VIEWS

Wagner Bianchi
Certified MySQL 5.0 Developer
Certified MySQL 5.0 Database Administrator
Certified MySQL 5.1 Cluster Database Administrator

Views

- Uma ***View*** ou visualização é uma tabela derivada de outras tabelas existentes no banco de dados, chamadas de ***based tables*** ou tabelas base;
- Também é composta de colunas e registros;
- A diferença é que os dados de uma ***View*** é o resultado de um mapeamento de colunas e registros de outras tabelas;
- Podem existir ***View***s de ***View***s;
- Existem Views materializáveis e não-materializáveis;
- Toda a definição de uma ***View*** é armazenado no dicionário de dados do MySQL, podendo ser recuperado com o seguinte **SELECT**:

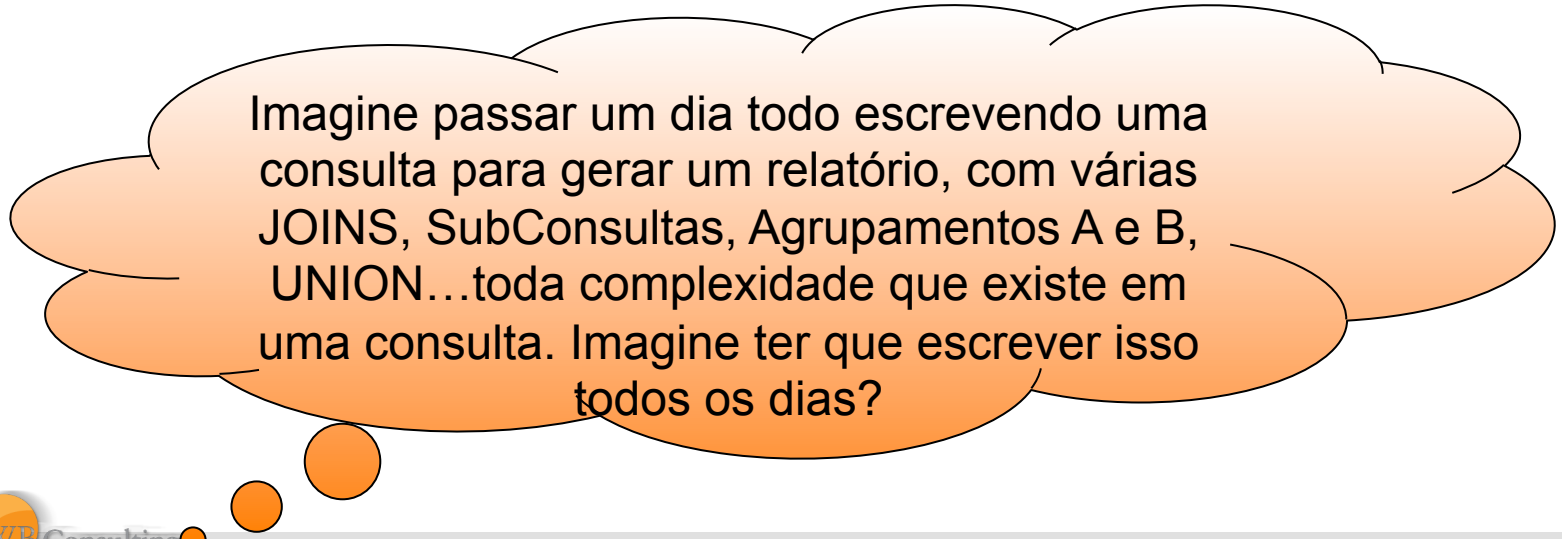
```
mysql> SELECT TABLE_NAME FROM INFORMATION_SCHEMA.VIEWS;
```

Views

- Segurança
  - As aplicações acessam ***View***s e não as tabelas;
  - Facilita o controle de acesso aos dados;
  - Usuários não são capazes de ver determinadas informações;
- Particionamento de Dados
  - Particionamento horizontal: Subconjunto de registros;
  - Particionamento vertical: Subconjunto de colunas;
- Facilidade o Acesso aos Dados
  - Consultas complexas pode ser mapeadas em uma ***View***;
  - O acesso aos dados pode ser feito a partir de uma ***View***, evitando a criação a criação de uma consulta complexa;

Views

- Independência Sobre a Definição de Dados
  - Alterações na estrutura física de dados não reflete na aplicação;
  - Cria camada entre a aplicação e o banco de dados;
  - Alteração no dicionário de dados podem ser mascaradas pela ***View***;
  - Em alguns casos não será possível manter a mesma estrutura;



Views

- Comando para criação de ***View***s

```
CREATE VIEW view_name(columns_name)
AS SELECT…
```

- O comando consiste em:
  - O nome para a ***View*** (view_name);
  - O comando SELECT que realizará o mapeamento de dados (*SELECT…*);
  - Opacionalmente, o nome das colunas da ***View***

Views

- Utilizando o banco de dados world, segue os exemplo:
  - Uma View mostrado uma tabela inteira:

```
mysql> CREATE VIEW vw_country AS SELECT * FROM Country;
Query OK, 0 rows affected (0.00 sec)
```

  - Uma View com um **SELECT** mais complexo:

```
mysql> CREATE VIEW vw_country_language
    -> AS
    -> SELECT a.Name, GROUP_CONCAT(b.language)
    -> FROM Country AS a INNER JOIN CountryLanguage AS b
    -> ON a.Code = b.CountryCode
    -> GROUP BY a.Name;
Query OK, 0 rows affected (0.02 sec)
```

Views

- Podemos renomear os nomes das colunas através de aliases – apelidos;

```
mysql> CREATE VIEW vw_country_names
    -> AS
    -> SELECT DISTINCT Name AS 'Países' FROM Country;
Query OK, 0 rows affected (0.00 sec)
```

- Podemos utilizar a lista de colunas para também renomear as colunas de uma ***View***;

```
mysql> CREATE VIEW vw_country_names (Paises)
    -> AS
    -> SELECT DISTINCT Name FROM Country;
Query OK, 0 rows affected (0.00 sec)
```

Views

- Para mostrar os dados de uma ***View***, já que ela é tida como uma tabela virtual, basta selecionarmos a mesma através do comando **`SELECT`**:

```
SELECT * FROM view_name ORDER BY 1 ASC;
```

- Na selecão precedente, utilizamos o **`ORDER BY`** para ordenar os resultados pela primeira coluna. Esse **`ORDER BY`** elimina um **`ORDER BY`** interno à ***View***;

Views

- Uma View pode ser criada a partir de qualquer comando **SELECT** válido, inclusive SubConsultas;
- Views podem ser atualizáveis ou não;
- Para alterar um View já existente, utilize os comandos **ALTER VIEW** ou **OR REPLACE**;

```
ALTER VIEW…
```

```
…ou ainda:
```

```
CREATE VIEW OR REPLACE view_name…
```

Exercício

- Artigo: Trabalhando com Views no MySQL

- Com base no assunto apresentando "***Views***", responda a **LISTA 9**.

- **Fonte de referência**: http://dev.mysql.com/doc/refman/5.0/en/views.html