



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

Licenciatura de Engenharia Informática e de Computadores

Introdução a Sistemas de Informação

Semestre de Inverno 2022-2023

Transporte de passageiros

Trabalho prático
(Fase 2)

Afonso Remédios, Matilde Pato e Nuno Datia

Planeamento

As datas importantes a recordar são:

- Lançamento do enunciado: **16 de Outubro 2022**
- Entrega intermédia (Fase 1): **14 de Outubro de 2022**
- Entrega intermédia (Fase 2): **2 de Dezembro de 2022**
- Entrega intermédia (Fase 3): **6 de Janeiro de 2023**

Cada entrega intermédia deve apresentar o relatório e código (se houver) referentes **exclusivamente** a essa fase. O relatório deve seguir um dos *templates* fornecidos, obrigatoriamente, sob pena de penalização. Este deve ser conciso e apresentar a justificação de todas as decisões tomadas (ver Critérios de Avaliação). A capa do relatório deve indicar a composição do grupo, a unidade curricular e a fase do trabalho que relata. Caso tenha adendas e/ou correcções a fazer a modelos já entregues, deve indicá-las de forma explícita no relatório seguinte.

O pdf (e, o zip) gerado deve seguir o nome da seguinte forma: 'TPNISI-2223SI-Grupo-**AP-NN**.ext' (AP representa o acrónimo do docente, NN representa um dígito, e 'ext' a extensão do ficheiro), e.g.: TP1ISI-2223SI-Grupo-mp-01.pdf.

16 de Outubro de 2022, Afonso Remédios, Matilde Pato e Nuno Datia

Objectivos de aprendizagem

No final da **segunda fase do trabalho**, os alunos devem ser capazes de:

- Utilizar correctamente a álgebra relacional, com os seus vários operadores, para expressar interrogações sobre um modelo relacional:
 - operadores relacionais unários: `select`, `project` e `rename`;
 - operadores relacionais binários: `division`;
 - operadores sobre conjuntos: `difference`, `intersect` e `union`;
 - operadores de junção: interna e externa;
 - operador de agregação;
- Utilizar correctamente SQL/DDL para criar as tabelas num sistema de gestão de bases de dados (SGDB);
- Garantir as restrições de integridade identificadas enumerando aquelas que têm de ser garantidas pelas aplicações;
- Inserir dados em lote através da cláusula SQL `INSERT`, garantindo que as restrições de integridade são cumpridas;
- Garantir a atomicidade de instruções, utilizando processamento transaccional;
- Utilizar correctamente os operadores da teoria dos conjuntos em PostgreSQL;
- Utilizar correctamente as cláusulas `INNER JOIN` e `OUTER JOIN`;
- Utilizar correctamente sub-interrogações correlacionadas;
- Utilizar correctamente funções de agregação;
- Utilizar correctamente a cláusula `HAVING`;
- Utilizar correctamente a cláusula `ORDER BY`;

- Utilizar correctamente o termo `DISTINCT`;
- Utilizar correctamente os predicados `IN` e `EXISTS`.

Após a realização da 1ª fase do trabalho, segue-se a implementação do modelo físico do sistema, i.e. deverá ser construído em PostgreSQL contemplando todas as restrições que consigam garantir na forma declarativa.

Nota: Deverão preencher a base de dados com informação necessária que permita, em seguida, realizar interrogações que apresentem resultados pertinentes. Na etapa de preenchimento da base de dados, os alunos deverão ter particular atenção ao cumprimento das **restrições de integridade, utilizando de forma adequada o controlo transaccional (a atomicidade)**.

Resultados pretendidos

Tendo em conta os objectivos de aprendizagem, deverão ser produzidos os seguintes resultados:

1. Construção do modelo físico do sistema, contemplando todas as restrições de integridade passíveis de ser garantidas declarativamente, assim como a **atomicidade** nas operações. O código PostgreSQL que permite:
 - (a) Criar o modelo físico (1 *script* autónomo): `createTable.sql`;
 - (b) Remover o modelo físico (1 *script* autónomo): `removeTable.sql`;
 - (c) Preenchimento inicial da base de dados (1 *script* autónomo): `insertTable.sql`.
Os dados introduzidos devem permitir validar todas as interrogações pedidas nesta fase do trabalho;
 - (d) Apagar todos os dados existentes e remover as tabelas (1 *script* autónomo): `deleteTable.sql`.
2. Considerando o esquema relacional obtido anteriormente e fornecido no final deste documento ("Adenda"), apresente as expressões em álgebra relacional (AR) que respondam às seguintes alíneas. Por forma a recorrer a todos os operadores, apresente **2 soluções alternativas** para a mesma questão.
 - (a) Para o cliente de nome "António Silva", pretende-se saber o número de identificação, o número de identificação fiscal (NIF), e as viagens que efectuou, nomeadamente a sua data e as horas de início e de fim, e o preço final desta.

