhz corsair | 68.5 | 3 |
| 834 | mem ddr3 2gb 1600mhz corsair | 71.5 | 3 |
| 835 | mem dimm 512 mb pc133 original brand | 13.5 | 3 |
| 836 | mem dimm 512 mb pc133 samsung | 14.5 | 3 |
| 837 | memoria para notebook 1gb pc2700 ddr kingston | 46.5 | 3 |
| 838 | memory card xd 2gb p/olympus | 17.75 | 3 |
| 1100 | nb mem ddr 512mb 266mhz corsair | 25.2 | 3 |
| 1101 | nb mem ddr 512mb 333mhz corsair | 24.75 | 3 |
| 1102 | nb mem ddr 512mb 400mhz corsair | 24.75 | 3 |
| 1103 | nb mem ddr 1gb 333mhz corsair | 49.2 | 3 |
| 1104 | nb mem ddr 1gb 400mhz corsair | 49.2 | 3 |
| 1105 | nb mem ddr2 512mb 667mhz original brand | 3 | 3 |
| 1106 | nb mem ddr2 512mb 667mhz samsung | 3 | 3 |
```

1107	nb mem ddr2 512mb 667mhz samsung	5.9	3
1108	nb mem ddr2 512mb 667mhz toshiba/compaq	3	3
1109	nb mem ddr2 1gb 533mhz corsair	27.75	3
1110	nb mem ddr2 1gb 555mhz toshiba/compaq	14.5	3
1111	nb mem ddr2 1gb 667 pc5300 original brand (hynix)	14.5	3
1112	nb mem ddr2 1gb 667 pc5300 samsung	14.5	3
1113	nb mem ddr2 1gb 667 pc5300 toshiba/compaq	14.5	3
1114	nb mem ddr2 1gb 667 pc6400 samsung	14.5	3
1115	nb mem ddr2 1gb 667mhz corsair	27.75	3
1116	nb mem ddr2 1gb 667mhz markvision	24.2	3
1117	nb mem ddr2 1gb 667mhz ocz	19.7	3
1118	nb mem ddr2 1gb 800mhz corsair	27.75	3
1119	nb mem ddr2 1gb 800mhz markvision	22.75	3
1120	nb mem ddr2 1gb 800mhz original brand	14.5	3
1121	nb mem ddr2 2gb 667mhz corsair	50.5	3
1122	nb mem ddr2 2gb 800mhz corsair	48.5	3
1123	nb mem ddr2 2gb 800mhz markvision	44.5	3
1124	nb mem ddr2 2gb 800mhz kingston	48.5	3
1125	nb mem ddr2 4gb 800mhz corsair	198.5	3
1126	nb mem ddr3 1gb pc8500 samsung	12.5	3
1127	nb mem ddr3 2gb 1066mhz corsair	54.2	3
1534	soundbridge pinnacle roku homemusic	77	NULL

Utilizar o '%' no início e no fim, significa que contenha. No caso, '%mem%' quer dizer que contenha 'mem'.

Observe que, devido à forma de cadastro dos dados, alguns produtos que não são memórias foram selecionados.

Como o MySQL não diferencia maiúsculas e minúsculas no like, tanto faz utilizar 'mem' ou 'MEM'.

Apesar de não ser muito comum, também podemos utilizar not like em um select.

2.1.21. Listando os dados de memórias kingston:

```
select * from produto p where (p.descricao like '%mem kingston%');
```

```
+-----+-----+-----+-----+
| codigo | descricao | preco | comissao |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Embora possa parecer natural, utilizar '%mem kingston%' significa que contenha exatamente 'mem kingston', com um espaço separando as duas palavras. Observe que nenhum produto foi encontrado.

```
select * from produto p where (p.descricao like '%mem%kingston%');
```

```
+-----+-----+-----+-----+
| codigo | descricao | preco | comissao |
+-----+-----+-----+-----+
| 799 | mem card micro sd 4gb kingston c/ 2 adaptador | 13.2 | 3 |
| 801 | mem card sd 4gb kingston | 12.75 | 3 |
| 802 | mem card sd 8gb kingston high capacity class 4 | 20.85 | 3 |
| 803 | mem card sd 16gb kingston high capacity class 4 | 42 | 3 |
| 804 | mem card sd 32gb kingston high capacity class 4 | 105 | 3 |
| 811 | mem ddr 512mb 400mhz kingston | 20.1 | 3 |
| 814 | mem ddr 1gb 400mhz kingston | 36.5 | 3 |
| 817 | mem ddr2 1gb 667mhz kingston | 27.6 | 3 |
| 820 | mem ddr2 1gb 800mhz kingston n6 | 28 | 3 |
| 824 | mem ddr2 1gb 667mhz kingston ecc | 29 | 3 |
| 827 | mem ddr2 2gb 667mhz kingston cl5 | 48.2 | 3 |
| 830 | mem ddr2 2gb 800mhz kingston | 49.5 | 3 |
| 837 | memoria para notebook 1gb pc2700 ddr kingston | 46.5 | 3 |
| 1124 | nb mem ddr2 2gb 800mhz kingston | 48.5 | 3 |
+-----+-----+-----+-----+
```

A solução mais usual é substituir cada espaço entre as palavras por um '%'. '%mem%kingston%' significa que primeiro contenha 'mem' e que depois, também contenha 'kingston'.

2.1.22. Filtrando vários vendedores pelo código:

O operador `in` é muito útil quando se deseja trabalhar com um subconjunto de dados.

```
select * from vendedor r where (r.codigo in (2,4,6));
select * from vendedor r where (r.codigo = 2) or (r.codigo = 4) or (r.codigo = 6);
```

codigo	nome	salario
2	Paulo Lopes Nunes	500
4	Cristiano Lopes Bezerra	550
6	Cristiane Nunes Brandao	500

```
select * from vendedor r where (r.codigo not in (2,4,6));
select * from vendedor r where (r.codigo != 2) and (r.codigo != 4) and (r.codigo != 6);
```

codigo	nome	salario
1	Mauricio Costa Quaresma	500
3	Patricia Menezes Silva	500
5	Michele Menezes Santos	550
7	Pedro Rosa Loureiro	850
8	Jose Souza Martins	550
9	Gabriel Albuquerque Menezes	500
10	Mauricio Bezerra Lopes	550

Em ambos exemplos, as requisições com e sem `in` são equivalentes e produzem exatamente a mesma resposta.

2.1.23. Filtrando produtos sem e com comissão:

Na grande maioria das vezes, os campos das tabelas são de preenchimento obrigatório, isto é, devem ser informados na operação de inclusão. Entretanto, pode ocorrer de um campo não ser de preenchimento obrigatório. Nestes casos, os SGBDs indicam o não preenchimento com o valor `NULL` (nulo ou vazio).

É o caso da comissão do produto. Um produto pode ou não possuir comissão. Se possuir, o código da comissão correspondente será armazenado. Se não possuir, o valor `NULL` será armazenado.

Para filtrar os produtos sem comissão:

```
select * from produto p where (p.comissao is null);
```

codigo	descricao	preco	comissao
1	acessorio kit p/ vga low profile	4.2	NULL
2	adaptador dvi-vga misc xfx	2.53	NULL
3	adaptador p/ carro de dc/ac aukson inversor 100watts 220 volts	20.5	NULL
4	adaptador p/ipod klipx icc-050 car charger	7.6	NULL
5	adaptador satellite usb 2. 0 al-09	1.7	NULL
...			
1869	wir. transmissor fm for iphone 3g- 0878 (silver)	15.75	NULL
1870	wir. transmissor fm orange or-668 (gray/white)	9.2	NULL

	1871		wir. transmissor logitech mp3 p/ entrada auxiliar de som		26.5		NULL	
	1872		wireless antena hyperlink 20dbi 090 setorial hg2420p-090		312		NULL	
	1873		wireless antena hyperlink 17dbi 120 setorial hg2417p-120		154		NULL	
+-----+-----+-----+-----+-----+-----+-----+-----+								

Para filtrar os produtos com comissão:

```
select * from produto p where (p.comissao is not null) order by p.comissao asc,p.codigo asc;
```

	codigo		descricao		preco		comissao	
	46		bolsa p/notebook 15.4 case logic snm-15 gray		34		1	
	47		bolsa p/notebook 15.4 case logic snm-15f gray		34		1	
	218		car alto falante p/carro lg las-6521 160w redondo		18.45		1	
	219		car alto falante p/carro lg las-6932 200w oval		23.57		1	
	220		car alto falante p/carro roadstar rs-6906 1600w		18.19		1	
...								
	70		calculadora hp 2220 office calc 100		10.6		2	
	71		calculadora hp 2221 office basic calc 200		18.5		2	
	72		calculadora hp financeira 10 bii - manual em espanhol		26		2	
	73		calculadora hp financeira 12c gold - manual portugues		61.5		2	
	74		calculadora hp financeira 17bii - manual em portugues e espanhol		79.2		2	
...								
	11		antena parabolica 45cm c/lmb probbox		19		3	
	12		antena parabolica 55cm c/lmb probbox		22		3	
	13		antena parabolica 60cm c/lmb bm		24.5		3	
	14		antena parabolica 60cm c/lmb shop+		29		3	
	15		antena parabolica 60cm s/lmb ecopower ku60		18		3	
...								
	238		car tela p/carro roadstar rs-1720 usb/sd/tv/17/ de teto		245		4	
	356		filmadora digital jvc everio gz-mg365bu (preto)		533		4	
	427		game console playstation 2 sony scph-90006 desbloqueado		146.5		4	
	428		game console playstation 3 sony 250gb/slim/go cech-2001b		404		4	
	429		game console playstation 3 sony slim c/hd 120gb americano (travado)		373		4	
...								
	19		ar multi split hometiter 12000+9000f/c 220v 50 hz		869		5	
	20		ar multi split hometiter 2x12000f/c 220v 50 hz		922.9		5	
	21		ar multi split hometiter 2x9000f/c 220v 50 hz		862.4		5	
	22		ar multi split hometiter 2x9000f/c 220v 60 hz br		862.4		5	
	23		ar split hometiter 18000f/c 220v 50 hz		449		5	
...								

Observe que, especificamente para NULL, não podem ser utilizados os operadores relacionais (=, != nem qualquer outro).

Apesar dos exemplos utilizarem apenas um ou outro operador e uma ou outra parte do select, nada impede que tudo seja combinado em uma única requisição.

```
select * from venda v where (v.dia between '2000-01-01' and '2000-01-31') and (v.vendedor in (1,3,5)) order by v.vendedor asc,v.dia asc limit 10 offset 0;
```

Vamos, agora, escrever requisições select mais aprimoradas, utilizando novos operadores e funções. Para mais detalhes sobre estes recursos, consulte o Guia de Consulta Rápida do MySQL ou o site do MySQL.

2.1.24. Simulando um novo salário para os vendedores:

Imagine que se queira simular um aumento de 15% no salário fixo mensal dos vendedores.

```
select *,r.salario*1.05 as novo_salario from vendedor r;
```

	codigo		nome		salario		novo_salario	
	1		Mauricio Costa Quaresma		500		525	

	2	Paulo Lopes Nunes	500	525
	3	Patricia Menezes Silva	500	525
	4	Cristiano Lopes Bezerra	550	577.5
	5	Michele Menezes Santos	550	577.5
	6	Cristiane Nunes Brandao	500	525
	7	Pedro Rosa Loureiro	850	892.5
	8	Jose Souza Martins	550	577.5
	9	Gabriel Albuquerque Menezes	500	525
	10	Mauricio Bezerra Lopes	550	577.5

Observe que o salário é mostrado como um número, e não com o formato usual para valores monetários. Para isto, podemos utilizar tanto a função `round` quanto a função `format`.

```
select r.codigo,r.nome,round(r.salario,2) as salario,round(r.salario*1.05,2) as novo_salario from vendedor r;
```

	codigo	nome	salario	novo_salario
	1	Mauricio Costa Quaresma	500.00	525.00
	2	Paulo Lopes Nunes	500.00	525.00
	3	Patricia Menezes Silva	500.00	525.00
	4	Cristiano Lopes Bezerra	550.00	577.50
	5	Michele Menezes Santos	550.00	577.50
	6	Cristiane Nunes Brandao	500.00	525.00
	7	Pedro Rosa Loureiro	850.00	892.50
	8	Jose Souza Martins	550.00	577.50
	9	Gabriel Albuquerque Menezes	500.00	525.00
	10	Mauricio Bezerra Lopes	550.00	577.50

```
select r.codigo,r.nome,format(r.salario,2) as salario,format(r.salario*1.05,2) as novo_salario from vendedor r;
```

	codigo	nome	salario	novo_salario
	1	Mauricio Costa Quaresma	500.00	525.00
	2	Paulo Lopes Nunes	500.00	525.00
	3	Patricia Menezes Silva	500.00	525.00
	4	Cristiano Lopes Bezerra	550.00	577.50
	5	Michele Menezes Santos	550.00	577.50
	6	Cristiane Nunes Brandao	500.00	525.00
	7	Pedro Rosa Loureiro	850.00	892.50
	8	Jose Souza Martins	550.00	577.50
	9	Gabriel Albuquerque Menezes	500.00	525.00
	10	Mauricio Bezerra Lopes	550.00	577.50

Repare que, embora pareça que as duas respostas são iguais, não são. O resultado de `format` é uma sequência de caracteres enquanto que o de `round` é um real.

2.1.25. Simulando um novo preço para memórias Kingston:

Imagine que se queira simular um desconto de 10% no preço das memórias Kingston.

```
select p.codigo,p.descricao,round(p.preco,2) as preco,p.comissao,round(p.preco*0.9,2) as novo_preco from produto p where (p.descricao like '%mem%kingston%');
```

	codigo	descricao	preco	comissao	novo_preco
	799	mem card micro sd 4gb kingston c/ 2 adaptador	13.20	3	11.88
	801	mem card sd 4gb kingston	12.75	3	11.48
	802	mem card sd 8gb kingston high capacity class 4	20.85	3	18.77
	803	mem card sd 16gb kingston high capacity class 4	42.00	3	37.80
	804	mem card sd 32gb kingston high capacity class 4	105.00	3	94.50

	811		mem ddr 512mb 400mhz kingston		20.10		3		18.09	
	814		mem ddr 1gb 400mhz kingston		36.50		3		32.85	
	817		mem ddr2 1gb 667mhz kingston		27.60		3		24.84	
	820		mem ddr2 1gb 800mhz kingston n6		28.00		3		25.20	
	824		mem ddr2 1gb 667mhz kingston ecc		29.00		3		26.10	
	827		mem ddr2 2gb 667mhz kingston cl5		48.20		3		43.38	
	830		mem ddr2 2gb 800mhz kingston		49.50		3		44.55	
	837		memoria para notebook 1gb pc2700 ddr kingston		46.50		3		41.85	
	1124		nb mem ddr2 2gb 800mhz kingston		48.50		3		43.65	
+-----+-----+-----+-----+-----+										

2.1.26. Outras formas de filtrar vendas pela data:

Imagine que seja necessário mostrar todos os dados das vendas de fevereiro de 2000.

```
select * from venda v where (extract(month from v.dia) = 2) and (extract(year from v.dia) = 2000);
select * from venda v where (cast(v.dia as char) like '2000-02-%');
```

+-----+-----+-----+-----+			
	codigo		vendedor dia
+-----+-----+-----+-----+			
	376		9 2000-02-01
	377		1 2000-02-01
	378		6 2000-02-01
	379		7 2000-02-01
	380		4 2000-02-01
	381		10 2000-02-01
	382		1 2000-02-01
	383		4 2000-02-01
	384		6 2000-02-01
...			
	697		7 2000-02-29
	698		10 2000-02-29
	699		1 2000-02-29
	700		3 2000-02-29
	701		6 2000-02-29
	702		10 2000-02-29
+-----+-----+-----+-----+			

No primeiro `select`, a função `extract` é utilizada, primeiro, para obter apenas o mês e, depois, para obter apenas o ano da data de venda. No segundo, a data da venda é transformada em uma sequência de caracteres com `cast` para depois poder ser utilizado o `like`.

Observe que os dois `select` são equivalentes e produzem a mesma resposta.

2.1.27. Mostrando a data da venda formatada como 'dia/mês/ano':

```
select *,concat(cast(extract(day from v.dia) as char),'/',cast(extract(month from v.dia) as char),'/',
cast(extract(year from v.dia) as char)) as dia_dma from venda v;
```

+-----+-----+-----+-----+				
	codigo		vendedor dia	dia_dma
+-----+-----+-----+-----+				
	1		3 2000-01-01	1/1/2000
	2		8 2000-01-01	1/1/2000
	3		8 2000-01-01	1/1/2000
	4		6 2000-01-01	1/1/2000
	5		1 2000-01-01	1/1/2000
	6		4 2000-01-01	1/1/2000
	7		4 2000-01-01	1/1/2000
	8		9 2000-01-01	1/1/2000
	9		9 2000-01-01	1/1/2000
	10		4 2000-01-01	1/1/2000
	11		8 2000-01-01	1/1/2000
	12		1 2000-01-01	1/1/2000
	13		3 2000-01-01	1/1/2000
	14		9 2000-01-02	2/1/2000
	15		10 2000-01-02	2/1/2000

	16	9	2000-01-02	2/1/2000
	17	10	2000-01-02	2/1/2000
	18	2	2000-01-02	2/1/2000
	19	10	2000-01-02	2/1/2000
	20	7	2000-01-02	2/1/2000
	21	9	2000-01-02	2/1/2000
	22	10	2000-01-02	2/1/2000
	23	10	2000-01-02	2/1/2000
	24	4	2000-01-02	2/1/2000
	25	5	2000-01-02	2/1/2000
			...	
	5655	10	2001-04-19	19/4/2001
	5656	8	2001-04-19	19/4/2001
	5657	3	2001-04-19	19/4/2001
	5658	2	2001-04-19	19/4/2001
	5659	2	2001-04-19	19/4/2001
	5660	5	2001-04-19	19/4/2001
	5661	7	2001-04-19	19/4/2001
	5662	3	2001-04-19	19/4/2001
	5663	1	2001-04-19	19/4/2001
	5664	3	2001-04-19	19/4/2001
	5665	4	2001-04-19	19/4/2001
	5666	4	2001-04-19	19/4/2001
	5667	8	2001-04-19	19/4/2001
	5668	8	2001-04-19	19/4/2001
	5669	4	2001-04-19	19/4/2001
	5670	6	2001-04-19	19/4/2001
	5671	6	2001-04-19	19/4/2001
	5672	1	2001-04-19	19/4/2001
	5673	7	2001-04-20	20/4/2001
	5674	9	2001-04-20	20/4/2001
	5675	10	2001-04-20	20/4/2001
	5676	1	2001-04-20	20/4/2001
	5677	3	2001-04-20	20/4/2001
	5678	7	2001-04-20	20/4/2001
	5679	7	2001-04-20	20/4/2001
	5680	2	2001-04-20	20/4/2001
	5681	4	2001-04-20	20/4/2001
	5682	1	2001-04-20	20/4/2001
	5683	8	2001-04-20	20/4/2001
	5684	1	2001-04-20	20/4/2001
	5685	6	2001-04-20	20/4/2001
	5686	3	2001-04-20	20/4/2001

Observe que, primeiro, foram extraídos valores inteiros correspondentes às partes dia, mês e ano da data utilizando `extract`. Após, os inteiros foram transformados em seqüências de caracteres utilizando `cast`. Por último, e só então, as seqüências de caracteres correspondentes ao dia, mês e ano foram concatenadas, juntamente com as barras separadoras, utilizando `concat` para formar uma única seqüência de caracteres formatada como 'dia/mês/ano'.

