

SGBD – SQL Server

Licenciatura em Engenharia Informática e Computação

Bases de Dados

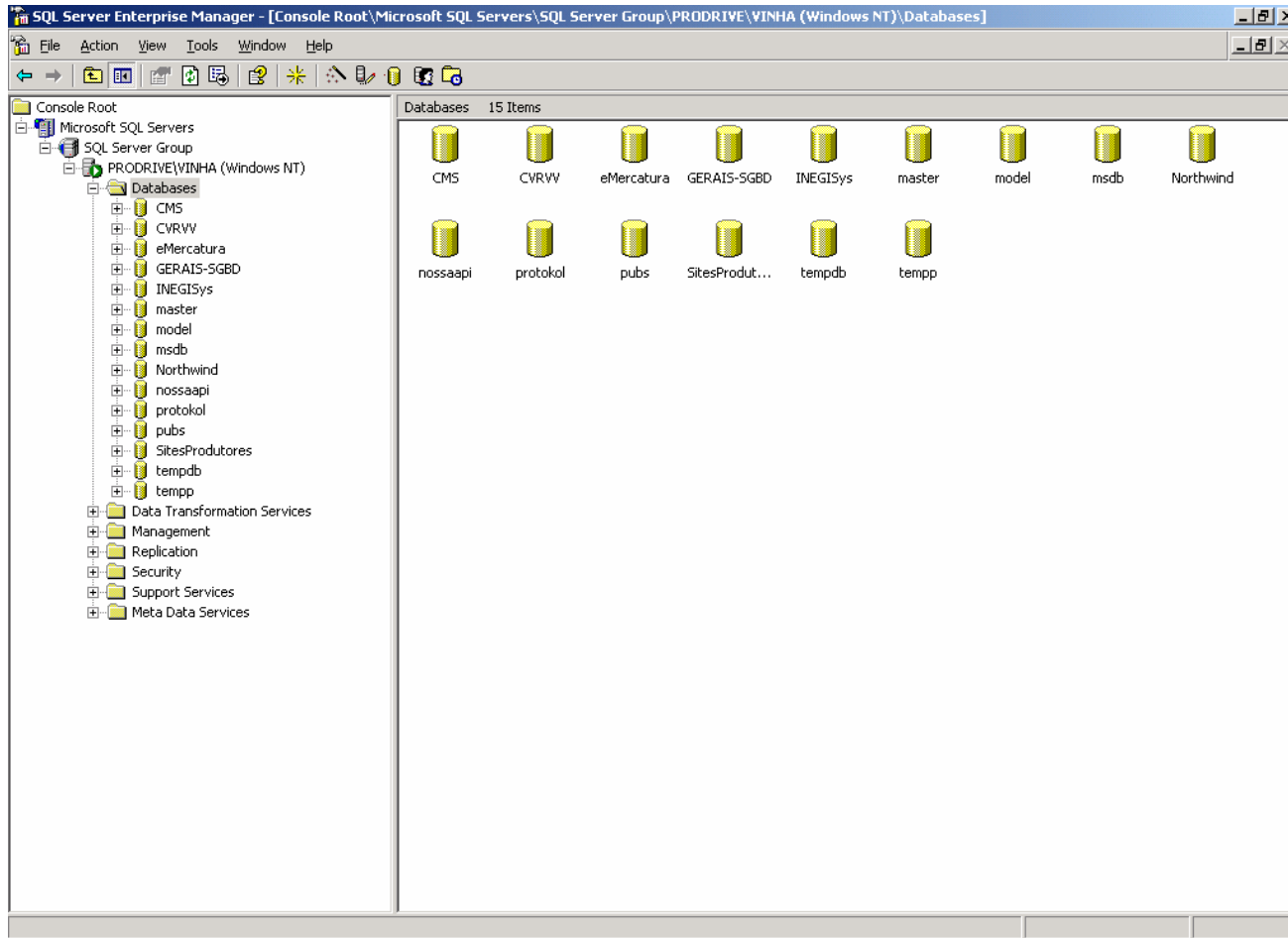
2003/04

Criação da Ligação ao SGBD

- Entrar no Microsoft SQL Server / Enterprise Manager
- Criar uma ligação ao servidor carregando com o botão da direita sobre SQL Server Group:
 - *New SQL Server Registration*
 - *Escolher/escrever servidor “DOLLY” e seleccionar*
 - *Escolher “Autenticação do SQL server”*

