

SQL

Structured Query Language



Ferramenta para:

Organizar

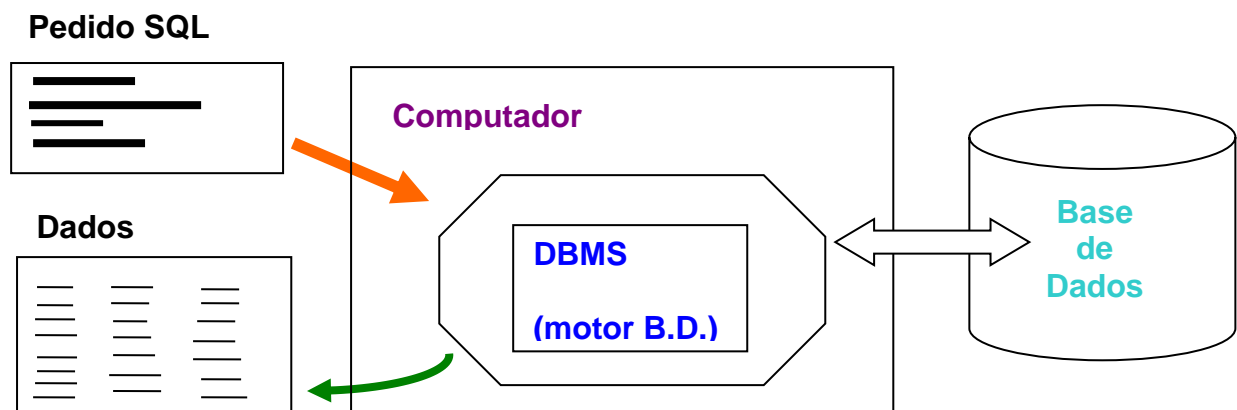
Gerir

Consultar

dados armazenados numa base de dados



Linguagem para interagir com base de dados relacionais



O SQL é usado para controlar todas as funções que o DBMS prevê para o utilizador:

- Definição dos dados
 - permite a definição da estrutura e organização dos dados e das suas relações
- Obtenção de dados
 - permite que uma aplicação obtenha os dados armazenados e que os use
- Manipulação dos dados
 - permite actualizações dos dados através das operações:
 - ☒ alterar
 - ☒ juntar
 - ☒ eliminar
- Controlo de acesso
 - permite restringir a capacidade de um utilizador de alterar, juntar ou eliminar dados
- Partilha de dados
 - Coordena a partilha de dados por utilizadores concorrentes
- Garantia de integridade dos dados

- Permite definir integridade evitando que a base de dados fique corrompida por alterações inconsistentes ou por falhas do sistema

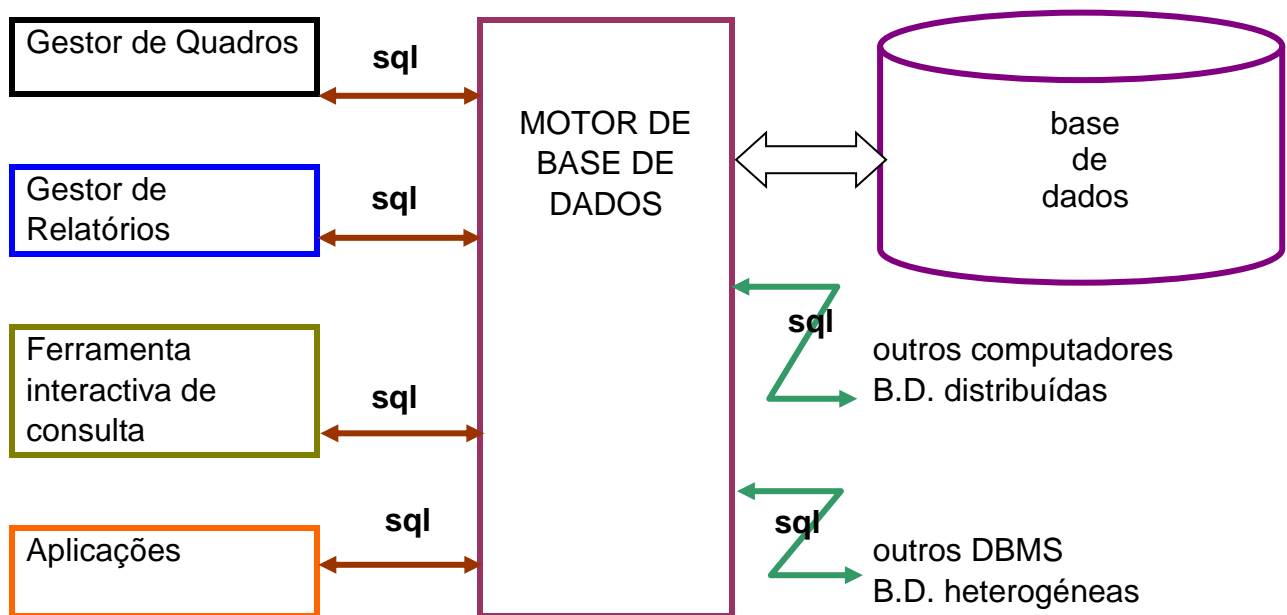
⇒ Não é uma linguagem completa

Não tem: `if`, `goto`, `do`, `for`, ...

