

SQL – APOSTILA

INTRODUÇÃO

Uma *linguagem de consulta* é a linguagem por meio da qual os usuários obtêm informações do banco de dados. Essas linguagens são, tipicamente, de nível mais alto que as linguagens de programação tradicionais. As linguagens de consulta podem ser categorizadas como procedurais ou não-procedurais. Em uma *linguagem procedural*, o usuário deve "ensinar" ao sistema a realização de uma seqüência de operações no banco de dados para obter o resultado desejado. Em uma linguagem *não-procedural*, o usuário descreve a informação desejada sem fornecer um procedimento específico para a obtenção dessas informações.

Os sistemas de banco de dados comerciais oferecem uma linguagem de consulta que incorpora elementos de ambos os enfoques: procedurais e não-procedurais.

Linguagens "puras": a álgebra relacional é procedural, enquanto o cálculo relacional de uma tupla e o cálculo relacional de um domínio são não-procedurais. Essas linguagens de consulta são concisas e formais, sem "o açúcar sintático" das linguagens comerciais, mas ilustram as técnicas fundamentais para a extração de dados do banco de dados.

A LINGUAGEM SQL

O nome "SQL" significa "Structured Query Language" - Linguagem Estruturada de Pesquisa. Essa linguagem, de grande utilização, teve seus fundamentos no modelo relacional de Codd (1970). Sua primeira versão recebeu o nome de SEQUEL ("Structured English Query Language"), sendo definida por D. D. CHAMBERLIN, entre outros, em 1974, nos laboratórios de pesquisa da IBM (Califórnia). Em 1975, foi implementado um protótipo de aplicação dessa nova linguagem. Entre 1976 e 1977, o SEQUEL foi revisado e ampliado, e teve seu nome alterado para "SQL" por razões jurídicas.

Com esta revisão foi posto em prática um projeto ambicioso da IBM chamado System R. Novas alterações foram introduzidas na SQL, graças às idéias apresentadas pelos diversos usuários do ambiente.

Devido ao sucesso dessa nova forma de consulta e manipulação de dados, dentro de um ambiente de banco de dados, a utilização da SQL foi se tornando cada vez maior. Com isso uma grande quantidade de SGBD's foi tendo como linguagem básica a SQL - SQL/DS e DB2 da IBM, ORACLE da Oracle Corporation, RDB da Digital, SYBASE da Sybase INC, e Microsoft® SQL Server™, entre outros.

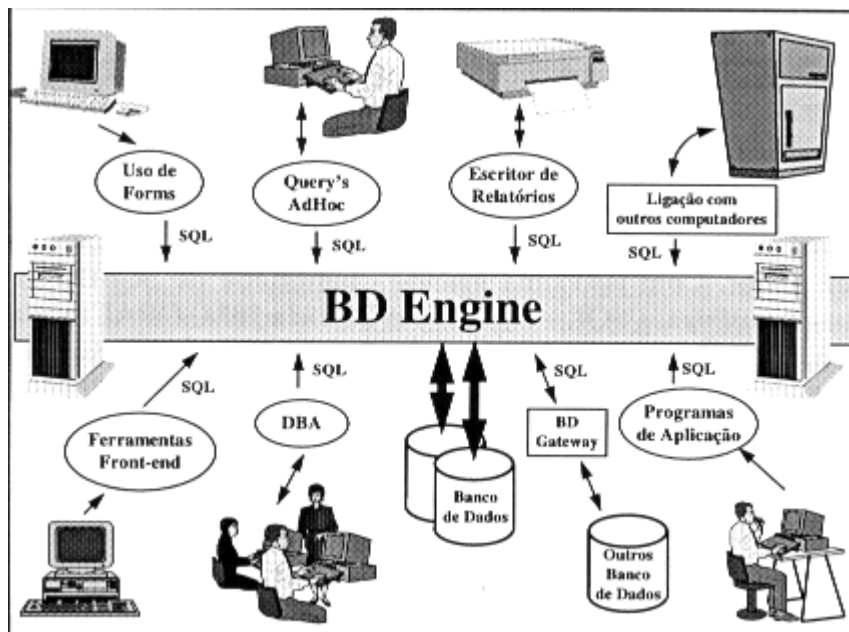
A SQL se tornou um padrão de fato, no mundo dos ambientes de banco de dados relacionais. Bastava agora se tornar de direito. Então, em 1982, o American National Standard Institute (ANSI) tornou a SQL padrão oficial de linguagem em ambiente relacional.

