Sistemas de Informação e Bases de Dados

2ª Parte do Projeto

Grupo 13

Diogo Proença, 75313 Diogo Martins, 75462
Bernardo Gomes, 75573

25 de Novembro de 2015

Conteúdo

1 Criação das tabelas da base de dados

Para a criação das tabelas na base de dados, as intruções de SQL utilizadas foram as seguintes:

```
create table Patient(
        number varchar(40),
        name varchar(255),
        address varchar(255),
        primary key(number));

create table PAN(
        domain varchar(255),
        phone varchar(20),
        primary key(domain));

create table Device(
        serialnum integer,
        manufacturer varchar(255),
        description varchar(255),
        primary key(serialnum, manufacturer));

create table Sensor(
        snum integer,
        manuf varchar(255),
        units varchar(255),
        primary key(snum, manuf),
        foreign key(snum, manuf)
                        references Device(serialnum, manufacturer));

create table Actuator(
        snum integer,
        manuf varchar(255),
        units varchar(255),
        primary key(snum, manuf),
        foreign key(snum, manuf)
                        references Device(serialnum, manufacturer));

create table Municipality(
        nut4code integer,
        name varchar(255),
        primary key(nut4code));

create table Period(
        start date,
        end date,
        check(start<=end),
        primary key(start, end));
```

```
create table Reading(
        snum integer,
        manuf varchar(255),
        datetime timestamp,
        value numeric(20,2),
        primary key(snum, manuf, datetime),
        foreign key(snum, manuf) references Sensor(snum, manuf));

create table Setting(
        snum integer,
        manuf varchar(255),
        datetime timestamp,
        value numeric(20,2),
        primary key(snum, manuf, datetime),
        foreign key(snum, manuf) references Actuator(snum, manuf));

create table Wears(
        start date,
        end date,
        patient varchar(255),
        pan varchar(255),
        check(start<=end),
        primary key(start, end, patient),
        foreign key(start, end) references Period(start, end),
        foreign key(patient) references Patient(number),
        foreign key(pan) references PAN(domain));

create table Lives(
        start date,
        end date,
        patient varchar(255),
        muni integer,
        check(start<=end),
        primary key(start, end, patient),
        foreign key(start, end) references Period(start, end),
        foreign key(patient) references Patient(number),
        foreign key(muni) references Municipality(nut4code));

create table Connects(
        start date,
        end date,
        snum integer,
        manuf varchar(255),
        pan varchar(255),
        check(start<=end),
        primary key(start, end, snum, manuf),
        foreign key(start, end) references Period(start, end),
        foreign key(snum, manuf) references Device(serialnum, manufacturer),
        foreign key(pan) references PAN(domain));
```

Relativamente às escolhas das variáveis o seguinte conjunto merece especial destaque:

- **variável *number* da tabela *Patient*** → ao termos levado a cabo pesquisa relativa a *Social Security Numbers* (SSN's) válidos, verificámos que este conjunto de números pode conter caracteres não numéricos (-"). Além deste facto, pode também começar pelo número "0", o que invalida o uso de variáveis *integer*, *numeric* e *decimal* caso contrário o número armazenado iria perder dígitos. Utiliza-se assim, a variável *varchar*;

- **variável *phone* da tabela *PAN*** → tendo em conta a existência de números de telefone com prefixo diferente em cada país (*e.g.* Portugal - +351), e não sendo necessário que o paciente tenha um número do país onde o *Medical Center* está localizado, utiliza-se a variável *varchar*;

- **variável *serialnum* da tabela *Device*** → utiliza-se a variável *integer*. No entanto, poder-se-ia considerar também do tipo *varchar* pelo facto de em alguns casos os números de série conterem caracteres. No entanto, não tendo sendo especificado nada no enunciado, e não tendo obtido nenhuma informação sobre os números de série de aparelhos médicos, considera-se o caso em que estes podem ser representados por um número;

- Para as tabelas *Sensor* e *Actuator* é utilizado o mesmo critério que a tabela anterior;

- **variável *nut4code* da tabela *Municipality*** → para esta tabela, considera-se apenas os códigos postais semelhantes a Portugal, identificados por quatro números, tal como o nome da variável indica;

- **variáveis *start* e *end* da tabela *Period*** → escolhe-se o tipo *date* pelo facto de as relações com esta entidade não necessitar de precisões ao nível HH:mm:ss;

- para as tabelas *Period*, *Wears*, *Lives* e *Connects* foi colocada uma verificação das datas *check(start<=end)* de forma a garantir que o período está consistente;

- **variável *datetime* das tabelas *Reading* e *Setting*** → escolhe-se o tipo *timestamp* pelo facto de as medições médicas necessitarem de precisões ao nível HH:mm:ss;

- **variável *value* das tabelas *Reading* e *Setting*** → escolhe-se o tipo *numeric* pelo facto de permitir precisão ao nível de casas decimais.

Inicialmente, acrescentam-se as seguintes instruções de forma a apagar eventuais tabelas com o mesmo nome antes da criação das novas:

```
drop table if exists Reading;
drop table if exists Setting;
drop table if exists Sensor;
drop table if exists Actuator;
drop table if exists Connects;
drop table if exists Device;
drop table if exists Lives;
drop table if exists Wears;
drop table if exists Patient;
drop table if exists PAN;
drop table if exists Municipality;
drop table if exists Period;
```

2 *Triggers* para prevenção de *overlapping periods*