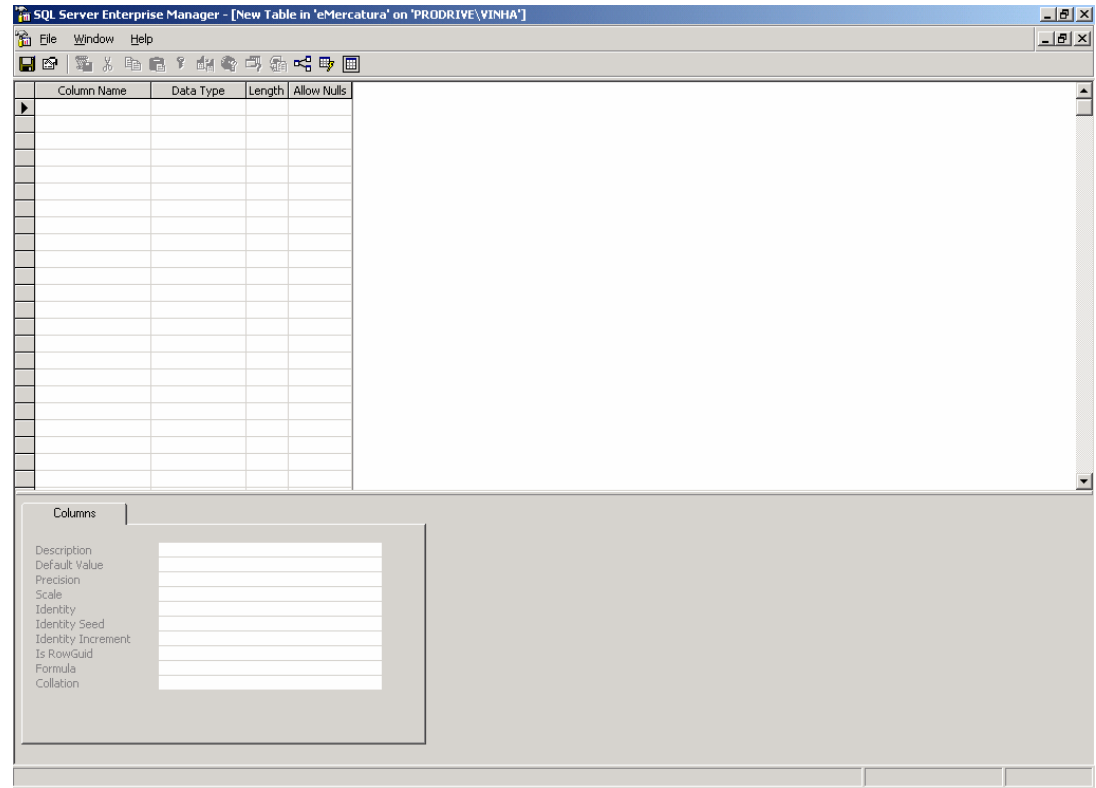
 - *Username: **BD03xx***
 - *Password: **BD03xx***
- Testar a ligação

