

Como aceder a bases de dados recorrendo à linguagem SQL ?

A linguagem SQL descreve-se pelos seguintes 10 pontos gerais:

1º;-SQL define-se por Linguagem de Consulta Estruturada (em inglês mais conhecida por Structured Query Language...) 2º;-SQL permite-lhe aceder a uma base de dados 3º;-SQL representa um padrão ANSI (em inglês American National Standards Institute) como linguagem computacional 4º;-SQL pode executar queries sobre uma base de dados 5º;-SQL pode retirar informação de uma base de dados 6º;-SQL pode inserir novos registos de informação numa base de dados 7º;-SQL pode apagar registos de informação numa base de dados 8º;-SQL pode actualizar registos de informação numa base de dados 9º;-SQL é uma linguagem bastante divulgada e conhecida 10º;-e mais importante : SQL é divertido e fácil de aprender!

A linguagem SQL corresponde a um padrão de linguagem computacional ANSI (organização particular americana, sem fins lucrativos, que tem por objectivo facilitar a padronização dos trabalhos dos seus membros) e que permite aceder e manipular Sistemas de Bases de Dados Relacionais - uma Base de Dados Relacional é um conceito abstracto que define maneiras de armazenar, manipular e recuperar dados estruturados unicamente na forma de tabelas, construindo assim uma base de dados.

No mercado informático de bases de dados, existem muitos exemplos de programas que actualmente usam esta linguagem, uns comerciais, outros código aberto, o que faz realçar a sua clara importância nesta área. Destacam-se por exemplo: Microsoft Access (ferramenta presente no pacote Microsoft Office, quem não conhece hoje em dia ?...)

DB2

Microsoft SQL Server

Oracle

SQLite

MySQL

etc ...

Agora, mencionei bases de dados... o que são realmente bases de dados? Qual a sua relação com a linguagem SQL ?

Uma Base de Dados representa um conjunto de dados com uma estrutura regular e organizados em aquilo que se designa por informação. Normalmente agrupa informações utilizadas para o mesmo fim. É mantida e gerida através de um pacote de software (programa) a que se dá o nome de Sistema Gestor de Bases de Dados - SGBD (em inglês Database Management System - DBMS).

Mas...e porquê usarmos bases de dados? Qual a sua utilidade ? Desde tempos imemoriais, o homem guardou em armários, gavetas (...) os dados importantes relativos às suas diversas actividades - as primeiras bases de dados! A partir dos meados do século XX, verificou-se um grande crescimento das organizações, o que provocou um aumento considerável no volume de dados nelas utilizados. Ao mesmo tempo, os computadores desenvolveram-se e tornaram-se mais acessíveis. Então, surge assim de forma natural, a introdução dos computadores nas organizações com o objectivo de servir de suporte às bases de dados nelas existentes. Numa 1ª fase surgem os sistemas de ficheiros e só depois são desenvolvidos os sistemas de bases de dados. Assim, com ambos se pretende obter acesso rápido aos dados, gerando informação necessária diversa às várias actividades. Os dados registam-se em ficheiros e a partir deles são gerados relatórios com a informação pretendida. Hoje em dia, quando nos referimos a bases de dados, queremos referir-nos às bases de dados em suporte informático. Podemos encontrar inúmeros exemplos de aplicações de bases de dados:
A sua agenda telefónica de casa

As Páginas Amarelas

A lista de colaboradores registada em Microsoft Access

Bases de dados de clientes de bancos, seguradoras

Base de dados dos CTT

Sites de comércio electrónico - e-Business

Aplicações CRM (Customer Relationship Management)

Aplicações ERP (Enterprise Resource Management)

Você, que está a ler este artigo, lembra-se de mais algum exemplo possível de base de dados ?... Mas não basta só ter uma base de dados. É preciso haver algo que a controle - SGBD : é um pacote de software que se destina a armazenar e gerir bases de dados. As bases de dados representam uma colecção de dados que se relacionam entre si. Porque se usa um sistema SGBD ? Independência de dados

Acesso eficiente

Tempo de desenvolvimento de aplicações reduzido; manutenção reduzida

Integridade e Segurança de Dados

Administração de dados - controlo uniforme de segurança, integridade e privacidade

