

Exercícios

01) Obter uma lista da movimentação dos produtos com a data, código do produto, descrição do produto, tipo do movimento ('E' para entrada e 'S' para saída) e quantidade ordenado por data e código do produto

```
SELECT entrada.data, produto.codigo, produto.descricao, 'E' AS  
tipo, entrada.quantidade FROM produto,entrada WHERE  
produto.codigo = entrada.produto UNION ALL SELECT saida.data,  
produto.codigo, produto.descricao, 'S' AS tipo, saida.quantidade  
FROM produto,saida WHERE produto.codigo=saida.produto ORDER BY  
data, codigo;
```

data	codigo	descricao	tipo	quantidade
2010-03-01	01	Caneta	E	5
2010-03-01	01	Caneta	E	15
2010-03-01	04	Borracha	E	15
2010-03-01	04	Borracha	E	5
2010-03-01	04	Borracha	E	5
2010-03-01	06	Caderno	E	10
2010-03-01	06	Caderno	E	30
2010-03-02	02	Lapis	E	15
2010-03-02	04	Borracha	S	8
2010-03-03	01	Caneta	S	3
2010-03-03	01	Caneta	E	5
2010-03-03	02	Lapis	S	1

Exercícios

2010-03-03	06	Caderno	S	6
2010-03-05	01	Caneta	S	4
2010-03-05	04	Borracha	S	1
2010-03-05	04	Borracha	E	10
2010-03-06	05	Regua	E	15
2010-03-07	06	Caderno	S	3
2010-03-08	02	Lapis	S	4
2010-03-08	02	Lapis	E	4
2010-03-08	05	Regua	E	10
2010-03-10	01	Caneta	S	2
2010-03-12	02	Lapis	E	8
2010-03-12	02	Lapis	S	8
2010-03-12	06	Caderno	S	5
2010-03-14	01	Caneta	S	1
2010-03-15	02	Lapis	E	7
2010-03-17	06	Caderno	S	3
2010-03-17	06	Caderno	S	2
2010-03-20	01	Caneta	E	4
2010-03-20	01	Caneta	S	2
2010-03-20	01	Caneta	E	6
2010-03-23	02	Lapis	S	3
2010-03-25	01	Caneta	S	3
2010-03-25	01	Caneta	S	3
2010-03-25	04	Borracha	S	8
2010-03-25	04	Borracha	E	5
2010-03-25	04	Borracha	S	4
2010-03-25	06	Caderno	S	4

Exercícios

2010-03-28	 02	 Lapis	 S	 	1
2010-03-28	 06	 Caderno	 S	 	2
2010-03-29	 04	 Borracha	 S	 	5

Exercícios

02) Listar o total de créditos e total de débitos da conta 01, cada total em uma linha

```
SELECT 'Creditos',SUM(valor) FROM lancamento WHERE  
valor>0 AND conta='01' UNION SELECT 'Debitos',SUM(valor)  
FROM lancamento WHERE valor<0 AND conta='01';
```

?column?		sum
-----+		
Creditos		2770.00
Debitos		-2610.00

Exercícios

03) Listar a descrição da fase e nome dos participantes do projeto 01 ordenados por fase, incluir a supervisão

```
SELECT 'Supervisao' AS fase, supervisor AS participante
FROM projeto WHERE codigo='01' UNION SELECT
fase.descricao AS fase, participante FROM participante,
fase WHERE fase.codigo=participante.fase AND projeto='01'
ORDER BY fase;
```

fase		participante
-----+-----		
Analise		Marcos
Analise		Sandra
Design		Paula
Programacao		Andre
Programacao		Silvia
Programacao		Cristian
Programacao		Barbara
Supervisao		Luis
Teste		Flavio
Teste		Natalia

