UPskill

Digital Business Community

INSTITUTO DO EMPREGO
E FORMAÇÃO PROFISSIONAL

UPskill - JAVA

Bases de Dados

UPskill 2020/2021

P.PORTO

isep Instituto Superior de Engenharia do Porto

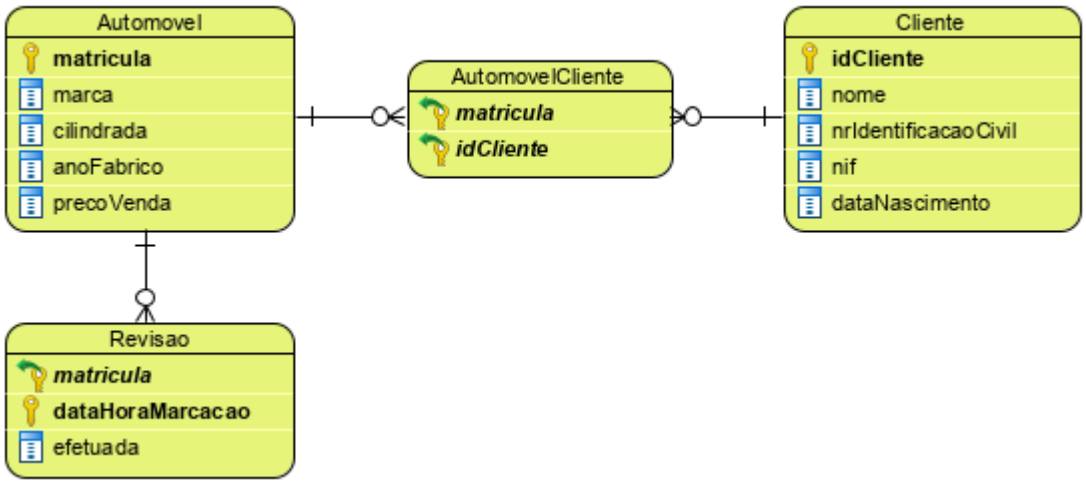
DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
Instituto Superior de Engenharia do Porto

AULA DE BASES DE DADOS

Dotar os alunos de exemplos similares aos que serão necessários para o desenvolvimento do 3º sprint.

1. CRIAÇÃO DO ESQUEMA DA BASE DE DADOS

Considerar o esquema relacional da figura seguinte e escrever todos os commandos SQL que permitem a criação desse esquema na base de dados.



1.1 Eliminar uma tabela

```
drop table Revisao;
```

1.2 Eliminar uma tabela e respetivas restrições de integridade associadas

```
drop table Revisao cascade constraints;
```

1.3 Considerações para a criação das tabelas

A. Tabela Automovel

1) Os valores do campo *matricula* têm que obedecer ao formato XX-AA-YY, XX-YY-AA ou AA-XX-YY, onde XX e YY são valores numéricos e AA são letras maiúsculas;
2) Os valores do campo *marca* não podem ser *NULL*;
3) Os valores do campo *cilindrada* têm de pertencer ao intervalo [*1000*, *6000*];
4) Os valores do campo *anoFabrico* têm de pertencer ao intervalo [*2000*, *ano atual*];
5) Os valores do campo *precoVenda* têm de ser numéricos positivos e ter no máximo 2 casas decimais.

B. Tabela Cliente

1) Os valores do campo *idCliente* têm de ser numéricos inteiros positivos e gerados automaticamente (auto-incrementados);
2) Os valores do campo *nome* não podem ser *NULL*;
3) Os valores do campo *nrIdentificacaoCivil* têm de ser numéricos inteiros positivos, ter no mínimo *6* algarismos e serem únicos. Valores *NULL* são permitidos;
4) Os valores do campo *nif* têm de ser numéricos inteiros positivos, ter sempre *9* dígitos e serem únicos. O valor NULL não é permitido.

C. Tabela Revisao

1) Os valores do campo *efetuada* só podem ser *S* e *N*, em maiúsculas ou minúsculas. Por omissão tem de ser *N*.