Enterprise Manager



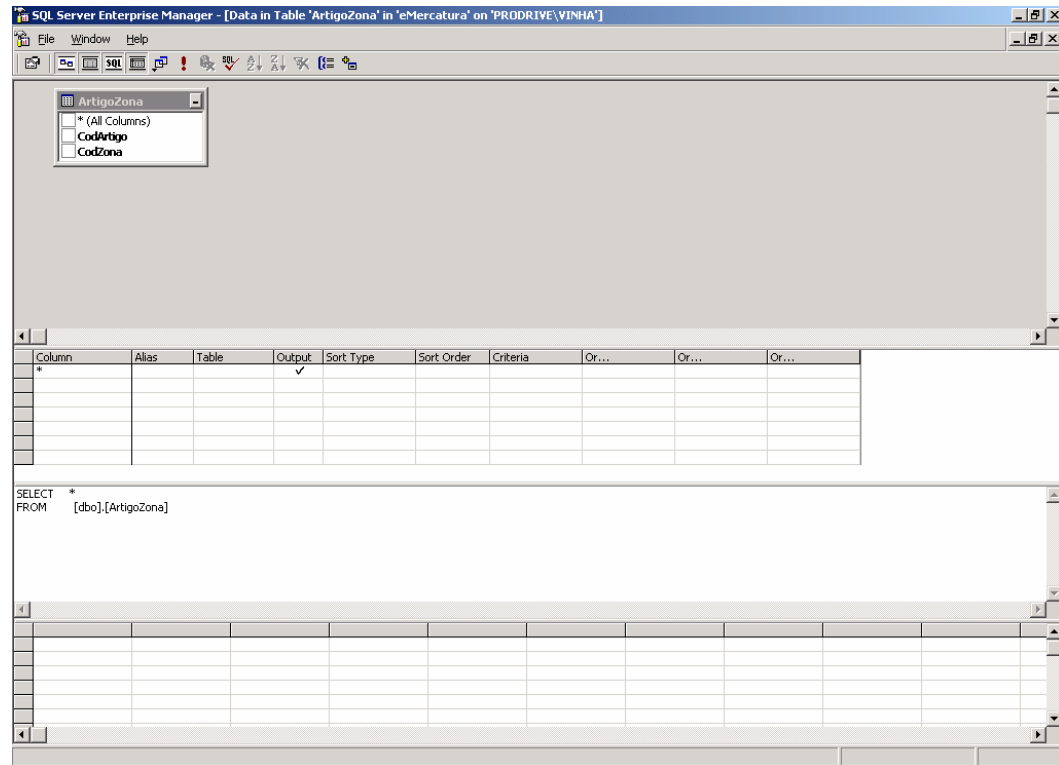
Criação de uma Tabela

- Escolher a BD
- Botão direito
- New → Table



O Interpretador de SQL

- Seleccionar uma tabela
- Botão direito
- Open Table → Query



Criação da tabela CLIENTE

- Criar a tabela **CLIENTE** com a seguinte estrutura:
 - `CodCliente, int, 4, NOT NULL`
 - `Titulo, varchar, 20`
 - `Nome, varchar, 200, NOT NULL`
 - `Morada, varchar, 200, NOT NULL`
 - `CodigoPostal, char, 8, NOT NULL`
 - `Localidade, varchar, 50, NOT NULL`
 - `Telefone, varchar, 15, NOT NULL`
- Definir `CodCliente` como chave primária

Criação da tabela CONTA

- Criar a tabela **CONTA** com a seguinte estrutura:
 - `CodConta, int, 4, NOT NULL`
 - `TipoConta, smallint, 2, NOT NULL`
 - `Saldo, float, 8, NOT NULL`
- Definir `CodConta` como chave primária

Criação da tabela TITULAR

- Criar a tabela **TITULAR** com a seguinte estrutura:
 - `CodConta, int, 4, NOT NULL`
 - `CodCliente, int, 4, NOT NULL`
 - `Titular, smallint, 2, NOT NULL`
- Definir `CodConta, CodCliente` como chaves primárias
- Definir relações

Criação da tabela MOVIMENTO

- Criar a tabela **MOVIMENTO** com a seguinte estrutura:
 - `CodMovimento, int, 4, IDENTITY`
 - `CodConta, int, 4, NOT NULL`
 - `Data, datetime, 8, NOT NULL`
 - `TipoMovimento, smallint, 2, NOT NULL`
 - `Montante, float, 8, NOT NULL`
 - `MovimentadoPor, int, 4`
- Definir `CodMovimento` como chave primária
- Definir relações

Chaves estrangeiras / relações

- Seleccionar “Diagrams”
- Botão direito / New DataBase Diagram
- Seleccionar as tabelas
- Desenhar o esquema

Inserção de Dados

```
INSERT INTO conta (codconta, tipoconta, saldo) VALUES (937261, 8, 1497.10)
```

```
INSERT INTO cliente (codcliente, titulo, nome, morada, codigopostal, localidade, telefone) VALUES (765076486, 'Sr.', 'Joaquim Silva', 'Av. dos Aliados, 125', 4000125, 'Porto', 222042132)
```

```
INSERT INTO titular (codconta, codcliente, titular) VALUES (937261, 765076486, 1)
```

Inserção e Actualização

```
INSERT INTO movimento (codconta, data,  
tipomovimento, montante, movimentadoPor)  
VALUES (937261, GETDATE(), 3, 1000,  
765076486)
```

```
UPDATE conta SET saldo = saldo + 1000  
WHERE codconta = 937261
```

Consulta de Dados

tabelaA

id	nome
1	João
2	Ana

tabelaB

id	cidade
1	Porto
2	Lisboa

```
SELECT * FROM tabelaA, tabelaB
```

tabelaA.id	tabelaA.nome	tabelaB.id	tabelaB.cidade
1	João	1	Porto
1	João	2	Lisboa
2	Ana	1	Porto
2	Ana	2	Lisboa

Consulta de Dados (JOIN)

```
SELECT * FROM tabelaA, tabelaB
```

É transformado em:

```
SELECT * FROM tabelaA CROSS JOIN tabelaB
```

→ É feita a multiplicação das tabelas.

Consulta de Dados

tabelaA.id	tabelaA.nome	tabelaB.id	tabelaB.cidade
1	João	1	Porto
1	João	2	Lisboa
2	Ana	1	Porto
2	Ana	2	Lisboa

tabelaA.nome	tabelaB.cidade
João	Porto
Ana	Lisboa

```
SELECT tabelaA.nome,
tabelaB.cidade FROM
tabelaA, tabelaB WHERE
tabelaA.id = tabelaB.id
```

Consulta de Dados (JOIN)

```
SELECT tabelaA.nome, tabelaB.cidade FROM  
tabelaA, tabelaB WHERE tabelaA.id = tabelaB.id
```

É transformado em:

```
SELECT * FROM tabelaA INNER JOIN tabelaB ON  
tabelaA.id = tabelaB.id
```

→ É feita a junção das tabelas nos campos quando os IDs são iguais.

