

02

## História do SQL

### Transcrição

[00:00] Vamos começar?

[00:01] Vamos falar um pouco da história da linguagem SQL. Para você que está tendo o primeiro contato com esse tipo de linguagem, e por isso está fazendo esse treinamento, é bom saber como ela surgiu, o motivo, o histórico dela.

[00:20] O SQL foi desenvolvido originalmente no início dos anos 70, nos laboratórios da IBM, na cidade de San Jose, na Califórnia, dentro de um projeto chamado System R, que tinha como objetivo demonstrar a viabilidade da implementação de um modelo relacional que estava sendo desenvolvido por um estudioso chamado Codd. O Codd imaginou uma forma estruturada de fazer consultas nos bancos de dados que estavam surgindo, que eram chamados de bancos de dados relacionais.

[01:03] Até aquele momento, os bancos de dados não tinham relacionamento entre as tabelas onde os dados eram armazenados. Esses bancos de dados antigos, os primeiros que surgiram, eram chamados de bancos de dados sequenciais, que possuem uma tripla de caracteres de textos que não permitia algum tipo de relacionamento entre eles.

[01:30] Com o surgimento desses bancos de dados relacionais, que se chamavam DBMSs, o Codd pensou em criar uma linguagem que fosse fácil para extrair dados, ou manipular não somente dados, mas as estruturas desse banco, aproveitando a característica de relacionamento entre eles.

[01:56] Mas não era só a IBM que trabalhava com isso. Já mais para o meio dos anos 80, a Oracle também estava procurando uma maneira fácil de manipular essas novas estruturas relacionais. E não somente ela, outras empresas estavam começando a fazer isso.

[02:14] Estamos falando aí de meio para o final dos anos 80. No final dos anos 80 e início dos 90, um órgão americano chamado ANSI, que vem da sigla American National Standard Institute, estipulou alguns padrões para as consultas dos bancos de dados relacionais.

[02:37] Foi criada então uma linguagem chamada SEQUEL, que vem do acrônico em inglês para Structured English Query Language, que traduzindo seria alguma coisa como linguagem de consulta estruturada em inglês.

[03:08] No inglês, geralmente não pronunciamos SQL, que seriam as letras SQL em inglês. Mas normalmente falamos em inglês que sabemos a linguagem SEQUEL. No português não. No português traduzimos para SQL.

[03:32] A linguagem SQL veio com o objetivo de padronizar a forma como os dados são consultados nesses bancos de dados relacionais. Hoje em dia, todos os bancos que se dizem relacionais adotam o padrão SEQUEL, mas esse padrão não é somente usado para consulta de dados. É utilizado também para criação, alteração, estrutura, manipulação da estrutura do banco de dados, forma como o banco de dados interage com a segurança e assim por diante.

[04:15] Podemos falar que existem algumas grandes vantagens do banco de dados relacional, e mais precisamente dessa padronização, usando a linguagem SEQUEL. A primeira é o aprendizado. Essa padronização de linguagem traz algumas vantagens. A primeira é que há um custo reduzido do aprendizado.

[04:45] O profissional que sabe SQL, por exemplo, de Oracle, facilmente vai utilizar o SQL de uma SQL, ou do SQL server da Microsoft. Algumas pequenas diferenças existem entre eles, principalmente na parte de funções, mas isso não é um grande problema para que um profissional se adapte facilmente de um banco de dados para outro.

[05:12] Outra vantagem é a portabilidade. Fica muito fácil você migrar sistemas que utilizam Oracle para SQL Server, ou para MYSQL, ou vice versa. Claro que se você é um programador e está desenvolvendo uma aplicação que consulta banco de dados relacional, quanto mais próximo do ANSI, ou seja, quanto mais você utiliza o SQL standard definido pelo ANSI, fica muito mais fácil depois fazer uma portabilidade. Evitar, por exemplo, aquelas funções específicas do banco e deixar isso para o seu programa fazer.