⇒ Contém cerca de 30 instruções

SQL → Parte integrante de um sistema de gestão de base de dados: uma ferramenta e uma linguagem para comunicar com o DBMS

Componentes típicos de um sistema de base de dados



Características do SQL

- 👉 Independência do fabricante
 - Centenas de implementações
 - Facilidade de migração

👉 Portabilidade

- microcomputadores



mainframes

👉 SQL padrão

- ansi 92: padrão actual

👉 Modelo relacional

- modelo linha/ coluna: intuitivo



sql fácil

👉 Linguagem descritiva de alto nível

- SQL, semelhante a frases em inglês
- **Descreve os dados a obter e não a forma de o fazer**
- Os nomes podem ser longos e portanto com significado acrescido

👉 Acesso interactivo à BD

- Permite obter respostas rápidas a problemas complexos

👉 Acesso programático à BD

- São usadas as mesmas instruções interactivas o que resulta numa grande facilidade em testar

👉 Diferentes vistas dos dados

- Permite isolar ou combinar dados

👉 Linguagem completa para base de dados

- Inicialmente só permitia consultas, actualmente a linguagem padrão aparece embutida em superconjuntos de SQL muito potentes enquanto linguagem de programação, com todo o tipo de controlo de fluxo de instruções e definições de tipos de dados (T-SQL, PL-SQL,)

👉 Definição dinâmica da estrutura

- Permite alterar a estrutura mesmo com utilizadores a usar a Base de dados

👉 Arquitectura Cliente/ Servidor

ABORDAGEM AO SQL

PERSPECTIVA GERAL

EXEMPLO DE BASE DE DADOS

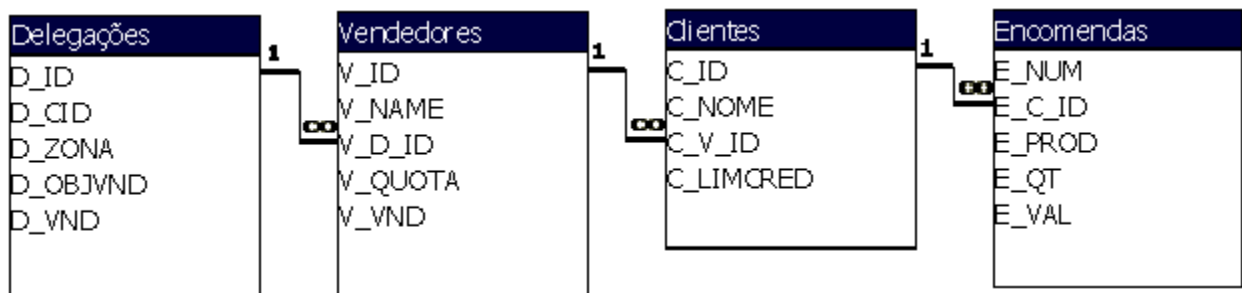


Tabela: Delegações

D_ID	D_CID	D_ZONA	D_OBJVND	D_VND
22	PORTO	Norte	3000	1200
21	LISBOA	Centro	5750	3700
11	FARO	Sul	8000	6300
43	BRAGA	Norte	3500	800

Tabela: Clientes

C_ID	C_NOME	C_V_ID	C_LIMCRED
2111	JCP, Lda	103	500
2102	AAA Investimentos, SA	107	650
2103	Carlos Costa & Cª, Lda	111	500
2123	João Dias - Representações	106	400
2107	Ana Lima - Soc Import, Lda	108	350
2115	AZ - Import/Export, SA	105	200
2101	Sá & Pereira, Lda	107	650
2112	R & J, Lda	107	450
2121	MegaCorp, Sa	106	250
2114	Alfa Beta, Lda	111	200
2124	Roma - Viagens, Sa	103	250
2108	Rest. Paga Pouco, Lda	108	500

Tabela: Vendedores

V_ID	V_NAME	V_D_ID	V_QUOTA	V_VND
103	Joao	22	3000	1200
105	Luis	21	3000	1500
111	Artur	11	5000	2000
107	Gil	43	3500	800
106	Ana	21	2750	3200
108	Paulo	11	3000	4300

Tabela: Encomendas

E_NUM	E_C_ID	E_PROD	E_QT	E_VAL
112961	2114	2A44L	7	315,0

113012	2111	41003	35	37,5
112989	2101	114	6	24,3
113051	2102	KX47	4	14,8
112968	2107	41004	34	108,8
113036	2112	4100Z	9	38,7
113045	2103	2A44R	10	56,8
112963	2108	112	28	327,6
113013	2124	XK47	1	9,3
113058	2103	114	10	40,5
112997	2114	4100Z	1	4,3
112983	2124	773C	6	31,8
113024	2114	4100X	20	74,0
113062	2103	4100Y	10	8,7
112979	2112	2A4R	6	24,6

1. Obter Dados

Instrução para obter dados: **SELECT**

```
SELECT D_CID, D_ID, D_VND  
FROM DELEGAÇÕES
```