Infelizmente, como todo padrão que se preze, existem hoje vários dialetos SQL, cada um, evidentemente, tentando ser mais padronizado que o outro.

O modelo relacional é constituído basicamente de tabelas, cada qual contendo linhas (registros, tuplas) e colunas. Os registros na tabela não são ordenados e sua localização se faz por meio de um campo-chave, ou seja, um campo que assume o papel de chave primária da tabela. É por intermédio dessa chave que se identifica uma, e somente uma, ocorrência do valor contido no campo.

Uma das razões da popularidade dos sistemas relacionais é a sua facilidade de manipulação e entendimento.

A linguagem SQL foi desenvolvida especialmente para o ambiente relacional, podendo ser adaptada a qualquer ambiente não relacional.

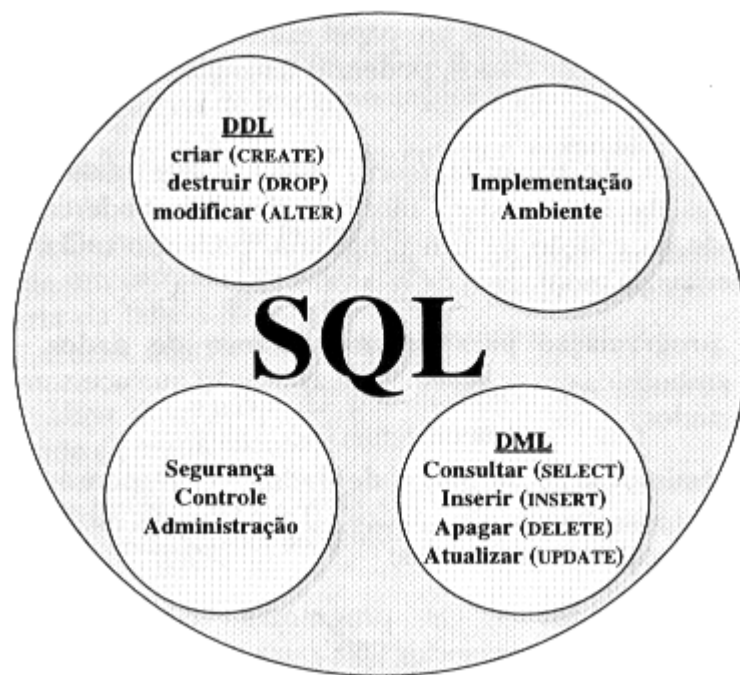


A idéia original da SQL só previa seu uso de forma interativa. Após sofrer alguns acréscimos, ela passou também a ter capacidade de ser utilizada em linguagens hospedeiras, tais como: COBOL, FORTRAN, "C", etc.

Atualmente, a linguagem SQL assume um papel muito importante nos sistemas de gerenciamento de banco de dados, podendo ter muitos enfoques, como apresenta a figura:

- **Linguagem interativa de consulta (*query AdHoc*)**- Por meio de comandos SQL, os usuários podem montar consultas poderosas sem a necessidade da criação de um programa, podendo utilizar *Forms* ou ferramentas de montagem de relatório;
- **Linguagem de programação para acesso a banco de dados** - Comandos SQL embutidos em programas de aplicação que acessam os dados armazenados;
- **Linguagem de administração de banco de dados** - O responsável pela administração do banco de dados (DBA) pode utilizar comandos SQL para realizar suas tarefas;
- **Linguagem cliente/servidor** - Os programas (cliente) dos computadores pessoais usam comandos SQL para se comunicarem por meio de uma rede local, compartilhando os dados armazenados em um único local (servidor). A arquitetura cliente/servidor minimiza o tráfego de dados pela rede;
- **Linguagem para banco de dados distribuído** - A SQL auxilia na distribuição dos dados por meio de vários nós conectados ao sistema de computação. Auxilia também na comunicação de dados com outros sistemas;
- **Caminho de acesso a outros bancos de dados em diferentes máquinas** - A SQL auxilia na conversão entre diferentes produtos de banco de dados colocados em diferentes máquinas (de micro até mainframe).