Acesso concorrente; recuperação de falhas

Um sistema SGBD disponibiliza linguagens de: Definição de dados: para criação e alteração da estrutura da base de dados (DDL - Data Definition Language). Consulta de dados: obter e processar os dados armazenados (DQL - Data Query Language). Manipulação de dados: para acrescentar dados novos e modificar dados existentes (DML - Data Manipulation Language). Para além disso, um sistema SGBD deve disponibilizar ainda como funcionalidades:

Modelos que definam o esquema dos dados armazenados no sistema: os quatro modelos mais conhecidos são: hierárquico, em rede, relacional, orientado a objectos. Estruturas de dados optimizadas, que possam manipular grandes quantidades de informação. Um mecanismo transaccional que garanta a consistência entre as operações dos dados armazenados. No mercado existem vários sistemas SGBDs, uns são código aberto (acesso livre mediante regras) como: MySQL e

PostgreSQL, etc outros comerciais (pagos) como: Oracle e

Microsoft SQL Server, etc... mas sem dúvida os mais utilizados são os SGBDs Relacionais, que permitem criar Bases de Dados Relacionais e o exemplo mais conhecido que se pode descobrir é o SGBD Microsoft Access, que faz parte do produto Microsoft Office!

Uma das funcionalidades de um sistema SGBD, como vimos, é disponibilizar modelos de dados. O que significam? O que são Modelos de Dados neste âmbito?

A Modelação de Dados é a actividade que, debruçando-se sobre a totalidade dos requisitos de informação de um sistema de informação, tenta encontrar um modelo que traduza a estrutura lógica dos dados que satisfaz esses requisitos. Por outras palavras mais simples: são esquemas abstractos de representação de informação, que representam as ligações entre os componentes que formam a base de dados – por exemplo são usados como meio para desenhar uma base de dados. Diversos modelos foram definidos ao longo do tempo, mas podemos realçar os 4 principais: Hierárquico, Redes,

Relacional (bastante usado actualmente!)

Orientado a Objectos. No entanto, outros ainda se podem destacar: Entidades e Relacionamentos, Lista invertida,

Relacional Estendido e Semi-Estruturado

Sendo o Modelo Relacional um dos mais usados do mercado, vamos analisá-lo um pouco:

Relacional – este tipo de modelo organiza os dados em uma estrutura de tabelas (relação). Para aceder aos dados nas tabelas utiliza-se uma linguagem declarativa (não procedural), a linguagem utilizada é SQL (um padrão). Para navegar é necessário conhecer a representação física da base de dados. A abordagem relacional separa a aplicação da implementação física da base de dados. Com isso, mudam as na representação física dos dados não causa impacto na aplicação, isto é importante.

Consultar este link para se ver um exemplo de um esquema relacional:

http://pedromesquita.alojamentogratico.com/geral.html?titulo=Exemplo_Relacional&url=http://pedromesquita.alojamentogratico.com/outrosmodulos/refs_externas/artigos/exemplo_relacional.jpg

Depois do Modelo Relacional ter sido definido, um número de linguagens foram desenvolvidas e usadas para implementar e gerir o SGBDR, nomeadamente a linguagem SQL. Esta linguagem está relacionada com o Modelo Relacional que é impossível discuti-la sem discutir o modelo! Sabemos então que uma base de dados relacional armazena dados em tabelas. Cada uma das colunas da tabela representa um atributo – por exemplo, para a tabela Autor temos: código autor, nome, telefone, etc. Cada linha de dados na tabela representa um registo. Uma única tabela com uma estrutura de colunas e registos, não representa por si uma base de dados relacional. As bases de dados relacionais possuem várias tabelas interligadas entre si, isto é um facto e o que acontece na realidade! As Chaves Primárias e Chaves Estrangeiras são 2 conceitos muito importantes em bases de dados relacionais! São estes 2 conceitos que permitem ligar tabelas relacionarem-se umas com as outras: - a chave primária é representada por uma ou mais colunas e identifica nessa tabela, a que pertence, univocamente um registo, ou seja, numa tabela relacional não pode haver dois registos com a mesma chave primária (por exemplo o nome de uma pessoa é uma chave primária pois é possível encontrar dois nomes iguais, mas o código de BI não é); - a chave estrangeira é representada por uma coluna que por sua vez referencia uma chave primária em uma outra tabela, daí o nome "chave estrangeira"... Voltando de novo à linguagem SQL...