D_CID	D_ID	D_VND
PORTO	22	1200
LISBOA	21	2700
FARO	11	3700
BRAGA	43	800

Indicam-se:

- os campos pretendidos
- tabela origem dados

Permite obter valores calculados:

```
SELECT V_NAME, V_VND, V_QUOTA, (V_VND-V_QUOTA)  
FROM Vendedores
```

V_NAME	V_VND	V_QUOTA	V_VND-V_QUOTA
Joao	1200	3000	-1800
Luis	1500	3000	-1500
Artur	2000	5000	-3000
Gil	800	3500	-2700
Ana	3200	2750	450
Paulo	4300	3000	1300

Permite obter valores seleccionados:

```
SELECT V_NAME, V_VND, V_QUOTA, (V_VND-V_QUOTA)
FROM Vendedores
WHERE V_VND < V_QUOTA
```

V_NAME	V_VND	V_QUOTA	V_VND-V_QUOTA
Joao	1200	3000	-1800
Luis	1500	3000	-1500
Artur	2000	5000	-3000
Gil	800	3500	-2700

De modo análogo poderemos saber quais as maiores encomendas, que Clientes as colocaram, de que Produto e em que Quantidade e podemos obter essa informação por ordem decrescente de valor:

```
SELECT E_NUM, E_C_ID, E_PROD, E_QT, E_VAL
FROM Encomendas
WHERE E_VAL>100
ORDER BY E_VAL DESC
```

E_NUM	E_C_ID	E_PROD	E_QT	E_VAL
112963	2108	112	28	327,6
112961	2114	2A44L	7	315,0
112968	2107	41004	34	108,8

2. Sumariar informação

Qual o valor médio das Encomendas?

```
SELECT Avg(E_VAL)
FROM Encomendas
```

Avg(E_VAL)
74,45

Qual o valor médio das Encomendas de um Cliente?

```
SELECT Avg(E_VAL)
FROM Encomendas
Where E_C_ID = 2112
```

Avg(E_VAL)
31,65

Qual o valor total das Encomendas de cada Cliente?

```
SELECT E_C_ID, Sum(E_VAL)
FROM Encomendas GROUP BY E_C_ID
```

E_C_ID	Sum (E_VAL)
2101	24,3
2102	14,8
2103	106
2107	108,8
2108	327,6
2111	37,5
2112	63,3
2114	393,3
2124	41,1

3. Inserir informação na Base de Dados

Instrução para inserir dados: **INSERT**

Foi criada uma nova delegação em Coimbra com objectivos de vendas de 4000 e código 27.

```
INSERT INTO Delegações ( D_ID, D_CID, D_ZONA, D_OBJVND, D_VND )
VALUES (27, "COIMBRA", "Centro", 4000, 0)
```

1 row inserted.

4. Apagar informação da Base de Dados

Instrução para apagar dados: **DELETE**

Se um Cliente decidiu mudar de Fornecedor pode ser removido da base:

```
DELETE C_NOME
FROM Clientes
WHERE C_NOME='MegaCorp, Sa'
```

1 row deleted.

De modo análogo se decidirmos remover os Vendedores com Vendas abaixo das Quotas poderemo fazê-lo com:

```
DELETE
FROM Vendedores
WHERE V_VND<V_QUOTA
```

4 rows deleted.

5 Actualizar informação da Base de Dados

Instrução para actualizar dados: **UPDATE**

Por exemplo para aumentar o Limite de Crédito de um Cliente:

```
UPDATE Clientes
SET C_LIMCRED = 750
WHERE C_NOME='R & J, Lda'
```

1 row updated.

Ou para aumentar a Quota de todos os Vendedores:

```
UPDATE Vendedores SET Vendedores.V_QUOTA = V_QUOTA+500
```

6 rows updated.

6. Proteger informação na Base de Dados

Instruções para proteger / desproteger dados: **GRANT/ REVOKE**

Por exemplo para permitir que um utilizador RITA possa inserir novos dados na tabela de Clientes:

```
GRANT INSERT  
ON Clientes  
TO Rita
```

Privilege granted.

E para que possa também actualizar e obter dados de Clientes:

```
GRANT UPDATE, SELECT  
ON Clientes  
TO Rita
```

Privilege granted.

Para que deixe de poder inserir Clientes:

```
REVOKE INSERT  
ON Clientes  
FROM Rita
```

Privilege granted.

E para que deixe de poder manipular Clientes de qualquer modo:

```
REVOKE ALL  
ON Clientes  
FROM Rita
```

Privilege granted.

7. Criação de Tabelas

Com o SQL é possível definir toda a estrutura de tabelas, campos, chaves e relações.

Por exemplo para criar na Base de Dados uma nova tabela para Produtos:

```
CREATE TABLE PRODUCTS  
(P_ID INTEGER,
```



```
P_DESCR CHAR(20) ,  
  P_M_ID INTEGER,  
  P_PRUNIT MONEY,  
P_QT_EXIST FLOAT)
```

Table created.

Também é possível remover uma tabela da Base de Dados:

```
DROP TABLE Temp
```

Table dropped.