Por ser uma linguagem de numerosas aplicações, a SQL pode manipular objetos de diferentes classes entre as funções de um SGBD:



- **Definição de dados (DDL)** - permite ao usuário a definição da estrutura e organização dos dados armazenados, e as relações que existem entre eles;
- **Manipulação de dados (DML)** - permite ao usuário ou a um programa de aplicação, a inclusão, remoção, seleção ou atualização de dados previamente armazenados no banco;
- **Controle de acesso** - protege os dados de manipulações não **autorizadas**;
- **Compartilhamento de dados** - coordena o compartilhamento dos dados por usuários concorrentes, sem contudo interferir na ação de cada um deles;
- **Integridade dos dados** - auxilia no processo de definição da integridade dos dados, protegendo contra corrupções, inconsistências e falhas do sistema de computação

Vantagens e Desvantagens da linguagem SQL

Com o uso e a padronização da SQL, algumas vantagens são diretas:

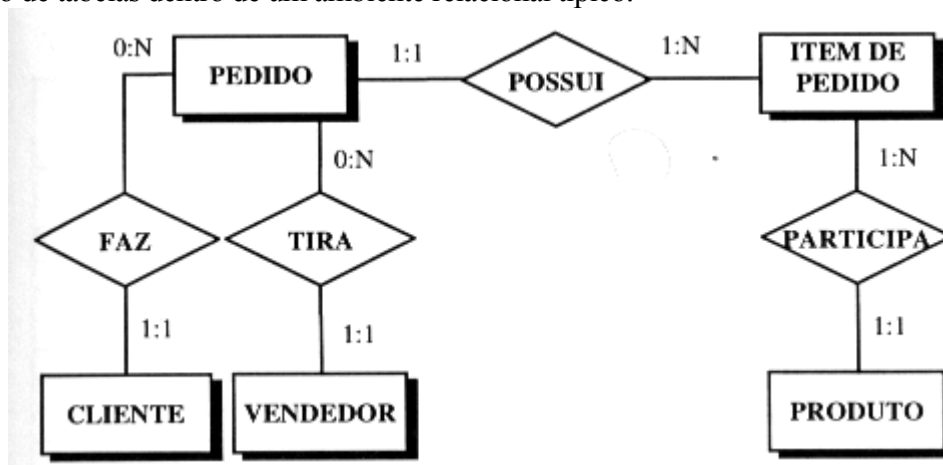
- **Independência de fabricante** - A SQL é oferecida em praticamente todos os SGBD's, e os que ainda não têm estão se encaminhando para lá. Com isso posso mudar de SGBD sem me preocupar com o novo que vai chegar;
- **Portabilidade entre computadores** - A SQL pode ser utilizada desde um computador pessoal, passando por uma estação de trabalho, até um computador de grande porte;
- **Redução dos custos com treinamento** - Baseado no item anterior, as aplicações podem se movimentar de um ambiente para o outro sem que seja necessária uma reciclagem da equipe de desenvolvimento;
- **Inglês estruturado de alto nível** - A SQL é formada por um conjunto bem simples de sentenças em inglês, oferecendo um rápido e fácil entendimento;
- **Consulta interativa** - A SQL provê um acesso rápido aos dados, fornecendo respostas ao usuário, a questões complexas, em minutos ou segundos;

- **Múltiplas visões dos dados** - A SQL permite ao criador do banco de dados levar diferentes visões dos dados a diferentes usuários;
- **Definição dinâmica dos dados** - Por meio da SQL, podem-se alterar, expandir ou incluir, dinamicamente, as estruturas dos dados armazenados com a máxima flexibilidade;

Apesar de todas essas vantagens, algumas críticas são dirigidas à SQL:

- A padronização leva a uma, natural, inibição da criatividade, pois quem desenvolve aplicações fica preso a soluções padronizadas, não podendo sofrer melhorias ou alterações;

Mesmo enfrentando alguns problemas e críticas, a linguagem SQL veio para ficar, auxiliando de forma bastante profunda a vida dos usuários e analistas no trabalho de manipulação dos dados armazenados em um banco de dados relacional. E é sobre esse auxílio que iremos tratar, mostrando comandos e funcionalidades da SQL, por meio de exemplos práticos. Não iremos mostrar todos os comandos, principalmente os que foram definidos para serem utilizados dentro de uma linguagem hospedeira (cursor); apresentaremos os comandos para criação, atualização, alteração, pesquisa e eliminação de tabelas dentro de um ambiente relacional típico.