A linguagem SQL é uma linguagem de bases de dados relacional. Por si só, não faz de uma base de dados um sistema SGBD! É apenas um meio de comunicação com o sistema SGBD para se poder consultar e modificar informação. Como linguagem, é formada por conjuntos de comandos escritos em inglês (devido a razões históricas e de universalidade de compreensão) que permitem de modo geral: consultar, inserir, actualizar e

apagar dados. Agora isto não significa que se possa escrever em texto livre uma operação – como por exemplo “Pull up the figures for last month sales" (“dá-me os números de vendas do último mês”)- a executar e esperar que o interpretador SQL entenda ! O que isto significa é que a linguagem SQL é uma linguagem mais fácil de aprender e perceber do que as outras linguagens computacionais conhecidas. A Linguagem SQL é muitas vezes referida como uma linguagem de bases de dados não-procedural. Isto significa na prática que quando executamos um comando de consulta da base de dados, não temos que indicar explicitamente ao SQL onde procurar os dados. É apenas suficiente “dizer” ao SQL quais os dados que queremos consultar. O sistema SGBD, esse sim, é responsável por localizar a informação requerida na base de dados, o que é muito útil para os utilizadores por estes não se terem de preocupar onde e como obter os dados. Linguagens de programação do tipo procedural, como o COBOL ou mesmo PASCAL, baseados nos modelos de dados Redes e Hierárquico, já obrigam os utilizadores a realizar isso!

Como jà vimos, uma Base de Dados Relacional contêm 1 ou mais tabelas SQL.

Uma Base de Dados SQL é uma Base de Dados Relacional. E como vimos também, cada tabela identifica-se por um nome e por um conjunto de campos a que se designam

‘colunas’ da tabela. Cada um destes campos tem um tipo associado : inteiro, decimal, booleano (verdadeiro ou falso), cadeia de caracteres (string), caracter, etc. As tabelas contêm dados, também designados por registos ou linhas de dados. A linguagem

SQL como DML – Data Manipulation Language (em português: Linguagem de Manipulação de Dados) :

A DML é um subconjunto da linguagem usado para selecionar (SELECT), inserir (INSERT), atualizar (UPDATE) e apagar (DELETE) dados. Destacam-se então os comandos: SELECT é o comando mais usado, permite efectuar consultas à base de dados através de queries. INSERT permite adicionar informação nova à base de dados, ou seja, permite inserir novos registos de informação. UPDATE permite actualizar informação na base de dados, ou seja, permite actualizar um (ou mais) registo de informação. DELETE permite apagar registos de informação da base de dados.

E ainda ...

BEGIN WORK (ou START TRANSACTION, dependendo da extensão SQL) pode ser usado para marcar o íncio de uma transação de base de dados que pode ser terminada ou não. COMMIT termina uma transacção com sucesso – todas as alterações feitas na base de dados são feitas! ROLLBACK cancela uma transacção – todas as alterações feitas na base de dados são canceladas e ignoradas! COMMIT e ROLLBACK interagem com ´reas de controlo como transacção e locação. Ambos terminam qualquer transacção aberta e libertam qualquer recurso ligado a dados. Na ausência de um BEGIN WORK ou uma declaração semelhante, a semântica de SQL é dependente da implementação.

A linguagem SQL como DDL – Data Definition Language (em português Linguagem de Definição de Dados) :

A DDL permite definir tabelas novas e elementos associados. A maioria das bases de dados de SQL comerciais tem extensões próprias no DDL. Destacam-se entre os comandos: `CREATE` cria um objecto (uma tabela por exemplo) dentro da base de dados. `DROP` apaga um objecto na base de dados. Alguns sistemas de base de dados usam o comando `ALTER`, que permite alterar um objecto, por exemplo, adicionando uma coluna a uma tabela existente.

