

## **Aula 4**

### **Sistema Gerenciador de Banco de Dados**

Prof. Leonel da Rocha

1

### **Conversa Inicial**

2

### **Manutenção SGBD**

- Nessa aula veremos alguns cuidados que devemos ter com um sistema gerenciador de banco de dados e o conteúdo por ele gerenciado
- Veremos como a manutenção de um banco é importante para manter os dados em boas condições

3

### **Manutenção SGBD**

- Abordaremos a importação e exportação de dados, tanto para outros bancos de dados de um mesmo SGBD quanto para outros gerenciadores e sistemas
- Vamos ver a replicação de dados que é uma ferramenta importante no gerenciamento de dados

4

### **Manutenção SGBD**

- Trataremos da migração de dados que possibilita fazer a transferência de base de dados para versões atualizadas de um SGBD ou de outros sistemas
- Para finalizar veremos rotinas de limpeza e ocupação de disco, que precisa de uma atenção especial, pois é nesse ambiente que o SGBD funciona

5

### **Manutenção de um SGBD**

6

### Manutenção de um SGBD

- O estado dos dados afeta sua disponibilidade e eficiência, sendo necessário mantê-los em boas condições
- É sob essa ótica que a manutenção de dados deve ser tratada
- O DBA deve cuidar e proteger os dados contra acessos não autorizados

7

### Manutenção de um SGBD

- A manutenção é um conjunto de atividades que visam garantir o bom funcionamento de um banco de dados
- As soluções de gerenciamento de um banco de dados exigem atualizações regulares para acomodar novas tecnologias e evitar riscos de segurança

8

### Manutenção de um SGBD

- Atividades de manutenção:
  - Backups
  - Exportação/importação de dados
  - Revisões de índices
  - Exclusão de objetos
  - Exclusão de dados redundantes ou inúteis
  - Expurgo de dados

9

### Manutenção de um SGBD

- Plano de manutenção do banco de dados:
  - Backup dos dados
  - Rotinas de desfragmentação dos índices
  - Coleta de estatísticas de desempenho
  - Mysqlcheck: verificar, reparar, otimizar e analisar tabelas
  - OPTIMIZE: reorganização e redução do tamanho de tabelas
  - CHECK TABLE: verificação de dados

10

### Importar/exportar dados

11

### Importar/exportar dados

- Práticas recomendadas a serem consideradas ao importar e exportar dados:
  - Minimizar o impacto no desempenho das exportações
  - Usar as sinalizações corretas ao criar um arquivo dump SQL
  - Compactar dados para reduzir custos
  - Reduzir processos de importação e exportação de longa duração
  - Verificar o banco de dados importado

12

### Importar/exportar dados

- **Minimize o impacto no desempenho das exportações**
  - Processo é executado enquanto o banco de dados está on-line
  - Quando os dados que estão sendo exportados são menores, é provável que o impacto seja mínimo
  - Com objetos maiores, é possível que o desempenho seja prejudicado

13

### Importar/exportar dados

- **Use as sinalizações corretas ao criar um arquivo dump SQL**
  - Se não forem usadas as sinalizações corretas ao exportar dados para um arquivo dump SQL, pode ocorrer uma falha na importação

14

### Importar/exportar dados

- **Comando para criação do arquivo dump SQL:**
  - `mysqldump --databases DATABASE_NAME -h INSTANCE_IP -u USERNAME -p \`
  - `--hex-blob --single-transaction --set-gtid-purged=OFF \`
  - `--default-character-set=utf8mb4 > SQL_FILE.sql`

15

### Importar/exportar dados

- **Compacte dados para reduzir custos**
  - O MySQL permite a importação e exportação de arquivos compactados e não compactados
  - A compactação pode economizar um espaço de armazenamento significativo e reduzir os custos com armazenamento, principalmente durante a exportação de grandes instâncias

16

### Importar/exportar dados

- **Reduza processos de importação e exportação de longa duração**
  - É possível diminuir o tempo necessário para concluir cada operação usando a função de importação ou exportação do MySQL com lotes menores de dados

17

### Importar/exportar dados

- **Verifique o banco de dados importado**
  - Após a conclusão de uma operação de importação é importante conectar-se ao banco de dados e executar os comandos apropriados para garantir que o conteúdo esteja correto
  - Por exemplo, conectar-se e listar os bancos de dados, tabelas e entradas específicas

18

### Replicação de dados

19

### Replicação de dados

- O objetivo de um mecanismo de replicação de dados é permitir a manutenção de várias cópias idênticas de um banco de dados
- Os benefícios da replicação de dados são a redundância, o que torna o sistema tolerante a falhas

20

### Replicação de dados

- Possibilidade de um balanceamento de carga do sistema, pois é possível distribuir o acesso entre as cópias
- Ter um *backup* dos dados, já que a replicação transfere parcial ou totalmente os dados de origem para um outro local