1.4 Criação das tabelas

```
create table Automovel(
 matricula char(8)
  constraint pkAutomovel primary key,
 marca varchar(40) constraint nnAutomovelMarca not null,
 cilindrada integer
  constraint nnAutomovelCilindrada not null
  constraint ckAutomovelCilindrada check(cilindrada between 1000 AND 6000),
 anoFabrico integer
  constraint nnAutomovelAnoFabrico not null
  constraint ckAutomovelAnoFabrico check (anoFabrico >= 2000),
 precoVenda number(10,2)
  constraint nnAutomovelPrecoVenda not null
  constraint ckAutomovelPrecoVenda check (precoVenda > 0),
 constraint ckAutomovelMatricula check (regexp_like(matricula, '^\d{2}-[A-Z]{2}-\d{2}$|^\d{2}-\d{2}-
[A-Z]{2}$|^[A-Z]{2}-\d{2}-\d{2}$'))
);

create table Cliente(
 idCliente integer generated as identity
  constraint pkClienteIdCliente primary key,
 nome varchar(40)
  constraint nnClienteNome not null,
 nrIdentificacaoCivil integer
```

```
 constraint ckClienteNrIdentificacaoCivil check(regexp_like(nrIdentificacaoCivil, '^\d{6,}$'))
 constraint ukClienteNrIdentificacaoCivil unique,
nif integer
 constraint nnClienteNif not null
 constraint ckClienteNif check(regexp_like(nif, '^\d{9}$'))
 constraint ukClienteNif unique,
dataNascimento date
 constraint nnClienteDataNascimento not null
);

create table AutomovelCliente (
 matricula char(8),
 idCliente integer,
 constraint pkAutomovelClienteMatriculaIdCliente primary key (matricula, idCliente)
);

create table Revisao(
 matricula char(8),
 dataHoraMarcacao date,
 efetuada char(1)
 constraint ckRevisaoEfetuada check(upper(efetuada) in ('S', 'N'))
);

alter table AutomovelCliente
 add constraint fkAutomovelClienteMatricula foreign key (matricula) references Automovel
(matricula);
alter table AutomovelCliente
 add constraint fkAutomovelClienteIdCliente foreign key (idCliente) references Cliente (idCliente);
alter table Revisao
 add constraint fkRevisaoMatricula foreign key (matricula) references Automovel (matricula);
```

1.5 Introdução de registos nas tabelas

Alguns dos exemplos têm erros (propositadamente), o objetivo é corrigi-los e executar todos os comandos

insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('45-PD-98', 'Mercedes', 2300, 2000, 34050);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('65-87-GR', 'Nissan', 1700, 2009, 23490.5);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('42-90-AS', 'Kia', 1300, 2008, 20870);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('BL-87-23', 'Volkswagen', 1100, 2017, 15600,75);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('83-QD-27', 'BMW', 2100, 2014, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values(XO-65-98, Toyota, 2100, 2010, 15940.123456789);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('av-23-27', 'Renault', 1100, 2009, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('23-av-27', 'Renault', 1100, 2009, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('23-28-av', 'Renault', 1100, 2009, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('23-bv-av', 'Renault', 1100, 2009, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('cv-bv-av', 'Renault', 1100, 2009, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('23-45-Av', 'Renault', 1100, 2009, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('23-78-aV', 'Renault', 1100, 2009, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('23-28-94', 'Renault', 1100, 2009, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('45-PD-98', 'BMW', 2100, 2014, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values(NULL, 'BMW', 2100, 2014, 35600);
insert into automovel(marca, cilindrada, anoFabrico, precoVenda) values('BMW', 2100, 2014, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('93-SC-27', null, 2100, 2014, 35600);
insert into automovel(matricula, cilindrada, anoFabrico, precoVenda) values('93-SC-27', 2100, 2014, 35600);

```
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('93-SC-27', 'BMW', 999, 2014, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('93-SC-27', 'BMW', 6001, 2014, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('93-SC-27', 'BMW', -2001, 2014, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('93-SC-27', 'BMW', 2100, 1999, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('93-SC-27', 'BMW', 2100, 2021, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('93-SC-27', 'BMW', 2100, -2003, 35600);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('93-SC-27', 'BMW', 2100, 2014, -10000.00);
insert into automovel(matricula, marca, cilindrada, anoFabrico, precoVenda) values('93-SC-27', 'BMW', 2100, 2014, 1234567890.5);


insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Sérgio Conceição', 987345, 105098124, TO_DATE('1974-11-15', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('António Oliveira', 937587, 104052455, DATE('1952-10-06', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Fernando Santos', NULL, 102000906, TO_DATE('1954-10-10'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Artur Jorge', 7098428, 100829087, TO_DATE('1946-22-13','yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Jesualdo Ferreira', NULL, 107559969, TO_DATE('1946-05-24','yyyy-mm-dd'));
insert into cliente(idCliente, nome, nrIdentificacaoCivil, nif, dataNascimento) values(10, 'Jorge Jesus', 870598, 105727913, TO_DATE('1954-06-24', 'yyyy-mm-dd'));
insert into cliente(idCliente, nome, nrIdentificacaoCivil, nif, dataNascimento) values(NULL, 'Jorge Jesus', 890146, 107559369, TO_DATE('1954-06-24', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values(NULL, 890146, 107559369, TO_DATE('1954-06-24', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Jorge Jesus', 937587, 107559369, TO_DATE('1954-06-24', 'yyyy-mm-dd'));
```

```
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Jorge Jesus', 89014, 107559369, TO_DATE('1954-06-24', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Jorge Jesus', -890146, 107559369, TO_DATE('1954-06-24', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Jorge Jesus', 987345, 107559369, TO_DATE('1954-06-24', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Jorge Jesus', 890146, NULL, TO_DATE('1954-06-24', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Jorge Jesus', 890146, -107559369, TO_DATE('1954-06-24', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Jorge Jesus', 890146, 10755936, TO_DATE('1954-06-24', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Jorge Jesus', 890146, 1075593692, TO_DATE('1954-06-24', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Jorge Jesus', 890146, 1075593692, TO_DATE('1954-16-24', 'yyyy-mm-dd'));
insert into cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values('Jorge Jesus', 890146, 1075593692, TO_DATE('1954-02-30', 'yyyy-mm-dd'));

insert into automovelCliente(matricula, idCliente) values('65-87-GR', 1);
insert into automovelCliente(matricula, idCliente) values('83-QD-27', 4);
insert into automovelCliente(matricula, idCliente) values('42-90-AS', 2);
insert into automovelCliente(matricula, idCliente) values('45-PD-98', 1);
insert into automovelCliente(matricula, idCliente) values('XO-65-98', 6);
insert into automovelCliente(matricula, idCliente) values('BL-87-23', 3);
insert into automovelCliente(matricula, idCliente) values('65-GR-87', 1);
insert into automovelCliente(matricula, idCliente) values(NULL, 1);
insert into automovelCliente(matricula, idCliente) values('65-GR-87', NULL);
insert into automovelCliente(matricula, idCliente) values(NULL, NULL);
insert into automovelCliente(matricula, idCliente) values('65-GR-87', 8);
insert into automovelCliente(matricula, idCliente) values('75-GR-87', 2);

insert into revisao(matricula, dataHoraMarcacao) values('65-87-GR', to_date('2018-10-04 09:00:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into revisao(matricula, dataHoraMarcacao) values('83-QD-27', to_date('2018-11-04 14:45:00', 'yyyy-mm-dd hh24:mi:ss'));
```

```
insert into revisao(matricula, dataHoraMarcacao) values('42-90-AS', to_date('2018-10-23 10:50:00', 'yyyy-mm-dd hh24:mi:ss'), 'N');
insert into revisao(matricula, dataHoraMarcacao, efetuada) values('XO-65-98', to_date('2018-12-01 18:30:00', 'yyyy-mm-dd hh24:mi:ss'), 'n');
insert into revisao(matricula, dataHoraMarcacao, efetuada) values('65-87-GR', to_date('2015-06-07 10:50:00', 'yyyy-mm-dd hh24:mi:ss'), 'S');
insert into revisao(matricula, dataHoraMarcacao, efetuada) values('XO-65-98', to_date('2016-11-22 12:20:00', 'yyyy-mm-dd hh24:mi:ss'), 'x');
insert into revisao(matricula, dataHoraMarcacao) values('42-90-AS', to_date('2018-10-23 10:50:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into revisao(matricula, dataHoraMarcacao) values(NULL, to_date('2018-10-23 10:50:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into revisao(matricula, dataHoraMarcacao) values('42-90-AS', NULL);
insert into revisao(matricula, dataHoraMarcacao) values(NULL, NULL);
insert into revisao(matricula, dataHoraMarcacao) values('28-90-AS', to_date('2020-10-23 10:50:00', 'yyyy-mm-dd hh24:mi:ss'));
insert into revisao(matricula, dataHoraMarcacao, efetuada) values('42-90-AS', to_date('2007-10-23 10:50:00', 'yyyy-mm-dd hh24:mi:ss'), 'x');
insert into revisao(matricula, dataHoraMarcacao, efetuada) values('42-90-AS', to_date('2007-10-23 10:50:00', 'yyyy-mm-dd hh24:mi:ss'), 'X');
insert into revisao(matricula, dataHoraMarcacao, efetuada) values('42-90-AS', to_date('2007-10-23 10:50:00', 'yyyy-mm-dd hh24:mi:ss'), NULL);
```

