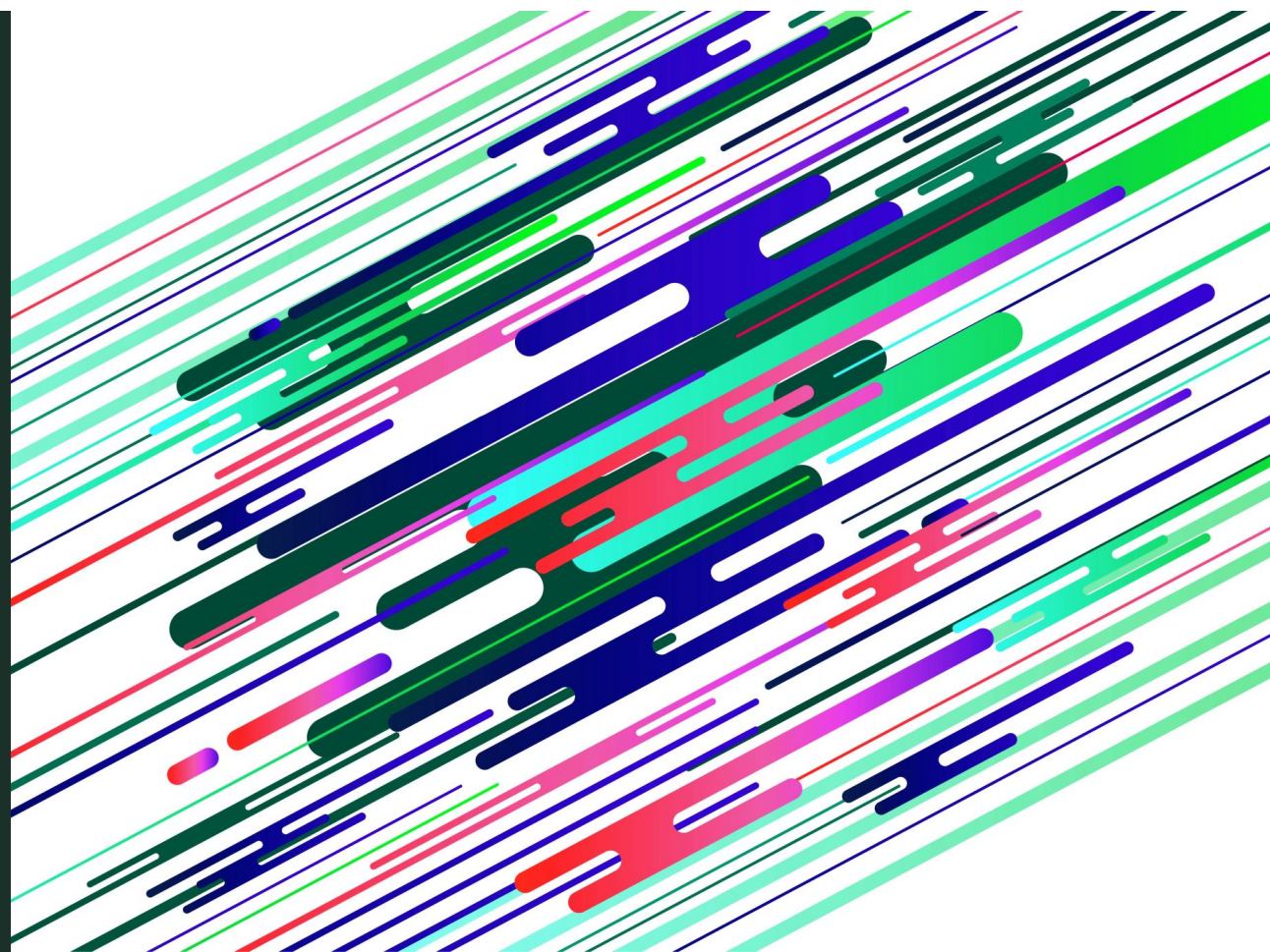


Professor: Jorge Baldez

# Jobs em SGBD MySQL

Introdução às rotinas automáticas no MySQL



# O que são Jobs?

- Jobs são tarefas automatizadas que são executadas em horários pré-determinados.
- No MySQL, essas tarefas são agendadas usando **Eventos**.
- Utilizados para manutenção, backup, processamento de dados, etc.

# Por que usar Jobs em SGBD?

- Automatização de tarefas repetitivas.
- Otimização de performance e limpeza de dados.
- Redução de erros humanos em tarefas manuais.
- Execução em horários não comerciais para minimizar impacto.

# Por que usar Jobs em SGBD?

Sintaxe Básica de um Job:

- CREATE EVENT: Comando para criar o job.
- ON SCHEDULE: Define quando o job será executado.
- DO: Código que será executado.

```
1  CREATE EVENT nome_do_evento
2  ON SCHEDULE AT 'YYYY-MM-DD HH:MM:SS'
3  DO
4      BEGIN
5          -- Código SQL
6      END;
```

# Exemplo Prático – Backup Diário

- Agendamento pode ser feito via **cron** em servidores Linux.
- Executado diariamente para garantir segurança dos dados.