[05:57] Longevidade. Se o seu sistema utiliza relatórios ou processos utilizando o SQL, temos uma garantia de que eles vão funcionar durante muito tempo, principalmente para as novas versões de banco de dados, porque vão estar sempre adaptados ao padrão ANSI. Seu sistema nunca vai ficar fora de funcionamento quando você for fazer um upgrade de banco de dados, por exemplo.

[06:33] Comunicação. O fato de todo mundo falar SQL permite que os sistemas possam facilmente se comunicar entre si. Processos de ETL ou de integração entre sistemas ficam muito mais fáceis de serem desenvolvidos, já que os dois falam o SQL padrão.

[06:56] Liberdade de escolha. Como há um padrão de linguagem, quando a empresa for selecionar o uso de um banco de dados relacional não vai ficar presa, por exemplo, à linguagem de comunicação, já que ela é muito parecida. A empresa, ao decidir o uso de um banco de dados relacional vai usar outros critérios de escolha, como por exemplo performance, hardware, custo, e assim por diante.

[07:30] Só que essa padronização também não tem só vantagens. Ela tem algumas desvantagens. Poucas, mas tem. A primeira é a falta de criatividade. O SQL possui alguns limites que podem não atender, por exemplo, as novas demandas de mercado. Principalmente com o advento das redes sociais e de grande volume de dados, chamado de big data. Ou seja, a necessidade de coletar dados que estão trafegando na internet.

[08:05] Tanto é assim que hoje estão surgindo outros bancos que utilizam padrões diferentes dos bancos de dados relacionais, que costumamos chamar de NOSEQUEL. Esses bancos NOSQL atendem melhor às demandas, por exemplo, de redes sociais, de tabelas de big data. Mas estamos falando de estruturas que fogem do padrão ANSI e que exigem um aprendizado muito focado.

[08:47] Outra coisa é a falta de estruturação da linguagem SQL. Ela não é uma linguagem estruturada, que possui ifs, whens, fors. Ou seja, comandos condicionais, como uma linguagem de programação normal.

[09:02] Para suprir essa falta de estruturação, os bancos de dados relacionais, Oracle, SQL, MYSQL criaram suas próprias linguagens internas que fazem esse tipo de estruturação usando a linguagem SQL. Mas vamos dizer que isso está fugindo um pouco do padrão ANSI.

[09:24] Esse padrão ANSI possui três grandes grupos de comandos. O primeiro deles são os DDLs, que vem de "linguagem de definição de dados". É a parte da linguagem SQL que permite a manipulação das estruturas do banco de dados. Criar um banco, tabelas, índice, apagar tabela, alterar índice, alterar política de crescimento de índice. Todos os comandos que mexem com a estrutura do banco de dados relacional são bancos do tipo DDL.

[10:13] O segundo grupo de comandos é chamado DML, que traduzindo para o português seria "linguagem de manipulação de dados". O DML é o grupo de comandos que tem como objetivo realmente gerenciar os dados. Incluir, alterar, excluir informações que estão dentro das estruturas do banco, como por exemplo as tabelas. Não somente fazer isso, mas também fazer consultas. Tirar informações das estruturas e exibir para os usuários.

[11:01] Finalmente, temos os comandos DCL, que traduzindo seria "linguagem de controle de dados". O DCL é um grupo de comandos que nos permite administrar o banco de dados, mas não a sua estrutura. Na administração do controle de acesso, gerenciar usuário, gerenciar o que cada usuário pode ou não ver. Gerenciar o banco a nível de estrutura, como vai ser a política de crescimento dele, como ele vai ser armazenado no disco, onde ele vai ser armazenado. Administrar os processos,

saber quantos processos estão sendo executados, e assim por diante. Controle de logs, por exemplo, também está dentro do conjunto de linguagens DCL.

[11:53] É isso. Quis dar nesse vídeo um apanhado geral. Não somente da história do SQL ou do SQL, e as características, vantagens e desvantagens e características dos comandos nessa linguagem.