- (b) Pretende-se obter os registos de todos os condutores (nome e número de identificação) que estão aptos a conduzir carros do dono de nome “Bruno Ferreira” mas que não estão aptos a conduzir carros do dono “Carlos Brito”.
 - (c) Apresente a lista do(s) cliente(s) (o id, o nome e o NIF), com mais viagens no ano de 2021¹.
 - (d) Pretende-se saber quem foram os condutores (o id, o nome e o NIF) que nunca efectuaram viagens.
 - (e) Liste os pares (nome de condutor, número de viagens) relativos ao ano de 2021.
3. Conceba, na linguagem PostgreSQL, as interrogações que produzam os resultados a seguir indicados, utilizando apenas uma instrução PostgreSQL. Guarde todas num *script* autónomo de nome “queries.sql”. **Para cada instrução deve ser também apresentada, em comentário, a descrição do raciocínio seguido.**
- (a) Implemente em PostgreSQL as interrogações pedidas na alínea 2 (Pode escolher uma das soluções que apresentou acima, desde que nesta fase consiga fazer uso de todos os operadores).
 - (b) Dado o número de identificação fiscal, de um proprietário, e o ano, apresente o número de viagens obtidas pelos seus carros.
 - (c) Dada a indicação de um ano, apresente os dados do condutor cujo total de viagens nesse ano teve o maior custo final acumulado. O resultado deve apresentar o nome, número de identificação e morada.
 - (d) Liste os pares (nome do condutor, total de viagens) que no ano de 2021 efectuaram um total de viagens inferior a 3.
 - (e) Crie uma vista CONDUTORVIAGENSPORANO que permita produzir, para cada condutor em cada ano, o total de viagens e a soma dos preços finais dessas viagens.
 - (f) Crie uma tabela para guardar as estatísticas relativas ao número de viagens anuais de cada cliente. Apresente o código SQL da criação da tabela e o código que popula a tabela com os dados de 2021. Deve garantir que não surgirão inconsistências nos dados.
4. Apresente o(s) comando(s) que permite(m): a) inserir em TIPOVEICULO o tuplo (4, Premium, 4); b) afectar este novo tuplo ao(s) veículo(s) já registado(s) na BD com o modelo RANGE ROVER. **Devem garantir atomicidade nas alterações.**

Todas as simplificações e optimizações realizadas ao modelo devem ser indicadas e justificadas.

¹Nota: Poderão fazer uso da função YEAR ()

PS.1 Sugere-se que consulte o manual do SGDB para obter informação sobre as funções de manipulação de datas.

Data limite para entrega: 2 de Dezembro de 2022 até às 23:59.

A entrega deve incluir um documento com as respostas de AR (em formato PDF) e o código PostgreSQL (com extensão .sql). Os ficheiros produzidos **deverão ser** comprimidos e enviados de forma electrónica através do Moodle.

Nota: Deve ser possível aferir cada um dos objectivos de aprendizagem no material que entregar.

LEIC

Modelo Relacional a usar no trabalho prático (fase 2)

Introdução a Sistemas de Informação

Semestre de Inverno 2022/2023

Versão 1.0

Modelo Relacional

Todos os atributos são obrigatórios nas relações, excepto quando indicado o contrário. As relações são apresentadas por ordem alfabética. **Nota:** Poderão não estar representadas todas as restrições de integridade, uma vez que esta componente também é avaliada. Mais, as restrições que não conseguiremos implementar no modelo físico serão asseguradas, posteriormente, na API que irão desenvolver na 3ª parte.

CLIENTEVIAGEM

CLIENTEVIAGEM(idpessoa, viagem).

Atributo	Tipo	Restrições Integridade
idpessoa	integer	FK referência de PESSOA.{id}.
viagem	integer	FK referência de VIAGEM.{idsistema}.

CONDHABILITADO

CONDHABILITADO(condutor, veiculo).

Atributo	Tipo	Restrições Integridade
condutor	integer	FK referência de CONDUTOR.{idpessoa}.
veiculo	integer	FK referência de VEICULO.{id}.

CONDUTOR

CONDUTOR(idpessoa, ncconducao, dtnascimento).

Atributo	Tipo	Restrições Integridade
idpessoa	integer	FK referência de PESSOA.{id}.
ncconducao	varchar(20)	Tem o formato CC-DDDDDDDD, onde C representa uma letra e D um inteiro. Valor único.

continua na próxima página

Atributo	Tipo	Restrições Integridade
dtnascimento	date	O condutor deve ter no mínimo 18 anos de idade à data actual.

CONTACTO

CONTACTO(idpessoa, ntelefone).

Atributo	Tipo	Restrições Integridade
idpessoa	integer	FK referência de PESSOA.{id}.
ntelefone	varchar(10)	Pode ser fixo ou móvel, inclui o prefixo do país.

CORVEICULO

CORVEICULO(veiculo, cor).

Atributo	Tipo	Restrições Integridade
veiculo	integer	FK referência de VEICULO.{id}.
cor	varchar(10)	

PERIODOACTIVO

PERIODOACTIVO(veiculo, condutor, dtinicio, dtfim, lat, long).