→ Tabelas referentes ao modelo de dados;

TABELA VENDEDOR

Código do vendedor	Nome do vendedor	Salário Fixo	Faixa de Comissão
209	José	1.800,00	C
111	Carlos	2.490,00	A
11	João	2.780,00	C
240	Antônio	9.500,00	C
720	Felipe	4.600,00	A
213	Jonas	2.300,00	A
101	João	2.650,00	C
310	Josias	870,00	B
250	Maurício	2.930,00	B

TABELA CLIENTE

Cód. do cliente	Nome do cliente	Endereço	Cidade	CEP	UF	CGC	IE
720	Ana	Rua 17 n. 19	Niterói	24358310	RJ	12113231/0001-34	2134
870	Flávio	Av. Pres. Vargas 10	São Paulo	22763931	SP	22534126/9387-9	4631
110	Jorge	Rua Caiapo 13	Curitiba	30078500	PR	14512764/9834-9	
222	Lúcia	Rua Itabira 123 Loja 9	Belo Horizonte	22124391	MG	28315213/9348-8	2985
830	Maurício	Av. Paulista 1236 sl/2345	São Paulo	3012683	SP	32816985/7465-6	9343
130	Edmar	Rua da Praia sn/	Salvador	30079300	BA	23463284/234-9	7121
410	Rodolfo	Largo da Lapa 27 sobrado	Rio de Janeiro	30078900	RJ	12835128/2346-9	7431
20	Beth	Av. Climério n. 45	São Paulo	25679300	SP	32485126/7326-8	9280
157	Paulo	Tv. Moraes c/3	Londrina		PR	32848223/324-2	1923
180	Lívio	Av. Beira Mar n. 1256	Florianópolis	30077500	SC	12736571/2347-4	
260	Susana	Rua Lopes Mendes 12	Niterói	30046500	RJ	21763571/232-9	2530
290	Renato	Rua Meireles n. 123 bl.2 sl.345	São Paulo	30225900	SP	13276571/1231-4	1820
390	Sebastião	Rua da Igreja n. 10	Uberaba	30438700	MG	32176547/213-3	9071
234	José	Quadra 3 bl. 3 sl. 1003	Brasília	22841650	DF	21763576/1232-3	2931

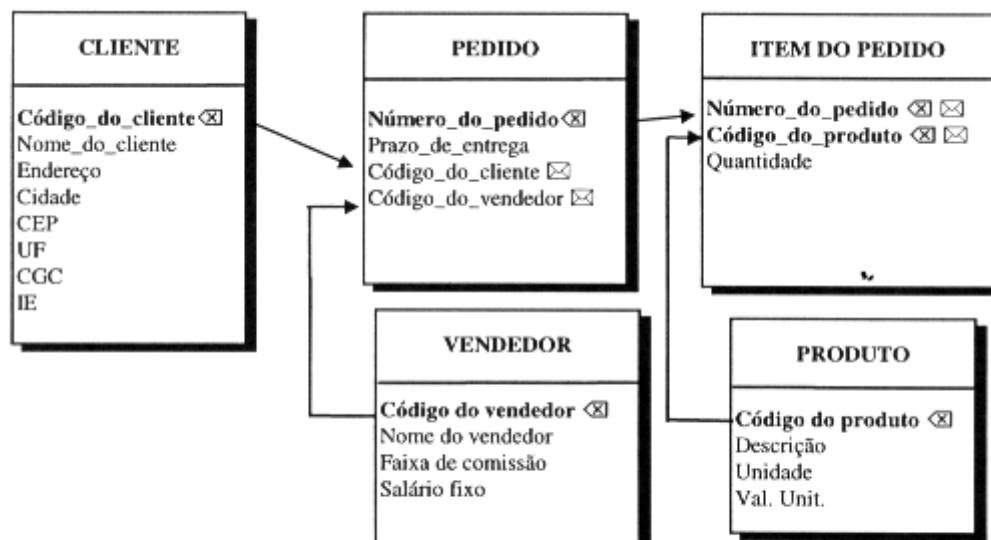
TABELA PEDIDO

Número do Pedido	Prazo de Entrega	Código do Cliente	Código do Vendedor
121	20	410	209
97	20	720	101
101	15	720	101
137	20	720	720
148	20	720	101
189	15	870	213
104	30	110	101
203	30	830	250
98	20	410	209
143	30	20	111
105	15	180	240
111	20	260	240
103	20	260	11
91	20	260	11
138	20	260	11
108	15	290	310
119	30	390	250
127	10	410	11

TABELA ITEM_DO_PEDIDO

Número do pedido	Código do produto	Quantidade
121	25	10
121	31	35
97	77	20
101	31	9

VISÃO GRÁFICA DO PEDIDO



☒ : Chave primária;

☒ : Chave estrangeira.