Uma outra forma de obter a mesma resposta seria utilizando a função de datas `date_format`.

```
select *,date_format(v.dia,'%d/%m/%Y') as dia_dma from venda v;
```

	codigo	vendedor	dia	dia_dma
	1	3	2000-01-01	01/01/2000
	2	8	2000-01-01	01/01/2000
	3	8	2000-01-01	01/01/2000
	4	6	2000-01-01	01/01/2000
	5	1	2000-01-01	01/01/2000
	6	4	2000-01-01	01/01/2000
	7	4	2000-01-01	01/01/2000
	8	9	2000-01-01	01/01/2000
	9	9	2000-01-01	01/01/2000
	10	4	2000-01-01	01/01/2000
	11	8	2000-01-01	01/01/2000
	12	1	2000-01-01	01/01/2000
	13	3	2000-01-01	01/01/2000

```

|      14 |      9 | 2000-01-02 | 02/01/2000 |
|      15 |     10 | 2000-01-02 | 02/01/2000 |
|      16 |      9 | 2000-01-02 | 02/01/2000 |
|      17 |     10 | 2000-01-02 | 02/01/2000 |
|      18 |      2 | 2000-01-02 | 02/01/2000 |
|      19 |     10 | 2000-01-02 | 02/01/2000 |
|      20 |      7 | 2000-01-02 | 02/01/2000 |
|      21 |      9 | 2000-01-02 | 02/01/2000 |
|      22 |     10 | 2000-01-02 | 02/01/2000 |
|      23 |     10 | 2000-01-02 | 02/01/2000 |
|      24 |      4 | 2000-01-02 | 02/01/2000 |
|      25 |      5 | 2000-01-02 | 02/01/2000 |
|
|      5655 |     10 | 2001-04-19 | 19/04/2001 |
|      5656 |      8 | 2001-04-19 | 19/04/2001 |
|      5657 |      3 | 2001-04-19 | 19/04/2001 |
|      5658 |      2 | 2001-04-19 | 19/04/2001 |
|      5659 |      2 | 2001-04-19 | 19/04/2001 |
|      5660 |      5 | 2001-04-19 | 19/04/2001 |
|      5661 |      7 | 2001-04-19 | 19/04/2001 |
|      5662 |      3 | 2001-04-19 | 19/04/2001 |
|      5663 |      1 | 2001-04-19 | 19/04/2001 |
|      5664 |      3 | 2001-04-19 | 19/04/2001 |
|      5665 |      4 | 2001-04-19 | 19/04/2001 |
|      5666 |      4 | 2001-04-19 | 19/04/2001 |
|      5667 |      8 | 2001-04-19 | 19/04/2001 |
|      5668 |      8 | 2001-04-19 | 19/04/2001 |
|      5669 |      4 | 2001-04-19 | 19/04/2001 |
|      5670 |      6 | 2001-04-19 | 19/04/2001 |
|      5671 |      6 | 2001-04-19 | 19/04/2001 |
|      5672 |      1 | 2001-04-19 | 19/04/2001 |
|      5673 |      7 | 2001-04-20 | 20/04/2001 |
|      5674 |      9 | 2001-04-20 | 20/04/2001 |
|      5675 |     10 | 2001-04-20 | 20/04/2001 |
|      5676 |      1 | 2001-04-20 | 20/04/2001 |
|      5677 |      3 | 2001-04-20 | 20/04/2001 |
|      5678 |      7 | 2001-04-20 | 20/04/2001 |
|      5679 |      7 | 2001-04-20 | 20/04/2001 |
|      5680 |      2 | 2001-04-20 | 20/04/2001 |
|      5681 |      4 | 2001-04-20 | 20/04/2001 |
|      5682 |      1 | 2001-04-20 | 20/04/2001 |
|      5683 |      8 | 2001-04-20 | 20/04/2001 |
|      5684 |      1 | 2001-04-20 | 20/04/2001 |
|      5685 |      6 | 2001-04-20 | 20/04/2001 |
|      5686 |      3 | 2001-04-20 | 20/04/2001 |
+-----+-----+-----+-----+

```

2.1.28. Mostrando o nome do dia da semana de uma venda:

Como a função `date_format` foi implementada para mostrar o nome do dia da semana em inglês (sunday, monday, ...), utilizar os formatos `'%a'` ou `'%W'` não será de muito auxílio. Mas `date_format` ainda pode ser utilizada para obter um número correspondente ao dia da semana.

```
select *,date_format(v.dia,'%w') as diasemana from venda v;
```

```

+-----+-----+-----+-----+
| codigo | vendedor | dia      | diasemana |
+-----+-----+-----+-----+
|      1 |      3 | 2000-01-01 | 6         |
|      2 |      8 | 2000-01-01 | 6         |
|      3 |      8 | 2000-01-01 | 6         |
|      4 |      6 | 2000-01-01 | 6         |
|      5 |      1 | 2000-01-01 | 6         |
|      6 |      4 | 2000-01-01 | 6         |
|      7 |      4 | 2000-01-01 | 6         |
|      8 |      9 | 2000-01-01 | 6         |
|      9 |      9 | 2000-01-01 | 6         |
|     10 |      4 | 2000-01-01 | 6         |
|     11 |      8 | 2000-01-01 | 6         |
|     12 |      1 | 2000-01-01 | 6         |
|     13 |      3 | 2000-01-01 | 6         |
|     14 |      9 | 2000-01-02 | 0         |

```


	15	10	2000-01-02	0
	16	9	2000-01-02	0
	17	10	2000-01-02	0
	18	2	2000-01-02	0
	19	10	2000-01-02	0
	20	7	2000-01-02	0
	21	9	2000-01-02	0
	22	10	2000-01-02	0
	23	10	2000-01-02	0
	24	4	2000-01-02	0
	25	5	2000-01-02	0
			...	
	5655	10	2001-04-19	4
	5656	8	2001-04-19	4
	5657	3	2001-04-19	4
	5658	2	2001-04-19	4
	5659	2	2001-04-19	4
	5660	5	2001-04-19	4
	5661	7	2001-04-19	4
	5662	3	2001-04-19	4
	5663	1	2001-04-19	4
	5664	3	2001-04-19	4
	5665	4	2001-04-19	4
	5666	4	2001-04-19	4
	5667	8	2001-04-19	4
	5668	8	2001-04-19	4
	5669	4	2001-04-19	4
	5670	6	2001-04-19	4
	5671	6	2001-04-19	4
	5672	1	2001-04-19	4
	5673	7	2001-04-20	5
	5674	9	2001-04-20	5
	5675	10	2001-04-20	5
	5676	1	2001-04-20	5
	5677	3	2001-04-20	5
	5678	7	2001-04-20	5
	5679	7	2001-04-20	5
	5680	2	2001-04-20	5
	5681	4	2001-04-20	5
	5682	1	2001-04-20	5
	5683	8	2001-04-20	5
	5684	1	2001-04-20	5
	5685	6	2001-04-20	5
	5686	3	2001-04-20	5

Existe ainda a função de datas `dayofweek`, com o mesmo propósito.

```
select *,dayofweek(v.dia) as diasemana from venda v;
```

	codigo	vendedor	dia	diasemana
	1	3	2000-01-01	7
	2	8	2000-01-01	7
	3	8	2000-01-01	7
	4	6	2000-01-01	7
	5	1	2000-01-01	7
	6	4	2000-01-01	7
	7	4	2000-01-01	7
	8	9	2000-01-01	7
	9	9	2000-01-01	7
	10	4	2000-01-01	7
	11	8	2000-01-01	7
	12	1	2000-01-01	7
	13	3	2000-01-01	7
	14	9	2000-01-02	1
	15	10	2000-01-02	1
	16	9	2000-01-02	1
	17	10	2000-01-02	1
	18	2	2000-01-02	1
	19	10	2000-01-02	1
	20	7	2000-01-02	1
	21	9	2000-01-02	1

	22	10	2000-01-02	1
	23	10	2000-01-02	1
	24	4	2000-01-02	1
	25	5	2000-01-02	1
	...			
	5655	10	2001-04-19	5
	5656	8	2001-04-19	5
	5657	3	2001-04-19	5
	5658	2	2001-04-19	5
	5659	2	2001-04-19	5
	5660	5	2001-04-19	5
	5661	7	2001-04-19	5
	5662	3	2001-04-19	5
	5663	1	2001-04-19	5
	5664	3	2001-04-19	5
	5665	4	2001-04-19	5
	5666	4	2001-04-19	5
	5667	8	2001-04-19	5
	5668	8	2001-04-19	5
	5669	4	2001-04-19	5
	5670	6	2001-04-19	5
	5671	6	2001-04-19	5
	5672	1	2001-04-19	5
	5673	7	2001-04-20	6
	5674	9	2001-04-20	6
	5675	10	2001-04-20	6
	5676	1	2001-04-20	6
	5677	3	2001-04-20	6
	5678	7	2001-04-20	6
	5679	7	2001-04-20	6
	5680	2	2001-04-20	6
	5681	4	2001-04-20	6
	5682	1	2001-04-20	6
	5683	8	2001-04-20	6
	5684	1	2001-04-20	6
	5685	6	2001-04-20	6
	5686	3	2001-04-20	6

Repare que, embora pareça que as duas respostas são iguais, não são. O resultado de `date_format` é um caracter de '0' a '6', enquanto que o de `dayofweek` é um inteiro de 1 a 7.

Agora, para obter o nome do dia da semana basta utilizar o operador `case`.

```
select *,case dayofweek(v.dia) when 1 then 'Domingo' when 2 then 'Segunda' when 3 then 'Terca' when 4 then 'Quarta' when 5 then 'Quinta' when 6 then 'Sexta' when 7 then 'Sabado' end as diasemana from venda v;
```

	codigo	vendedor	dia	diasemana
	1	3	2000-01-01	Sabado
	2	8	2000-01-01	Sabado
	3	8	2000-01-01	Sabado
	4	6	2000-01-01	Sabado
	5	1	2000-01-01	Sabado
	6	4	2000-01-01	Sabado
	7	4	2000-01-01	Sabado
	8	9	2000-01-01	Sabado
	9	9	2000-01-01	Sabado
	10	4	2000-01-01	Sabado
	11	8	2000-01-01	Sabado
	12	1	2000-01-01	Sabado
	13	3	2000-01-01	Sabado
	14	9	2000-01-02	Domingo
	15	10	2000-01-02	Domingo
	16	9	2000-01-02	Domingo
	17	10	2000-01-02	Domingo
	18	2	2000-01-02	Domingo
	19	10	2000-01-02	Domingo
	20	7	2000-01-02	Domingo
	21	9	2000-01-02	Domingo
	22	10	2000-01-02	Domingo

	23	10	2000-01-02	Domingo
	24	4	2000-01-02	Domingo
	25	5	2000-01-02	Domingo
			...	
	5655	10	2001-04-19	Quinta
	5656	8	2001-04-19	Quinta
	5657	3	2001-04-19	Quinta
	5658	2	2001-04-19	Quinta
	5659	2	2001-04-19	Quinta
	5660	5	2001-04-19	Quinta
	5661	7	2001-04-19	Quinta
	5662	3	2001-04-19	Quinta
	5663	1	2001-04-19	Quinta
	5664	3	2001-04-19	Quinta
	5665	4	2001-04-19	Quinta
	5666	4	2001-04-19	Quinta
	5667	8	2001-04-19	Quinta
	5668	8	2001-04-19	Quinta
	5669	4	2001-04-19	Quinta
	5670	6	2001-04-19	Quinta
	5671	6	2001-04-19	Quinta
	5672	1	2001-04-19	Quinta
	5673	7	2001-04-20	Sexta
	5674	9	2001-04-20	Sexta
	5675	10	2001-04-20	Sexta
	5676	1	2001-04-20	Sexta
	5677	3	2001-04-20	Sexta
	5678	7	2001-04-20	Sexta
	5679	7	2001-04-20	Sexta
	5680	2	2001-04-20	Sexta
	5681	4	2001-04-20	Sexta
	5682	1	2001-04-20	Sexta
	5683	8	2001-04-20	Sexta
	5684	1	2001-04-20	Sexta
	5685	6	2001-04-20	Sexta
	5686	3	2001-04-20	Sexta

2.1.29. Outra forma de listar os dados das memórias Kingston:

```
select * from produto p where (locate('mem',p.descricao) > 0) and (locate('kingston',p.descricao) > 0);
```

codigo	descricao	preco	comissao
799	mem card micro sd 4gb kingston c/ 2 adaptador	13.2	3
801	mem card sd 4gb kingston	12.75	3
802	mem card sd 8gb kingston high capacity class 4	20.85	3
803	mem card sd 16gb kingston high capacity class 4	42	3
804	mem card sd 32gb kingston high capacity class 4	105	3
811	mem ddr 512mb 400mhz kingston	20.1	3
814	mem ddr 1gb 400mhz kingston	36.5	3
817	mem ddr2 1gb 667mhz kingston	27.6	3
820	mem ddr2 1gb 800mhz kingston n6	28	3
824	mem ddr2 1gb 667mhz kingston ecc	29	3
827	mem ddr2 2gb 667mhz kingston c15	48.2	3
830	mem ddr2 2gb 800mhz kingston	49.5	3
837	memoria para notebook 1gb pc2700 ddr kingston	46.5	3
1124	nb mem ddr2 2gb 800mhz kingston	48.5	3

Neste caso, foi utilizada a função `locate` para procurar por 'mem' e por 'kingston' nas descrições dos produtos.

2.1.30. Listando os valores distintos de salário de vendedores:

```
select r.salario from vendedor r;
```

```
+-----+
| salario |
+-----+
|      500 |
|      500 |
|      500 |
|      550 |
|      550 |
|      500 |
|      850 |
|      550 |
|      500 |
|      550 |
+-----+
```

O `select` acima lista o valor do salário de cada vendedor. Observe que há repetição de valores.

Para indicar que valores repetidos devem ser suprimidos na resposta de um `select`, devemos utilizar o operador `distinct`.

```
select distinct r.salario from vendedor r;
```

```
+-----+
| salario |
+-----+
|      500 |
|      550 |
|      850 |
+-----+
```

Repare que, com `distinct`, apenas os três valores distintos de salário existentes na tabela `vendedor` são mostrados, e, sem `distinct`, os salários de todos os dez vendedores na tabela `vendedor` são mostrados, com repetição dos valores.

2.1.31. Listando os produtos que pagam 5% ou mais de comissão:

Começamos listando todos os produtos:

```
select * from produto p;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| codigo | descricao | preco | comissao |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | acessorio kit p/ vga low profile | 4.2 | NULL |
| 2 | adaptador dvi-vga misc xfx | 2.53 | NULL |
| 3 | adaptador p/ carro de dc/ac aukson inversor 100watts 220 volts | 20.5 | NULL |
| 4 | adaptador p/ipod klipx icc-050 car charger | 7.6 | NULL |
| 5 | adaptador satelite usb 2. 0 al-09 | 1.7 | NULL |
...
| 1869 | wir. transmissor fm for iphone 3g- 0878 (silver) | 15.75 | NULL |
| 1870 | wir. transmissor fm orange or-668 (gray/white) | 9.2 | NULL |
| 1871 | wir. transmissor logitech mp3 p/ entrada auxiliar de som | 26.5 | NULL |
| 1872 | wireless antena hyperlink 20dbi 090 setorial hg2420p-090 | 312 | NULL |
| 1873 | wireless antena hyperlink 17dbi 120 setorial hg2417p-120 | 154 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Observando a resposta, é preciso perceber que um produto não tem o valor da comissão, e sim o código de uma comissão (que pode até ser nulo). É na tabela comissão que estão os valores de comissão que devem ser pagos.

O `select` que lista todas as comissões que pagam mais de 5% é:

```
select * from comissao c where (c.comissao >= 5);
```

codigo	comissao
1	10
2	5

Agora, vamos voltar ao básico. O que é a resposta deste `select`? É um conjunto, contendo todos os dados das comissões que pagam 5% ou mais. Conjunto é a palavra chave.

A vantagem de tratar a resposta de um `select` como um conjunto é, então, que o operador `in`, específico para conjuntos, pode ser utilizado.

Sabendo isto, basta então, adaptar o `select` que lista todos os produtos para filtrar aqueles cujo código da comissão pertença à resposta do `select` que lista todas as comissões que pagam 5% ou mais, ou seja, unir os dois `select` com `in`.

É muito importante perceber que a ligação entre os dois `select` é feita pelo código da comissão, que se chama comissão na tabela produto e código na tabela comissão.

Conforme discutido, não são necessários todos os dados das comissões que pagam 5% ou mais, basta o código:

```
select c.codigo from comissao c where (c.comissao >= 5);
```

codigo
1
2

E o `select` final, pronto, é:

```
select p.descricao from produto p where (p.comissao in (select c.codigo from comissao c where (c.comissao >= 5)));
```

descricao
bolsa p/notebook 15.4 case logic snm-15 gray
bolsa p/notebook 15.4 case logic snm-15f gray
calculadora hp 2220 office calc 100
calculadora hp 2221 office basic calc 200
calculadora hp financeira 10 bii - manual em espanhol
calculadora hp financeira 12c gold - manual portugues
calculadora hp financeira 17bii - manual em portugues e espanhol
calculadora hp grafica 50g (manual portugues)
...
vga pciexp.1gb ddr5 hd4890 850m 256bit ati tv/dual dvi/hdtv xfx
vga pciexp.1gb ddr5 hd4890 850m 256bit ati tv/dual dvi/hdtv xfx
vga pciexp.1gb ddr5 hd4890 850m 256bit ati tv/dual dvi/hdtv xfx (gtx285)
vga pciexp.1gb ddr5 hd5750 700m 128bit dual dvi/tvo/hdmi sapphire
vga pciexp.1gb ddr5 hd5750 700m 128bit ati tv/dual dvi/hdtv/hdmi xfx
vga pciexp.1gb ddr5 hd5750 710m vapor-x dual dvi/tvo/hdmi sapphire
vga pciexp.1gb ddr5 hd5770 850m 128bit ati hdmi/dual dvi/hdtv xfx
vga pciexp.1gb ddr5 hd5850 725m 256bit ati tv/dual dvi/hdtv/hdmi xfx
vga pciexp.1gb ddr5 hd5850 755m ati tv/dual dvi/hdtv/hdmi xfx
vga pciexp.1gb ddr5 hd5850 765m 256bit ati tv/dual dvi/hdtv/hdmi black xfx
vga pciexp.1gb ddr5 hd5870 850m 256bit ati tv/dual dvi/hdtv xfx
vga pciexp.1gb ddr5 hd5870 875m 256bit ati cypress xt/dual dvi/hdmi xfx
vga pciexp.2gb ddr3 gf gts250 738m dual dvi/hdtv ecs

Neste tipo de construção, com um `select` dentro de outro, o `select` mais interno é denominado `subselect`.

2.1.32. Listando os vendedores que efetuaram vendas em 15/01/2000:

Novamente, uma operação de conjuntos é necessária. Precisaremos da coluna nome na tabela vendedor e da coluna dia na tabela venda. Observando a descrição das tabelas, percebemos que, neste caso, a ligação entre as tabelas é o campo código do vendedor, que se chama código na tabela vendedor e vendedor na tabela venda.

O `select` que lista os códigos dos vendedores das vendas efetuadas em 15/01/2000 é:

```
select v.vendedor from venda v where (v.dia = '2000-01-15');
```

```
+-----+
| vendedor |
+-----+
|      10 |
|       6 |
|       1 |
|       5 |
|       8 |
|       7 |
|       9 |
|       3 |
|      10 |
|       1 |
|       6 |
+-----+
```

Para obter a resposta desejada basta, agora, listar os nomes dos vendedores cujos códigos pertençam à resposta do `select` que lista as vendas efetuadas em 15/01/2000.

O `select` que dá a resposta desejada é:

```
select r.nome from vendedor r where (r.codigo in (select v.vendedor from venda v where (v.dia = '2000-01-15')));
```

```
+-----+
| nome |
+-----+
| Mauricio Costa Quaresma |
| Patricia Menezes Silva |
| Michele Menezes Santos |
| Cristiane Nunes Brandao |
| Pedro Rosa Loureiro |
| Jose Souza Martins |
| Gabriel Albuquerque Menezes |
| Mauricio Bezerra Lopes |
+-----+
```

2.1.33. Listando os vendedores que efetuaram vendas entre 10/01/2000 e 15/01/2000:

Este exemplo é similar ao anterior. Apenas é utilizado um intervalo de datas ao invés de uma única data.

```
select r.nome from vendedor r where (r.codigo in (select v.vendedor from venda v where (v.dia between '2000-01-10' and '2000-01-15')));
```

```
+-----+
| nome |
+-----+
| Mauricio Costa Quaresma |
| Paulo Lopes Nunes |
| Patricia Menezes Silva |
| Cristiano Lopes Bezerra |
+-----+
```

Michele Menezes Santos	
Cristiane Nunes Brandao	
Pedro Rosa Loureiro	
Jose Souza Martins	
Gabriel Albuquerque Menezes	
Mauricio Bezerra Lopes	
+-----+	

2.1.34. Listando os produtos vendidos entre 10/01/2000 e 15/01/2000:

Para obter a resposta desejada, é necessário utilizar dois campos de ligação. Não existe uma coluna de ligação em comum entre o produto e a venda. É o item que faz esta ponte. É necessário, então, primeiro, ligar a venda ao item e, depois, o item ao produto.

Começando pela tabela venda, filtramos as vendas no intervalo de interesse.