1.6 Exemplos sobre utilização de cursores (usando blocos anónimos)

```
declare
 cursor c is select matricula, dataHoraMarcacao from Revisao;
 r c%rowtype;
begin
 open c;  --executar o select e colocar em memória o conjunto obtido
 loop
  fetch c into r;
  if c%found then
   dbms_output.put_line('Matrícula: '||r.matricula||', data de marcação: '||r.dataHoraMarcacao);
  else
   exit;
  end if;
```

```
 end loop;
 close c;
end;
/

declare
 cursor c is select matricula, dataHoraMarcacao from Revisao;
 r c%rowtype;
begin
 open c;  --executar o select e colocar em memória o conjunto obtido
 loop
  fetch c into r;
  exit when c%notfound;
  dbms_output.put_line('Matrícula: '||r.matricula||', data de marcação: '||r.dataHoraMarcacao);
 end loop;
 close c;
end;
/

--for cursor
begin
 for r in (select matricula, dataHoraMarcacao from Revisao)
 loop
  dbms_output.put_line('Matrícula: '||r.matricula||', data de marcação: '||r.dataHoraMarcacao);
 end loop;
end;
/

--cursor com parâmetros
declare
 cursor c(p_ano int) is
  select matricula, marca
   from Automovel
  where anoFabrico = p_ano;
 r c%rowtype;
begin
 open c(2009);  --executar o select e colocar em memória o conjunto obtido
```

```
 loop
  fetch c into r;
  exit when c%notfound;
  dbms_output.put_line('Matrícula: '||r.matricula||', marca: '||r.marca);
 end loop;
 close c;
 open c(2014);  --executar o select e colocar em memória o conjunto obtido
 loop
  fetch c into r;
  exit when c%notfound;
  dbms_output.put_line('Matrícula: '||r.matricula||', marca: '||r.marca);
 end loop;
 close c;
end;
/
```

1.7 Procedimentos que devolvem cursores em parâmetros de output

```
create procedure procGetAutomoveis(p_out out sys_refcursor) is
begin
 open p_out for
  select * from Automovel;
end;
/

declare
 v_cur sys_refcursor;
 r Automovel%rowtype;
begin
 procGetAutomoveis(v_cur);
 fetch v_cur into r;
 while v_cur%found loop
  dbms_output.put_line('Matrícula: '||r.matricula||', marca: '||r.marca);
  fetch v_cur into r;
 end loop;
 close v_cur;
```

```
end;
/
```

1.8 Funções que retornam cursores

```
create function funcGetAutomoveis return sys_refcursor is
 v_ret sys_refcursor;
begin
 open v_ret for
  select * from Automovel;
 return v_ret;
end;
/

declare
 v_cur sys_refcursor;
 r Automovel%rowtype;
begin
 v_cur := funcGetAutomoveis;
 fetch v_cur into r;
 while v_cur%found loop
  dbms_output.put_line('Matrícula: '||r.matricula||', marca: '||r.marca);
  fetch v_cur into r;
 end loop;
 close v_cur;
end;
/

select funcGetAutomoveis from dual;
```

1.9 Função para obter o número de veículos fabricados num determinado ano

```
create or replace function getNumeroVeiculosAnoFabrico(p_ano int) return int
is
 v_ret int;
begin
 select count(*) into v_ret
```

```
  from Automovel
 where anofabrico = p_ano;
return v_ret;
end;
/

select getNumeroVeiculosAnoFabrico(2009) from dual;
```

1.10 Função para obter a data da última revisão de uma determinada viatura.

```
create or replace function getDataUltimaRevisao(p_matricula automovel.matricula%type) return
date
is
 v_ret date;
begin
 select max(dataHoraMarcacao) into v_ret
  from Revisao
 where matricula = p_matricula;
return v_ret;
end;
/

select getDataUltimaRevisao('65-87-GR') from dual;
```

1.11 Função para obter um conjunto com toda a informação sobre os automóveis que um determinado cliente possui.

```
create or replace function getAutomoveisCliente(p_cliente cliente.idCliente%type) return
sys_refcursor
is
 v_ret sys_refcursor;
begin
 open v_ret for
  select a.*
   from Automovel a join AutomovelCliente b on (a.matricula = b.matricula)
  where b.idCliente = p_cliente;
 return v_ret;
```

```
end;
/

select getAutomoveisCliente(1) from dual;
```