RI: O período só pode ser criado para um par Condutor-Veículo apto a conduzir o veículo.

Atributo	Tipo	Restrições Integridade
veiculo	integer	FK referência de VEICULO.{id}.
condutor	integer	FK referência de CONDUTOR.{idpessoa}.
dtinicio	datetime	Tem o formato “aaaa-mm-dd hh:mm:ss”. Valor deve ser inferior à data actual.
dtfim	datetime	Tem o formato “aaaa-mm-dd hh:mm:ss”. Valor superior a dtinicio. O atributo pode tomar valor NULL.

continua na próxima página

Atributo	Tipo	Restrições Integridade
lat	decimal(6,2)	Corresponde à latitude numa coordenada GPS, da última localização do veículo.
long	decimal(6,2)	Corresponde à longitude numa coordenada GPS, da última localização do veículo.

PESSOA

PESSOA(id, noident, nif, nproprio, apelido, morada, codpostal, localidade, atrdisc).

Atributo	Tipo	Restrições Integridade
id	integer	Foi adicionado este atributo como identificador de PESSOA por questões de simplificação. Valor sequencial.
noident	char(12)	Número de identificação (cartão de cidadão ou passaporte). Por questões de simplificação, optou-se por considerar apenas um dos números para identificação. Chave candidata.
nif	char(12)	Número de identificação fiscal. Outra chave candidata.
nproprio	varchar(10)	
apelido	varchar(15)	
morada	varchar(150)	
codpostal	integer	Valor que contém 7 dígitos, que compõem o código postal sem hífen.
localidade	varchar(150)	
atrdisc	char(2)	Atributo discriminante que contém "P" para representar o proprietário, "C" para condutor e "CL" a clientes.

PROPRIETARIO

PROPRIETARIO(idpessoa, dtnascimento).

Atributo	Tipo	Restrições Integridade
idpessoa	integer	FK referência de PESSOA.{id}.

continua na próxima página

Atributo	Tipo	Restrições Integridade
dtnascimento	date	A pessoa deve ter no mínimo 18 anos de idade à data actual.

TIPOVEICULO

TIPOVEICULO(tipo, nlugares, multiplicador, designacao).

Atributo	Tipo	Restrições Integridade
tipo	integer	Valor sequencial.
nlugares	integer	O valor é positivo e inferior a 8.
multiplicador	numeric(1)	
designacao	char(10)	Poderá tomar como valores "Normal, XL, Luxo".

VEICULO

VEICULO(id, matricula, tipo, modelo, marca, ano, proprietario).

Atributo	Tipo	Restrições Integridade
id	integer	Foi adicionado este atributo como identificador de VEICULO por questões de simplificação. Valor sequencial.
matricula	varchar(10)	Tem o formato "CCDDCC" ou "DDCCDD", onde C representa uma letra e D um dígito. Chave candidata.
tipo	integer	FK referência a TIPOVEICULO.{tipo}.
modelo	varchar(10)	
marca	varchar(10)	
ano	integer	A idade do veículo deve ser inferior a 5 (inclusivé) à data do registo.
proprietario	integer	FK referência a PROPRIETARIO.{idpessoa}. Cada proprietário tem no máximo 20 veículos.

VIAGEM

VIAGEM(idsistema, inicio, hfim, dtviagem, valestimado, valfinal, latinicio, longinicio, latfim, longfim, classificacao, veiculo, condutor, dtinicio).

Atributo	Tipo	Restrições Integridade
idsistema	integer	Valor sequencial.
inicio	time	Tem o formato "hh:mm:ss".
hfim	time	Tem o formato "hh:mm:ss". Valor superior a inicio. Poderá estar a NULL se a viagem estiver a decorrer.
dtviagem	date	Tem o formato "aaaa-mm-dd".
valestimado	decimal(5,2)	Corresponde ao valor estimado em euros.
valfinal	decimal(5,2)	Corresponde ao valor final em euros. Poderá estar a NULL se a viagem estiver a decorrer.
latinicio	decimal(6,2)	Corresponde à latitude numa coordenada GPS.
longinicio	decimal(6,2)	Corresponde à longitude numa coordenada GPS.
latfim	decimal(6,2)	Corresponde à latitude numa coordenada GPS. Deve ser diferente de latinicio. Deverá estar a NULL se hfim também o estiver.
longfim	decimal(6,2)	Corresponde à longitude numa coordenada GPS. Deve ser diferente de longinicio. Deverá estar a NULL se hfim também o estiver.
classificacao	integer	Corresponde à classificação dada pelo cliente após a viagem. Valores possíveis 1 a 5.
veiculo	integer	FK referência de PERIODOACTIVO.{veiculo}.
condutor	integer	FK referência de PERIODOACTIVO.{condutor}.
dtinicio	datetime	FK referência de PERIODOACTIVO.{dtinicio}.

15/11/2022, Afonso Remédios, Matilde Pato e Nuno Datia