```
select v.codigo from venda v where (v.dia between '2000-01-10' and '2000-01-15');
```

+-----+
codigo
+-----+
96
97
98
99
100
...
160
161
162
163
164
+-----+

Em seguida, na tabela item, filtramos os itens destas vendas:

```
select i.produto from item i where (i.venda in (select v.codigo from venda v where (v.dia between '2000-01-10' and '2000-01-15')));
```

+-----+
produto
+-----+
8
18
113
932
1065
1122
1687
1854
465
506
...
1085
1265
878
908
1038
1055
1455
1456
1600
223
+-----+

Por último, na tabela produto, filtramos os produtos destes itens:

```
select p.descricao from produto p where (p.codigo in (select i.produto from item i where (i.venda in (select v.codigo from venda v where (v.dia between '2000-01-10' and '2000-01-15'))))));
```

```
+-----+
| descricao |
+-----+
| adaptador p/ impressao tel100-plu trendnet usb em uso de rede |
| antena parabolica 80cm s/lnb shop+ |
| camera foto polaroid i1035 10.1 black |
| mouse ps/2 a5 satellite optical (preto) |
| mp7-celular mox mo-829 tv/phone/mp3/mp4/tela 2.0 8.0 mpixel /4 bandas (black) |
| nb mem ddr2 2gb 800mhz corsair |
| vga pci 512 mb ddr2 gf 6200a 350m 64bit vga/dvi/hdtv xfx |
| wir. mk mini pci cartao router board r52h a/b/g 320mw |
| gaveta p/ hd sata 2.5/3.5 usb 2.0 ext. satellite ax-dk226 (branco) |
| hd ext. usb 2.5 320 gb samsung s2 portable blue |
|
| mp7-celular x tech xt n6310 tv/4b/2chip/ vermelho |
| pda smartphone kaiomy aegis 1183 2.8/4bandas/460mhz/128mb/wi-fi lan/win mobile 6.1 portugues |
| midia dvd-r 4.7 gb kaiomy 120 min c/logo 16x speed 50 pcs |
| monit. lcd 20 samsung 2043nwx wide preto |
| mp3 - 8gb apple ipod nano black mc031zy/a |
| mp7-celular fujilink fu-91t tv/phone/mp3/mp4/4 bandas (branco) |
| proc. intel 775p core 2 quad q8400 2.66 4mb/1333 mhz box |
| proc. intel 775p core 2 quad q9400 2.66 6mb/1333mhz box |
| teclado i-tech virtual laser infrared (full-size qwerty) para pda/blackberry |
| car alto falante p/carro roadstar-rs-6908- 1800w |
+-----+
```

2.1.35. Contando a quantidade de memórias da marca Kingston:

```
select count(*) as quantidade from produto p where (p.descricao like '%mem%kingston%');
```

```
+-----+
| quantidade |
+-----+
|          14 |
+-----+
```

A função de agregação `count(*)` conta a quantidade de linhas na resposta produzida por um `select`.

2.1.36. Calculando a média dos preços das memórias Kingston:

```
select round(avg(p.preco),2) as preco_medio from produto p where (p.descricao like '%mem%kingston%');
```

```
+-----+
| preco_medio |
+-----+
|          37.69 |
+-----+
```

A função de agregação `avg` calcula a média dos valores de uma coluna da resposta produzida por um `select`.

2.1.37. Encontrando o menor preço de uma memória Kingston:

```
select min(p.preco) as menor_preco from produto p where (p.descricao like '%mem%kingston%');
```

```
+-----+
| menor_preco |
+-----+
|          12.75 |
+-----+
```

A função de agregação `min` encontra o menor valor de uma coluna da resposta produzida por um `select`.

2.1.38. Encontrando o maior preço de uma memória Kingston:

```
select max(p.preco) as maior_preco from produto p where (p.descricao like '%mem%kingston%');
```

```
+-----+
| maior_preco |
+-----+
|          105 |
+-----+
```

A função de agregação `max` encontra o maior valor de uma coluna da resposta produzida por um `select`.

2.1.39. Contando a quantidade de vendas entre 10/01/2000 e 15/01/2000 do vendedor Maurício Costa Quaresma, cujo código é 1:

Primeiro vamos listar todas as vendas do vendedor no período de interesse:

```
select * from venda v where (v.dia between '2000-01-10' and '2000-01-15') and (v.vendedor = 1);
```

```
+-----+-----+-----+
| codigo | vendedor | dia      |
+-----+-----+-----+
| 102    | 1         | 2000-01-10 |
| 108    | 1         | 2000-01-10 |
| 119    | 1         | 2000-01-11 |
| 127    | 1         | 2000-01-12 |
| 128    | 1         | 2000-01-12 |
| 134    | 1         | 2000-01-12 |
| 142    | 1         | 2000-01-13 |
| 144    | 1         | 2000-01-13 |
| 152    | 1         | 2000-01-14 |
| 156    | 1         | 2000-01-15 |
| 163    | 1         | 2000-01-15 |
+-----+-----+-----+
```

Em seguida, basta utilizar `count` para contar a quantidade de linhas da resposta:

```
select count(*) as quantidade from venda v where (v.dia between '2000-01-10' and '2000-01-15') and (v.vendedor = 1);
```

```
+-----+
| quantidade |
+-----+
|          11 |
+-----+
```

2.1.40. Contando a quantidade de salários distintos dos vendedores:

```
select count(r.salario) as quantidade from vendedor r;
```

```
+-----+
| quantidade |
+-----+
|          10 |
+-----+
```

```
select count(distinct r.salario) as quantidade from vendedor r;
```

```
+-----+
| quantidade |
+-----+
|           3 |
+-----+
```

Repare que `count` sem `distinct` não nos dá a resposta desejada. Sem `distinct`, a resposta é 10, pois esta é a quantidade de vendedores (e salários) na tabela `vendedor`. Com `distinct`, a resposta é 3, pois apenas os valores distintos de salário foram contados.

2.1.41. Listando os vendedores que recebem o menor salário:

Primeiro, precisamos determinar qual o valor do menor salário:

```
select min(r.salario) from vendedor r;
```

```
+-----+
| min(r.salario) |
+-----+
|           500 |
+-----+
```

Em seguida, utilizamos o `in` para filtrar apenas os vendedores com salário igual ao menor salário:

```
select * from vendedor r where r.salario in (select min(r.salario) from vendedor r);
```

```
+-----+-----+-----+
| codigo | nome                | salario |
+-----+-----+-----+
|      1 | Mauricio Costa Quaresma |      500 |
|      2 | Paulo Lopes Nunes      |      500 |
|      3 | Patricia Menezes Silva  |      500 |
|      6 | Cristiane Nunes Brandao |      500 |
|      9 | Gabriel Albuquerque Menezes |      500 |
+-----+-----+-----+
```

2.1.42. Contando a quantidade de unidades do produto camera webcam genius eye 320, cujo código é 155, vendidas em outubro de 2000:

Primeiro, vamos listar as vendas do produto no intervalo de interesse.

```
select * from item i where (i.produto = 155) and (i.venda in (select v.codigo from venda v where (cast(v.dia as char) like '2000-10-%')));
```

```
+-----+-----+-----+
| venda | produto | quantidade |
+-----+-----+-----+
| 3372 | 155 | 1 |
| 3486 | 155 | 5 |
| 3527 | 155 | 1 |
| 3541 | 155 | 1 |
| 3589 | 155 | 1 |
+-----+-----+-----+
```

Neste caso, não se quer contar a quantidade de linhas, mas sim, somar a quantidade vendida de cada produto em uma venda. A função de agregação `sum` deve ser utilizada, e não a função `count`.

```
select sum(i.quantidade) as unidades from item i where (i.produto = 155) and (i.venda in (select v.codigo from venda v where (cast(v.dia as char) like '2000-10-%')));
```

```
+-----+
| unidades |
+-----+
|          9 |
+-----+
```

A função de agregação `sum` calcula o somatório dos valores de uma coluna da resposta produzida por um `select`.

2.1.43. Contando a quantidade de vendedores, por salário:

Neste exemplo, precisaremos utilizar mais uma parte importante do `select`, o `group by`. Como o nome já diz, o `group by` possibilita agrupar as linhas da resposta de um `select` segundo um critério de agrupamento e aplicar uma função de agregação a cada um dos grupos.

No caso em questão, o critério de agrupamento é o valor do salário e a função de agregação é a de contagem.

```
select r.salario,count(*) as quantidade from vendedor r group by r.salario;
```

```
+-----+-----+
| salario | quantidade |
+-----+-----+
|      500 |           5 |
|      550 |           4 |
|      850 |           1 |
+-----+-----+
```

Para entendê-lo melhor, vamos simular seu funcionamento. Quando enviamos o `select` acima ao MySQL, é o mesmo que dizer:

- 1) Ordene os dados pelo salário, em ordem crescente;
- 2) Agrupe as linhas com salários iguais;
- 3) Aplique a função de agregação `count` em cada grupo e,
- 4) Remova repetições de valores em cada grupo, deixando apenas um.

Acompanhe passo-a-passo o funcionamento do `select` com `group by` na figura 5 abaixo.

Passo 1	Passo 2	Passo 3	Passo 4																																																			
<table><tr><th>Salário</th></tr><tr><td>500</td></tr><tr><td>500</td></tr><tr><td>500</td></tr><tr><td>500</td></tr><tr><td>500</td></tr><tr><td>550</td></tr><tr><td>550</td></tr><tr><td>550</td></tr><tr><td>550</td></tr><tr><td>850</td></tr></table>	Salário	500	500	500	500	500	550	550	550	550	850	<table><tr><th>Salário</th><td rowspan="5">Grupo 1</td></tr><tr><td>500</td></tr><tr><td>500</td></tr><tr><td>500</td></tr><tr><td>500</td></tr><tr><td>550</td><td rowspan="4">Grupo 2</td></tr><tr><td>550</td></tr><tr><td>550</td></tr><tr><td>550</td></tr><tr><td>850</td><td>Grupo 3</td></tr></table>	Salário	Grupo 1	500	500	500	500	550	Grupo 2	550	550	550	850	Grupo 3	<table><tr><th>Salário</th><th>count(*)</th><td rowspan="5">Grupo 1</td></tr><tr><td>500</td><td rowspan="4">5</td></tr><tr><td>500</td></tr><tr><td>500</td></tr><tr><td>500</td></tr><tr><td>550</td><td rowspan="4">4</td><td rowspan="4">Grupo 2</td></tr><tr><td>550</td></tr><tr><td>550</td></tr><tr><td>550</td></tr><tr><td>850</td><td>1</td><td>Grupo 3</td></tr></table>	Salário	count(*)	Grupo 1	500	5	500	500	500	550	4	Grupo 2	550	550	550	850	1	Grupo 3	<table><tr><th>Salário</th><th>count(*)</th><td rowspan="3">Grupo 1</td></tr><tr><td>500</td><td>5</td></tr><tr><td>550</td><td>4</td></tr><tr><td>850</td><td>1</td><td>Grupo 3</td></tr></table>	Salário	count(*)	Grupo 1	500	5	550	4	850	1	Grupo 3
Salário																																																						
500																																																						
500																																																						
500																																																						
500																																																						
500																																																						
550																																																						
550																																																						
550																																																						
550																																																						
850																																																						
Salário	Grupo 1																																																					
500																																																						
500																																																						
500																																																						
500																																																						
550	Grupo 2																																																					
550																																																						
550																																																						
550																																																						
850	Grupo 3																																																					
Salário	count(*)	Grupo 1																																																				
500	5																																																					
500																																																						
500																																																						
500																																																						
550	4	Grupo 2																																																				
550																																																						
550																																																						
550																																																						
850	1	Grupo 3																																																				
Salário	count(*)	Grupo 1																																																				
500	5																																																					
550	4																																																					
850	1	Grupo 3																																																				

Figura 5 - Funcionamento do `group by`

É importante colocar que, quando se utiliza `group by`, apenas as colunas necessárias para a aplicação da função de agregação devem ser recuperadas. No exemplo, a resposta é produzida apenas com o salário e o resultado de `count`, sem outros dados de empregado.

Embora, neste exemplo, tenha sido utilizada a função `count`, qualquer função de agregação pode ser utilizada com o `group by`.

2.1.44. Contando a quantidade de vendas, por vendedor, em janeiro de 2000:

Neste caso, basta filtrar as vendas efetuadas em janeiro de 2000 e, em seguida, utilizar `group by` juntamente com `count` para agrupar os dados destas vendas pelo código do vendedor, contar a quantidade de linhas em cada grupo (vendas de cada vendedor) e obter a resposta desejada.

```
select v.vendedor, count(*) as quantidade from venda v where (cast(v.dia as char) like '2000-01-%') group by v.vendedor order by v.vendedor asc;
```

+-----+-----+	
vendedor	quantidade
+-----+-----+	
1	41
2	38
3	43
4	41
5	32
6	33
7	31
8	42
9	38
10	36
+-----+-----+	

2.1.45. Contando a quantidade de unidades vendidas, por produto, em 10/01/2000:

Para isto, primeiro precisamos perceber que precisamos de dados de duas tabelas diferentes. A quantidade de unidades vendidas de um produto é armazenada na tabela `item`. Já a data da venda é armazenada na tabela `venda`. Para obter os itens de uma venda, precisamos fazer uma ligação entre estas tabelas. Esta ligação é o código da venda, que se chama `venda` na tabela `item` e `código` na tabela `venda`. Isto posto, basta filtrar as vendas efetuadas em 10/01/2000 para, via `in`, obter todos os itens relacionados à estas vendas.

Agora, tendo filtrado apenas os itens das vendas efetuadas em 10/01/2000, vamos utilizar o `group by` juntamente com `sum` para agrupar os dados pelo código do produto, somar a quantidade de unidades vendidas de cada produto e obter a resposta desejada.

Repare que não se quer contar a quantidade de vezes que cada produto foi vendido, e sim, obter a quantidade de unidades vendidas de cada produto. Por isto, foi utilizado `sum` e não `count`.

```
select i.produto, sum(i.quantidade) as unidades from item i where (i.venda in (select v.codigo from venda v where (v.dia = '2000-01-10')))) group by i.produto order by i.produto asc;
```

+-----+-----+	
produto	unidades
+-----+-----+	
8	1
18	1
23	1
77	1
83	1
113	1
116	2
160	1

	201	5
	220	1
	223	1
	247	1
	266	1
	302	1
	307	1
	327	1
	336	1
	354	1
	361	1
	391	2
	430	1
	448	1
	452	1
	453	1
	465	2
	470	4
	471	1
	479	1
	506	1
	510	1
	557	1
	592	1
	599	1
	614	1
	631	5
	633	1
	679	4
	691	4
	745	1
	764	1
	767	1
	771	3
	785	2
	796	1
	821	5
	864	1
	907	1
	926	1
	932	1
	954	1
	1029	1
	1065	1
	1122	2
	1126	1
	1156	1
	1169	1
	1204	1
	1208	1
	1222	1
	1228	1
	1251	1
	1258	1
	1276	1
	1340	1
	1401	4
	1414	1
	1418	1
	1449	1
	1451	1
	1485	5
	1544	1
	1549	1
	1560	1
	1604	1
	1605	1
	1606	1
	1621	1
	1684	1
	1687	1
	1696	1
	1731	1
	1737	1
	1739	1
	1746	1

	1750		1	
	1785		5	
	1820		3	
	1835		1	
	1849		1	
	1854		1	
	1870		3	
	1871		1	
+-----+-----+				

2.1.46. Contar a quantidade de vendas, por vendedor e por data, entre 01/01/2000 e 05/01/2000:

```
select v.vendedor,v.dia,count(*) as quantidade from venda v where (v.dia between '2000-01-01' and '2000-01-05') group by v.vendedor,v.dia order by v.dia asc,v.vendedor asc;
```

+-----+-----+-----+			
	vendedor	dia	quantidade
+-----+-----+-----+			
	1	2000-01-01	2
	3	2000-01-01	2
	4	2000-01-01	3
	6	2000-01-01	1
	8	2000-01-01	3
	9	2000-01-01	2
	2	2000-01-02	1
	4	2000-01-02	1
	5	2000-01-02	1
	7	2000-01-02	1
	9	2000-01-02	3
	10	2000-01-02	5
	1	2000-01-03	1
	3	2000-01-03	3
	4	2000-01-03	2
	5	2000-01-03	4
	6	2000-01-03	1
	7	2000-01-03	1
	8	2000-01-03	2
	9	2000-01-03	1
	10	2000-01-03	1
	1	2000-01-04	2
	4	2000-01-04	2
	9	2000-01-04	1
	1	2000-01-05	3
	4	2000-01-05	1
	5	2000-01-05	1
	7	2000-01-05	1
	8	2000-01-05	1
+-----+-----+-----+			

Este `select` não é diferente dos anteriores com `group by`. A única diferença é que, neste caso, duas colunas estão sendo utilizadas como critério de agrupamento. Os grupos são formados, então, pelas linhas que possuem, simultaneamente, os mesmos valores de código do vendedor e de data da venda, ou seja, por vendedor e por data.

2.1.47. Listando apenas os vendedores que efetuaram mais de 3 vendas por dia entre 01/01/2000 e 05/01/2000:

Com base no `select` para contar a quantidade de vendas, por vendedor e por data, entre 01/01/2000 e 05/01/2000, basta então filtrar apenas aqueles vendedores que efetuaram mais de 3 vendas. Mas isto não pode ser feito no `where`. Para filtrar com base nos resultados das funções de agregação, deve ser utilizado mais uma parte do `select`, o `having`.

```
select v.vendedor,v.dia,count(*) as quantidade from venda v where (v.dia between '2000-01-01' and '2000-01-05') group by v.vendedor,v.dia having (count(*) > 3) order by v.dia asc,v.vendedor asc;
```

vendedor	dia	quantidade
10	2000-01-02	5
5	2000-01-03	4

Observe que a condição do `having` deve ser escrita com a função de agregação, e não com o apelido dado à coluna na resposta.

2.1.48. Outra forma de contar a quantidade de vendas, por vendedor, em janeiro de 2000:

No exemplo 2.1.44, foi utilizado o `select` abaixo:

```
select v.vendedor,count(*) as quantidade from venda v where (cast(v.dia as char) like '2000-01-%') group by v.vendedor order by v.vendedor asc;
```

vendedor	quantidade
1	41
2	38
3	43
4	41
5	32
6	33
7	31
8	42
9	38
10	36

Mas e se for necessário o nome do vendedor, ao invés de seu código? Uma das formas de fazer isto é com um `subselect`, que também pode ser utilizado para totalizar dados. No `select` abaixo, para cada vendedor é executado o `subselect` de contagem. Observe que o código do vendedor, no nível mais externo, é utilizado como parte do filtro, no nível mais interno.

```
select r.nome,(select count(*) from venda v where (v.vendedor = r.codigo) and (cast(v.dia as char) like '2000-01-%')) as quantidade from vendedor r;
```

nome	quantidade
Mauricio Costa Quaresma	41
Paulo Lopes Nunes	38
Patricia Menezes Silva	43
Cristiano Lopes Bezerra	41
Michele Menezes Santos	32
Cristiane Nunes Brandao	33
Pedro Rosa Loureiro	31
Jose Souza Martins	42
Gabriel Albuquerque Menezes	38
Mauricio Bezerra Lopes	36

2.1.49. Calculando a quantidade média de vendas, por vendedor, em 2000:

Em muitos casos, é útil utilizar a resposta de um `select` como tabela no `from` de outro `select`. A utilização da resposta de um `select` como uma tabela temporária intermediária facilita muito a solução de problemas mais complexos.

O `select` abaixo conta a quantidade de vendas, por vendedor, em 2000.

```
select v.vendedor, count(*) as quantidade_vendas from venda v where (extract(year from v.dia) = 2000) group by v.vendedor;
```

vendedor	quantidade_vendas
1	420
2	424
3	475
4	418
5	436
6	448
7	438
8	417
9	440
10	466

Mas para obter a quantidade média de vendas por vendedor no período, é necessário somar todas as quantidades de vendas e dividir pela quantidade de vendedores. E isto não pode ser feito em um único `select`.

Para isto, vamos utilizar a resposta do `select` acima como tabela no `from` de outro `select`.

```
select * from (select v.vendedor, count(*) as quantidade_vendas from venda v where (extract(year from v.dia) = 2000) group by v.vendedor) as t;
```

vendedor	quantidade_vendas
1	420
2	424
3	475
4	418
5	436
6	448
7	438
8	417
9	440
10	466

Repare que a resposta do `subselect` agora é conhecida como `t`, e pode ser utilizada como se fosse uma tabela normal. Para calcular, então, a quantidade média de vendas por vendedor no período basta utilizar a função `avg` sobre a coluna `quantidade_vendas` de `t`:

```
select round(avg(t.quantidade_vendas),2) as quantidade_media_vendas from (select v.vendedor, count(*) as quantidade_vendas from venda v where (extract(year from v.dia) = 2000) group by v.vendedor) as t;
```

quantidade_media_vendas
438.20

Uma outra maneira, também utilizando `subselect`, de produzir a mesma resposta:

```
select (select count(*) from venda v where (extract(year from v.dia) = 2000))/(select count(*) from
vendedor r) as quantidade_media_vendas;
```

quantidade_media_vendas
438.2000

Nesta terceira etapa, vamos, escrever requisições `select` ainda mais aprimoradas, utilizando junções entre várias tabelas.

2.1.50. Listando o nome do vendedor que efetuou cada venda em 30/05/2000:

O `select` abaixo lista os códigos dos vendedores que efetuaram vendas em 30/05/2000:

```
select * from venda v where (v.dia = '2000-05-30');
```

codigo	vendedor	dia
1728	10	2000-05-30
1729	7	2000-05-30
1730	9	2000-05-30
1731	2	2000-05-30
1732	3	2000-05-30

Obter os nomes dos vendedores que efetuaram vendas neste dia é relativamente fácil:

```
select r.nome from vendedor r where (r.codigo in (select v.vendedor from venda v where (v.dia = '2000-05-30')));
```

nome
Paulo Lopes Nunes
Patricia Menezes Silva
Pedro Rosa Loureiro
Gabriel Albuquerque Menezes
Mauricio Bezerra Lopes

Mas e se for necessário mostrar tudo isto junto? Como mostrar o nome do vendedor juntamente com os dados da venda? Utilizando junções. As junções permitem acrescentar colunas à resposta de um `select`.

Entretanto, antes de utilizar uma junção, devemos entender muito bem o que há por trás de seu funcionamento, o produto cartesiano.

O produto cartesiano de dois conjuntos A e B quaisquer é o conjunto de todas as combinações possíveis de elementos do conjunto A com elementos do conjunto B.

Imagine que o conjunto A é composto pelos números inteiros 1, 2 e 3, e que o conjunto B é composto pelas letras 'A' e 'B'. Neste caso, o produto cartesiano de A e B são os pares (1,'A'), (1,'B'), (2,'A'), (2,'B'), (3,'A') e (3,'B').

Se A e B são tabelas, o produto cartesiano é o conjunto de todas as combinações possíveis das linhas de A com as linhas de B, ou seja, a primeira linha de A combinada com cada linha de B, a segunda linha de A combinada com cada linha de B e assim por diante, até a última linha de A.

Vamos a um exemplo mais prático para visualizar melhor o produto cartesiano. O `select` abaixo produz o produto cartesiano entre os vendedores e as vendas realizadas em 30/05/2000.