21

### Replicação de dados

- Com o MySQL, é possível realizar um tipo de replicação conhecida como *Master-Slave*, em que temos um servidor atuando como *master* e um ou mais servidores atuando como *slave*
- O servidor *master* armazena em um *log* de alteração todos os comandos de atualizações do banco de dados; a partir disso, todas as alterações ocorridas no servidor *master* são replicadas para os servidores *slave*

22

### Replicação de dados

- Vantagens da replicação de dados:
  - Equilíbrio de carga, pois a replicação não precisa de uma largura de banda muito alta, sendo possível ainda iniciar e parar conforme a necessidade
  - Além de ser possível para os servidores trabalharem distantes um do outro

23

### Replicação de dados

- Vantagens da replicação de dados:
  - Possibilidade de implementar a distribuição de dados, em que será possível distribuir as consultas de leitura por meio de diversos servidores
  - Com isso, o mesmo dado pode estar armazenado em todos os servidores envolvidos na replicação

24

### Replicação de dados

- Vantagens da replicação de dados:
  - Pode ser uma técnica de *backup*, lembrando, porém, que um *slave* não é exatamente um *backup* e não deve ser um substituto para os *backups* tradicionais
  - Permite uma alta disponibilidade dos dados, já que, se um servidor falhar, teremos outro para substituí-lo

25

### Migração de dados

26

### Migração de dados

- O processo da migração de dados pode ser complexo, já que existe muitas bases de dados diferentes
- Além da dificuldade em transferir os dados entre os dois SGBD, também contribuirá muito na complexidade da migração o tipo de dados das tabelas envolvidas no processo

27

### Migração de dados

- Datas e booleanos podem dar problemas ao passar de um SGBD para outro, porque podem ser tratados de maneiras diferentes
- Números decimais apresentam problemas com uma precisão diferente

28

### Migração de dados

- Alguns processos de migração possuem maior facilidade
- Se a nossa base de dados origem estiver em Access a migração será fácil, pois o MySQL dispõe de um *driver* ODBC para sistemas Windows, que permite conectar Access com o próprio MySQL e realizar de maneira fácil a migração

29

### Migração de dados

- Para migrar dados do SQL Server para o MySQL, podemos utilizar o Access como ponte entre esses dois SGBDs
- O Access permite selecionar uma base de dados do SQL Server e depois exportamos para o MySQL

30

### Migração de dados

- Para outros SGBDs que possuam conexão via driver ODBC não haverá maiores problemas para conectá-la com Access, de maneira similar como se conecta com MySQL
- Podemos então utilizar o Access para exportar os dados, para o MySQL por exemplo

31

### Migração de dados

- Se não temos Access, ou a base de dados original não tem *driver* ODBC, ou se o processo de migração não funcionou corretamente, outra possibilidade é exportar os dados para um arquivo de texto separados por vírgulas (CSV)
- Muitas bases de dados têm ferramentas para exportar os dados das tabelas para arquivos de texto, os quais podem ser inseridos no MySQL

32

### Migração de dados

- Em algumas circunstâncias, possivelmente precisaremos criar algum *script* que realize as mudanças necessárias nos dados para adequá-los aos tipos de dados que são tratados de forma diferentes entre os SGBDs
- Valores booleanos – 0/1 ou v/f
- Datas – “dd/mm/aaa” ou “aaaa/mm/dd”

33

### Verificação de ocupação de espaço em disco

34

### Verificação de ocupação de espaço em disco

- O bom funcionamento de um SGBD depende de muitos fatores, entre eles, o espaço em disco é um item muito importante
- Veremos aqui como o MySQL trata e se comporta com discos sem espaço

35

### Verificação de ocupação de espaço em disco

- Quando ocorre uma condição de disco sem espaço, o MySQL faz algumas verificações importantes que veremos a seguir:
  - É verificada, a cada minuto, a existência de espaço suficiente para gravar os dados
  - Se houver espaço suficiente, ele continua gravando sem interromper o funcionamento

36

#### Verificação de ocupação de espaço em disco

- De seis em seis minutos, é gravada uma entrada no *log* de arquivo avisando sobre a condição de disco cheio
- Para continuar quando o disco está cheio, é preciso liberar espaço suficiente em disco para inserir todos os registros
- Para abortar um processo de gravação, é preciso enviar um `mysqladmin kill` para o procedimento

37

#### Verificação de ocupação de espaço em disco

- Outro processo pode estar esperando pelas tabelas que provocaram a condição de disco cheio
- Se vários processos estão bloqueados, matar o que está esperando pela condição de disco cheio vai permitir os outros de continuarem

38

#### Verificação de ocupação de espaço em disco

- Com a utilização do comando `TRUNCATE TABLE`, é possível recuperar o espaço em disco do sistema operacional
- A capacidade de truncar tabelas e devolver espaço em disco ao sistema operacional também significa que os *backups* físicos podem ser menores

39