

Considere o esquema relacional:

Departamento (DepNum, Nome, Local)

Empregado (EmpNum, Nome, Categoria, Salario, DepNum)

Projecto (ProjNum, Designacao, Fundos)

Atribuicao (EmpNum, ProjNum, Funcao)

Onde, em cada relação, os atributos sublinhados correspondem à chave primária.

1. Criação da base de dados

A partir do esquema relacional (conjunto de relações) pretende-se construir uma base de dados em *Microsoft SQL Server*, que vamos designar por *DBProject*.

Para esta tarefa será necessário estabelecer uma ligação com o servidor de base de dados, usando para o efeito o programa *SQL Server Management Studio* e o utilizador *sa* (*system administrator*).

Pretende-se que execute as seguintes tarefas:

- a) Use um motor de busca para procurar o comando CREATE DATABASE no SQL SERVER e estude os resultados devolvidos.
- b) Use o editor do programa *SQL Server Management Studio* para abrir o ficheiro (script) com a designação *BD-CreateDB.sql*.
Estude o código, e após adaptação, crie a base de dados.
Verifique que dois novos ficheiros (um para os dados e outro para o *log*) foram criados no disco local e verifique o seu tamanho.
Verifique a uma nova base de dados está disponível na lista de bases de dados do servidor de bases de dados.
- c) Ligue-se à base de dados *DBProject* e verifique que a nova base de dados já está equipada com vários objetos. Navegue sobre esses objetos.

2. Criação das tabelas

Pretende-se transformar o conjunto de relações, obtido durante a fase de modelação, num conjunto de tabelas na base de dados. Para isso é necessário dar nomes às tabelas (é frequente manter o nome das relações correspondentes) e aos seus campos (é frequente manter o nome dos atributos das relações) e também decidir acerca do tipo de dados de cada campo e do seu tamanho.

Por exemplo:

Departamento (DepNum, Nome, Local)		
DepNum	→	Integer
Nome	→	30 Chars
Local	→	30 Char

Empregado (EmpNum, Nome, Categoria, Salario, DepNum)

EmpNum	→	Integer
Nome	→	50 Chars
Categoria	→	30 Chars
Salario	→	Float
DepNum	→	Integer

Projecto(ProjNum, Designacao, Fundos)

ProjNum	→	Integer
Designacao	→	30 Chars
Fundos	→	Float

Atribuicao(EmpNum, ProjNum, Funcao)

EmpNum	→	Integer
Pro Num	→	Integer
Funcao	→	30 Chars

Finalmente, é necessário usar o comando da SQL DDL (*Data Definition Language*) apropriado para criar cada tabela dentro da base de dados *DBProject*.

Pretende-se que execute as seguintes tarefas:

- Use um motor de busca para procurar o comando CREATE TABLE no SQL SERVER e estude os resultados devolvidos.
- Estude os tipos de dados disponíveis no *SQL Server*, usando para o efeito um motor de busca (visite, por exemplo, http://www.w3schools.com/sql/sql_datatypes.asp).
- Use o editor do programa *SQL Server Management Studio* para abrir o ficheiro (script) com a designação ***BD-CreateTables.sql***.
Estude o código associado à criação de cada tabela. Pesquise, usando o motor de busca, o que significa cada elemento do comando CREATE TABLE, nomeadamente, CONSTRAINT, CHECK, DEFAULT, NOT NULL, PRIMARY KEY, FOREIGN KEY, REFERENCES, ON UPDATE CASCADE, ON DELETE NO ACTION.
Execute o *script* e verifique que foram criadas as tabelas pretendidas.
- Use um motor de busca para procurar o comando ALTER TABLE no SQL SERVER e estude os resultados devolvidos.

3. Inserção de dados iniciais

Pretende-se também inserir alguns dados nas tabelas. O SQL também ajuda nesta etapa (neste caso usando comandos DML - *Data Manipulation Language*).

Pretende-se que execute as seguintes tarefas:

- Use um motor de busca para procurar o comando INSERT INTO no SQL SERVER e estude os resultados devolvidos.
- Use o editor do programa *SQL Server Management Studio* para abrir o ficheiro (script) com a designação ***BD-InsData.sql***.
Estude o código e execute o *script*. Verifique que é emitida uma mensagem de erro. Identifique, no código, a origem da mensagem de erro e estude-a.
Verifique que as tabelas contêm dados.