Procedures

MySql

Prof. Ricardo Satoshi

Procedimentos Armazenados (Stored Procedures)

- Um procedimento armazenado **[stored procedure]** é um conjunto de comandos SQL que são compilados e armazenados no servidor.
- A vantagem de usar **procedimentos armazenados** é que eles podem encapsular rotinas de uso freqüente no próprio servidor, e estarão disponíveis para todas as aplicações.
- Parte da lógica do sistema pode ser armazenada no próprio banco de dados, em vez de ser codificada várias vezes em cada aplicação.

Criando Stored Procedures pelo DDL

Para criar um procedimento use o comando CREATE PROCEDURE.

Por exemplo, o procedimento abaixo recebe um parâmetro de entrada (in sex) e mostra todos os alunos com o sexo informado:

```
create procedure Buscasexo (in sex char(1))
    select nome, sobrenome from tabalunos
    where sexo=sex ;
```

Note que os parâmetros são sempre declarados logo após o nome do procedimento. Um procedimento pode ter zero ou mais parâmetros. Declara-se o nome do procedimento, e a seguir o nome e o tipo (de dados) do parâmetro : **in** para entrada e **out** para saida.

Executando procedimentos armazenados

- Para executar um procedimento usa-se o comando CALL. Por exemplo, execute o procedimento anterior da seguinte forma:

```
CALL Buscasexo('f')
```

- O resultado será as linhas da tabela tabalunos que tiverem o sexo passado como parâmetro.

- Deste modo o procedimento pode buscar tanto mulheres como homens, dependendo do parâmetro.

Comandos para uso em procedimentos armazenados

É possível criar procedimentos **mais complexos**, formados por mais de uma linha de comando.

Nesses casos é necessário informar ao mysql um novo delimitador para que ele não entenda o ponto e vírgula **(;)** como término do comando.

Um procedimento pode ter uma lógica de execução que decida qual sequência de comandos usar dependendo de uma ou outra situação.

Para tanto podem ser definidas variáveis locais ao procedimento.

Comandos para uso em procedimentos armazenados (cont)

```
DELIMITER $
CREATE PROCEDURE Buscasexo(in s char(1))
begin
  declare contagem integer;
  declare mensagem char(100);
  set contagem=(select count(sexo) from tabalunos
  where sexo=s);
  if (contagem =0) then
     begin
     set mensagem = concat('Nao ha alunos com
 o sexo "', s,'"');
     SELECT MENSAGEM;
     end;
  ELSE
  SELECT nome, sobrenome from tabalunos where
  sexo=s;
  END IF;
  end$
DELIMITER ;
```

Comandos para uso em procedimentos armazenados (cont)

- O comando DECLARE declara variáveis internas ao procedimento. No caso, **contagem** é uma variável do tipo integer e **mensagem** do tipo char(100).
- Note que quando você usa um comando SELECT, o resultado pode ser colocado numa variável, como contagem acima.
- Para que se possa atribuir um valor a uma variável usa-se o comando SET. Exemplo: SET variável = 0.
- Os comandos BEGIN e END são usados para definir um bloco de comandos que passa a ser tratada como se fosse um comando único. No caso acima, eles são necessários para poder executar dois comandos dentro do IF (o SET e o SELECT).

Outros exemplos

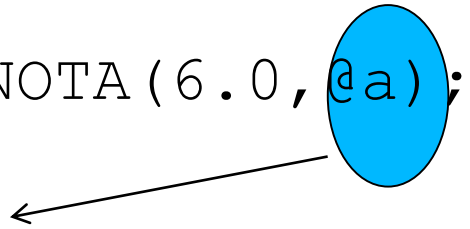
```
CREATE PROCEDURE BuscaMedia(in M numeric(5,2))
   SELECT CONCAT(NOME,' ',SOBRENOME) AS NOME,
         round((nota1+nota2)/2,2) as media
         FROM TABALUNOS WHERE (nota1+nota2)/2 < M;
```

Mostra nome, sobrenome dos alunos com média inferior ao valor passado por parâmetro

```
CALL BuscaMedia(6.5);

CALL BuscaMedia(8);
```

Outros exemplos – parâmetro de saída

```
Delimiter &

CREATE PROCEDURE BuscaNota(in N numeric(5,2),
                           out s integer)
Begin

  SELECT CONCAT(NOME,' ',SOBRENOME) AS NOME,

       round((nota1+nota2)/2,2) as media

       FROM TABALUNOS WHERE (nota1+nota2)/2 < N;

   set s=(select count(*) from tabalunos

            where (nota1+nota2)/2 < N);

End&
Delimiter ;


CALL BUSCANOTA(6.0,@a);

SELECT @a;
```



Procedimentos armazenados (cont)

- Um procedimento armazenado pode conter quaisquer comandos do SQL.
- O exemplo a seguir mostra um procedimento que atualiza a nota1 dos alunos caso esta nota seja inferior a 6

```
CREATE PROCEDURE AlteraNota1()
   UPDATE TABALUNOS SET NOTA1 = NOTA1+0.5
         WHERE NOTA1 < 6;
```

Procedures

Para saber os nomes dos procedimentos criados em um banco de dados, use:

```
SELECT name FROM mysql.proc  WHERE type='procedure'
and DB= <'nome_banco'>;
```

Para saber como um procedimento foi criado use:

```
SHOW CREATE PROCEDURE <nome_procedure>;
```