Consulta de Dados (JOIN)

```
SELECT * FROM tabelaA LEFT JOIN tabelaB ON  
tabelaA.id = tabelaB.id
```

➔ Mostra todos os elementos da tabelaA, mesmo quando não têm relações na tabelaB:

Consulta de Dados

tabelaA

id	nome
1	João
2	Ana
3	Maria

tabelaB

id	cidade
1	Porto
2	Lisboa

```
SELECT * FROM tabelaA LEFT JOIN tabelaB ON
tabelaA.Id = tabelaB.Id
```

tabelaA.id	tabelaA.nome	tabelaB.id	tabelaB.cidade
1	João	1	Porto
2	Ana	2	Lisboa
3	Maria		

Consulta de Dados

```
SELECT codconta, saldo FROM  
conta WHERE codconta = 937261
```

```
SELECT c.titulo, c.nome FROM  
cliente c, titular t WHERE  
t.codconta = 937261 AND  
c.codcliente = t.codcliente  
ORDER BY c.nome
```

Consulta de Dados

```
SELECT tipoconta, MAX(saldo) FROM  
conta GROUP BY tipoconta HAVING  
MAX(saldo) > 2000
```

```
SELECT t.cliente, COUNT(t.conta) AS  
contas FROM titular t, conta c WHERE  
c.saldo > 2000 AND t.conta = c.conta  
GROUP BY t.cliente HAVING  
COUNT(t.conta) > 1
```

Eliminação de Dados

```
DELETE FROM movimento
```

```
DELETE FROM movimento WHERE codconta  
= 937261
```

Vistas

- Criar uma vista para mostrar todos os clientes com contas em que o saldo é zero:
 - Botão direito em *views / New View*
 - *Seleccionar tabelas e alterar o SQL gerado...*

ou

```
CREATE VIEW vista AS
  SELECT Cliente.Titulo, Cliente.Nome
  FROM Cliente INNER JOIN
  Titular ON Cliente.CodCliente = Titular.CodCliente
  INNER JOIN Conta ON Titular.CodConta =
  Conta.CodConta
  WHERE (Conta.Saldo = 0)
```

Restrições

- Para inserir/alterar restrições:
 - Alterar a tabela (botão direito / design table)
 - Ícone “Manage Constrains”
- Exercício:
 - Definir que o tipo de conta tem de ser superior a 0 e menor ou igual que 999

Gatilhos

```
CREATE TRIGGER actualiza_saldo ON movimento FOR INSERT
AS
IF EXISTS(select * from inserted)
BEGIN
    UPDATE conta
    SET saldo = saldo + i.montante FROM inserted i
    WHERE conta.codconta = i.codconta
END
```

```
CREATE TRIGGER actualiza_movimentos ON movimento FOR
UPDATE, DELETE
AS
RAISERROR('Não pode actualizar nem eliminar itens da
tabela MOVIMENTO!', 16, 1)
ROLLBACK TRANSACTION
```


Disparo de um gatilho

SQL Server Enterprise Manager - [Data in Table 'Movimento' in 'banco']


File Window Help

SQL

CodMovimento	CodConta	Data	TipoMovimento	Montante	MovimentadoPor
1	937261	14-12-2003 20:44:	3	1000	765076486
3	123	10-10-2003	1	500	1
5	123	15-10-2004	1	10000	1
6	123	02-02-2004	1	1000	1

* [asterisk icon]

SQL Server Enterprise Manager



[Microsoft][ODBC SQL Server Driver][SQL Server]Não pode actualizar nem eliminar itens da tabela MOVIMENTO!

OK Help

Importação / Exportação de dados

- Útil para migrar bases de dados de uma tecnologia para outra ou para sincronizar bases de dados remotas
- Start / Programs / SQL Server / Import and Export Data

Importação / Exportação de dados

- Seleccionar Data Source:
 - Microsoft OLE DB Provider for Oracle

- Definir ligação do Oracle
 - Servidor
 - Porta
 - Login

- Definir objectos a transferir

Ligação DSN

- É uma forma simples de ligar-se a uma BD usando *drivers* do Windows.
- Painel de Controlo / Ferramentas administrativas / Fontes de Dados (ODBC)
- Exercício: Criar uma ligação DSN à BD

Ligação através de JAVA

- Uma *Query* JDBC a uma BD necessita dos seguintes componentes:
 - Driver
 - String de ligação
 - Connection
 - Statement
 - ResultSet