



Instituto Superior de Engenharia de Lisboa

Ano Lectivo 2010/2011 – Semestre Inverno

Sistemas de Informação 2

1ª Aula Prática

Professor: Felipe Profírio

Data de Entrega: 29 de Março de 2011

Autores:

Cláudia Crisóstomo nº 32142

Nuno Cancelo nº 31401

Nuno Sousa nº 33595

Índice

Enunciado.....	3
Parte I.....	4
Alínea a).....	4
Alínea b).....	5
Parte II.....	6
Alínea b).....	6
Alínea c).....	6
Alínea d).....	7

Enunciado

I

Na rede de sensores do Observatório-X, um sensor é caracterizado por um nome único, e.g., SENSOR001 (9 caracteres) e por dois valores reais opcionais que especificam os limites inferior e superior do intervalo de valores que se consideram usuais. Existe um sistema de recolha de leituras que é executado regularmente. Cada leitura é caracterizada por um identificador inteiro único (gerado automaticamente) e inclui a informação do sensor, o instante de leitura e o respectivo valor. No Observatório-X são efectuados relatórios que incluem leituras com valores fora da gama espectável.

- a) Projecte o repositório de dados contemplando os níveis: conceptual, lógico e físico.
- b) Construa, em T-SQL, um procedimento armazenado que receba o identificador (válido) de um sensor e que gere uma leitura. Admita que os valores podem estar dentro ou fora da gama espectável.
- c) Construa, em T-SQL, o código para obter os relatórios das leituras.

II

No Observatório-X não eram consideradas as situações de alarme relativas a leituras fora da gama de valores esperados. Por isso, a empresa decidiu implementar a geração de alarmes. Os alarmes devem ser mantidos no sistema, mesmo que os valores das leituras e/ou dos sensores a que dizem respeito deixem de existir. Um alarme é gerado sempre que uma leitura sai fora da gama espectável de valores do respectivo sensor. Um alarme é caracterizado pela identificação do respectivo sensor, contém o tipo de alarme ('min' para valor inferior ao limite inferior e 'max' para valor superior ao limite superior) e o instante de colheita da leitura associada.

- a) Projecte o novo repositório de dados contemplando os níveis: conceptual, lógico e físico.
- b) Construa em T-SQL o código que permita realizar a geração de um alarme recorrendo à utilização de procedimentos armazenados.
- c) Construa em T-SQL o código que permita realizar a geração de um alarme recorrendo à utilização de gatilhos.
- d) Comente as soluções apresentadas, comparando questões relacionadas com a ocultação das regras de negócio e com o impacto em I das alterações introduzidas em II.

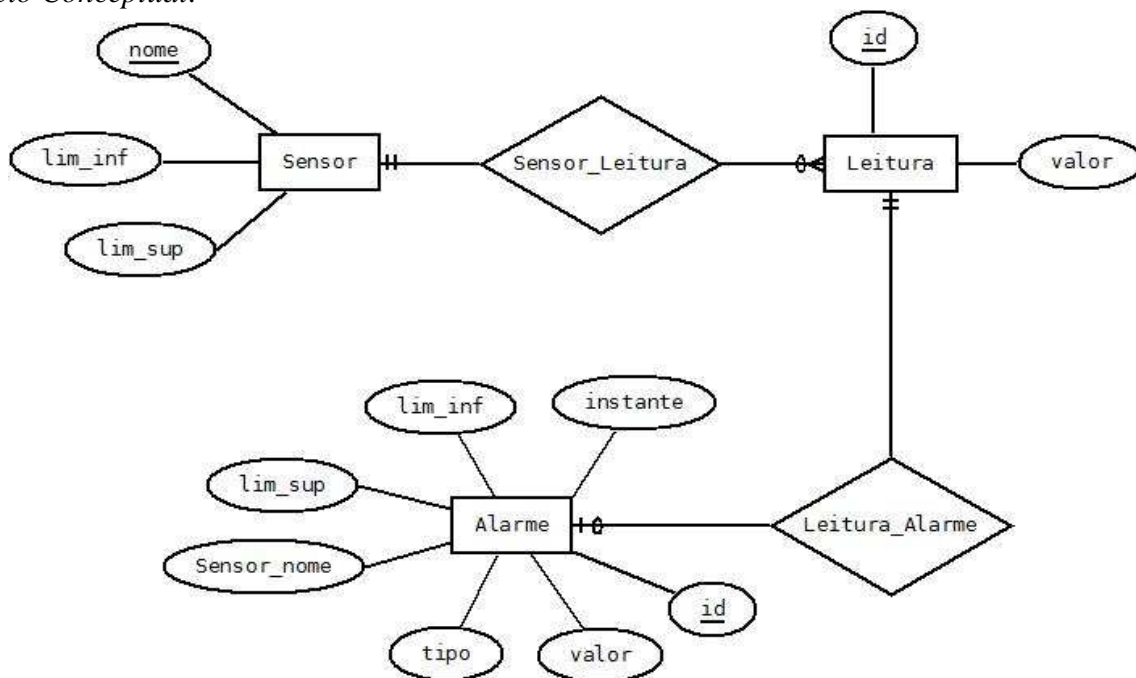
Parte I

Alínea a)

Projecte o novo repositório de dados contemplando os níveis: conceptual, lógico e físico.

O enunciado apresentado separa o problema em duas partes, por isso, para simplificar a resolução do mesmo optámos por desenvolver apenas um modelo conceptual, lógico e físico que abrange todo o problema.