1.12 Função para criar um cliente e devolver o respetivo idCliente (gerado automaticamente)

```
create or replace function createCliente(
 p_nome cliente.nome%type
, p_nidc cliente.nrIdentificacaoCivil%type
, p_nif  cliente.nif%type
, p_data cliente.dataNascimento%type) return cliente.idCliente%type
is
 v_id cliente.idCliente%type;
begin
 insert into Cliente(nome, nrIdentificacaoCivil, nif, dataNascimento) values(p_nome, p_nidc, p_nif,
p_data) returning idCliente into v_id;
 return v_id;
end;
/

declare
 v_id int;
begin
 v_id := createCliente('Teste', 121212, 121212121, to_date('1950-01-01', 'yyyy-mm-dd'));
 dbms_output.put_line('Novo id cliente: '||v_id);
end;
/
```

1.13 Acrescentar a data de compra à tabela Automovel.

```
alter table Automovel add(dataCompra date);
```

1.14 Procedimento para criar um automóvel (compra por parte de um cliente) e associá-lo a um cliente.

```
create or replace procedure createAutomovel(
 p_matricula automovel.matricula%type
, p_marca Automovel.marca%type
, p_cilindrada Automovel.cilindrada%type
, p_ano Automovel.anoFabrico%type
, p_preco Automovel.precoVenda%type
, p_data Automovel.dataCompra%type
, p_cliente Cliente.idCliente%type)
is
 v_count int;
 ex_cliente exception;
begin
 --Validar o cliente
 select count(*) into v_count
  from Cliente
  where idCliente = p_cliente;
 if v_count = 0 then
  raise ex_cliente;
 end if;
 insert into Automovel(matricula, marca, cilindrada, anoFabrico, precoVenda, dataCompra)
 values(p_matricula, p_marca, p_cilindrada, p_ano, p_preco, p_data);
 insert into AutomovelCliente(matricula, idCliente) values(p_matricula, p_cliente);
exception
 when ex_cliente then
  raise_application_error(-20001, 'Cliente inexistente');
end;
/

begin
 createAutomovel('11-11-AA', 'BMW', 2500, 2020, 30000, sysdate, 1);
end;
/
```

1.15 Função para obter a próxima hora livre para uma revisão, a partir de uma determinada data. Uma revisão demora (normalmente) 60 minutos. A oficina funciona nos dias úteis entre as 8:00 e as 19:00, e não sendo possível obter um horário para esse dia, deve obter o dia útil seguinte. A hora é automaticamente definida como sendo 1 hora após a hora da revisão anterior (de qualquer outro veículo) nesse dia.

```
create or replace function getNextDataHoraRevisao(p_data date) return date is
 v_dia date;
 v_ret date;
 v_hora int;
 v_count int;
 v_proprio_dia boolean := true;
begin
 --Obter dia útil.
 v_dia := trunc(p_data);
 v_hora := to_char(p_data, 'hh24')+1;
 <<main_loop>>
 loop
  if to_char(v_dia, 'D') in (2, 3, 4, 5, 6) then
   if (not v_proprio_dia) or (v_hora < 8) then
    v_hora := 8;
   end if;
   while v_hora <= 18 loop
    select count(*) into v_count
     from Revisao
     where dataHoraMarcacao between v_dia+(v_hora/24) and v_dia+(v_hora+1/24);
    if v_count = 0 then
     v_ret := v_dia+(v_hora/24);
     exit main_loop;
    else
     v_hora := v_hora + 1;
    end if;
   end loop;
  end if;
  v_dia := v_dia + 1;
  v_proprio_dia := false;
 end loop;
```

```
 return v_ret;
end;
/

select getNextDataHoraRevisao(trunc(sysdate)+30) from dual;
```

1.16 Trigger para criar a primeira revisão de um automóvel. A primeira revisão é criada automaticamente quando o automóvel é comprado (30 dias após a data da compra).

```
create or replace trigger trgPrimeiraRevisao after insert on Automovel for each row
declare
 v_datahora date;
begin
 v_datahora := trunc(:new.dataCompra)+30; --30 dias após a data de compra
 v_datahora := getNextDataHoraRevisao(v_datahora); --Primeira hora livre desse dia (ou dias seguintes)
 insert into Revisao(matricula, dataHoraMarcacao, efetuada) values(:new.matricula, v_datahora, 'n');
end;
/

begin
 createAutomovel('11-11-DD', 'BMW', 2500, 2020, 30000, sysdate, 1);
end;
/
```

1.17 Trigger para impedir que a data de nascimento de um cliente seja maior do que a data atual.

```
create or replace trigger trgValidaCliente after insert or update on Cliente for each row
begin
 if trunc(:new.dataNascimento) >= trunc(sysdate) then
  raise_application_error(-20000, 'Erro na data de nascimento do cliente');
 end if;
end;
/
```