```
select * from venda v,vendedor r where (v.dia = '2000-05-30');
```

codigo	vendedor	dia	codigo	nome	salario
1728	10	2000-05-30	1	Mauricio Costa Quaresma	500
1728	10	2000-05-30	2	Paulo Lopes Nunes	500
1728	10	2000-05-30	3	Patricia Menezes Silva	500
1728	10	2000-05-30	4	Cristiano Lopes Bezerra	550
1728	10	2000-05-30	5	Michele Menezes Santos	550
1728	10	2000-05-30	6	Cristiane Nunes Brandao	500
1728	10	2000-05-30	7	Pedro Rosa Loureiro	850
1728	10	2000-05-30	8	Jose Souza Martins	550
1728	10	2000-05-30	9	Gabriel Albuquerque Menezes	500
1728	10	2000-05-30	10	Mauricio Bezerra Lopes	550
1729	7	2000-05-30	1	Mauricio Costa Quaresma	500
1729	7	2000-05-30	2	Paulo Lopes Nunes	500
1729	7	2000-05-30	3	Patricia Menezes Silva	500
1729	7	2000-05-30	4	Cristiano Lopes Bezerra	550
1729	7	2000-05-30	5	Michele Menezes Santos	550
1729	7	2000-05-30	6	Cristiane Nunes Brandao	500
1729	7	2000-05-30	7	Pedro Rosa Loureiro	850
1729	7	2000-05-30	8	Jose Souza Martins	550
1729	7	2000-05-30	9	Gabriel Albuquerque Menezes	500
1729	7	2000-05-30	10	Mauricio Bezerra Lopes	550
1730	9	2000-05-30	1	Mauricio Costa Quaresma	500
1730	9	2000-05-30	2	Paulo Lopes Nunes	500
1730	9	2000-05-30	3	Patricia Menezes Silva	500
1730	9	2000-05-30	4	Cristiano Lopes Bezerra	550
1730	9	2000-05-30	5	Michele Menezes Santos	550
1730	9	2000-05-30	6	Cristiane Nunes Brandao	500
1730	9	2000-05-30	7	Pedro Rosa Loureiro	850
1730	9	2000-05-30	8	Jose Souza Martins	550
1730	9	2000-05-30	9	Gabriel Albuquerque Menezes	500
1730	9	2000-05-30	10	Mauricio Bezerra Lopes	550
1731	2	2000-05-30	1	Mauricio Costa Quaresma	500
1731	2	2000-05-30	2	Paulo Lopes Nunes	500
1731	2	2000-05-30	3	Patricia Menezes Silva	500
1731	2	2000-05-30	4	Cristiano Lopes Bezerra	550
1731	2	2000-05-30	5	Michele Menezes Santos	550
1731	2	2000-05-30	6	Cristiane Nunes Brandao	500
1731	2	2000-05-30	7	Pedro Rosa Loureiro	850
1731	2	2000-05-30	8	Jose Souza Martins	550
1731	2	2000-05-30	9	Gabriel Albuquerque Menezes	500
1731	2	2000-05-30	10	Mauricio Bezerra Lopes	550
1732	3	2000-05-30	1	Mauricio Costa Quaresma	500
1732	3	2000-05-30	2	Paulo Lopes Nunes	500
1732	3	2000-05-30	3	Patricia Menezes Silva	500
1732	3	2000-05-30	4	Cristiano Lopes Bezerra	550
1732	3	2000-05-30	5	Michele Menezes Santos	550
1732	3	2000-05-30	6	Cristiane Nunes Brandao	500
1732	3	2000-05-30	7	Pedro Rosa Loureiro	850
1732	3	2000-05-30	8	Jose Souza Martins	550
1732	3	2000-05-30	9	Gabriel Albuquerque Menezes	500
1732	3	2000-05-30	10	Mauricio Bezerra Lopes	550

Observe que cada um dos vendedores foi associado a cada uma das venda no período, e não apenas aquele vendedor que efetuou a venda. É por este motivo que a resposta tem 50 linhas (10 vendedores vezes 5 vendas em 30/05/2000).

Mas então, como fazer para associar apenas o vendedor que efetuou a venda com a venda? Com uma condição de junção. É a condição de junção que controla quais linhas do produto cartesiano devem fazer parte da resposta final e quais devem ser ignoradas.

Neste caso, assim como quando foi utilizado `in`, a ligação é feita pelo código do vendedor, que se chama código na tabela vendedor e vendedor na tabela venda. Entretanto, para a junção, uma condição é necessária. Conhecidos os campos de ligação, observando a resposta do produto cartesiano percebemos que as linhas que nos interessam são aquelas em que `(v.vendedor = r.codigo)`.

O `select` final que produz a resposta desejada é:

```
select * from venda v,vendedor r where (v.dia = '2000-05-30') and (v.vendedor = r.codigo);
```

codigo	vendedor	dia	codigo	nome	salario
1728	10	2000-05-30	10	Mauricio Bezerra Lopes	550
1729	7	2000-05-30	7	Pedro Rosa Loureiro	850
1730	9	2000-05-30	9	Gabriel Albuquerque Menezes	500
1731	2	2000-05-30	2	Paulo Lopes Nunes	500
1732	3	2000-05-30	3	Patricia Menezes Silva	500

Neste ponto é importante ressaltar, novamente, o que faz uma junção. As junções permitem acrescentar colunas à resposta de um `select`. Observe que a resposta contém, ao mesmo tempo, as colunas da tabela venda e as colunas da tabela vendedor, utilizando o campo código do vendedor como ligação. E mais. Embora seja resultado do produto cartesiano entre vendedores e vendas, devido à condição de junção, a resposta contém apenas as linhas que associam corretamente uma venda com o vendedor que a efetuou.

Utilizar várias tabelas separadas por vírgulas no `from` é uma das formas de realizar junções. A outra forma é com `join`. Toda junção com `from` pode ser reescrita com `join` e vice-versa. Para que a junção fique explícita, vamos preferir a escrita com `join` à com `from`.

O `select`, com `join`, que produz a resposta desejada é:

```
select * from venda v join vendedor r on (v.vendedor = r.codigo) where (v.dia = '2000-05-30');
```

codigo	vendedor	dia	codigo	nome	salario
1728	10	2000-05-30	10	Mauricio Bezerra Lopes	550
1729	7	2000-05-30	7	Pedro Rosa Loureiro	850
1730	9	2000-05-30	9	Gabriel Albuquerque Menezes	500
1731	2	2000-05-30	2	Paulo Lopes Nunes	500
1732	3	2000-05-30	3	Patricia Menezes Silva	500

Observe que todos os elementos (as tabelas, a condição do filtro e a condição da junção) da junção com `from` estão presentes na junção com `join`, apenas foi utilizada uma sintaxe diferente.

A figura 4 lista os tipos de junções possíveis. A mais comum é a junção interna, ou `inner join`, utilizada no exemplo acima. Observe que o uso da palavra `inner` é opcional.

2.1.51. Outra forma de listar os produtos vendidos entre 10/01/2000 e 15/01/2000:

Anteriormente, no exemplo 2.1.34, foi utilizado o `select` abaixo, com `in`, para obter a resposta desejada.

```
select p.descricao from produto p where (p.codigo in (select i.produto from item i where (i.venda in (select v.codigo from venda v where (v.dia between '2000-01-10' and '2000-01-15'))))));
```

Com base no mesmo raciocínio, podemos, agora, escrever um `select` com `join` que produz exatamente a mesma resposta. Acompanhe, passo-a-passo, a construção da resposta final.

Passo 1: Filtrar as vendas no intervalo de interesse.

```
select * from venda v where (v.dia between '2000-01-10' and '2000-01-15');
```

codigo	vendedor	dia
96	8	2000-01-10
97	7	2000-01-10
98	7	2000-01-10
99	7	2000-01-10
100	10	2000-01-10
101	3	2000-01-10
102	1	2000-01-10
103	6	2000-01-10
104	9	2000-01-10
105	7	2000-01-10
106	2	2000-01-10
107	5	2000-01-10
108	1	2000-01-10
109	2	2000-01-10
110	10	2000-01-10
111	7	2000-01-10
112	4	2000-01-10
113	3	2000-01-11
114	9	2000-01-11
115	4	2000-01-11
116	9	2000-01-11
117	8	2000-01-11
118	2	2000-01-11
119	1	2000-01-11
120	5	2000-01-11
121	10	2000-01-11
122	5	2000-01-11
123	4	2000-01-11
124	10	2000-01-11
125	6	2000-01-12
126	7	2000-01-12
127	1	2000-01-12
128	1	2000-01-12
129	9	2000-01-12
130	9	2000-01-12
131	3	2000-01-12
132	3	2000-01-12
133	4	2000-01-12
134	1	2000-01-12
135	4	2000-01-12
136	6	2000-01-12
137	10	2000-01-12
138	7	2000-01-12
139	8	2000-01-13
140	8	2000-01-13
141	2	2000-01-13
142	1	2000-01-13
143	8	2000-01-13
144	1	2000-01-13
145	9	2000-01-14
146	3	2000-01-14
147	6	2000-01-14
148	10	2000-01-14
149	3	2000-01-14
150	2	2000-01-14
151	5	2000-01-14
152	1	2000-01-14
153	8	2000-01-14
154	10	2000-01-15

	155	6	2000-01-15
	156	1	2000-01-15
	157	5	2000-01-15
	158	8	2000-01-15
	159	7	2000-01-15
	160	9	2000-01-15
	161	3	2000-01-15
	162	10	2000-01-15
	163	1	2000-01-15
	164	6	2000-01-15

Passo 2: Acrescentar as colunas da tabela item, mas apenas as linhas corretamente associadas à cada venda.

```
select * from venda v join item i on (i.venda = v.codigo) where (v.dia between '2000-01-10' and '2000-01-15');
```

codigo	vendedor	dia	venda	produto	quantidade
96	8	2000-01-10	96	8	1
96	8	2000-01-10	96	1122	2
96	8	2000-01-10	96	1065	1
96	8	2000-01-10	96	1687	1
96	8	2000-01-10	96	113	1
96	8	2000-01-10	96	932	1
96	8	2000-01-10	96	18	1
96	8	2000-01-10	96	1854	1
97	7	2000-01-10	97	465	1
98	7	2000-01-10	98	1126	1
98	7	2000-01-10	98	1418	1
98	7	2000-01-10	98	679	3
98	7	2000-01-10	98	506	1
98	7	2000-01-10	98	1156	1
...					
162	10	2000-01-15	162	1265	1
162	10	2000-01-15	162	501	1
162	10	2000-01-15	162	633	1
162	10	2000-01-15	162	1085	1
162	10	2000-01-15	162	378	1
162	10	2000-01-15	162	960	1
162	10	2000-01-15	162	256	1
163	1	2000-01-15	163	1038	1
163	1	2000-01-15	163	1600	5
163	1	2000-01-15	163	1456	1
163	1	2000-01-15	163	878	1
163	1	2000-01-15	163	1455	1
163	1	2000-01-15	163	908	1
163	1	2000-01-15	163	1055	1
164	6	2000-01-15	164	223	1

Passo 3: Acrescentar as colunas da tabela produto, mas apenas as linhas corretamente associadas à cada item.

```
select * from venda v join item i on (i.venda = v.codigo) join produto p on (i.produto = p.codigo) where (v.dia between '2000-01-10' and '2000-01-15');
```

codigo	vendedor	dia	venda	produto	quantidade	codigo	descricao	preco	comissao
96	8	2000-01-10	96	8	1	8	adaptador p/ impressao tel00-plu trendnet usb em uso de rede	63.25	NULL
96	8	2000-01-10	96	1122	2	1122	nb mem ddr2 2gb 800mhz corsair	48.5	3
96	8	2000-01-10	96	1065	1	1065	mp7-celular mox mo-829 tv/phone/mp3/mp4/tela 2.0 8.0 mpixel /4 bandas (black)	60.5	2
96	8	2000-01-10	96	1687	1	1687	vga pci 512 mb ddr2 gf 6200a 350m 64bit vga/dvi/hdvtv xfx	38.5	2
96	8	2000-01-10	96	113	1	113	camera foto polaroid i035 10.1 black	66	3
96	8	2000-01-10	96	932	1	932	mouse ps/2 a5 satellite optical (preto)	5.6	NULL
96	8	2000-01-10	96	18	1	18	antena parabolica 80cm s/lmb shop	27.5	3
96	8	2000-01-10	96	1854	1	1854	wir. mk mini pci cartao router board r52h a/b/g 320mw	59	NULL
97	7	2000-01-10	97	465	1	465	gaveta p/ hd sata 2.5/3.5 usb 2.0 ext. satellite ax-dk226 (branco)	23.5	2
98	7	2000-01-10	98	1126	1	1126	nb mem ddr3 1gb pc8500 samsung	12.5	3
98	7	2000-01-10	98	1418	1	1418	pos leitor laser kaiomy hbs-610 contact omni usb	217	3
98	7	2000-01-10	98	679	3	679	impressora samsung laser ml-1915	107	3
98	7	2000-01-10	98	506	1	506	hd ext. usb 2.5 320 gb samsung s2 portable blue	86	4
98	7	2000-01-10	98	1156	1	1156	nb acer as5738g-6536 core 2 duo 2.2/4gb/500/dvdrw/15.6/linux/webcam	881	5
...									
162	10	2000-01-15	162	1265	1	1265	pda smartphone kaiomy aegis i183 2.8/4bandas/460mhz/128mb/wi-fi lan/win mobile 6.1 portugues	157.5	4
162	10	2000-01-15	162	501	1	501	hd ext. usb 2.5 160 gb samsung s2 portable pink	65.5	4
162	10	2000-01-15	162	633	1	633	hub switch 24p. 10/100 - encore enh924	45.5	2
162	10	2000-01-15	162	1085	1	1085	mp7-celular x tech xt n6310 tv/4b/2chip/ vermelho	83	2
162	10	2000-01-15	162	378	1	378	fonte gabinete atx p4 corsair 950watts-real ps/2 tx series	185	1
162	10	2000-01-15	162	960	1	960	mouse ps/2 usb satellite a15 opt.ver/min	6.3	NULL
162	10	2000-01-15	162	256	1	256	carregador de pilha mo-806 3 in 1-bi-volt mox	2.35	1
163	1	2000-01-15	163	1038	1	1038	mp3 - 8gb apple ipod nano black mc031zy/a	155	2
163	1	2000-01-15	163	1600	5	1600	teclado i-tech virtual laser infrared (full-size qwerty) para pda/blackberry	167	NULL
163	1	2000-01-15	163	1456	1	1456	proc. intel 775p core 2 quad q9400 2.66 6mb/1333mhz box	209	3

163	1	2000-01-15	163	878	1	878	midia dvd-r 4.7 gb kaiomy 120 min c/logo 16x speed 50 pcs	9.35	NULL
163	1	2000-01-15	163	1455	1	1455	proc. intel 775p core 2 quad q8400 2.66 4mb/1333 mhz box	183	3
163	1	2000-01-15	163	908	1	908	monit. lcd 20 samsung 2043nwx wide preto	182	4
163	1	2000-01-15	163	1055	1	1055	mp7-celular fujilink fu-91t tv/phone/mp3/mp4/4 bandas (branco)	58	2
164	6	2000-01-15	164	223	1	223	car alto falante p/cairo roadstar-rs-6908- 1800w	19.28	1

Passo 4: Listar apenas os nomes, sem repetições, dos produtos vendidos no intervalo de interesse.

```
select distinct p.descricao from venda v join item i on (i.venda = v.codigo) join produto p on (i.produto = p.codigo) where (v.dia between '2000-01-10' and '2000-01-15');
```

```
+-----+-----+-----+
| descricao |
+-----+-----+-----+
| adaptador p/ impressao tel100-plu trendnet usb em uso de rede |
| antena parabolica 80cm s/lmb shop+ |
| camera foto polaroid i1035 10.1 black |
| mouse ps/2 a5 satellite optical (preto) |
| mp7-celular mox mo-829 tv/phone/mp3/mp4/tela 2.0 8.0 mpixel /4 bandas (black) |
| nb mem ddr2 2gb 800mhz corsair |
| vga pci 512 mb ddr2 gf 6200a 350m 64bit vga/dvi/hdvtv xfx |
| wir. mk mini pci cartao router board r52h a/b/g 320mw |
| gaveta p/ hd sata 2.5/3.5 usb 2.0 ext. satellite ax-dk226 (branco) |
| hd ext. usb 2.5 320 gb samsung s2 portable blue |
|
| fonte gabinete atx p4 corsair 950watts-real ps/2 tx series |
| hd ext. usb 2.5 160 gb samsung s2 portable pink |
| mouse ps/2 usb satellite a15 opt.ver/min |
| mp7-celular x tech xt n6310 tv/4b/2chip/ vermelho |
| pda smartphone kaiomy aegis 1183 2.8/4bandas/460mhz/128mb/wi-fi lan/win mobile 6.1 portuges |
| monit. lcd 20 samsung 2043nwx wide preto |
| mp7-celular fujilink fu-91t tv/phone/mp3/mp4/4 bandas (branco) |
| proc. intel 775p core 2 quad q8400 2.66 4mb/1333 mhz box |
| proc. intel 775p core 2 quad q9400 2.66 6mb/1333mhz box |
| teclado i-tech virtual laser infrared (full-size qwerty) para pda/blackberry |
+-----+-----+-----+
```

Na realidade, aqui cabe uma observação muito importante. Um `select` com `in` sempre pode ser reescrito como um `select` com `join`, mas não o contrário.

2.1.52. Outra forma de contar a quantidade de vendas entre 10/01/2000 e 15/01/2000 do vendedor Maurício Costa Quaresma:

Anteriormente, no exemplo 2.1.39, foi preciso informar, além do nome do vendedor, seu código. E o motivo é simples. Não existe uma coluna na tabela venda para o nome do vendedor. E nem deveria. Basta o código do vendedor para saber quem efetuou cada venda. Não há razão para se gastar espaço com duplicação de dados, armazenando o nome do vendedor também na tabela venda, uma vez que, indiretamente, o código do vendedor nos permite determinar o nome de quem efetuou a venda.

Para resolver o problema sabendo apenas o nome do vendedor, precisamos utilizar junções. Acompanhe, passo-a-passo, a construção da resposta final.

Passo 1: Filtrar as vendas no intervalo de interesse.

```
select * from venda v where (v.dia between '2000-01-10' and '2000-01-15');
```

```
+-----+-----+-----+
| codigo | vendedor | dia |
+-----+-----+-----+
| 96 | 8 | 2000-01-10 |
| 97 | 7 | 2000-01-10 |
| 98 | 7 | 2000-01-10 |
| 99 | 7 | 2000-01-10 |
| 100 | 10 | 2000-01-10 |
| 101 | 3 | 2000-01-10 |
| 102 | 1 | 2000-01-10 |
```

	103	6	2000-01-10	
	104	9	2000-01-10	
	105	7	2000-01-10	
	106	2	2000-01-10	
	107	5	2000-01-10	
	108	1	2000-01-10	
	109	2	2000-01-10	
	110	10	2000-01-10	
	111	7	2000-01-10	
	112	4	2000-01-10	
	113	3	2000-01-11	
	114	9	2000-01-11	
	115	4	2000-01-11	
	116	9	2000-01-11	
	117	8	2000-01-11	
	118	2	2000-01-11	
	119	1	2000-01-11	
	120	5	2000-01-11	
	121	10	2000-01-11	
	122	5	2000-01-11	
	123	4	2000-01-11	
	124	10	2000-01-11	
	125	6	2000-01-12	
	126	7	2000-01-12	
	127	1	2000-01-12	
	128	1	2000-01-12	
	129	9	2000-01-12	
	130	9	2000-01-12	
	131	3	2000-01-12	
	132	3	2000-01-12	
	133	4	2000-01-12	
	134	1	2000-01-12	
	135	4	2000-01-12	
	136	6	2000-01-12	
	137	10	2000-01-12	
	138	7	2000-01-12	
	139	8	2000-01-13	
	140	8	2000-01-13	
	141	2	2000-01-13	
	142	1	2000-01-13	
	143	8	2000-01-13	
	144	1	2000-01-13	
	145	9	2000-01-14	
	146	3	2000-01-14	
	147	6	2000-01-14	
	148	10	2000-01-14	
	149	3	2000-01-14	
	150	2	2000-01-14	
	151	5	2000-01-14	
	152	1	2000-01-14	
	153	8	2000-01-14	
	154	10	2000-01-15	
	155	6	2000-01-15	
	156	1	2000-01-15	
	157	5	2000-01-15	
	158	8	2000-01-15	
	159	7	2000-01-15	
	160	9	2000-01-15	
	161	3	2000-01-15	
	162	10	2000-01-15	
	163	1	2000-01-15	
	164	6	2000-01-15	
	+-----+			

Passo 2: Acrescentar as colunas da tabela vendedor, mas apenas as linhas corretamente associadas à cada venda.

