STRUCTURED QUERY LANGUAGE

A linguagem SQL surgiu em meados da década de 70, sendo resultado de um estudo de Edgar Frank Codd, membro do laboratório de pesquisa da IBM em San Jose, Califórnia. Este estudo tinha foco em desenvolver uma linguagem que adapta-se ao modelo relacional. O primeiro sistema de BD baseado em SQL tornou-se comercial no final dos anos 70 juntamente com outros sistema de BD's relacionais. O sucesso da linguagem SQL foi tão grande que obrigou o ANSI (American National Standarts Institute), a padronizar as implementações da linguagem, assim, nos dias de hoje, a maior parte de BD's seguem criteriosamente esta padronização, podendo ter algumas variações, mais mesmo assim não afetando na padronização global da linguagem tornando assim a portabilidade mais fácil, se seguida de forma adequada pelo DBA.

Já em meados da década de 80, a ANSI (American National Standard Institute) e a ISO (International Organization for Standardization), duas agências reguladoras, fizeram o trabalho de padronizar. A primeira normalização do SQL foi em 1986, com diversas atualizações seguintes, que a tornaram o que a gente conhece hoje como a ferramenta para definição e manipulação de BD's utilizada em grande parte dos BD existente, tais como MySQL, SQLServer, Firebird dentre outros. Saber SQL é fundamental para lidar com bancos de dados relacionais, já que as principais plataformas de bancos de dados dessa categoria a utilizam.

Quais as vantagens de usar SQL?

A maioria dos sistemas de bancos de dados usa SQL, é então fundamental a uma pessoa profissional de dados capacitada saber manejá-la. Oracle, MySQL, SQLServer, PostgreSQL, Informix, Microsoft Access são alguns dos sistemas de gerenciamento mais populares que a utilizam. Trata-se de uma linguagem estável, amplamente usada desde sua criação, lá na década de 70.

Além disso, exatamente por ser muito difundida, é essencial à pessoa profissional que pretende se diferenciar no mercado de trabalho. A SQL ainda se destaca, sobretudo, por ser fácil de aprender. Baseia-se em comandos simples, quase intuitivos. É uma linguagem vasta e completa, com a qual é possível definir, manipular, consultar, controlar e fazer transações no banco de dados.

Quais os tipos/subconjuntos da linguagem SQL?

A SQL possui cinco subconjuntos, responsáveis pelas operações de definição, manipulação, consulta, transação e controle. Veremos a seguir cada um deles.

DDL: linguagem de definição de dados

A DDL (Data Definition Language) engloba os comandos de definição do banco de dados. Interagem com os objetos do banco. São eles:

- **CREATE** - De modo geral, o comando Create cria objetos. Pode ser usado para criar desde novos bancos de dados completamente zerados a tabelas específicas. No exemplo, estamos criando uma tabela para os dados de funcionários.

  CREATE TABLE funcionários;

- **DROP -** O comando Drop exclui objetos do banco de dados. Essa remoção de tabelas envolve todas as linhas, privilégios e índices. Na aplicação, o drop não necessita de nenhuma cláusula adicional.

  DROP DATABASE empresa_matriz;

- **ALTER** – O comando Alter altera objetos já existentes, seja modificando, excluindo ou adicionando. No exemplo a seguir, alteramos a tabela estudantes excluindo a coluna nome. No mesmo caso, vemos também o uso do DROP.

  ALTER TABLE funcionarios

DROP COLUMN nome;

- **COMENT -** É uma função usada para fazer um comentário explicativo ou impedir a execução de uma linha de SQL pelo sistema. Esse princípio, a possibilidade de fazer comentários, é comum a outras linguagens de programação, representado por diferentes símbolos.
  Existem duas opções de COMMENT no SQL: simbolizado por dois traços (—), estando tudo posterior a eles na linha automaticamente anulado na execução do código, ou por meio da utilização de um asterisco e uma barra (/*), que necessita ser aberto antes do início do comentário, e fechado, após o fim deste. O primeiro, de dois traços, é usado para comentários de apenas uma linha. O segundo, o comentário multilinhas, para aqueles com duas ou mais. Vejamos exemplos:

  -- atencao para aniversario

  /* Dois funcionários comemoram aniversario em data diferente da que foram registrados */

  RENAME - É um comando bastante simples, sua função é renomear objetos. Na sintaxe, é preciso citar qual objeto sofrerá a alteração e qual será o novo nome.

  RENAME TABLE estudantes TO alunos;

DQL: linguagem de consulta de dados

A DQL (Data Query Language) é o subconjunto responsável por comandos de consulta aos dados armazenados. Dentro dele, encontramos apenas o comando Select. É importante observar que em alguns materiais acadêmicos essa instrução aparece incorporada no conjunto DML, que veremos mais a seguir**.**

SELECT - Esse comando é um dos mais importantes da SQL, pois é ele quem possibilita a consulta a dados de uma tabela. De modo geral, o Select recupera dados de determinado lugar. Os dados recuperados pelo Select são armazenados em uma nova tabela, chamada conjunto de resultados. É um comando que tem a possibilidade de ser estruturado de forma a fazer consultas mais simples ou mais complexas.

SELECT aniversario FROM estudantes;

DML: linguagem de manipulação de dados

A DML (Data Manipulation Language) corresponde aos comandos de manipulação dos dados. Composta de apenas três comandos, envolve interações de armazenamento, modificação, exclusão, inserção e atualização.

