



una

O MELHOR
CENTRO UNIVERSITÁRIO
PRIVADO DE BH

Fonte: MEC

una

O MELHOR
CENTRO UNIVERSITÁRIO
PRIVADO DE BH
Fonte: MEC



Professor Leonardo Larback



Funções

Para facilitar a manipulação dos dados, o mysql oferece algumas funções pré-definidas. Vejamos algumas:

- **Funções de agregação:**

AVG() Retorna a média dos valores passados como argumentos.

COUNT() Retorna a quantidade de linhas retornadas.

MAX() Retorna o maior valor encontrado.

MIN() Retorna o menor valor encontrado.

SUM() Retorna a soma dos valores.

DISTINCT() Elimina dados duplicados em uma consulta

Em geral essas funções são utilizadas em conjunto com a clausula *group by*:

```
select count(*) from pessoas group by sexo
```



Funções

- **Funções de data e hora** (<http://dev.mysql.com/doc/refman/5.0/en/date-and-time-functions.html>)

curdate() Retorna a data corrente do sistema.

curtime() Retorna a hora corrente do sistema.

now() Retorna a data e a hora corrente [A função *now* retorna a hora em que a query começou a ser executada, a função *sysdate()* retorna a hora exata da execução. `select sysdate(), sleep(2), sysdate();`]

last_day() Retorna o último dia do mês da data informada como parâmetro.

date() Extrai a data de uma expressão datetime.

day() Extrai o dia de uma expressão datetime.

month() Extrai o mês de uma expressão datetime.

year() Extrai o ano de uma expressão datetime.

hour() Extrai a hora de uma expressão datetime.

minute() Extrai os minutos de uma expressão datetime.

second() Extrai os segundos de uma expressão datetime.



Funções

- **Funções de data e hora** (<http://dev.mysql.com/doc/refman/5.0/en/date-and-time-functions.html>)

monthname() Retorna o nome do mês.

dayname() Retorna o nome do dia da semana.

dayofweek() Retorna o número do dia da semana.

dayofyear() Retorna o número do dia do ano (1 - 366).

adddate() Adiciona N à uma data.

Exemplos:

- `select now();`
- `select date(now());`
- `select day(now());`
- `select adddate(now(), interval 10 DAY);`
- `select dayofyear(now());`
- `select dayofweek(now());`



Funções

- **Funções para manipulação de strings** (<http://dev.mysql.com/doc/refman/5.0/en/string-functions.html>)
 - ascii()* Retorna o valor numérico do caracter informado.
 - bin()* Retorna uma string contendo o número informado em binário.
 - conv()* Retorna uma string convertida da base x para y.
 - char()* Retorna o caracter correspondente ao número informado.
 - char_length()* Retorna o número de caracteres informados.
 - lower()* Retorna o argumento passado em minúsculo.
 - upper()* Retorna o argumento passado em maiúsculo.
 - reverse()* Retorna o argumento invertido.
 - replace()* Retorna uma string onde todas as ocorrências de *string1* são substituídas por *string2*.
 - substr()* Retorna N caracteres de uma string.



Funções

- **Funções para manipulação de strings** (<http://dev.mysql.com/doc/refman/5.0/en/string-functions.html>)

Exemplos:

- `select ascii('c');`
- `select char(97);`
- `select bin(ascii('l'));`
- `select conv(11,2,10);`
- `select lower("PEDRO");`
- `select upper("pedro");`
- `select reverse('cabral');`
- `select replace('cabral','a','4');`
- `select substr('cabral',4,2);`



Funções

- **Funções informativas** (

<http://dev.mysql.com/doc/refman/5.0/en/information-functions.html>)

database() Retorna o nome do banco de dados corrente.

last_insert_id() Retorna o último valor gerado em um campo AUTOINCREMENT.

user() Retorna o nome do usuário logado.

version() Retorna uma string indicando a versão do servidor Mysql instalado.



Funções

Prática orientada

Conecte-se à máquina do professor via SSH – o professor informará os dados necessários – Se desejar fazer os testes em sua própria máquina deverá criar um banco de dados e as tabelas utilizadas. Também será sua responsabilidade cadastrar os dados.