```
select * from venda v join vendedor r on (v.vendedor = r.codigo) where (v.dia between '2000-01-10' and '2000-01-15');
```

codigo	vendedor	dia	codigo	nome	salario
96	8	2000-01-10	8	Jose Souza Martins	550
97	7	2000-01-10	7	Pedro Rosa Loureiro	850
98	7	2000-01-10	7	Pedro Rosa Loureiro	850
99	7	2000-01-10	7	Pedro Rosa Loureiro	850
100	10	2000-01-10	10	Mauricio Bezerra Lopes	550
101	3	2000-01-10	3	Patricia Menezes Silva	500
102	1	2000-01-10	1	Mauricio Costa Quaresma	500
103	6	2000-01-10	6	Cristiane Nunes Brandao	500
104	9	2000-01-10	9	Gabriel Albuquerque Menezes	500
105	7	2000-01-10	7	Pedro Rosa Loureiro	850
106	2	2000-01-10	2	Paulo Lopes Nunes	500
107	5	2000-01-10	5	Michele Menezes Santos	550
108	1	2000-01-10	1	Mauricio Costa Quaresma	500
109	2	2000-01-10	2	Paulo Lopes Nunes	500
110	10	2000-01-10	10	Mauricio Bezerra Lopes	550
111	7	2000-01-10	7	Pedro Rosa Loureiro	850
112	4	2000-01-10	4	Cristiano Lopes Bezerra	550
113	3	2000-01-11	3	Patricia Menezes Silva	500
114	9	2000-01-11	9	Gabriel Albuquerque Menezes	500
115	4	2000-01-11	4	Cristiano Lopes Bezerra	550
116	9	2000-01-11	9	Gabriel Albuquerque Menezes	500
117	8	2000-01-11	8	Jose Souza Martins	550
118	2	2000-01-11	2	Paulo Lopes Nunes	500
119	1	2000-01-11	1	Mauricio Costa Quaresma	500
120	5	2000-01-11	5	Michele Menezes Santos	550
121	10	2000-01-11	10	Mauricio Bezerra Lopes	550
122	5	2000-01-11	5	Michele Menezes Santos	550
123	4	2000-01-11	4	Cristiano Lopes Bezerra	550
124	10	2000-01-11	10	Mauricio Bezerra Lopes	550
125	6	2000-01-12	6	Cristiane Nunes Brandao	500
126	7	2000-01-12	7	Pedro Rosa Loureiro	850
127	1	2000-01-12	1	Mauricio Costa Quaresma	500
128	1	2000-01-12	1	Mauricio Costa Quaresma	500
129	9	2000-01-12	9	Gabriel Albuquerque Menezes	500
130	9	2000-01-12	9	Gabriel Albuquerque Menezes	500
131	3	2000-01-12	3	Patricia Menezes Silva	500
132	3	2000-01-12	3	Patricia Menezes Silva	500
133	4	2000-01-12	4	Cristiano Lopes Bezerra	550
134	1	2000-01-12	1	Mauricio Costa Quaresma	500
135	4	2000-01-12	4	Cristiano Lopes Bezerra	550
136	6	2000-01-12	6	Cristiane Nunes Brandao	500
137	10	2000-01-12	10	Mauricio Bezerra Lopes	550
138	7	2000-01-12	7	Pedro Rosa Loureiro	850
139	8	2000-01-13	8	Jose Souza Martins	550
140	8	2000-01-13	8	Jose Souza Martins	550
141	2	2000-01-13	2	Paulo Lopes Nunes	500
142	1	2000-01-13	1	Mauricio Costa Quaresma	500
143	8	2000-01-13	8	Jose Souza Martins	550
144	1	2000-01-13	1	Mauricio Costa Quaresma	500
145	9	2000-01-14	9	Gabriel Albuquerque Menezes	500
146	3	2000-01-14	3	Patricia Menezes Silva	500
147	6	2000-01-14	6	Cristiane Nunes Brandao	500
148	10	2000-01-14	10	Mauricio Bezerra Lopes	550
149	3	2000-01-14	3	Patricia Menezes Silva	500
150	2	2000-01-14	2	Paulo Lopes Nunes	500
151	5	2000-01-14	5	Michele Menezes Santos	550
152	1	2000-01-14	1	Mauricio Costa Quaresma	500
153	8	2000-01-14	8	Jose Souza Martins	550
154	10	2000-01-15	10	Mauricio Bezerra Lopes	550
155	6	2000-01-15	6	Cristiane Nunes Brandao	500
156	1	2000-01-15	1	Mauricio Costa Quaresma	500
157	5	2000-01-15	5	Michele Menezes Santos	550
158	8	2000-01-15	8	Jose Souza Martins	550
159	7	2000-01-15	7	Pedro Rosa Loureiro	850
160	9	2000-01-15	9	Gabriel Albuquerque Menezes	500

161	3	2000-01-15	3	Patricia Menezes Silva	500
162	10	2000-01-15	10	Mauricio Bezerra Lopes	550
163	1	2000-01-15	1	Mauricio Costa Quaresma	500
164	6	2000-01-15	6	Cristiane Nunes Brandao	500

Passo 3: Filtrar apenas as vendas do vendedor Maurício Costa Quaresma no intervalo de interesse.

```
select * from venda v join vendedor r on (v.vendedor = r.codigo) where (v.dia between '2000-01-10' and '2000-01-15') and (r.nome = 'Mauricio Costa Quaresma');
```

codigo	vendedor	dia	codigo	nome	salario
102	1	2000-01-10	1	Mauricio Costa Quaresma	500
108	1	2000-01-10	1	Mauricio Costa Quaresma	500
119	1	2000-01-11	1	Mauricio Costa Quaresma	500
127	1	2000-01-12	1	Mauricio Costa Quaresma	500
128	1	2000-01-12	1	Mauricio Costa Quaresma	500
134	1	2000-01-12	1	Mauricio Costa Quaresma	500
142	1	2000-01-13	1	Mauricio Costa Quaresma	500
144	1	2000-01-13	1	Mauricio Costa Quaresma	500
152	1	2000-01-14	1	Mauricio Costa Quaresma	500
156	1	2000-01-15	1	Mauricio Costa Quaresma	500
163	1	2000-01-15	1	Mauricio Costa Quaresma	500

Passo 4: Contar a quantidade de vendas do vendedor Maurício Costa Quaresma no intervalo de interesse.

```
select count(*) as quantidade from venda v join vendedor r on (v.vendedor = r.codigo) where (v.dia between '2000-01-10' and '2000-01-15') and (r.nome = 'Mauricio Costa Quaresma');
```

quantidade
11

2.1.53. Outra forma de contar a quantidade de unidades do produto camera webcam genius eye 320 vendidas em outubro de 2000:

Este exemplo é parecido com o acima. Anteriormente, no exemplo 2.1.42, foi preciso informar, além do nome do produto, seu código, visto que não existe, nem na tabela venda, nem na tabela item, uma coluna para a descrição do produto vendido. E nem deveria existir. Novamente, não há razão para se gastar espaço com duplicação de dados, armazenando a descrição do produto nestas tabelas.

Para resolver o problema sabendo apenas a descrição do produto, precisamos utilizar junções. Acompanhe, passo-a-passo, a construção da resposta final.

Passo 1: Filtrar as vendas no intervalo de interesse.

```
select * from venda v where (cast(v.dia as char) like '2000-10-%');
```

codigo	vendedor	dia
3312	7	2000-10-01
3313	10	2000-10-01
3314	2	2000-10-01
3315	8	2000-10-01
3316	2	2000-10-01
3317	6	2000-10-01
3318	7	2000-10-01
3319	5	2000-10-01
3320	6	2000-10-01

	3321	6	2000-10-02
	3322	1	2000-10-02
	3323	9	2000-10-02
	3324	5	2000-10-02
	3325	1	2000-10-02
	...		
	3651	2	2000-10-30
	3652	8	2000-10-30
	3653	6	2000-10-30
	3654	6	2000-10-30
	3655	6	2000-10-30
	3656	5	2000-10-30
	3657	3	2000-10-30
	3658	8	2000-10-30
	3659	10	2000-10-30
	3660	6	2000-10-30
	3661	1	2000-10-30
	3662	6	2000-10-30
	3663	6	2000-10-30
	3664	4	2000-10-31
	3665	9	2000-10-31
	3666	4	2000-10-31
	3667	5	2000-10-31
	3668	9	2000-10-31
	3669	4	2000-10-31
	3670	8	2000-10-31
	3671	8	2000-10-31
	3672	6	2000-10-31
	3673	7	2000-10-31
	3674	1	2000-10-31
	3675	1	2000-10-31

Passo 2: Acrescentar as colunas da tabela item, mas apenas as linhas corretamente associadas à cada venda.

```
select * from venda v join item i on (i.venda = v.codigo) where (cast(v.dia as char) like '2000-10-%');
```

codigo	vendedor	dia	venda	produto	quantidade
3312	7	2000-10-01	3312	1381	5
3312	7	2000-10-01	3312	161	1
3313	10	2000-10-01	3313	550	1
3313	10	2000-10-01	3313	1167	1
3313	10	2000-10-01	3313	886	1
3314	2	2000-10-01	3314	129	1
3314	2	2000-10-01	3314	1202	1
3314	2	2000-10-01	3314	1530	1
3314	2	2000-10-01	3314	1764	1
3314	2	2000-10-01	3314	922	1
3314	2	2000-10-01	3314	527	5
3314	2	2000-10-01	3314	740	1
3314	2	2000-10-01	3314	1601	4
3314	2	2000-10-01	3314	1217	2
3315	8	2000-10-01	3315	709	1
3315	8	2000-10-01	3315	355	1
3315	8	2000-10-01	3315	10	3
3315	8	2000-10-01	3315	1076	1
3316	2	2000-10-01	3316	454	1
3316	2	2000-10-01	3316	227	3
3316	2	2000-10-01	3316	432	3
3316	2	2000-10-01	3316	183	1
3316	2	2000-10-01	3316	1170	1
3316	2	2000-10-01	3316	515	1
3316	2	2000-10-01	3316	1072	1
3316	2	2000-10-01	3316	945	1
3316	2	2000-10-01	3316	349	1
3316	2	2000-10-01	3316	880	5
3317	6	2000-10-01	3317	560	1
3317	6	2000-10-01	3317	225	1
3317	6	2000-10-01	3317	1153	1
3317	6	2000-10-01	3317	1155	1
3317	6	2000-10-01	3317	803	1
3317	6	2000-10-01	3317	1524	1

	3318	7	2000-10-01	3318	1771	1
	3318	7	2000-10-01	3318	327	5
	3318	7	2000-10-01	3318	302	1
	3318	7	2000-10-01	3318	500	1
	3318	7	2000-10-01	3318	658	1
	3318	7	2000-10-01	3318	1204	1
	3318	7	2000-10-01	3318	1275	1
	3318	7	2000-10-01	3318	1058	1
	3318	7	2000-10-01	3318	998	1
	3318	7	2000-10-01	3318	352	1
	3319	5	2000-10-01	3319	1343	1
	3319	5	2000-10-01	3319	1705	1
	3319	5	2000-10-01	3319	1819	2
	3319	5	2000-10-01	3319	1736	1
	3319	5	2000-10-01	3319	1360	1
	3319	5	2000-10-01	3319	788	1
	3319	5	2000-10-01	3319	1394	1
	3319	5	2000-10-01	3319	1782	4
	3319	5	2000-10-01	3319	1805	4
	3319	5	2000-10-01	3319	782	1
	3320	6	2000-10-01	3320	943	1
	3320	6	2000-10-01	3320	1507	1
	3320	6	2000-10-01	3320	1852	1
	3320	6	2000-10-01	3320	291	1
	3320	6	2000-10-01	3320	1367	4
	3320	6	2000-10-01	3320	1276	1
	3320	6	2000-10-01	3320	1409	1
	3320	6	2000-10-01	3320	1754	4
	3320	6	2000-10-01	3320	384	5
	3320	6	2000-10-01	3320	1491	1
				...		
	3664	4	2000-10-31	3664	570	2
	3664	4	2000-10-31	3664	166	1
	3665	9	2000-10-31	3665	1362	1
	3665	9	2000-10-31	3665	840	1
	3665	9	2000-10-31	3665	1178	1
	3665	9	2000-10-31	3665	1862	1
	3665	9	2000-10-31	3665	962	1
	3665	9	2000-10-31	3665	182	1
	3665	9	2000-10-31	3665	1217	1
	3665	9	2000-10-31	3665	1187	1
	3665	9	2000-10-31	3665	84	1
	3665	9	2000-10-31	3665	709	4
	3666	4	2000-10-31	3666	928	1
	3666	4	2000-10-31	3666	1730	3
	3666	4	2000-10-31	3666	839	1
	3666	4	2000-10-31	3666	1652	1
	3666	4	2000-10-31	3666	329	1
	3666	4	2000-10-31	3666	1673	1
	3666	4	2000-10-31	3666	408	1
	3666	4	2000-10-31	3666	1124	1
	3666	4	2000-10-31	3666	498	1
	3667	5	2000-10-31	3667	956	1
	3667	5	2000-10-31	3667	1031	1
	3667	5	2000-10-31	3667	335	1
	3667	5	2000-10-31	3667	744	1
	3668	9	2000-10-31	3668	1489	1
	3668	9	2000-10-31	3668	1716	1
	3669	4	2000-10-31	3669	817	1
	3669	4	2000-10-31	3669	630	1
	3669	4	2000-10-31	3669	1584	1
	3669	4	2000-10-31	3669	1728	3
	3669	4	2000-10-31	3669	800	1
	3670	8	2000-10-31	3670	392	1
	3670	8	2000-10-31	3670	1122	1
	3670	8	2000-10-31	3670	1845	1
	3670	8	2000-10-31	3670	164	1
	3670	8	2000-10-31	3670	475	1
	3671	8	2000-10-31	3671	376	1
	3671	8	2000-10-31	3671	137	1
	3671	8	2000-10-31	3671	1144	1
	3671	8	2000-10-31	3671	1743	1
	3671	8	2000-10-31	3671	1306	1
	3671	8	2000-10-31	3671	868	1
	3671	8	2000-10-31	3671	1834	1
	3671	8	2000-10-31	3671	962	1

			3671		8	2000-10-31	3671	1264		2	
			3671		8	2000-10-31	3671	543		1	
			3672		6	2000-10-31	3672	1528		1	
			3672		6	2000-10-31	3672	304		1	
			3672		6	2000-10-31	3672	484		1	
			3672		6	2000-10-31	3672	22		1	
			3672		6	2000-10-31	3672	844		3	
			3673		7	2000-10-31	3673	597		1	
			3673		7	2000-10-31	3673	1526		1	
			3673		7	2000-10-31	3673	1642		1	
			3674		1	2000-10-31	3674	616		1	
			3674		1	2000-10-31	3674	1131		5	
			3674		1	2000-10-31	3674	151		1	
			3674		1	2000-10-31	3674	687		1	
			3674		1	2000-10-31	3674	240		1	
			3674		1	2000-10-31	3674	1731		1	
			3674		1	2000-10-31	3674	697		3	
			3674		1	2000-10-31	3674	570		1	
			3675		1	2000-10-31	3675	1057		4	
			3675		1	2000-10-31	3675	401		5	
			3675		1	2000-10-31	3675	1496		2	
			3675		1	2000-10-31	3675	1646		1	
			3675		1	2000-10-31	3675	1049		1	
			3675		1	2000-10-31	3675	990		5	
			3675		1	2000-10-31	3675	1867		2	
			3675		1	2000-10-31	3675	601		1	

Passo 3: Acrescentar as colunas da tabela produto, mas apenas as linhas corretamente associadas à cada item.