Modelo Conceptual:



Legenda 1: Modelo Entidade-Associação.

Modelo Lógico:

Sensor(nome[PK], lim_inf, lim_sup)

Leitura(id[PK], valor, nomeSensor)

Alarme(id[PK], instante, lim_inf, lim_sup, Sensor_nome, tipo, valor)

Modelo Físico:

```
IF db_id('SI2_API') is null
    CREATE DATABASE AP1;
GO

USE SI2_API;
set dateformat dmy;

if exists(select *
          from INFORMATION_SCHEMA.TABLES
          where TABLE_NAME = 'Leitura')
drop table dbo.Leitura;
```

```

if exists(select *
          from INFORMATION_SCHEMA.TABLES
          where TABLE_NAME = 'Sensor')
drop table dbo.Sensor;

if exists(select *
          from INFORMATION_SCHEMA.TABLES
          where TABLE_NAME = 'Alarme')
drop table dbo.Alarme;

create table Sensor(
    name char(9) not null,
    limitInf decimal(10) not null,
    limitSup decimal(10) not null,
    constraint pk_Sensor primary key (name)
)

create table Leitura(
    id int identity(1,1),
    sensorName char(9) not null,
    valor decimal(10) not null,
    instanteLeitura datetime not null,
    constraint pk_Leitura primary key(id),
    constraint fk_Leitura foreign key(sensorName) references Sensor(name)
)

create table Alarme(
    id int identity(1,1),
    sensorName char(9) not null,
    sensorLimInf decimal not null,
    sensorLimSup decimal not null,
    leituraValor decimal(10) not null,
    instanteLeitura datetime not null,
    /*tipo char(3) not null,*/
    constraint pk_Alarme primary key(id),
    /*constraint ck_Alarme_Tipo check(tipo='min' or tipo='max')*/
)

```

Alínea b)

Construa, em T-SQL, um procedimento armazenado que receba o identificador (válido) de um sensor e que gere uma leitura. Admita que os valores podem estar dentro ou fora da gama espectável.

```

IF EXISTS(SELECT * FROM sys.objects WHERE type = 'P' AND name = 'geraLeitura')
    DROP PROCEDURE geraLeitura
go
create procedure geraLeitura(@sensor char(9))
as
begin transaction
insert into dbo.Leitura(sensorName, valor, instanteLeitura) values(@sensor, 20 * rand(), SYSDATETIME())
commit

```

Alínea c)

Construa, em T-SQL, o código para obter os relatórios das leituras.

```

SELECT @@TOTAL_READ AS 'Reads'

```

Parte II

Alínea b)

Construa em T-SQL o código que permita realizar a geração de um alarme recorrendo à utilização de procedimentos armazenados.

```
create procedure geraAlarme(@sensorName char(9), @sensorLimInf decimal, @sensorLimSup decimal,
                           @leituraValor decimal(10))
as
begin transaction
    declare @tipo char(3)
    set @tipo = ""
    if (@leituraValor > @sensorLimSup)
    begin
        set @tipo = 'max'
    end
    if (@leituraValor < @sensorLimInf)
    begin
        set @tipo = 'min'
    end
    if (@tipo != "")
    begin
        insert into dbo.Alarme values(@sensorName, @sensorLimInf, @sensorLimSup, @leituraValor,
SYSDATETIME(), @tipo)
    end
commit
```

Com a implementação de um procedimento armazenado é necessário invocar explicitamente o mesmo para que a inserção de novos tuplos em Alarme seja efectuado.

Alínea c)

Construa em T-SQL o código que permita realizar a geração de um alarme recorrendo à utilização de gatilhos.

```
IF EXISTS(SELECT * FROM sys.objects WHERE type = 'TR' AND name = 'alarmeLeitura')
    DROP TRIGGER alarmeLeitura
go
create trigger alarmeLeitura
on Leitura
after INSERT
as
    declare @valor decimal(10), @sensor char(9), @inf decimal(10), @sup decimal(10), @instante
datetime
    select @valor = valor, @sensor = sensorName, @instante=instanteLeitura from INSERTED
    select @inf = limitInf, @sup = limitSup from Sensor
        where (Sensor.name = @sensor)

    begin
        if( @valor > @sup OR @valor < @inf)
```

```
insert into dbo.Alarme(  
    sensorName,  
    sensorLimInf,  
    sensorLimSup,  
    leituraValor,  
    instanteLeitura)  
values(@sensor,  
    @inf,  
    @sup,  
    @valor,  
    @instante)  
  
end
```

Alínea d)

Comente as soluções apresentadas, comparando questões relacionadas com a ocultação das regras de negócio e com o impacto em I das alterações introduzidas em II.

Através das soluções desenvolvidas nas alíneas b) e c) verificámos que a utilização de um procedimento armazenado não é benéfico porque é necessário efectuar a invocação explícita do mesmo (geraAlarme) aquando da inserção de um tuplo Leitura, por outro lado, o gatilho, alarmeLeitura, é executado pelo Sql Server sempre que ocorra a inserção de um tuplo na tabela Leitura.

Assim, concluímos que a utilização de gatilhos é mais fiável ao invés de procedimentos armazenados devido à garantia de execução do mesmo (gatilho) pelo Sql Server, enquanto que execução de procedimentos armazenados está dependente do algoritmo desenvolvido pelo programador, o que pode originar falhas por parte do programador.