Exercícios

04) Listar o total de horas de supervisão dos projetos

```
SELECT SUM(horas_supervisao) FROM projeto;
```

sum

86

Exercícios

05) Listar a quantidade de automóveis

```
SELECT COUNT(*) FROM automovel;
```

count

15

Exercícios

06) Obter o número de entradas do produto '02'

```
SELECT count(*) FROM entrada WHERE produto='02';
```

count

4

Exercícios

07) Listar o número de programadores participantes do projeto 02

```
SELECT COUNT(*) FROM participante WHERE fase='3' AND  
projeto='02';
```

count

2

Exercícios

08) Listar a quantidade de automóveis produzidos no Brasil

```
SELECT COUNT(*) FROM automovel WHERE pais='Brasil';
```

count

7

Exercícios

09) Obter o número de entradas para cada produto

```
SELECT produto, count(*) FROM entrada GROUP BY produto;
```

produto	count
02	4
06	2
04	5
05	2
01	5

Exercícios

10) Listar o país e o total de veículos produzidos no país

```
SELECT pais,COUNT(*) AS automoveis FROM automovel GROUP BY pais;
```

pais	automoveis
Argentina	5
Mexico	3
Brasil	7

Exercícios

11) Obter a soma das entradas para cada produto

```
SELECT produto, SUM(quantidade) FROM entrada GROUP BY  
produto;
```

produto		sum
-----+-----		
02		34
06		40
04		40
05		25
01		35

Exercícios

12) Listar o país e total de valor pago nos automóveis fabricados no país

```
SELECT país,SUM(preco) AS pago FROM automovel GROUP BY país;
```

país	pago
Argentina	103000.00
Mexico	84500.00
Brasil	145500.00

Exercícios

13) Listar o nome do fabricante e o total de automóveis de cada fabricante

```
SELECT fabricante.nome AS fabricante, COUNT(*) AS  
automoveis FROM automovel, fabricante WHERE  
fabricante.codigo=automovel.fabricante GROUP BY  
fabricante.nome;
```

fabricante	 	automoveis
-----+-----		
Fiat	 	3
Volkswagen	 	5
Chevrolet	 	3
Ford	 	4

Exercícios

14) Listar a descrição da plataforma e o total de horas gasto com fases dessa plataforma

```
SELECT plataforma.descricao AS plataforma,  
SUM(participante.horas) AS horas FROM participante,  
plataforma WHERE plataforma.codigo =  
participante.plataforma GROUP BY plataforma.descricao;
```

plataforma		horas
Android		338
iOS		307

Exercícios

15) Listar a descrição do projeto, descrição da plataforma e o total de horas gasto com fases dessa plataforma para cada projeto

```
SELECT projeto.descricao AS projeto,  
plataforma.descricao, SUM(participante.horas) AS horas  
FROM projeto, participante, plataforma WHERE  
participante.projeto=projeto.codigo AND plataforma.codigo  
= participante.plataforma GROUP BY projeto.codigo,  
plataforma.descricao;
```

projeto	descricao	horas
-----+-----+		
Organizador Financeiro	iOS	94
Agenda Pessoal	Android	97
Organizador Financeiro	Android	97
Catalogo de Colecoes	iOS	62
Editor de Fotos	Android	81
Editor de Fotos	iOS	88
Organizacao de Eventos	Android	63
Organizacao de Eventos	iOS	63

Exercícios

16) Listar o nome do fabricante e o total do valor de venda dos automóveis do fabricante

```
SELECT fabricante.nome AS fabricante,SUM(valor) AS valor  
FROM venda,automovel,fabricante WHERE automovel.codigo =  
venda.automovel AND fabricante.codigo =  
automovel.fabricante GROUP BY fabricante.nome;
```

fabricante	 	valor
-----+-----		
Fiat	 	52500.00
Volkswagen	 	170000.00
Chevrolet	 	44500.00
Ford	 	61100.00

Exercícios

17) Obter a média das quantidades das saídas de cada produto, ordenado pelo código do produto

```
SELECT produto, AVG(quantidade) FROM saida GROUP BY  
produto ORDER BY produto;
```

produto	avg
01	2.5714285714285714
02	3.4000000000000000
04	5.2000000000000000
06	3.5714285714285714