```
select * from venda v join item i on (i.venda = v.codigo) join produto p on (i.produto = p.codigo) where (cast(v.dia as char) like '2000-10-%');
```

codigo	vendedor	dia	venda	produto	quantidade	codigo	descricao	preco	comissao
3312	7	2000-10-01	3312	1381	5	1381	placa pci sabrent uf6 4 portas usb 2.0 + 2 fire wire ieee 1394 com cabo firewire 6pin to 4pin	10,85	NULL
3312	7	2000-10-01	3312	161	1	161	camera webcam kaiomy talkcam-v 1.3 black/silver	8	3
3313	10	2000-10-01	3313	550	1	550	hd ext. usb 3.5 1tb iomega prestige hdd 34275	129	4
3313	10	2000-10-01	3313	1167	1	1167	nb acer as7720-6712/ intel core 2 duo 1.66/2gb/250/dvdrw/17/black/vista/webcam	792	5
3313	10	2000-10-01	3313	886	1	886	midia dvd-rw mini philips 10pc p/film 60min/2.8gb 2xspeed	15,95	NULL
3314	2	2000-10-01	3314	129	1	129	camera foto samsung digimax sl620 12mp silver	161	3
3314	2	2000-10-01	3314	1202	1	1202	nb gateway ml63lu amd turion 2.0/4gb/320/dvdrw 15.4 whpx64/ wcam	576	5
3314	2	2000-10-01	3314	1530	1	1530	software microsoft win. vista business 32bit oem espanhol. + licencia **original**	133,5	NULL
3314	2	2000-10-01	3314	1764	1	1764	videocamera kaiomy usb ext.14tvpnc-u2 tv/fm palm/n ntsc c/control	34,2	NULL
3314	2	2000-10-01	3314	922	1	922	mouse mini labtec optical glow black	21,5	NULL
3314	2	2000-10-01	3314	527	5	527	hd ext. usb 2.5 320 gb samsung s2 portable red	86,5	4
3314	2	2000-10-01	3314	740	1	740	nb am3 asrock 780gh/128m hdm/som/vga/lan/sata/ddr3	96	3
3314	2	2000-10-01	3314	1602	4	1602	teclado logitech cordless desktop comfort laser 967692	85	NULL
3314	2	2000-10-01	3314	1217	2	1217	nb hp dv4-1430us rb intel core 2 duo 2.1/4gb/320/dvdrw 14.1 vista wcam(white)	715	5
3315	8	2000-10-01	3315	709	1	709	maleta p/notebook 14 hnr 249 kaiba	12,43	NULL
3315	8	2000-10-01	3315	355	1	355	estab. well 2000va bivolt p/02 maquina	35	1
3315	8	2000-10-01	3315	10	3	10	adaptador satellite usb 2.0 al-08	1,6	NULL
3315	8	2000-10-01	3315	1076	1	1076	mp7-celular mox mo-873 tv/phone/mp3/mp4/tela 2.4 9.0 mpixel /4 bandas (gray)	85	2
3316	2	2000-10-01	3316	454	1	454	gaveta p/ hd nb sata dr.hank 2.5 usb ah-2520 2.0 ext.	8,5	2
3316	2	2000-10-01	3316	227	3	227	car amplificador p/carro roadstar rs-580v vacuum amp	103,82	2
3316	2	2000-10-01	3316	432	3	432	game joyetek fujitwork playstation 2 gamepad (red)	8,2	NULL
3316	2	2000-10-01	3316	183	1	183	camera webcam v-gear talkcam live 1.3 black	6	3
3316	2	2000-10-01	3316	1170	1	1170	nb acer aspire aoa110-1564/atom 1.6/512mb/hd8gb/8.9/blue/linux/webcam/3cel.	239	5
3316	2	2000-10-01	3316	515	1	515	hd ext. usb 2.5 160 gb samsung s2 portable red	69,5	4
3316	2	2000-10-01	3316	1072	1	1072	mp7-celular mox mo-873 tv/phone/mp3/mp4/tela 2.28.0 mpixel /4 bandas(black)	81	2
3316	2	2000-10-01	3316	945	1	945	mouse ps/2 omega 3d net.	1,25	NULL
3316	2	2000-10-01	3316	349	1	349	estab. fgt- 1000kva bivolt p/02 maquina	28	1
3316	2	2000-10-01	3316	880	5	880	midia dvd-r 4,7 gb philips 120 min c/logo 16x speed 25 pcs(capa)	7,48	NULL
3317	6	2000-10-01	3317	560	1	560	hd nb sata 2.5 500 gb samsung 5400rpm	44,5	3
3317	6	2000-10-01	3317	225	1	225	car alto falante p/carro roadstar-rs-1692 800w	11,2	1
3317	6	2000-10-01	3317	1153	1	1153	nb acer as5738-6294 intel core 2 duo 2.2/3gb/320/dvdrw/15.6 linux c/cam	698	5
3317	6	2000-10-01	3317	1155	1	1155	nb acer as5738-6294 intel core 2 duo 2.2/4gb/500/dvdrw/15.6 linux c/cam	765	5
3317	6	2000-10-01	3317	803	1	803	memoria sd 16gb kingston high capacity classe 4	802	3
3317	6	2000-10-01	3317	1524	1	1524	reprodutor carcd/usb multimedia bt-701p c/control	32,5	NULL
3318	7	2000-10-01	3318	1771	1	1771	wir. adaptador pci nexxt nwl22nt12 54mbps	11,1	NULL
3318	7	2000-10-01	3318	327	5	327	dvd player philips dvp-3350 usb	42,5	2
3318	7	2000-10-01	3318	302	1	302	cooler omega xp-2200 base cobre n82	4,4	NULL
3318	7	2000-10-01	3318	500	1	500	hd ext. usb 2.5 160 gb samsung s2 portable blue	65,5	4
3318	7	2000-10-01	3318	658	1	658	impressora hp col laserjet 4700n 31/31ppm 160mb 110v	1695	3
3318	7	2000-10-01	3318	1204	1	1204	nb hp 6730b intel core 2 duo 2.26/2048/160/dvdrw/14.1 vista c/cam	1086	5
3318	7	2000-10-01	3318	1275	1	1275	pendrive 16gb supertalent pico-c nickel usb 2.0	39	2
3318	7	2000-10-01	3318	1058	1	1058	mp7-celular fujilink fu-915 tv/phone/mp3/mp4/4 bandas (preto)	58	2
3318	7	2000-10-01	3318	998	1	998	mouse usb satellite all opt. azul/mini	7,75	NULL
3318	7	2000-10-01	3318	352	1	352	estab. microfins 1000 bivolt p/02 maquina	22,5	1
3319	5	2000-10-01	3319	1343	1	1343	pendrive 8gb supertalent sud-s8gb(silver/preto)	18,3	2
3319	5	2000-10-01	3319	1705	1	1705	vga pciexp. 512 mb ddr2 hdg4670 650m vga/dvi/hdmi sapphire	67	2
3319	5	2000-10-01	3319	1819	2	1819	wir. ap d-link dwl-2100ap 108mbps	63,5	5
3319	5	2000-10-01	3319	1736	1	1736	vga pciexp.lgb ddr3 hd4670 750m 128bit vga/dvi/hdmi sapphire	84,5	2
3319	5	2000-10-01	3319	1360	1	1360	pendrive 8gb kingston mini pink dtman	20,4	2
3319	5	2000-10-01	3319	788	1	788	nb p4 4775 intel dg3lpr som/vga/sata/gblan/ddr2 1066mhz	69	3
3319	5	2000-10-01	3319	1394	1	1394	porta retrato digital jwin 7 jpl147	42,5	NULL
3319	5	2000-10-01	3319	1782	4	1782	wir. adaptador usb kaiomy wa-150u pro 150mbps n lite antenna direcional 6dbi ralink	24	NULL
3319	5	2000-10-01	3319	1805	4	1805	wir. antena hyperlink 24dbi 2.4ghz paraboloic hg2424g	48,75	NULL
3319	5	2000-10-01	3319	782	1	782	nb p4 4775 ace g1it-m som/vga/lan/sata/ddr2/core - proc. quad core	43,75	5
3320	6	2000-10-01	3320	943	1	943	mouse ps/2 mtek net.opt.ms-b130 branco	3,5	NULL
3320	6	2000-10-01	3320	1507	1	1507	rede pci 10/100 3com 3c905c-tx-m c/cabo	9,8	NULL
3320	6	2000-10-01	3320	1852	1	1852	wir. apresentador laser pointer p 2,4gz infiniter lr-12r	31,9	NULL
3320	6	2000-10-01	3320	291	1	291	celular kaiomy segis 806 camera 2.0mp/3.0touch/2chips/2gb/3bandas(mp9)	89,5	3
3320	6	2000-10-01	3320	1367	4	1367	pilha recarr.mo-aa3000b2 c/2aa 3000mah mox	1,4	NULL
3320	6	2000-10-01	3320	1276	1	1276	pendrive 16gb supertalent pico-c silver usb 2.0	38,75	2
3320	6	2000-10-01	3320	1409	1	1409	pos leitor bematech laser aguilas-a2400 ps2 preto	296	3
3320	6	2000-10-01	3320	1754	4	1754	video pixelview play tv box6 c/contr ext. (1680x1050)	43,75	NULL
3320	6	2000-10-01	3320	384	5	384	fonte gabinete atx p4 mtek 650w 24/20 pinos pci-e 6 pin box	23,6	1
3320	6	2000-10-01	3320	1491	1	1491	receptor p/tv digital az-america s810 usb/ hdm /gravador pvr	128	3
3664	4	2000-10-31	3664	570	2	570	hd nb sata ref 2.5 300 gb fujitsu 5400rpm	52,25	3
3664	4	2000-10-31	3664	166	1	166	camera webcam satellite multi usb 2.0mp preto (wb-cl5)	21	3
3665	9	2000-10-31	3665	1362	1	1362	pilha alkalina mo-aab4alk c/4aa-mox	0,8	NULL
3665	9	2000-10-31	3665	840	1	840	microfone dr-hank ep-b960 headphoness	6	NULL
3665	9	2000-10-31	3665	1782	1	1782	nb acer aspire d550-i610/atom 1.6/1gb/160/10.1/red/winxp/webcam/6cel.	446	5
3665	9	2000-10-31	3665	1862	1	1862	wir. mk placa router board rb/450	70,5	NULL
3665	9	2000-10-31	3665	962	1	962	mouse ps/2 usb satellite a9 opt.silver	5,8	NULL
3665	9	2000-10-31	3665	182	1	182	camera webcam v-gear talkcam beauty 1.3 white	6,75	3
3665	9	2000-10-31	3665	1217	1	1217	nb hp dv4-1430us rb intel core 2 duo 2.1/4gb/320/dvdrw 14.1 vista wcam(white)	715	5
3665	9	2000-10-31	3665	1187	1	1187	nb acer e-machines eme627-5436 and athlon 1.6/2gb/160/dvdrw 15.6	451	5
3665	9	2000-10-31	3665	84	1	84	camera foto olympus tough-6000 10 mega pixel yellow/prova dagua	247	3

3665	9	2000-10-31	3665	709	4	709	maleta p/notebook 14 hnr 249 kaiba	12.43	NULL	
3666	4	2000-10-31	3666	928	1	928	mouse pad personalizado (stampa-varias>	1.15	NULL	
3666	4	2000-10-31	3666	1730	3	1730	vga pciexp.lgb ddr3 gf 9600gt 650m 256bit ddr3 vga/dvi/hdmi ecs	106	2	
3666	4	2000-10-31	3666	839	1	839	microfone logitech headphone noise canceling direct	15.5	NULL	
3666	4	2000-10-31	3666	1652	1	1652	tel panasonic kxtg9332 c/viva voz / secretaria 110v	71	NULL	
3666	4	2000-10-31	3666	329	1	329	dvd player portatil mox mo-dvd7007 7 c/tv widescreen (preto)	95	2	
3666	4	2000-10-31	3666	1673	1	1673	ups bca 650v 220v-60hz	65.5	3	
3666	4	2000-10-31	3666	408	1	408	gabinete p4 kit mtek mt7518 4baiais preto/prata	39.5	1	
3666	4	2000-10-31	3666	1124	1	1124	nb mem ddr2 2gb 800mhz kingston	48.5	3	
3666	4	2000-10-31	3666	498	1	498	hd ext. usb 1.8 120 gb samsung sl mini portable black	81	4	
3667	5	2000-10-31	3667	956	1	956	mouse ps/2 usb satellite a13 opt.azul mini	7	NULL	
3667	5	2000-10-31	3667	1031	1	1031	mp3 - 2gb mox mo-98 lcd 0.9 black	17.5	2	
3667	5	2000-10-31	3667	335	1	335	dvd player portatil mox mo-dvd9009 9 c/tv widescreen (vermelho)	125	2	
3667	5	2000-10-31	3667	744	1	744	nb am3 asus m4a785-m som/gblan/sata/vga/dvi/hdmi/ddr2 (green)	84.5	3	
3668	9	2000-10-31	3668	1489	1	1489	receptor p/tv digital az-america s807 usb/dv-out /gravador pvr	122	3	
3668	9	2000-10-31	3668	1716	1	1716	vga pciexp. 512 mb ddr3 gf gts250 738m dual dvi/hdvt xfx	150.5	2	
3669	4	2000-10-31	3669	937	1	937	mem ddr2 1gb 667mhz kingston	27.6	3	
3669	4	2000-10-31	3669	630	1	630	hub switch 24p. 10/100 nexxt nw223nxt54	55	2	
3669	4	2000-10-31	3669	1584	1	1584	speaker thonet vander 103-110 hoch 110v preto	119	NULL	
3669	4	2000-10-31	3669	1728	3	1728	vga pciexp.lgb ddr2 hd4670 750m 128bit ati tv/dual dvi/hdvt xfx	73.5	2	
3669	4	2000-10-31	3669	800	1	800	mem card micro sd 16gb markvision c/ adaptador	48.2	2	
3670	8	2000-10-31	3670	392	1	392	fonte universal x tech p/ notebook e monitor sc902 para carro e energia	29	1	
3670	8	2000-10-31	3670	1122	1	1122	nb mem ddr2 2gb 800mhz corsair	48.5	3	
3670	8	2000-10-31	3670	1845	1	1845	wir. ap/router tp-link tl-wr84lnd 4lan 1wan 300mbps	33.2	NULL	
3670	8	2000-10-31	3670	164	1	164	camera webcam omega silver	6.3	3	
3670	8	2000-10-31	3670	475	1	475	gaveta para cd-rom + hd 5.25 usb 2.0 external ax-521 satellite	23.5	2	
3671	8	2000-10-31	3671	376	1	376	fonte gabinete atx p4 corsair 450w real ps/2 vx us version	87	1	
3671	8	2000-10-31	3671	137	1	137	camera foto sony dsc-w270 12.1 black	230	3	
3671	8	2000-10-31	3671	1144	1	1144	nb acer as534-5950 amd athlon 1.6/2gb/160/dvdrw/15.6c/cam/ w7hp	450	2	
3671	8	2000-10-31	3671	1743	1	1743	vga pciexp.lgb ddr5 hd5770 850m 128bit ati hdmi/dual dvi/hdvt xfx	205.5	2	
3671	8	2000-10-31	3671	1306	1	1306	pendrive 4gb markvision usb2 resistente agua/queda	11.3	2	
3671	8	2000-10-31	3671	868	1	868	midia cd-r 700 mb.kaionoy 80 min.c/logo 52x speed (unidade em envelope)	0.3	NULL	
3671	8	2000-10-31	3671	1834	1	1834	wir. ap/router kaionoy 4lan 1wan 54mbps	22.2	NULL	
3671	8	2000-10-31	3671	962	1	962	mouse ps/2 usb satellite a9 opt.silver	5.8	NULL	
3671	8	2000-10-31	3671	1264	2	1264	pda smartphone kaionoy aegis 1183 2.8/4bandas/460mhz/128mb/wi-fi lan (b/g)/win mobile 6.1 espanhol	158.5	4	
3671	8	2000-10-31	3671	543	1	543	hd ext. usb 2.5 640 gb samsung s2 portable red	129	4	
3672	6	2000-10-31	3672	1528	1	1528	scanner hp scanjet g2710 - photosmart 2400x480dpi	97	2	
3672	6	2000-10-31	3672	304	1	304	cooler p/notebook 17ext.2fan usb evercool np-501-ar	19.5	NULL	
3672	6	2000-10-31	3672	484	1	484	gps orange navigation g420slim c/mapa touchscreen (white)	131	4	
3672	6	2000-10-31	3672	22	1	22	ar multi split hometiter 2x9000f/c 220v 60 hz br	862.4	5	
3672	6	2000-10-31	3672	844	3	844	microfone mult. 2 em 1 dr. hank ep-n220	4.75	NULL	
3673	7	2000-10-31	3673	597	1	597	hd sataii ref 320 gb seagate 7200 16mb	40.5	3	
3673	7	2000-10-31	3673	1526	1	1526	scanner canon lide 100 2400 x 4800 dpi 4ez buttons/48bit/usb hi-speed	54.5	2	
3673	7	2000-10-31	3673	1642	1	1642	teclado usb satellite ak-180 numerico	9.2	NULL	
3674	1	2000-10-31	3674	616	1	616	hub 4p. usb 2.0 a-hub01 satellite (preto)	4.7	1	
3674	1	2000-10-31	3674	1131	5	1131	nb acer as4540-1110 amd athlon ii x2 2.0/2gb/250/dvdrw/14 c/cam/linux	545	5	
3674	1	2000-10-31	3674	151	1	151	camera webcam dr hank vc-p073-cp1b 2.0 black	12	3	
3674	1	2000-10-31	3674	687	1	687	leitor card reader 2lxl markvision usb	4.75	NULL	
3674	1	2000-10-31	3674	240	1	240	car tocs cd p/carro roadstar sr-5260 mp3/bluetooth/anti shock	53.5	2	
3674	1	2000-10-31	3674	1731	1	1731	vga pciexp.lgb ddr3 gf gt240 550m 128bit dual dvi/hdmi xfx	112.5	2	
3674	1	2000-10-31	3674	697	3	697	m/f adsl/roteador d-link 500-b	29.2	NULL	
3674	1	2000-10-31	3674	570	1	570	hd nb sata ref 2.5 300 gb fujitsu 5400rpm	52.25	3	
3675	1	2000-10-31	3675	1057	4	1057	mp7-celular fujilink fu-91t tv/phone/mp3/mp4/4 bandas (prata)	58	2	
3675	1	2000-10-31	3675	401	1	401	gabinete p4 kit mtek mt7503 4baiais preto/ prata portugues	39.5	1	
3675	1	2000-10-31	3675	1496	2	1496	rede conversor d-link dmc-700sc w/mm 550m	58.5	NULL	
3675	1	2000-10-31	3675	1646	1	1646	teclado usb satellite multimedia ak-818 portugues (aluminio)	23	NULL	
3675	1	2000-10-31	3675	1049	1	1049	mp6-celular mox mo-698 4band lcd2.6 mp3/mp4 2 chip(vermelho)	73.5	2	
3675	1	2000-10-31	3675	990	5	990	mouse usb logitech v150 p/ notebook	33.2	NULL	
3675	1	2000-10-31	3675	1867	2	1867	wir. print server tp-link tl-wps510u usb 2.0 54mbps	45.5	NULL	
3675	1	2000-10-31	3675	601	1	601	hd sataii ref 1000 gb seagate 7200	79	3	

Passo 4: Filtrar apenas as vendas do produto no intervalo de interesse.

```
select * from venda v join item i on (i.venda = v.codigo) join produto p on (i.produto = p.codigo) where (cast(v.dia as char) like '2000-10-%') and (p.descricao = 'camera webcam genius eye 320');
```

codigo	vendedor	dia	venda	produto	quantidade	codigo	descricao	preco	comissao
3372	8	2000-10-06	3372	155	1	155	camera webcam genius eye 320	20.75	3
3486	9	2000-10-15	3486	155	5	155	camera webcam genius eye 320	20.75	3
3527	1	2000-10-19	3527	155	1	155	camera webcam genius eye 320	20.75	3
3541	9	2000-10-20	3541	155	1	155	camera webcam genius eye 320	20.75	3
3589	10	2000-10-24	3589	155	1	155	camera webcam genius eye 320	20.75	3

Passo 5: Calcular a quantidade de unidades vendidas do produto no intervalo de interesse.

```
select sum(i.quantidade) as unidades from venda v join item i on (i.venda = v.codigo) join produto p on (i.produto = p.codigo) where (cast(v.dia as char) like '2000-10-%') and (p.descricao = 'camera webcam genius eye 320');
```

unidades
9

2.1.54. Mais outra forma de contar a quantidade de vendas, por vendedor, em janeiro de 2000:

No exemplo 2.1.44, o select abaixo foi utilizado para resolver o problema.

```
select v.vendedor,count(*) as quantidade from venda v where (cast(v.dia as char) like '2000-01-%') group by v.vendedor order by v.vendedor asc;
```

vendedor	quantidade
1	41
2	38
3	43

	4		41	
	5		32	
	6		33	
	7		31	
	8		42	
	9		38	
	10		36	
+-----+				

Observe que a resposta contém os códigos dos vendedores, mas não seus nomes. Não é confortável para nós trabalhar com códigos. É sempre preferível utilizar nomes ou descrições. Como não existe uma coluna com o nome do vendedor na tabela venda, para uma resposta com os nomes dos vendedores ao invés de seus códigos, o `select` abaixo foi utilizado no exemplo 2.1.48.

```
select r.nome,(select count(*) from venda v where (v.vendedor = r.codigo) and (cast(v.dia as char) like '2000-01-%')) as quantidade from vendedor r;
```

+-----+	
nome	quantidade
+-----+	
Mauricio Costa Quaresma	41
Paulo Lopes Nunes	38
Patricia Menezes Silva	43
Cristiano Lopes Bezerra	41
Michele Menezes Santos	32
Cristiane Nunes Brandao	33
Pedro Rosa Loureiro	31
Jose Souza Martins	42
Gabriel Albuquerque Menezes	38
Mauricio Bezerra Lopes	36
+-----+	

Entretanto, existe ainda uma segunda forma de se produzir uma resposta com os nomes dos vendedores. Isto pode ser feito combinando as colunas da tabela venda e da tabela vendedor através de uma junção. O `select` abaixo faz isto para produzir a resposta desejada.

```
select r.nome,count(*) as quantidade from venda v join vendedor r on (v.vendedor = r.codigo) where (cast(v.dia as char) like '2000-01-%') group by r.nome order by r.nome asc;
```

+-----+	
nome	quantidade
+-----+	
Cristiane Nunes Brandao	33
Cristiano Lopes Bezerra	41
Gabriel Albuquerque Menezes	38
Jose Souza Martins	42
Mauricio Bezerra Lopes	36
Mauricio Costa Quaresma	41
Michele Menezes Santos	32
Patricia Menezes Silva	43
Paulo Lopes Nunes	38
Pedro Rosa Loureiro	31
+-----+	

Na realidade, assim como neste exemplo, vários dos itens anteriores podem ser reescritos utilizando junções. O mais importante a notar é que, se foi utilizado um filtro `where` sem junção, este filtro também será utilizado com a junção. E o mesmo vale para as funções de agregação (`min`, `max`, `count`, `sum` e `avg`) e também para `group by`, `having` e `order by`.

2.1.55. Listando as descrições dos produtos não vendidos em janeiro de 2000:

Para resolver este problema, precisamos utilizar, em um mesmo `select`, `join` e `in`. Acompanhe, passo-a-passo, a construção da resposta final.

Passo 1: Filtrar os códigos dos produtos vendidos em janeiro de 2000.

```
select distinct i.produto from venda v join item i on (i.venda = v.codigo) where (cast(v.dia as char) like '2000-01-%');
```

```
+-----+
| produto |
+-----+
|      9 |
|     130 |
|     213 |
|     304 |
|     402 |
|     1220 |
|     1776 |
|     1035 |
|     1052 |
|     1333 |
|      ... |
|     1830 |
|      478 |
|      647 |
|      793 |
|      703 |
|       71 |
|      249 |
|     1503 |
|     1569 |
|     1637 |
+-----+
```

Passo 2: Listar as descrições dos produtos não vendidos em janeiro de 2000.