INSERT - Essa instrução insere dados a uma ou mais tabelas. Na sua estrutura, deve ser acompanhada de INTO. Vejamos um exemplo:

INSERT INTO estudantes (matricula, nome, aniversario) values (1776, Joana, 21 abr. 1987);

UPDATE - Atualiza os dados existentes em uma ou mais tabelas. Deve ser usado com a cláusula WHERE, para que se saiba em que linha será a atualização dos dados. Caso seja utilizado sem o WHERE, atualiza todos os registros. Como exemplo do segundo caso:

UPDATE estudantes;

DELETE - Exclui os registros de uma tabela ou mais. Quando não acompanhado de uma cláusula, todas as linhas são removidas.

DELETE FROM funcionários WHERE matricula = 1345;

DCL: linguagem de controle de dados

Esse subconjunto do SQL envolve comandos relacionados à segurança do banco de dados. A DCL (Data Control Language) controla o acesso aos dados, tanto concedendo privilégio de acesso, quanto retirando a permissão do usuário ou usuária.

GRANT - Fornece a determinada pessoa o privilégio de acesso dentro do banco de dados. No exemplo a seguir, estamos permitindo a Luiz consultar os dados da tabela estudantes.

GRANT SELECT ON estudantes TO Luiz;

REVOKE - Esse comando retira os privilégios de acesso. Ou seja, faz a operação inversa ao GRANT, negando a permissão. A seguir, vamos desfazer o que fizemos com GRANT. Para isso, utilizaremos o FROM.

REVOKE SELECT ON estudantes FROM Luiz;

DTL ou TCL: linguagem de transação de dados

A DTL ou TCL (Data Transaction Language) é um subconjunto do SQL para transação de dados. A DTL envolve gerenciamento e controle de transações.

BEGIN/SET TRANSACTION - Tanto o comando BEGIN TRANSACTION quanto o SET TRANSACTION indicam o início de uma transação. Devem ser usados imediatamente no começo do código, registrando que tudo que vem abaixo faz parte da mesma transação. A diferença entre BEGIN TRANSACTION e SET TRANSACTION está que na segunda pode-se atribuir especificações a respeito daquela transação, como, por exemplo, se será apenas para leitura.

COMMIT - Se a instrução BEGIN/ SET TRANSACTION inicia uma transação, a COMMIT a finaliza. O comando indica o fim de cada transação, salvando o que foi feito na transação atual. O COMMIT aparece no final daquela transação em específico, fechando o que foi aberto pelo BEGIN/SET TRANSACTION. A seguir, vejamos um exemplo de início e fim de uma transação, usando os comandos BEGIN TRANSACTION e COMMIT.

BEGIN TRANSACTION
DELETE FROM funcionários WHERE matricula = 1776;
COMMIT;

ROLLBACK - O comando Rollback reverte uma transação. Na prática, restaura o banco de dados desde a última vez que o comando COMMIT foi aplicado, garantindo apenas até onde as alterações já foram salvas. É um comando crucial, que pode ser usado em situações de erro.

SAVEPOINT - O SAVEPOINT define um ponto de salvamento dentro de uma transação, funciona como um "ponto de segurança". Tudo que é anterior a ele não pode ser descartado com o comando ROLLBACK, apenas o que vem após.

Vantagens da linguagem SQL

SQL tem muitas vantagens, o que o torna popular e altamente exigido. É uma linguagem confiável e eficiente usada para se comunicar com o banco de dados. Algumas vantagens do SQL são as seguintes:

Processamento de consulta mais rápido - Grande quantidade de dados é recuperada de forma rápida e eficiente. Operações como inserção, exclusão e manipulação de dados também são feitas em quase nenhum momento.

Sem habilidades de codificação - Para recuperação de dados, não é necessário um grande número de linhas de código. Todas as palavras-chave básicas como SELECT, INSERT INTO, UPDATE, etc são usadas e também as regras sintáticas não são complexas no SQL, o que o torna uma linguagem amigável.

Linguagem padronizada - devido à documentação e ao longo estabelecimento ao longo dos anos, fornece uma plataforma uniforme em todo o mundo para todos os seus usuários.

Portátil - Pode ser utilizado em programas em PCs, servidores, laptops independentes de qualquer plataforma (Sistema Operacional, etc). Além disso, ele pode ser incorporado a outros aplicativos de acordo com a necessidade / requisito / uso.

Linguagem interativa - Fácil de aprender e entender, as respostas a perguntas complexas podem ser recebidas em segundos.

Múltiplas visualizações de dados

Desvantagens do SQL

Embora o SQL tenha muitas vantagens, ainda existem algumas desvantagens.

Várias desvantagens do SQL são as seguintes:

Interface complexa - SQL tem uma interface difícil que deixa poucos usuários desconfortáveis ao lidar com o banco de dados.

Custo - algumas versões são caras e, portanto, os programadores não podem acessá-las.

Controle parcial - devido a regras de negócios ocultas, o controle completo não é dado ao banco de dados.

Conclusão

A linguagem SQL tem grande importância, pois com ela acessamos e manipulamos registros dentro um banco de dados. Armazenar e manipular informações de modo que elas possam ser usadas nas aplicações são habilidades relevantes para um programador. O uso da linguagem SQL ajuda na criação de diversas soluções, desde sistemas simples até os mais complexos.

Referencias Bibliográficas

- ELMASRI, R.; NAVATHE, S.B. Sistemas de Banco de Dados. 6ª Edição. São Paulo. Pearson. 2011.
- <https://docs.ufpr.br/~ademirlp/SQL.ppt>. Acessado em: 07 de Setembro de 2022.
- <https://dev.mysql.com/doc/refman/8.0/en/>. Acessado em: 07 de setembro de 2022.