- `#!/bin/bash`
- `mysqldump -u user -p password meu_banco > /backup_dir/backup.sql`

# Limpeza de Dados Antigos

## Job para Excluir Registros Antigos:

- Remove registros com mais de 30 dias da tabela logs.
- Mantém o banco de dados eficiente e enxuto.

```
CREATE EVENT limpar_dados_antigos
ON SCHEDULE EVERY 1 DAY
DO
  DELETE FROM logs WHERE data_criacao < NOW() - INTERVAL 30 DAY;
```

# Atualização Automática de Estatísticas

## **Job para Atualizar Estatísticas Diariamente:**

Calcula o total de vendas diárias.

Atualiza automaticamente a tabela de estatísticas.

```
CREATE EVENT atualizar_estatisticas
ON SCHEDULE EVERY 1 DAY
DO
  UPDATE estatisticas SET total_vendas = (SELECT COUNT(*) FROM vendas WHERE data_venda = CURDATE());
```

# Notificação de Usuários Inativos

Job para Notificar Usuários Inativos:

- Notifica usuários que estão inativos por mais de 60 dias.
- Envia notificações automáticas para incentivar o uso.

```
CREATE EVENT notificar_inativos
ON SCHEDULE EVERY 1 WEEK
DO
  INSERT INTO notificacoes (usuario_id, mensagem)
  SELECT id, 'Inativo por 60 dias' FROM usuarios WHERE ultimo_acesso < NOW() - INTERVAL 60 DAY;
```



# Otimização de Tabelas

## Job para Otimizar Tabelas Mensalmente

- Reduz fragmentação e melhora a performance do banco de dados.
- Executado regularmente para manter o sistema rápido.

```
CREATE EVENT otimizar_tabelas  
ON SCHEDULE EVERY 1 MONTH  
DO  
    OPTIMIZE TABLE usuarios, vendas;
```

# Mais exemplos

Job para Backup Diário do Banco de Dados:

```
#!/bin/bash

# Script para backup de um banco de dados MySQL
DB_NAME="meu_banco"
BACKUP_DIR="/path/to/backup/directory"
DATE=$(date +%Y-%m-%d)
USER="meu_usuario"
PASSWORD="minha_senha"

# Executa o comando de dump do MySQL
mysqldump -u $USER -p$PASSWORD $DB_NAME > $BACKUP_DIR/backup_$DB_NAME_$DATE.sql

# Comando para agendar via cron no servidor Linux (executa diariamente às 2h da manhã)
# 0 2 * * * /path/to/backup_script.sh
```

# Mais exemplos

Job para Limpar Dados Antigos:

Neste exemplo:

- A rotina é agendada para ser executada a cada dia, uma hora após o momento de criação do evento.
- Exclui registros na tabela logs que têm mais de 30 dias.

```
DELIMITER //
```

```
CREATE EVENT limpar_dados_antigos  
ON SCHEDULE EVERY 1 DAY  
STARTS CURRENT_TIMESTAMP + INTERVAL 1 HOUR  
DO  
    BEGIN  
        DELETE FROM logs WHERE data_criacao < NOW() - INTERVAL 30 DAY;  
    END //
```

```
DELIMITER ;
```

# Mais exemplos

## Job para Atualizar Estatísticas

O job é executado diariamente à meia-noite, e atualiza uma tabela de estatísticas com o total de vendas e clientes do dia.

```
DELIMITER //
```

```
CREATE EVENT atualizar_estatisticas  
ON SCHEDULE EVERY 1 DAY  
STARTS '2024-10-11 00:00:00'  
DO  
  BEGIN  
    UPDATE estatisticas  
    SET total_vendas = (SELECT COUNT(*) FROM vendas WHERE data_venda = CURDATE()),  
        total_clientes = (SELECT COUNT(DISTINCT cliente_id) FROM vendas WHERE data_venda = CURDATE());  
  END //
```

```
DELIMITER ;
```

# Mais exemplos

Job para Otimização de Tabelas:

O job é agendado para ser executado mensalmente e otimiza as tabelas usuarios, vendas e produtos.

```
DELIMITER //
```

```
CREATE EVENT atualizar_estatisticas  
ON SCHEDULE EVERY 1 DAY  
STARTS '2024-10-11 00:00:00'  
DO  
  BEGIN  
    UPDATE estatisticas  
    SET total_vendas = (SELECT COUNT(*) FROM vendas WHERE data_venda = CURDATE()),  
        total_clientes = (SELECT COUNT(DISTINCT cliente_id) FROM vendas WHERE data_venda = CURDATE());  
  END //
```

```
DELIMITER ;
```

# Mais exemplos

Job para Transferência  
de Dados Arquivados

O job transfere dados da tabela vendas com mais de 1 ano de antiguidade para a tabela vendas\_arquivo, mantendo a tabela principal mais leve.

```
DELIMITER //
```

```
CREATE EVENT mover_dados_antigos  
ON SCHEDULE EVERY 1 MONTH  
DO  
    BEGIN  
        INSERT INTO vendas_arquivo SELECT * FROM vendas WHERE data_venda < NOW() - INTERVAL 1 YEAR;  
        DELETE FROM vendas WHERE data_venda < NOW() - INTERVAL 1 YEAR;  
    END //
```

```
DELIMITER ;
```