```
select p.descricao from produto p where (p.codigo not in (select distinct i.produto from venda v join item i on (i.venda = v.codigo) where (cast(v.dia as char) like '2000-01-%')));
```

```
+-----+
| descricao |
+-----+
| acessorio kit p/ vga low profile |
| adaptador dvi-vga misc xfx |
| adaptador p/ carro de dc/ac aukson inversor 100watts 220 volts |
| adaptador satellite usb 2. 0 al-08 |
| antena parabolica 45cm c/lmb probbox |
| antena parabolica 60cm c/lmb bm |
| antena parabolica 75cm c/lmb prosat |
| ar cortina de ar hometiter fm-1209 - 220v 60hz 96cm |
| bateria p/ camera genius dv5700/dv610 np-60 (us-n) |
| bateria p/ camera recarr.mox mo-fe1 3.6v 450/mah p/ sony |
|
| wir. apresentador laser r800 logitech |
| wir. apresentador laser pointer infiniter lr3 |
| wir. mk caixa indoor case (ca/600 ) rb/600 |
| wir. mk mini pci cartao router board r52n a/b/g/n dual band |
| wir. mk placa router board rb/411 |
| wir. mk placa router board rb/433ah nivel5 |
| wir. mk placa router board rb/450 |
| wir. print server tp-link tl-wps510u usb 2.0 54mbps |
| wir. transmissor fm column ipda mc100 (pink) |
| wir. transmissor fm for iphone 3g- 0878 (silver) |
+-----+
```

2.1.56. Listando todos produtos com o valor da comissão correspondente:

A cada produto da tabela produto pode estar associada uma ou nenhuma comissão. Neste caso, comissão é um campo de preenchimento não obrigatório, que pode ser deixado vazio (nulo). E tem mais. Ao invés de utilizar diretamente o valor percentual da comissão na tabela produto, foi criada a tabela comissão, específica. Devido a esta decisão, os valores percentuais das comissões estão armazenados na tabela comissão e um código de comissão é utilizado na tabela produto para indicar, indiretamente, o valor da comissão percentual de um produto. O campo de ligação entre a tabela produto e a tabela comissão é o código da comissão, que se chama comissão na tabela produto e código na tabela comissão.

Com base na explicação acima e no que foi discutido sobre junções internas até agora, uma solução seria:

```
select * from produto p join comissao c on (p.comissao = c.codigo);
```

codigo	descricao	preco	comissao	codigo	comissao
46	bolsa p/notebook 15.4 case logic snm-15 gray	34	1	1	10
47	bolsa p/notebook 15.4 case logic snm-15f gray	34	1	1	10
70	calculadora hp 2220 office calc 100	10.6	2	2	5
71	calculadora hp 2221 office basic calc 200	18.5	2	2	5
72	calculadora hp financeira 10 bii - manual em espanhol	26	2	2	5
73	calculadora hp financeira 12c gold - manual portugues	61.5	2	2	5
74	calculadora hp financeira 17bii - manual em portugues e espanhol	79.2	2	2	5
75	calculadora hp grafica 50g (manual portugues)	128	2	2	5
11	antena parabolica 45cm c/lnb probbox	19	3	3	3
12	antena parabolica 55cm c/lnb probbox	22	3	3	3
...					
1743	vga pciexp.lgb ddr5 hd5770 850m 128bit ati hdmi/dual dvi/hdvt xfx	205.5	2	2	5
1744	vga pciexp.lgb ddr5 hd5850 725m 256bit ati tv/dual dvi/hdvt/hdmi xfx	315	2	2	5
1745	vga pciexp.lgb ddr5 hd5850 755m ati tv/dual dvi/hdvt/hdmi xfx	340	2	2	5
1746	vga pciexp.lgb ddr5 hd5850 765m 256bit ati tv/dual dvi/hdvt/hdmi black xfx	357	2	2	5
1747	vga pciexp.lgb ddr5 hd5870 850m 256bit ati tv/dual dvi/hdvt xfx	445	2	2	5
1748	vga pciexp.lgb ddr5 hd5870 875m 256bit ati cypress xt/dual dvi/hdmi xfx	466	2	2	5
1749	vga pciexp.2gb ddr3 gf gts250 738m dual dvi/hdvt ecs	220	2	2	5
1819	wir. ap d-link dwl-2100ap 108mbps	63.5	5	5	1
1820	wir. ap d-link dwl-7100ap a/b/g 108mbps	120	5	5	1
1821	wir. ap linksys wap2000 54mbps	178	5	5	1

Observe atentamente a resposta. Foram listados apenas os produtos que possuem comissão. Onde estão os produtos que não possuem comissão? Onde estão os produtos cujo código de comissão é nulo?

Por definição, a junção interna ignora quaisquer linhas do produto cartesiano que não satisfaçam a condição de junção, evitando, corretamente, quaisquer associações indevidas. Neste exemplo, isto ocorre quando o código da comissão no produto é diferente do código da comissão na comissão. Mas e quando o valor do código da comissão de um produto for nulo? Na junção interna, estas linhas também são ignoradas. Por isto, nenhum produto sem comissão aparece na resposta anterior.

Para permitir que valores nulos sejam admitidos na resposta de uma junção, devemos utilizar outro tipo de junção, a junção externa (`outer join`).

Os tipos de junção externa são à esquerda (`left`), à direita (`right`) e completa (`full`). Sendo a junção externa completa equivalente a soma da resposta de uma junção à esquerda com a de uma junção à direita.

Uma junção externa à esquerda funciona assim. Primeiro, é realizada uma junção interna. Depois, são incluídas na resposta todas as linhas da tabela à esquerda do `join` cujos campos presentes na condição de junção possuem valor nulo. O mesmo ocorre em uma junção externa à direita, mas para os campos da tabela à direita do `join`.

Vamos ver isto funcionando na prática. Para listar todos produtos com o valor da comissão correspondente, precisam também ser incluídos na resposta as linhas da tabela produto cujos códigos de comissão sejam nulo. Como o `select` com junção interna foi escrito com a tabela produto à esquerda da palavra `join`, basta transformar a junção interna em uma junção externa à esquerda acrescentando a palavra `left`.

```
select * from produto p left join comissao c on (p.comissao = c.codigo);
```

codigo	descricao	preco	comissao	codigo	comissao
1	acessorio kit p/ vga low profile	4.2	NULL	NULL	NULL
2	adaptador dvi-vga misc xfx	2.53	NULL	NULL	NULL
3	adaptador p/ carro de dc/ac aukson inversor 100watts 220 volts	20.5	NULL	NULL	NULL
4	adaptador p/ipod klipx icc-050 car charger	7.6	NULL	NULL	NULL
5	adaptador satellite usb 2.0 al-09	1.7	NULL	NULL	NULL
6	adaptador satellite usb 2.0 al-10	1.8	NULL	NULL	NULL
7	adaptador data switch non power 2 in 1 port paralelo ap201n	15.5	NULL	NULL	NULL
8	adaptador p/ impressao tel100-plu trendnet usb em uso de rede	63.25	NULL	NULL	NULL
9	adaptador p/ipod yamaha yds10 dock	107	NULL	NULL	NULL
10	adaptador satellite usb 2.0 al-08	1.6	NULL	NULL	NULL
11	antena parabolica 45cm c/lnb probbox	19	3	3	3
12	antena parabolica 55cm c/lnb probbox	22	3	3	3
13	antena parabolica 60cm c/lnb bm	24.5	3	3	3
14	antena parabolica 60cm c/lnb shop+	29	3	3	3
15	antena parabolica 60cm s/lnb ecopower ku60	18	3	3	3
16	antena parabolica 75cm c/lnb prosat	31.5	3	3	3

17		antena parabólica 80cm c/lnb evolution dish/shop		34.5		3		3		3	
18		antena parabólica 80cm s/lnb shop+		27.5		3		3		3	
19		ar multi split hometiter 12000+9000f/c 220v 50 hz		869		5		5		1	
20		ar multi split hometiter 2x12000f/c 220v 50 hz		922.9		5		5		1	
21		ar multi split hometiter 2x9000f/c 220v 50 hz		862.4		5		5		1	
22		ar multi split hometiter 2x9000f/c 220v 60 hz br		862.4		5		5		1	
23		ar split hometiter 18000f/c 220v 50 hz		449		5		5		1	
24		ar cortina de ar hometiter fm-1209 - 220v 60hz 96cm		103.5		5		5		1	
...											
1864		wir. nanostation2 ubiquiti 2.4ghz 10dbi ns2		76.5		NULL		NULL		NULL	
1865		wir. nanostation5 ubiquiti 5ghz 14dbi ns5		96		NULL		NULL		NULL	
1866		wir. pcmcia rede d-link dwa-645 n 300mbps		55		NULL		NULL		NULL	
1867		wir. print server tp-link tl-wps510u usb 2.0 54mbps		45.5		NULL		NULL		NULL	
1868		wir. transmissor fm column ipda mc100 (pink)		11.5		NULL		NULL		NULL	
1869		wir. transmissor fm for iphone 3g- 0878 (silver)		15.75		NULL		NULL		NULL	
1870		wir. transmissor fm orange or-668 (gray/white)		9.2		NULL		NULL		NULL	
1871		wir. transmissor logitech mp3 p/ entrada auxiliar de som		26.5		NULL		NULL		NULL	
1872		wireless antena hyperlink 20dbi 090 setorial hg2420p-090		312		NULL		NULL		NULL	
1873		wireless antena hyperlink 17dbi 120 setorial hg2417p-120		154		NULL		NULL		NULL	

Observe que, agora, a resposta lista todos os produtos e possui 1873 linhas, ao invés das 1201 linhas da resposta com junção interna.

Para simplificar a resposta, vamos recuperar apenas as colunas descrição e preço do produto e comissão da comissão. Repare que a coluna comissão utilizada é o valor da comissão percentual e não o código da comissão.

```
select p.descricao,p.preco,c.comissao from produto p left join comissao c on (p.comissao = c.codigo);
```

descricao	preco	comissao
acessorio kit p/ vga low profile	4.2	NULL
adaptador dvi-vga misc xfx	2.53	NULL
adaptador p/ carro de dc/ac aukson inversor 100watts 220 volts	20.5	NULL
adaptador p/ipod klipx icc-050 car charger	7.6	NULL
adaptador satellite usb 2. 0 al-09	1.7	NULL
adaptador satellite usb 2. 0 al-10	1.8	NULL
adaptador data switch non power 2 in 1 port paralelo ap201n	15.5	NULL
adaptador p/ impressao tel00-plu trendnet usb em uso de rede	63.25	NULL
adaptador p/ipod yamaha yds10 dock	107	NULL
adaptador satellite usb 2. 0 al-08	1.6	NULL
antena parabólica 45cm c/lnb probbox	19	3
antena parabólica 55cm c/lnb probbox	22	3
antena parabólica 60cm c/lnb bm	24.5	3
antena parabólica 60cm c/lnb shop+	29	3
antena parabólica 60cm s/lnb ecopower ku60	18	3
antena parabólica 75cm c/lnb prosat	31.5	3
antena parabólica 80cm c/lnb evolution dish/shop	34.5	3
antena parabólica 80cm s/lnb shop+	27.5	3
ar multi split hometiter 12000+9000f/c 220v 50 hz	869	1
ar multi split hometiter 2x12000f/c 220v 50 hz	922.9	1
ar multi split hometiter 2x9000f/c 220v 50 hz	862.4	1
ar multi split hometiter 2x9000f/c 220v 60 hz br	862.4	1
ar split hometiter 18000f/c 220v 50 hz	449	1
ar cortina de ar hometiter fm-1209 - 220v 60hz 96cm	103.5	1
...		
wir. nanostation2 ubiquiti 2.4ghz 10dbi ns2	76.5	NULL
wir. nanostation5 ubiquiti 5ghz 14dbi ns5	96	NULL
wir. pcmcia rede d-link dwa-645 n 300mbps	55	NULL
wir. print server tp-link tl-wps510u usb 2.0 54mbps	45.5	NULL
wir. transmissor fm column ipda mc100 (pink)	11.5	NULL
wir. transmissor fm for iphone 3g- 0878 (silver)	15.75	NULL
wir. transmissor fm orange or-668 (gray/white)	9.2	NULL
wir. transmissor logitech mp3 p/ entrada auxiliar de som	26.5	NULL
wireless antena hyperlink 20dbi 090 setorial hg2420p-090	312	NULL
wireless antena hyperlink 17dbi 120 setorial hg2417p-120	154	NULL

Mas ainda não obtemos a resposta desejada. Como fazer para mostrar o valor 0 quando um produto não pagar comissão? Isto pode ser feito com um `case`.

```
select p.descricao,p.preco,case when (c.comissao is null) then 0 else c.comissao end as comissao from
produto p left join comissao c on (p.comissao = c.codigo);
```

descricao	preco	comissao
acessorio kit p/ vga low profile	4.2	0
adaptador dvi-vga misc xfx	2.53	0
adaptador p/ carro de dc/ac aukson inversor 100watts 220 volts	20.5	0
adaptador p/ipod klipx icc-050 car charger	7.6	0
adaptador satelite usb 2. 0 al-09	1.7	0
adaptador satelite usb 2. 0 al-10	1.8	0
adaptador data switch non power 2 in 1 port paralelo ap201n	15.5	0
adaptador p/ impressao tel100-plu trendnet usb em uso de rede	63.25	0
adaptador p/ipod yamaha yds10 dock	107	0
adaptador satelite usb 2. 0 al-08	1.6	0
antena parabolica 45cm c/lnb probbox	19	3
antena parabolica 55cm c/lnb probbox	22	3
antena parabolica 60cm c/lnb bm	24.5	3
antena parabolica 60cm c/lnb shop+	29	3
antena parabolica 60cm s/lnb ecopower ku60	18	3
antena parabolica 75cm c/lnb prosat	31.5	3
antena parabolica 80cm c/lnb evolution dish/shop	34.5	3
antena parabolica 80cm s/lnb shop+	27.5	3
ar multi split hometiter 12000+9000f/c 220v 50 hz	869	1
ar multi split hometiter 2x12000f/c 220v 50 hz	922.9	1
ar multi split hometiter 2x9000f/c 220v 50 hz	862.4	1
ar multi split hometiter 2x9000f/c 220v 60 hz br	862.4	1
ar split hometiter 18000f/c 220v 50 hz	449	1
ar cortina de ar hometiter fm-1209 - 220v 60hz 96cm	103.5	1
...		
wir. nanostation2 ubiquiti 2.4ghz 10dbi ns2	76.5	0
wir. nanostation5 ubiquiti 5ghz 14dbi ns5	96	0
wir. pcmcia rede d-link dwa-645 n 300mbps	55	0
wir. print server tp-link tl-wps510u usb 2.0 54mbps	45.5	0
wir. transmissor fm column ipda mcl100 (pink)	11.5	0
wir. transmissor fm for iphone 3g- 0878 (silver)	15.75	0
wir. transmissor fm orange or-668 (gray/white)	9.2	0
wir. transmissor logitech mp3 p/ entrada auxiliar de som	26.5	0
wireless antena hyperlink 20dbi 090 setorial hg2420p-090	312	0
wireless antena hyperlink 17dbi 120 setorial hg2417p-120	154	0

Pronto. Agora obtemos a resposta desejada.

2.1.57. Contando a quantidade de produtos, por comissão:

O `select`, sem junção, abaixo resolve o problema.

```
select p.comissao,count(*) as quantidade from produto p group by p.comissao;
```

comissao	quantidade
NULL	672
1	129
2	351
3	443
4	133
5	145

Repare que, para os produtos que não pagam comissão (código da comissão é nulo), o resultado da contagem é 672, mas que não aparece nenhum valor de comissão associado.

Para tornar a resposta mais legível, podemos utilizar, como ponto de partida, o `select` com `case` e `left join` do exemplo acima. Partindo dele, podemos pensar em utilizar `group by` para agrupar a resposta pelo valor da comissão e em seguida utilizar a função de contagem em cada grupo. O `select` que produziria a resposta desejada seria:

```
select case when (c.comissao is null) then 0 else c.comissao end as comissao,count(*) as quantidade from
produto p left join comissao c on (p.comissao = c.codigo) group by comissao;
```

Nas versões anteriores do MySQL, não podíamos utilizar o apelido de uma coluna para fazer o `group by`. O MySQL acusava erro ao se tentar executar o `select` acima. Uma forma de contornar o problema era transformar a resposta do `select` do exemplo acima em uma tabela intermediária temporária e aplicar o `group by` nela, conforme mostrado abaixo. Atualmente o comando acima é suportado.

```
select t.comissao,count(*) as quantidade from (select p.descricao,p.preco,case when (c.comissao is null)
then 0 else c.comissao end as comissao from produto p left join comissao c on (p.comissao = c.codigo)) as t
group by t.comissao order by t.comissao asc;
```

comissao	quantidade
0	672
1	145
2	133
3	443
5	351
10	129

Nesta última parte, vamos explorar ainda mais as potencialidades do `select`, utilizando todas as partes da figura 4 com várias tabelas ao mesmo tempo.

2.1.58. Listando todos os processadores de dois núcleos:

Assim como existe uma maneira para acrescentar colunas de tabelas diferentes em uma resposta (`join`), existe também uma maneira para acrescentar linhas de tabelas diferentes em uma resposta. Isto pode ser feito com uniões (`union`).

Como os dados de todos os processadores de dois núcleos estão na mesma tabela (`produto`), o `select` simples, abaixo, pode ser utilizado para obter a resposta desejada.

```
select * from produto p where (p.descricao like 'proc%amd%x2%') or (p.descricao like 'proc%intel%duo%');
```

codigo	descricao	preco	comissao
1429	proc. amd am2 athlon64 x2 5200 1m/2.7ghz box	72	3
1431	proc. amd am3 athlon ii x2 240/2.8 ghz - 2 mb	65.5	3
1432	proc. amd am3 athlon ii x2 245/2.9 ghz - 2 mb 65w	69.5	3
1433	proc. amd am3 athlon ii x2 250/3.0 ghz - 2 mb 65w	78.5	3
1434	proc. amd am3 athlon ii x2 545/3.0 ghz - 7 mb 85w	98	3
1448	proc. intel 775p core 2 duo e7400 2.80ghz 3mb/1066mhz box	122.5	3
1449	proc. intel 775p core 2 duo e7500 2.93ghz 3mb/1066mhz box	130.5	3
1450	proc. intel 775p core 2 duo e8400 3.0ghz/ 6mb/1333mhz box	182	3
1451	proc. intel 775p core 2 duo e8500 3.16ghz/ 6mb/1333mhz box	203	3
1452	proc. intel 775p core 2 duo e8600 3.33ghz/ 6mb/1333mhz box	293	3

Entretanto, também pode ser utilizado uma união para produzir a mesma resposta.

```
(select p.descricao from produto p where (p.descricao like 'proc%amd%x2%'))
union
(select p.descricao from produto p where (p.descricao like 'proc%intel%duo%'));
```

descricao
proc. amd am2 athlon64 x2 5200 1m/2.7ghz box
proc. amd am3 athlon ii x2 240/2.8 ghz - 2 mb

```

| proc. amd am3 athlon ii x2 245/2.9 ghz - 2 mb 65w |
| proc. amd am3 athlon ii x2 250/3.0 ghz - 2 mb 65w |
| proc. amd am3 athlon ii x2 545/3.0 ghz - 7 mb 85w |
| proc. intel 775p core 2 duo e7400 2.80ghz 3mb/1066mhz box |
| proc. intel 775p core 2 duo e7500 2.93ghz 3mb/1066mhz box |
| proc. intel 775p core 2 duo e8400 3.0ghz/ 6mb/1333mhz box |
| proc. intel 775p core 2 duo e8500 3.16ghz/ 6mb/1333mhz box |
| proc. intel 775p core 2 duo e8600 3.33ghz/ 6mb/1333mhz box |
+-----+

```

Observe que foram utilizados dois `select` independentes. O primeiro filtra apenas os processadores da linha AMD, enquanto que o segundo, filtra apenas processadores da linha Intel. Como já foi dito, a união é utilizada quando se deseja acrescentar linhas, geralmente de tabelas diferentes, em uma resposta. Neste caso, a resposta contém as linhas do primeiro `select` e, em seguida, as linhas do segundo.

Embora a utilização de uniões pareça fácil, duas observações importantes devem ser feitas para o seu correto uso. Primeiro, dificilmente uma união será utilizada com `select *`. Apenas algumas colunas devem ser recuperadas. Segundo, devemos ter muito cuidado com o casamento de tipos entre as colunas de cada `select`. O tipo da primeira coluna do primeiro `select` deve ser o mesmo da primeira coluna do segundo `select`. O tipo da segunda coluna do primeiro `select` deve ser o mesmo da segunda coluna do segundo `select`. E assim por diante. Não se pode misturar os tipos.

Para um outro exemplo utilizando várias tabelas, imagine um hospital, com tabelas específicas para armazenar os dados de médicos, enfermeiros, funcionários e pacientes. Imagine também que fosse necessário gerar uma única lista contendo todas as pessoas cujo primeiro nome é Pedro. Apenas o `select` abaixo produziria a resposta desejada.

```

(select m.nome from medico m where (m.nome like 'pedro%'))
union
(select e.nome from enfermeiro e where (e.nome like 'pedro%'))
union
(select f.nome from funcionario f where (f.nome like 'pedro%'))
union
(select p.nome from paciente p where (p.nome like 'pedro%'));

```

2.1.59. Calcular o valor total da venda cujo código é 40:

Primeiro, devemos lembrar que o valor total de uma venda é informação, e não dado. E que apenas dados devem ser armazenados. Observe que em nenhuma tabela do banco loja existe um campo para o valor total de cada venda. Portanto, o valor total de cada venda deve ser calculado com base nos dados desta venda. Vamos fazer isto então.

Primeiro, vamos listar todos os dados necessários para se calcular o valor total relacionados à venda de código 40. Precisaremos do preço unitário de cada produto e da quantidade de unidades vendidas em cada item.

```

select *
from venda v
  join item i on (i.venda = v.codigo)
  join produto p on (i.produto = p.codigo)
where (v.codigo = 40);

```

	codigo	vendedor	dia	venda	produto	quantidade	codigo	descricao	preco	comissao
	40	3	2000-01-03	40	58	1	58	cabo de vga 15 pinos p/monitor/projetores - 10 metros	20.5	NULL
	40	3	2000-01-03	40	82	2	82	camera foto olympus tough-6000 10 mega pixel green/prova dagua	247	3
	40	3	2000-01-03	40	372	1	372	fonte gabinete atx p4 asvotek av-600 24 pin 400w real	32	1
	40	3	2000-01-03	40	782	5	782	mb p4 s775 ecs g3lt-m scm/vga/lan/sata/ddr2/core - proc. quad core	46.6	3
	40	3	2000-01-03	40	956	4	956	mouse ps/2 usb satellite al3 opt.azul mini	7	NULL
	40	3	2000-01-03	40	1372	1	1372	pilha recarr.mo-aa4800b2 c/2aa 4800mah mox	1.7	NULL
	40	3	2000-01-03	40	1468	3	1468	projektor kaiomy 10 lumens asteria 10(mini-projetor)	179	4
	40	3	2000-01-03	40	1550	1	1550	speaker satellite usb as-667u (preto)	6.2	NULL
	40	3	2000-01-03	40	1592	1	1592	speaker thonet vander 112-110 spiel 110v preto	28.5	NULL
	40	3	2000-01-03	40	1743	1	1743	vga pciexp.1gb ddr5 hd5770 850m 128bit ati hdmi/dual dvi/hdtv xfx	205.5	2

Para calcular o subtotal de cada produto basta multiplicar o preço unitário do produto pela quantidade de unidades vendidas em cada item. O `select` abaixo faz isto.