[illegible]

 ALTER INDEX permite alterar a estrutura de um índice. DROP INDEX permite remover a estrutura de um índice.

 CREATE VIEW permite criar uma vista – uma vista é uma tabela virtual baseada no resultado de pesquisa de um SELECT.

[illegible]

como DDL , exemplos de comandos : CREATE TABLE Cliente

```
(  
    Nome char(50),  
    Morada char(50),  
    Cidade char(50),  
    Pais char(25),  
    Data_Nascimento date
```

); - cria (CREATE) uma tabela de nome 'Cliente' com os seguintes campos: Nome (tipo cadeia de caracteres, 50 caracteres), Morada (tipo cadeia de caracteres, 50 caracteres), Cidade (tipo cadeia de caracteres, 50 caracteres), Pais (tipo cadeia de caracteres, 25 caracteres) e Data_Nascimento (data). DROP TABLE Cliente - remove/apaga (DROP) a tabela de nome 'Cliente' da base de dados.

A linguagem SQL como DCL - Data Control Language (em português: Linguagem de Controlo de Dados) :

A DCL controla os aspectos de autorização de dados e licenças para controlar quem tem acesso para consultar ou manipular dados na base de dados, envolve portanto questões de segurança.

Destacam-se os comandos principais: GRANT autoriza o utilizador a realizar determinada operação: consulta, actualização, etc... REVOKE realiza o inverso do comando GRANT, ou seja, retira a autorização. A linguagem SQL como DQL - Data Query Language (em português: Linguagem de Consulta de Dados) : A DQL foi aqui analisada e diz respeito ao comando que permite extrair dados da base de dados, ou seja, o comando SELECT.

Exemplos: SELECT * FROM autores - selecciona todos os campos de todos os registos da tabela autores. SELECT nome, email, telefone, estado FROM autores - selecciona os campos nome, email, telefone e estado de todos os registos da tabela autores. SELECT nome, email, estado FROM autores WHERE estado = 'SP' - selecciona os campos nome, email e estado apenas dos registos cujo estado é SP. SELECT nome, email, estado, idade FROM autores WHERE idade = '23' AND estado = 'SP' - selecciona os campos nome, email, estado e idade apenas dos registos cuja idade é 23 e o estado é SP. SELECT nome, email, estado, idade FROM autores WHERE estado = 'SP' OR estado = 'AM' - selecciona os campos nome, email, estado e idade apenas dos autores onde o estado é SP ou AM.

A linguagem SQL como DCL , exemplos de comandos :

GRANT SELECT ON TABLE dept TO USER scott

- é autorizada (GRANT) a permissão de SELECT sobre a tabela 'dept' (ON TABLE) para o utilizador 'scott' (TO USER). GRANT SELECT, UPDATE ON TABLE dept TO scott - é autorizada (GRANT) a permissão de SELECT e UPDATE sobre a tabela 'dept' (ON TABLE) para o utilizador 'scott' (TO USER).

REVOKE SELECT ON TABLE dept FROM Scott - nega (REVOKE) a permissão de SELECT na tabela 'dept' (ON TABLE) para o utilizador 'Scott' (FROM). A linguagem SQL como DQL , exemplos de comandos :

SELECT * FROM autores - selecciona todos os campos de todos os registos da tabela autores. SELECT nome, email, telefone, estado FROM autores - selecciona os campos nome, email, telefone e estado de todos os registos da tabela autores. SELECT nome, email, estado FROM autores WHERE estado = 'SP' - selecciona os campos nome, email e estado apenas dos registos cujo estado é SP. SELECT nome, email, estado, idade FROM autores WHERE idade = '23' AND estado = 'SP' - selecciona os campos nome, email, estado e idade apenas dos registos cuja idade é 23 e o estado é SP. SELECT nome, email, estado, idade FROM autores WHERE estado = 'SP' OR estado = 'AM' - selecciona os campos nome, email, estado e idade apenas dos autores onde o estado é SP ou AM.

Como disse, a linguagem SQL é divertida e fácil de aprender.

Boas queries de SQL a vocês todos !

Os meus cumprimentos,

Pedro Mesquita (<http://pedromesquita.pt.to>)

Sobre o Autor

(consultar site pessoal)