

Aula Prática 11

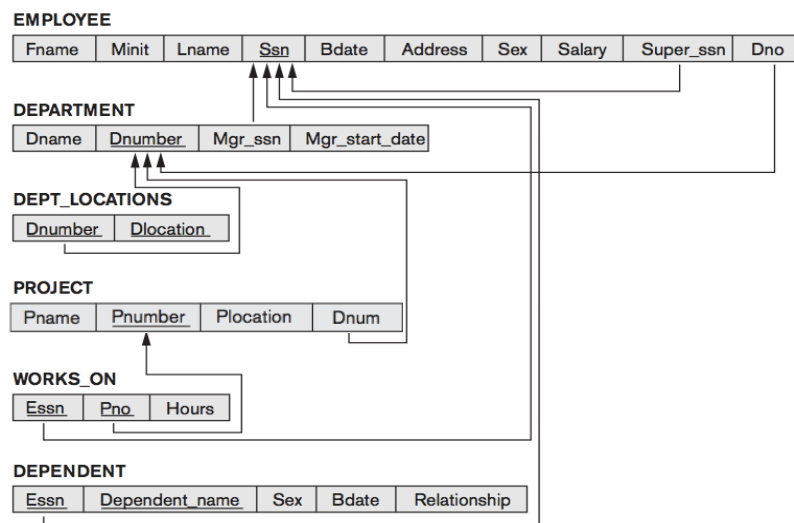
Objetivos

SQL Programming:

Stored Procedure, User-Defined Function (UDF), Trigger e Cursor.

Problema 11.1

Tome como referência a base de dados relacional da Empresa utilizada em guiões anteriores:



- Construa um stored procedure que aceite o ssn de um funcionário, que o remova da tabela de funcionários, que remova as suas entradas da tabela works_on e que remova ainda os seus dependentes. Que preocupações adicionais devem ter no storage procedure para além das referidas anteriormente?
- Crie um stored procedure que retorne um record-set com os funcionários gestores de departamentos, assim como o ssn e número de anos (como gestor) do funcionário mais antigo dessa lista.
- Construa um trigger que não permita que determinado funcionário seja definido como gestor de mais do que um departamento.
- Crie um trigger que não permita que determinado funcionário tenha um vencimento superior ao vencimento do gestor do seu departamento. Nestes casos, o trigger deve ajustar o salário do funcionário para um valor igual ao salário do gestor menos uma unidade.
- Crie uma UDF que, para determinado funcionário (ssn), devolva o nome e localização dos projetos em que trabalha.
- Crie uma UDF que, para determinado departamento (dno), retorne os funcionários com um vencimento superior à média dos vencimentos desse departamento;
- Crie uma UDF que, para determinado departamento, retorne um record-set com os

projetos desse departamento. Para cada projeto devemos ter um atributo com seu o orçamento mensal de mão de obra e outra coluna com o valor acumulado do orçamento.

Nota: parta do princípio que um funcionário trabalha 40 horas por semana para o cálculo do custo da sua afetação ao projeto.

Exemplo: `select * from dbo.employeeDeptHighAverage(3);`

pname	pnumber	plocation	dnum	budget	totalbudget
Aveiro Digital	1	Aveiro	3	2625,00	2625,00
Dicooogle	3	Aveiro	3	1337,50	3962,50
GOP&CS	4	Aveiro	3	781,25	4743,75

{3 row(s) affected}

Recomendação: utilize um cursor.

- h) Crie um trigger que, quando se elimina um departamento, este passe para uma tabela `department_deleted` com a mesma estrutura da `department`. Caso esta tabela não exista então deve criar uma nova e só depois inserir o registo. Deve utilizar um trigger do tipo *instead of*. Porquê? Qual a desvantagem de utilizar um trigger *after* neste caso?

Utilize a seguinte instrução para verificar se determinada tabela existe:

```
IF (EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES
            WHERE TABLE_SCHEMA = 'myschema' AND TABLE_NAME = 'mytable'))
```

- i) Relativamente aos stored procedure e UDFs, enumere as suas mais valias e as características que as distingue. Dê exemplos de situações em que se deve utiliza cada uma destas ferramentas;