Exercícios

18) Obter a maior quantidade saída de cada produto, ordenada por produto

```
SELECT produto, MAX(quantidade) FROM saída GROUP BY  
produto ORDER BY produto;
```

produto		max
-----	+	-----
01		4
02		8
04		8
06		6

Exercícios

19) Obter os produtos e total de saídas para os produtos que tiveram total de saídas maior que 20, ordenado por código do produto

```
SELECT produto, SUM(quantidade) FROM saida GROUP BY  
produto HAVING SUM(quantidade) > 20 ORDER BY produto;
```

produto		sum
04		26
06		25

Exercícios

20) Listar o nome das revendas e total do valor de venda para as revendas com total de vendas acima de R\$ 50.000,00

```
SELECT revenda.nome AS fabricante,SUM(valor) AS valor  
FROM venda,revenda WHERE revenda.codigo=venda.revenda  
GROUP BY revenda.nome HAVING SUM(valor)>50000;
```

fabricante	 	valor
-----+-----		
Alameda	 	50100.00
Paraíso	 	81500.00
Cabana	 	70000.00

Exercícios

21) Listar a descrição dos produtos que tiveram mais de 3 entradas

```
SELECT produto.descricao AS produto FROM entrada,produto  
WHERE produto.codigo=entrada.produto GROUP BY  
produto.descricao HAVING(COUNT(*)) > 3;
```

produto

Lapis

Caneta

Borracha

Exercícios

22) Listar o total de horas de cada fase dos projetos (incluindo supervisão) ordenados pelo total de horas em ordem decrescente

```
SELECT 'Supervisao' AS fase, SUM(horas_supervisao) AS  
horas FROM projeto UNION SELECT fase.descricao AS fase,  
SUM(horas) FROM participante, fase WHERE fase.codigo =  
participante.fase GROUP BY fase.descricao ORDER BY horas  
DESC;
```

fase		horas
-----+-----		
Programacao		462
Teste		183
Analise		95
Supervisao		86
Design		30

Exercícios

23) Listar a descrição do projeto e o custo total do projeto

```
SELECT projeto.descricao AS projeto, SUM(horas_supervisao  
* custo_hora)+(SELECT SUM(horas*custo_hora) FROM  
participante WHERE projeto=projeto.codigo) AS custo FROM  
projeto GROUP BY projeto.codigo;
```

projeto	custo
Editor de Fotos	20915.00
Agenda Pessoal	13346.00
Catalogo de Colecoes	8325.00
Organizador Financeiro	28110.00
Organizacao de Eventos	15215.00

Exercícios

24) Obter o código do produto, data e quantidade(s) da(s) ultima(s) saída(s) de cada produto, considerar apenas a data da saída

```
SELECT produto,data,quantidade FROM saida WHERE  
data=( SELECT MAX(data) FROM saida AS foo WHERE  
produto=saida.produto);
```

produto		data		quantidade
-----	+	-----	+	-----
01		2010-03-25		3
01		2010-03-25		3
02		2010-03-28		1
04		2010-03-29		5
06		2010-03-28		2

Exercícios

25) Obter o código, descrição do produto, total de entradas e total de saídas do produtos, ordenado pela descrição dos produtos

```
SELECT codigo, descricao, ( SELECT SUM(quantidade) FROM
entrada WHERE produto=produto.codigo ) AS entradas,
( SELECT SUM(quantidade) FROM saida WHERE produto =
produto.codigo ) AS saidas FROM produto ORDER BY
descricao;
```

codigo	descricao	entradas	saidas
04	Borracha	40	26
06	Caderno	40	25
01	Caneta	35	18
03	Compasso		
02	Lapis	34	17
05	Regua	25	

Exercícios

26) Obter o código, descrição e o saldo da movimentação dos produtos, ordenados pelo código dos produtos

```
SELECT codigo, descricao, COALESCE( ( SELECT  
SUM(quantidade) FROM entrada WHERE produto=produto.codigo  
, 0 ) - COALESCE( ( SELECT SUM(quantidade) FROM saida  
WHERE produto=produto.codigo ), 0 ) AS saldo FROM produto  
ORDER BY codigo;
```

codigo	descricao	saldo
-----+-----+-----		
01	Caneta	17
02	Lapis	17
03	Compasso	0
04	Borracha	14
05	Regua	25
06	Caderno	15

