

12 – Introdução

- Os primeiros banco de dados possuíam como objetivo principal apenas a correta armazenagem dos dados. Todas as regras de negócio eram implementadas na aplicação por meio de uma linguagem de programação qualquer.
- Atualmente, os principais sistemas gerenciadores de banco de dados, como o MySQL, PostgreSQL, Oracle, entre outros, passaram a contar com recursos para programação e criação de sub-rotinas, possibilitando que partes da regra de negócio possa ser implementada dentro do próprio banco de dados.
- Para implementação das sub-rotinas armazenadas, cada sistema gerenciador de banco de dados possui uma linguagem de programação interna que possibilita que os desenvolvedores implementem as rotinas.

12.1 – Introdução

- O Oracle Database foi o sistema gerenciador de banco de dados pioneiro no desenvolvimento de rotinas armazenadas, já implementando este recurso em meados de 1990. Hoje a sua linguagem procedural denominada **PL/SQL** é considerada a mais completa entre os BD's.
- O SQL Server da Microsoft também possui uma excelente linguagem para implementação de sub-rotinas no banco de dados. Ela é conhecida atualmente como **Transact-SQL**, sendo muito respeitada entre os desenvolvedores.
- O MySQL e o PostgreSQL começaram a ter recursos para implementação de sub-rotinas mais recentemente. O MySQL disponibilizou a linguagem **PL/MySQL** a partir de sua versão 5.0, no início dos anos 2000.

12.2 – Rotinas Armazenadas

- Os programas armazenados, também conhecido como rotinas ou módulos armazenados, são programas armazenados e executados dentro do próprio servidor de banco de dados.
- Quando o programa armazenado é executado, ele utiliza os recursos computacionais do próprio banco de dados (espaço de memória, processo e threads do banco de dados).
- Os três principais tipos de programas armazenados são:
 1. **Stored Procedure:** Rotina que aceita parâmetros de entrada e saída, capaz de manipular os dados armazenados no banco de dados. Para ser executada el precisa ser chamada.
 2. **Stored Function:** São semelhantes as stored procedures, mas o seu resultado de execução retorna um valor. Ela pode ser usada juntamente com as instruções SQL nativas do BD.
 3. **Triggers:** Rotinas ativadas em resposta a um evento ocorrido dentro do banco de dados. Normalmente, uma trigger será invocada em resposta a uma operação DML (INSERT, UPDATE, DELETE) feita em uma determinada tabela do banco de dados.

12.3 – Vantagens das Rotinas Armazenadas

- Existem uma infinidade de linguagens de programação que podem ser utilizadas pelos desenvolvedores. Entretanto, trazer parte da programação para dentro do banco de dados possui algumas vantagens, são elas:
 1. A utilização de programas armazenados pode deixar a aplicação mais segura, pois o fato de estar localizada no interior do banco de dados evita acesso sem autorização.
 2. Os programas armazenados oferecem mecanismos para abstrair as rotinas de acesso aos dados, podendo acessar diretamente (sem intermediários) as tabelas e estruturas do banco de dados.
 3. Os programas armazenados podem reduzir o tráfego na rede, visto que boa parte da execução dos dados será feita dentro do próprio servidor de banco de dados.

12.3 – Vantagens das Rotinas Armazenadas

4. Trazer parte da programação para dentro do banco de dados pode melhorar o desempenho global da aplicação. Isso porque o processamento dos dados será feito dentro do servidor banco de dados, sem a necessidade da aplicação se conectar ao BD diversas vezes.
5. Como as rotinas estão dentro do banco de dados, as solicitações dos usuários irão trafegar menos vezes na rede, reduzindo o tempo de resposta.
6. O uso de programas armazenados podem, em algumas circunstâncias, melhorar a portabilidade da aplicação. Como a programação está dentro do banco de dados, se torna mais fácil a troca da aplicação ou da linguagem de programação.
7. Programas armazenados podem ser usados por múltiplas aplicações, agilizando o processo de desenvolvimento.

12.4 – Problemas das Rotinas Armazenadas

- Embora a utilização de rotinas armazenadas possa oferecer vantagens, em algumas situações podem não ser a melhor alternativa, comprometendo o desempenho da aplicação. A seguir, serão listadas as principais desvantagens dos programas armazenados no banco de dados:
 1. Os programas armazenados são mais eficientes e rápidos do que as linguagens de programação quando se trata de acesso a registros armazenados no banco de dados, visto que o acesso é direto, sem a necessidade de comunicação por rede.

Mas quando se trata de cálculos matemáticos, manipulação de string, processamento de algoritmos complexos, entre outros, as linguagens de programação possuem melhor desempenho.
 2. Algumas aplicações podem ter a implementação de sua regra de negócio fragmentada em rotinas armazenadas no banco de dados e linguagem de programação. Essa fragmentação pode deixar o processo de manutenção mais complexo.

12.4 – Problemas das Rotinas Armazenadas

3. Embora a linguagem SQL seja padronizada, os programas armazenados possuem grandes diferenças na forma de implementação entre os diversos banco de dados. Desta forma, um procedimento feito no Oracle provavelmente não irá funcionar no SQL Server ou no Postgre.

Desta forma, quando a programação é feita no banco de dados, será difícil efetuar a troca do mesmo, visto que a aplicação terá que ser praticamente refeita.

4. As linguagens de programação, via de regra, ainda possuem mais recursos do que os oferecidos pelos banco de dados no quesito programação.

5. Toda a execução será feita no servidor de banco de dados, exigindo que o servidor seja eficiente e tenha recursos computacionais para executar todas as operações necessárias.

12.5 – Observações

- Trazer toda a regra de negócio para dentro do banco de dados certamente terá mais desvantagens do que vantagens.
- Entretanto, implementar algumas rotinas dentro do banco de dados pode melhorar de forma efetiva o desempenho da aplicação. Como exemplo, pode-se pensar em rotinas de atualização de estoque, geração de valor de comissão, etc.
- Geralmente, é vantajoso implementar no banco de dados rotinas que manipulam os dados armazenados nas tabelas.
- A implementação de rotinas armazenadas exige a utilização de uma ferramenta (IDE) gráfica. Trabalhar com sub-rotinas no terminal acaba deixando o processo de desenvolvimento mais complexo.