Abra o mysql: **mysql -u root -p**

Selecione o banco de dados una: **use una;**

Confira qual banco de dados está selecionado: **select database();**

Confira qual a versão do SGBD instalado: **select version();**

Confira qual o usuário atualmente logado: **select user();**

Entre as tabelas existentes no banco de dados, encontra-se a tabela gastos. Esta tabela será utilizada para os nossos próximos testes e possui a seguinte estrutura:

```
mysql> desc gastos;
```

Field	Type	Null	Key	Default	Extra
data	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP
descricao	varchar(100)	YES		NULL	
valor	float(5,2)	YES		NULL	
controle	varchar(200)	YES		NULL	



Funções

Prática orientada

A tabela gastos já possui alguns registros cadastrados. Vamos descobrir quantos:

```
select count(*) from gastos;
```

A tabela possui vários registros com a mesma descrição, vejamos quais as descrições existentes:

```
select distinct(descricao) from gastos;
```

Vejamos quanto totaliza a coluna valores:

```
select sum(valor) from gastos;
```

Agora, vejamos quanto foi gasto em cada item descrito:

```
select descricao, sum(valor) from gastos group by descricao;
```

Agora, vejamos a média geral dos valores gastos:

```
select avg(valor) from gastos;
```

Para vermos a média gasta por descrição, devemos agrupar os dados:

```
select descricao, avg(valor) from gastos group by descricao;
```



Funções

Prática orientada

Agora, vejamos qual o maior valor cadastrado:

```
select max(valor) from gastos;
```

Para sabermos qual registro possui esse valor, devemos utilizar subconsultas:

```
select * from gastos where valor=(select max(valor) from gastos);
```

Atenção: Desenvolvedores menos experientes tendem a pensar que a consulta `select descricao, max(valor) from gastos;` tem o mesmo resultado da subconsulta acima, o que **NÃO** é correto.

O mesmo raciocínio aplicado para descobrir o maior valor gasto e o registro com esse valor, pode ser aplicado para descobrirmos o menor valor gasto e o registro com esse valor – usando a função `min()`.

```
select min(valor) from gastos;
```

```
select * from gastos where valor=(select min(valor) from gastos);
```



Funções

Prática orientada

Agora, vamos descobrir quantas vezes cada descrição ocorre:

```
select count(*), descricao from gastos group by descricao;
```

Se você precisar acrescentar alguma condição em cima de um valor agregado, deve usar a cláusula **having** e não o usual **where** (<http://www.mysqltutorial.org/mysql-having.aspx>). Por exemplo, digamos que queremos repetir a consulta anterior mas limitando a exibição aos registros que tenham mais de uma ocorrência:

```
select count(*), descricao from gastos group by descricao having count(*)>1;
```

Se desejarmos, podemos acrescentar “rotulos” para as colunas utilizando a cláusula **AS**:

```
select count(*) as quantidade, descricao from gastos group by descricao  
having quantidade>1;
```



Funções

Prática orientada

Continuando nossos testes, vamos exibir as descrições em letras maiúsculas:

```
select upper(descricao) from gastos;
```

Podemos ver os valores em binário:

```
select descricao, bin(valor) from gastos;
```

Podemos ver o nome do mês em que ocorreu o lançamento:

```
select monthname(data) as mes, descricao, valor from gastos;
```

Você pode voltar alguns slides e criar seus próprios testes para as funções aqui apresentadas. Além disso, não deixe de seguir os links apontados para uma leitura complementar.

Para terminarmos, você já deve ter percebido (se está usando o computador do professor) que o campo controle está encriptado. Ele foi encriptado utilizando a função password(), entretanto, existem inúmeras outras funções de criptografia disponíveis no mysql - <http://dev.mysql.com/doc/refman/5.5/en/encryption-functions.html> - Apenas para uma melhor compreensão, vamos cadastrar um novo gasto:

```
insert into gastos (descricao,valor,controle) values ('Cafe na Una',10,password('CafeNaUna'));
```



Funções

Prática orientada

A encriptação utilizada pela função password é one-way, ou seja, não é possível revertê-la. Assim, se você desejar saber qual o valor guardado em um campo encriptado, deverá fornecer a palavra-chave, criptografá-la e comparar com o valor armazenado no banco:

```
select * from gastos where  
controle=password('CafeNaUna');
```