Exercícios

27) Obter o código, descrição e o saldo da movimentação dos produtos que tem saldo menor ou igual a 15, ordenados pela descrição dos produtos

```
SELECT * FROM ( SELECT codigo, descricao,  
COALESCE( ( SELECT SUM(quantidade) FROM entrada WHERE  
produto=produto.codigo ), 0 ) - COALESCE( ( SELECT  
SUM(quantidade) FROM saida WHERE produto = produto.codigo  
, 0 ) AS saldo FROM produto ) AS foo WHERE saldo <= 15  
ORDER BY descricao;
```

codigo	descricao	saldo
-----	+-----	+-----
04	Borracha	14
06	Caderno	15
03	Compasso	0

Exercícios

28) Obter o código, descrição, data e saldo da movimentação do produto na data ordenado por produto e data

```
SELECT produto.codigo, produto.descricao, foo.data,  
SUM(foo.quantidade) AS saldo FROM produto,( SELECT data,  
produto, quantidade FROM entrada UNION ALL SELECT data,  
produto, -1*quantidade FROM saida ) AS foo WHERE  
produto.codigo=foo.produto GROUP BY produto.codigo,  
foo.data ORDER BY produto.codigo, foo.data;
```

codigo	descricao	data	saldo
01	Caneta	2010-03-01	20
01	Caneta	2010-03-03	2
01	Caneta	2010-03-05	-4
01	Caneta	2010-03-10	-2
01	Caneta	2010-03-14	-1
01	Caneta	2010-03-20	8
01	Caneta	2010-03-25	-6
02	Lapis	2010-03-02	15
02	Lapis	2010-03-03	-1
02	Lapis	2010-03-08	0

Exercícios

02	Lapis	2010-03-12	0
02	Lapis	2010-03-15	7
02	Lapis	2010-03-23	-3
02	Lapis	2010-03-28	-1
04	Borracha	2010-03-01	25
04	Borracha	2010-03-02	-8
04	Borracha	2010-03-05	9
04	Borracha	2010-03-25	-7
04	Borracha	2010-03-29	-5
05	Regua	2010-03-06	15
05	Regua	2010-03-08	10
06	Caderno	2010-03-01	40
06	Caderno	2010-03-03	-6
06	Caderno	2010-03-07	-3
06	Caderno	2010-03-12	-5
06	Caderno	2010-03-17	-5
06	Caderno	2010-03-25	-4
06	Caderno	2010-03-28	-2

Exercícios

29) Obter o código, descrição e total dos salários das diretorias com total de salários maior que \$12000,00

```
SELECT diretoria.codigo, diretoria.descricao,  
SUM(funcionario.salario) FROM diretoria, funcionario,  
secao WHERE diretoria.codigo=secao.diretoria AND  
secao.codigo=funcionario.secao GROUP BY diretoria.codigo  
HAVING SUM(funcionario.salario) > 12000;
```

codigo		descricao		sum
PRD		Producao		13900.00
PES		Pessoal		15900.00

Exercícios

30) Obter o total dos maiores salários de cada diretoria

```
SELECT SUM(salarios) FROM ( SELECT  
MAX(funcionario.salario) AS salarios FROM diretoria,  
funcionario, secao WHERE diretoria.codigo=secao.diretoria  
AND secao.codigo=funcionario.secao GROUP BY  
diretoria.codigo ) AS foo;
```

sum

12900.00

Exercícios

31) Obter para cada produto o código e saldo da movimentação na primeira quinzena e na segunda quinzena do mês 03/2010.

```
SELECT produto, SUM(CASE WHEN data>='2010/03/01' AND  
data<='2010/03/15' THEN quantidade ELSE 0 END) AS  
primeira, SUM(CASE WHEN data > '2010/03/15' AND  
data<='2010/03/31' THEN quantidade ELSE 0 END) AS segunda  
FROM ( SELECT produto, data, quantidade FROM entrada  
UNION ALL SELECT produto, data, -quantidade FROM saida )  
AS foo GROUP BY produto;
```

produto	primeira	segunda
02	21	-4
06	26	-11
04	26	-12
05	25	0
01	15	2