```
select i.produto,p.descricao,i.quantidade,p.preco,round(p.preco*i.quantidade,2) as subtotal
from venda v
      join item i on (i.venda = v.codigo)
      join produto p on (i.produto = p.codigo)
where (v.codigo = 40);
```

produto	descricao	quantidade	preco	subtotal
58	cabo de vga 15 pinos p/monitor/projetores - 10 metros	1	20.5	20.50
82	camera foto olympus tough-6000 10 mega pixel green/prova agua	2	247	494.00
372	fonte gabinete atx p4 asvotek av-600 24 pin 400w real	1	32	32.00
782	mb p4 s775 ecs g31t-m som/vga/lan/sata/ddr2/core - proc. quad core	5	46.6	233.00
956	mouse ps/2 usb satellite a13 opt.azul mini	4	7	28.00
1372	pilha recarr.mo-aa4800b2 c/2aa 4800mah mox	1	1.7	1.70
1468	projektor kaiomy 10 lumens asteria 10(mini-projetor)	3	179	537.00
1550	speaker satellite usb as-667u (preto)	1	6.2	6.20
1592	speaker thonet vander 112-110 spiel 110v preto	1	28.5	28.50
1743	vga pciexp.1gb ddr5 hd5770 850m 128bit ati hdmi/dual dvi/hdtv xfx	1	205.5	205.50

Para obter o valor total agora, basta somar todos os subtotais:

```
select v.codigo,round(sum(p.preco*i.quantidade),2) as total
from venda v
      join item i on (i.venda = v.codigo)
      join produto p on (i.produto = p.codigo)
where (v.codigo = 40)
group by v.codigo;
```

codigo	total
40	1586.40

Uma observação muito importante cabe neste momento. Embora o valor total da venda de código 40 também possa ser calculado via código, isto gastaria muito mais tempo, muito mais memória e produziria muito mais transferências de dados entre o SGBD e a aplicação. Lembrem que nem sempre o SGBD e a aplicação estão hospedados no mesmo computador.

```
inicialize o total t com zero
selecione todos os itens da tabela item cujo código da venda seja 40 e armazene no vetor vi
para cada elemento i do vetor vi faça
    selecione o produto da tabela produto cujo código seja o mesmo do produto no elemento i e armazene em p
    calcule o subtotal s
    some o subtotal s ao total t
fim para
mostre o total t
```

É realmente importante entender que utilizar um `select`, por mais complexo que seja, é sempre mais vantajoso e eficiente do que utilizar um código com vários `para-faça` e vários `select` mais simples.

2.1.60. Calcular o valor total de cada venda em janeiro de 2000:

Utilizando o mesmo raciocínio do exemplo 2.1.59, podemos calcular o valor total de todas as vendas de janeiro de 2000. O `select` abaixo produz a resposta desejada.

```
select v.codigo,round(sum(p.preco*i.quantidade),2) as total
from venda v
      join item i on (i.venda = v.codigo)
      join produto p on (i.produto = p.codigo)
where (cast(v.dia as char) like '2000-01-%')
group by v.codigo
order by v.codigo asc;
```

codigo	total
1	1478.00
2	643.55
3	1715.60
4	967.67
5	1399.41
6	727.60
7	42.65
8	1065.90
9	804.53
10	9.50
...	
366	987.10
367	10.60
368	2808.40
369	882.00
370	22.55
371	706.27
372	3732.92
373	1448.56
374	12.43
375	4163.25

2.1.61. Calculando o valor da comissão do vendedor Paulo Lopes Nunes em janeiro de 2000:

O raciocínio necessário para calcular quanto o vendedor Paulo Lopes Nunes deve receber de comissão em janeiro de 2000 é muito parecido com o para calcular o valor total de cada venda no mesmo período. A única diferença é que também precisaremos considerar o valor da comissão que cada produto paga. Como a comissão é um número de 0 a 100 que representa um valor percentual, é necessário dividir o valor da comissão por 100. O `select` abaixo produz a resposta desejada.

```
select r.codigo,r.nome,round(sum(p.preco*i.quantidade*c.comissao/100),2) as comissao
from venda v
      join item i on (i.venda = v.codigo)
      join produto p on (i.produto = p.codigo)
      join comissao c on (p.comissao = c.codigo)
      join vendedor r on (v.vendedor = r.codigo)
where (cast(v.dia as char) like '2000-01-%') and (r.nome = 'Paulo Lopes Nunes')
group by r.codigo;
```

codigo	nome	comissao
2	Paulo Lopes Nunes	736.69

2.1.62. Calculando o valor das comissões em janeiro de 2000, por vendedor:

Sabendo como calcular quanto um determinado vendedor vai receber de comissão em um mês, basta repetir o processo sem determinar um vendedor para produzir a resposta desejada.

```
select r.codigo,r.nome,round(sum(p.preco*i.quantidade*c.comissao/100),2) as comissao
from venda v
      join item i on (i.venda = v.codigo)
      join produto p on (i.produto = p.codigo)
      join comissao c on (p.comissao = c.codigo)
      join vendedor r on (v.vendedor = r.codigo)
where (cast(v.dia as char) like '2000-01-%')
group by r.codigo
order by r.codigo asc;
```

codigo	nome	comissao
1	Mauricio Costa Quaresma	850.08
2	Paulo Lopes Nunes	736.69
3	Patricia Menezes Silva	935.68
4	Cristiano Lopes Bezerra	903.14
5	Michele Menezes Santos	713.14
6	Cristiane Nunes Brandao	569.81
7	Pedro Rosa Loureiro	657.77
8	Jose Souza Martins	767.65
9	Gabriel Albuquerque Menezes	1144.85
10	Mauricio Bezerra Lopes	814.97

2.1.63. Calculando o salário final de cada vendedor em janeiro de 2000:

Para finalmente calcular o salário final de cada vendedor em um mês, basta somar à comissão calculada o valor do salário fixo de cada vendedor. O `select` abaixo faz isto.

```
select r.codigo,r.nome,round(r.salario,2) as salario,
      round(sum(p.preco*i.quantidade*c.comissao/100),2) as comissao,
      round(r.salario+sum(p.preco*i.quantidade*c.comissao/100),2) as salario_final
from venda v
      join item i on i.venda = v.codigo
      join produto p on i.produto = p.codigo
      join comissao c on p.comissao = c.codigo
      join vendedor r on v.vendedor = r.codigo
where (cast(v.dia as char) like '2000-01-%')
group by r.codigo,r.salario
order by r.codigo asc;
```

codigo	nome	salario	comissao	salario_final
1	Mauricio Costa Quaresma	500.00	850.08	1350.08
2	Paulo Lopes Nunes	500.00	736.69	1236.69
3	Patricia Menezes Silva	500.00	935.68	1435.68
4	Cristiano Lopes Bezerra	550.00	903.14	1453.14
5	Michele Menezes Santos	550.00	713.14	1263.14
6	Cristiane Nunes Brandao	500.00	569.81	1069.81
7	Pedro Rosa Loureiro	850.00	657.77	1507.77
8	Jose Souza Martins	550.00	767.65	1317.65
9	Gabriel Albuquerque Menezes	500.00	1144.85	1644.85
10	Mauricio Bezerra Lopes	550.00	814.97	1364.97

Com este exemplo, acabamos de explorar as potencialidades do `select`.

2.2. CREATE TABLE

É a instrução de criação de tabelas. Sua sintaxe mais usual é:

```
CREATE TABLE nome_tabela (
    nome_coluna tipo_dado [ [ NOT ] NULL ] [ DEFAULT expressão_padrão ] [ CHECK ( expressão ) ]
    [ UNIQUE ] [ AUTO_INCREMENT ] [ REFERENCES nome_tabela_referência ( nome_coluna_referência ) ]
    [, ...]
    [ PRIMARY KEY ( nome_coluna [, ...] ) ]
    [ FOREIGN KEY ( nome_coluna [, ...] ) REFERENCES nome_tabela_referência ( nome_coluna_referência [, ...]
) ]
)
legenda:
[ ]   opcional
{ | } seleção
...   repetição
```

Figura 6 - Sintaxe do create table

2.2.1. Criando as tabelas do banco loja:

```
create table comissao (
    codigo integer not null auto_increment,
    comissao float not null check ((comissao >= 0) and (comissao <= 100)),
    primary key (codigo)
);

create table vendedor (
    codigo integer not null auto_increment,
    nome varchar(50) not null,
    salario float not null check (salario > 0),
    primary key (codigo)
);

create table produto (
    codigo integer not null auto_increment,
    descricao varchar(200) not null,
    preco float not null check (preco > 0),
    comissao integer references comissao(codigo),
    primary key (codigo)
);

create table venda (
    codigo integer not null auto_increment,
    vendedor integer not null references vendedor(codigo),
    dia date not null default current_date(),
    primary key (codigo)
);

create table item (
    venda integer not null references venda(codigo),
    produto integer not null references produto(codigo),
    quantidade integer not null check (quantidade > 0) default 1,
    primary key (venda,produto)
);
```

Vamos analisar as instruções. Primeiro, à cada tabela é associada um nome único. Cada tabela pode ter um ou mais campos. Cada campo é de um e somente um tipo de dado. Campos do tipo `integer` podem ser auto incrementados. Cada campo pode ser de preenchimento obrigatório (`not null`) ou não. Cada campo pode possuir valor padrão (`default`) ou não. Cada campo pode possuir validação (`check`) ou não. Cada campo pode ou não fazer referência (`references`) a um campo correlato de uma outra tabela. Cada tabela deve possuir uma chave primária formada por um ou mais de seus campos.

Vamos então à explicação de cada uma das partes da instrução `create table`. Começamos pelos tipos de dados. A tabela 4 mostra alguns tipos de dados suportados pelo MySQL:

Tipo	Descrição
BOOLEAN	true, false
TINYINT	-128 até 127
SMALLINT	-32768 até 32767
INT	-2147483648 até 2147483647
BIGINT	-9223372036854775808 até 9223372036854775807
FLOAT	-3.402823466E+38 até 3.402823466E+38
DOUBLE	-2.2250738585072014E-308 até 2.2250738585072014E-308
TIME	formato 'HH:MM:SS'
DATE	formato 'AAAA-MM-DD'
TIMESTAMP	formato 'AAAA-MM-DD HH:MM:SS'
CHAR (M)	Literal de tamanho fixo, com tamanho máximo M, com espaços à direita.
VARCHAR (M)	Literal de tamanho variável, com tamanho máximo M.
TEXT	Literal de tamanho 65535.

Tabela 4 - Tipos de dados

O `not null` deve ser utilizado para explicitar que um campo é de preenchimento obrigatório. Observe que a grande maioria dos campos do banco `loja` é de preenchimento obrigatório. Apenas o campo comissão na tabela `produto` não é de preenchimento obrigatório, uma vez que um produto pode não pagar comissão em venda.

O `auto_increment` é geralmente utilizado para codificar, no banco de dados, objetos do mundo real que não possuem algum tipo de codificação própria, como código de barras, associados. Quando um campo do tipo inteiro é definido como auto incrementado, na próxima instrução de inclusão de dados o MySQL automaticamente calcula o próximo valor somando um ao último valor incluído.

O `check` deve ser utilizado para validar os valores permitidos em um campo. Alguns exemplos. Se o campo representa o salário de um funcionário, não podem existir salários negativos. Se o campo representa o sexo de uma pessoa, os únicos valores aceitos podem ser 'M' para homens e 'F' para mulheres. Se o campo representa um valor percentual, não podem existir valores menores do que 0 e maiores do que 100.

O `default` deve ser utilizado para atribuir valores aos campos para os quais não foram explicitamente fornecidos valores em uma instrução de inclusão de dados. No banco `loja`, se nenhuma data for explicitamente fornecida na inclusão de uma venda, o valor padrão `current_date()` será utilizado. O mesmo vale para a quantidade de unidades vendidas de um produto em uma venda, na tabela `item`.

O `references` deve ser utilizado para criar uma dependência, ou vínculo, entre campos correlatos de duas tabelas diferentes. Campos de tabelas diferentes são correlatos quando representam, ou significam, a mesma coisa. Na prática, quando uma referência é estabelecida, não podem existir, no campo da tabela referenciadora, valores que não existam no campo da tabela referenciada. O campo da tabela referenciadora também é conhecido como chave estrangeira. Observe o campo vendedor da tabela `venda`, que referencia o campo código da tabela `vendedor`. Apesar de não possuírem o mesmo nome, os campos são correlatos pois significam a mesma coisa, um código de vendedor. Na prática, esta dependência significa que uma venda

só pode existir se o valor do seu campo vendedor for um dos valores do campo código na tabela vendedor ou, de outra forma, à cada venda deve estar associado um vendedor.

A chave primária, ou `primary key`, serve para identificar, de forma única, uma e somente uma linha de uma tabela. Em outras palavras, isto significa que não podem existir duas linhas com a mesma chave primária em uma tabela. E mais. Cada tabela deve possuir uma, e somente uma, chave primária. Excepcionalmente, ocorre uma tabela sem chave primária. Além disto, na grande maioria das vezes, as entidades do problema sendo tratado já possuem um sistema único de identificação, ou código, que pode ser utilizado como chave primária. No caso dos vendedores, este código poderia ser o CPF. Como todos os vendedores possuem um e somente um CPF e um CPF identifica um e apenas um vendedor, por consequência, não podem existir dois vendedores com o mesmo CPF.

Por último, é importante perceber que existe uma ordem correta para a criação das tabelas de um banco de dados. Primeiro, devem ser criadas as tabelas independentes, que não referenciam campos em outras tabelas. Somente depois é que devem ser criadas as tabelas dependentes, que referenciam um ou mais campos em outras tabelas. Uma tabela dependente, com chaves estrangeiras, só pode ser criada após todas as outras tabelas independentes terem sido criadas. Para ficar mais claro, observe a ordem de criação das tabelas do banco `loja`. As tabelas independentes são comissão e vendedor. As tabelas dependentes são produto, que depende de comissão; venda, que depende de vendedor e item, que depende de venda e produto.

2.3. INSERT

É a instrução de inserção de dados. Sua sintaxe mais usual é:

```
INSERT INTO nome_tabela ( nome_coluna [, ...] ) VALUES ( expressão [, ...] )
legenda:
[ ]   opcional
{ | } seleção
...   repetição
```

Figura 7 - Sintaxe do `insert`

2.3.1. Incluindo um novo vendedor:

Imagine a contratação de um novo vendedor chamado Pedro Costa Rocha, com salário inicial de \$500,00. O `insert` que cadastra seus dados na tabela vendedor é:

```
insert into vendedor (nome,salario) values ('Pedro Costa Rocha',500.00);
```

Se o campo código da tabela vendedor não fosse auto incrementado e o código disponível fosse 12345, o `insert` correto seria:

```
insert into vendedor (codigo,nome,salario) values (12345,'Pedro Costa Rocha',500.00);
```

Observe que campos auto incrementados não devem aparecer na lista de nomes de campos nem na lista de valores correspondentes.

2.3.2. Incluindo um novo produto:

Imagine o início da venda do produto mem ddr3 2gb 1066mhz kingston, com preço de \$55,00. Para inclusão deste novo produto, duas são as situações possíveis, dependendo se o produto pagará ou não comissão. No caso de o produto não pagar comissão, deve ser utilizado o insert abaixo:

```
insert into produto (descricao,preco) values ('mem ddr3 2gb 1066mhz kingston',55.00);
```

Já no caso do produto pagar comissão, primeiro deve ser feita uma consulta à tabela comissão para descobrir qual o código correspondente à comissão que o produto vai pagar. Considerando que este novo produto pague 5% de comissão, o código da comissão correspondente é 2. Sabendo isto, neste caso, o insert correto é:

```
insert into produto (descricao,preco,comissao) values ('mem ddr3 2gb 1066mhz kingston',55.00,2);
```

Observe que, nos dois casos, o campo código do produto não aparece no insert, já que é auto incrementado.

2.3.3. Incluindo uma nova venda:

Imagine que a data atual é 01/01/2010 e que o vendedor Paulo Lopes Nunes (código 2) está realizando a venda de 1 unidade de game console playstation 2 sony scph-90006 desbloqueado (código 427), 2 unidades de game joystick fujiwork playstation 2 gamepad (blue) (código 431) e 1 unidade de mp4 - 16gb apple ipod nano silver mc060zy/a (código 1043). Da forma como uma venda foi representada, dois passos são necessários para cadastrar todos os dados:

Passo 1: Cadastrar a venda.

Para cadastrar esta venda, basta utilizar o insert abaixo:

```
insert into venda (vendedor) values (2);
```

Novamente, como código da venda é auto incrementado, não aparece. Mas e a data da venda? Porque também não apareceu? Neste caso, porque foi utilizado o valor padrão para o campo dia. Sem utilizar o valor padrão para dia, a data da venda deve ser explicitada:

```
insert into venda (vendedor,dia) values (2,'2010-01-01');
```

Passo 2: Cadastrar cada item da venda.

Chegamos ao ponto de associar cada um dos produtos e quantidades à venda. Mas a qual venda? Primeiro precisamos descobrir qual código foi dado à venda que acabamos de incluir. O select abaixo faz isto:

```
select max(v.codigo) from venda v where (v.dia = '2010-01-01') and (v.vendedor = 2);
```

Considerando que o código resultante tenha sido 1205, os insert abaixo, então, cadastram os itens:

```
insert into item (venda,produto,quantidade) values (1205,427,1);  
insert into item (venda,produto,quantidade) values (1205,431,2);  
insert into item (venda,produto,quantidade) values (1205,1043,1);
```

2.4. UPDATE

É a instrução de alteração de dados. Sua sintaxe mais usual é:

```
UPDATE nome_tabela
  SET nome_coluna = expressão [, ...]
  [ WHERE condição ]
legenda:
  [ ]   opcional
  { | } seleção
  ...   repetição
```

Figura 8 - Sintaxe do update

Observe que no update, assim como no select, também pode ser utilizado where. Tudo o que foi apresentado para o where do select também se aplica ao where do update.

2.4.1. Alterando o preço do produto game console playstation 2 sony scph-90006 desbloqueado (código 427):

Imagine que seja necessário remarcar o preço deste produto, de \$146.50 para \$153.40. O update abaixo faz isto:

```
update produto set preco = 153.40 where (codigo = 427);
```

2.4.2. Alterando o preço das memórias Kingston:

Imagine que, conforme o exemplo 2.1.25, se deseja efetivar um desconto de 10% no preço das memórias Kingston. O update abaixo faz isto:

```
update produto set preco = preco*0.9 where (p.descricao like '%mem%kingston%');
```

Finalizando, uma observação muito importante deve ser feita quanto ao uso do update. Embora alguns SGBDs permitam, nunca se deve alterar o valor da chave primária de um registro, sob o risco de comprometer a integridade, a coerência e a consistência dos dados.

2.5. DELETE

É a instrução de exclusão de dados. Sua sintaxe mais usual é:

```
DELETE FROM nome_tabela
  [ WHERE condição ]
legenda:
  [ ]   opcional
  { | } seleção
  ...   repetição
```

Figura 9 - Sintaxe do delete

Observe que no delete, assim como no select, também pode ser utilizado where. Tudo o que foi apresentado para o where do select também se aplica ao where do delete.

2.5.1. Excluindo o vendedor Paulo Lopes Nunes (código 2):

O `delete` que faria isto é o abaixo:

```
delete from vendedor where (codigo = 2);
```

Mas isto não pode ser feito! Como ficariam as vendas efetuadas (que referenciam) este vendedor? Neste caso, o vendedor não pode ser efetivamente excluído sob pena de deixar o banco de dados inconsistente. Uma possível solução seria criar mais um campo que indicasse se o vendedor está ativo ou inativo. Excluir um vendedor seria, então, apenas torná-lo inativo, mantendo seus dados cadastrados no banco de dados. Este mesmo raciocínio vale para todas as outras tabelas.

Entretanto, se realmente for necessário excluir um vendedor, o procedimento correto seria, primeiro, excluir em todas as outras tabelas, as linhas que referenciem este vendedor e, por último, excluir o vendedor da tabela vendedor. A sequência de `delete` abaixo faria isto:

```
delete from venda where (vendedor = 2);  
delete from vendedor where (codigo = 2);
```

Mas isto causa mais um problema. E os itens das vendas que foram excluídas? Não podem existir itens sem uma venda e, portanto, devem ser excluídos também! E isto não pode ser feito com apenas um `delete`!

2.5.2. Excluindo todos os vendedores:

O `delete` que faria isto é o abaixo:

```
delete from vendedor;
```

Novamente, é importante ressaltar que excluir um vendedor, ou qualquer entidade que seja referenciada por outras, causa uma cascata de problemas.

2.6. DROP TABLE

É a instrução de destruição de tabelas. Sua sintaxe mais usual é:

```
DROP TABLE nome_tabela  
legenda:  
[ ]   opcional  
{ | } seleção  
...   repetição
```

Figura 10 - Sintaxe do `drop table`

Assim como o `delete`, o `drop table` deve ser utilizado com responsabilidade e apenas depois que todas as dependências referenciais forem resolvidas.

2.6.1. Excluindo a tabela vendedor:

O `drop table` que faria isto é o abaixo:

```
drop table vendedor;
```

Observe que existe uma grande diferença entre excluir todos as linhas da tabela vendedor (`delete from vendedor`) e excluir a tabela vendedor do banco de dados (`drop table vendedor`).

Com estes últimos exemplos, acabamos de apresentar os comandos básicos da linguagem SQL.