

Relatório do Trabalho Prático 1

Instituto Superior de Engenharia do Porto

Base de Dados

2020/2021

1181743 Guilherme Daniel

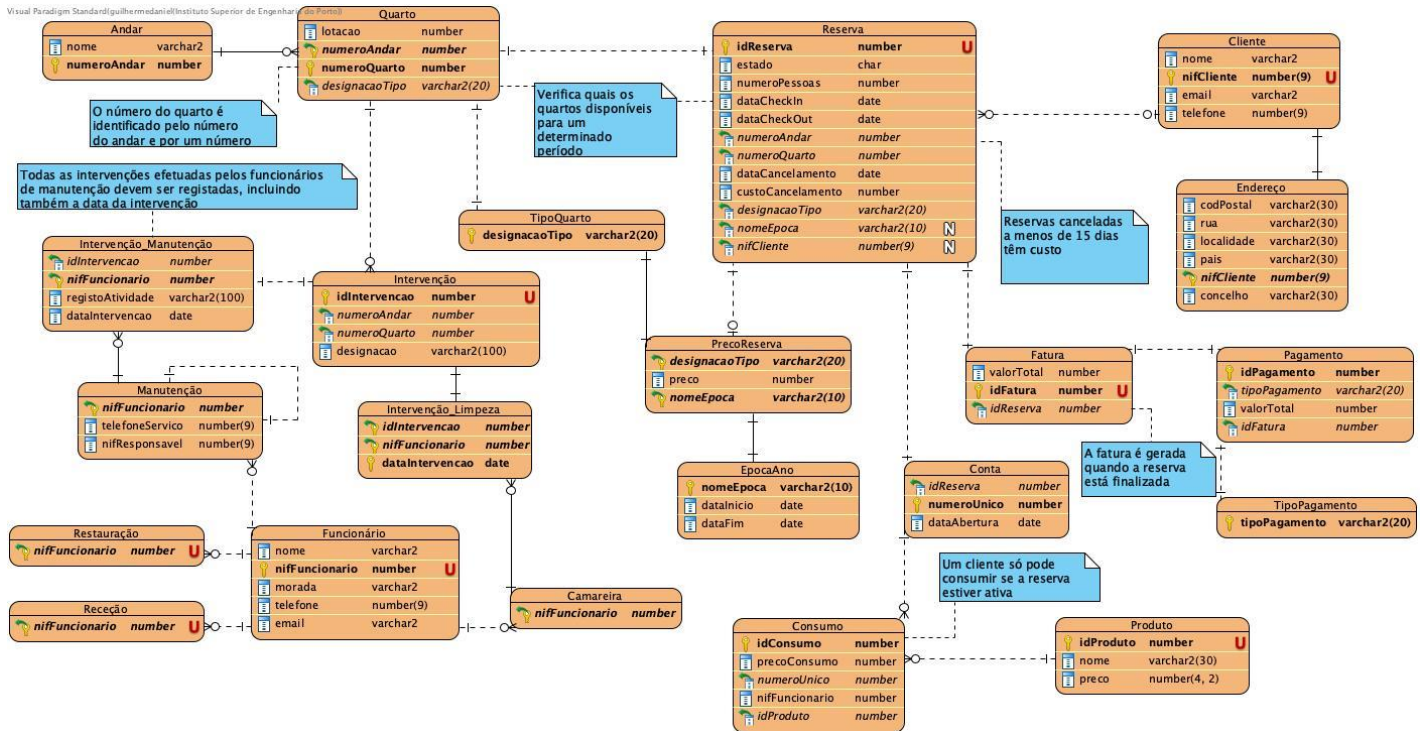
1181853 Kevin Sousa

1171589 Lucas Sousa

Licenciatura em Engenharia Informática

Novembro de 2020

Modelo Relacional:



Decisões tomadas em relação ao modelo relacional:

Uma **reserva** tem um e um só **quarto** associado, e cada quarto tem um e um só tipo de quarto (**tipoQuarto**). Tem também um preço, a ser pago no final. Este preço (**PrecoReserva**), depende de um **tipo de quarto** e da **época** em que se encontra. Quem efetua uma **reserva** é um **cliente**, e um **cliente** pode efetuar **várias reservas**. Cada **cliente** tem um **endereço** associado, que é onde este habita.

Quando um cliente se encontra hospedado no hotel, pode fazer **vários consumos de produtos no frigo-bar**, pelo que foi criada uma entidade **Produto**, que contém as informações de todos os produtos, e depois uma entidade **Consumo**, encarregue de registar os produtos consumidos pelo cliente, registo tal feito por uma camareira. A cada consumo está associada uma **conta**, e uma conta pode ter vários consumos, sendo que um cliente pode efetuar vários consumos em dias diferentes.

No hotel existem também vários funcionários. Sabendo que a informação essencial é comum a todos os funcionários, foi criada uma entidade apenas designada **Funcionario**, a esta entidade **conectam os diferentes tipos: restauração, manutenção, camareira e receção**.

Os dois funcionários a destacar são o de **manutenção** e a **camareira**. O funcionário de **manutenção** pode (ou não), ter um **responsável**, que corresponde também a um funcionário de manutenção, daí a ligação para ele próprio. O **funcionário camareira e manutenção** podem **efetuar intervenções**, limpeza e manutenção, respetivamente. Foi criada uma entidade **Intervenção**, de modo a ser possível obter o identificador, número do quarto e a designação de cada intervenção, e a esta foi ligada uma intervenção limpeza e manutenção.

A **intervenção limpeza** não necessita de ser registada, contudo contém uma **dataIntervencao** para saber quando foi efetuada. O **funcionário de manutenção** é obrigado a registar a intervenção, e a data em que foi efetuada.

Por fim, é gerada uma **fatura** no fim da estadia, com todos os consumos efetuados e com o valor do quarto, que é paga (**pagamento**), de uma ou várias maneiras (**tipoPagamento**).

Consultas:

Todo o código das consultas a serem demonstradas encontram-se no ficheiro submetido, com o nome **CONSULTAS**.

De forma a provar a criação das tabelas, restrições triviais e respetivos atributos, serão apresentadas algumas consultas consideradas relevantes para a elaboração do trabalho. Inicialmente, serão mostradas apenas listagens simples, apenas das entidades principais, para mostrar que realmente estas estão registadas na base de dados.

Consulta de Andares, Tipos de Quarto e Quartos:

DESIGNACAOTIPO
1 presidencial
2 single
3 suite
4 superior
5 twin

Figura 1- Tipos de quarto

NOME	NUMEROANDAR
1 paraíso	1
2 luxo	2

Figura 2- Andares

ID	LOTACAO	NUMEROANDAR	NUMEROQUARTO	DESIGNACAOTIPO
1	1	3	1	10001single
2	2	4	1	10002single
3	3	1	1	10003single
4	4	3	1	10004single
5	5	2	1	10005single
6	6	1	1	10006single
7	7	2	1	10007single
8	8	3	1	10008single
9	9	1	1	10009single
10	10	1	1	10010single
11	11	4	1	10011twin
12	12	6	1	10012twin
13	13	10	1	10013twin
14	14	4	1	10014twin
15	15	5	1	10015twin
16	16	9	1	10016twin
17	17	6	1	10017twin
18	18	2	1	10018twin
19	19	3	1	10019twin
20	20	7	1	10020twin
21	21	10	1	10021superior
22	22	12	1	10022superior
23	23	7	1	10023superior
24	24	6	1	10024superior
25	25	5	1	10025superior
26	26	9	1	10026superior
27	27	6	1	10027superior
28	28	4	1	10028superior
29	29	3	1	10029superior
30	30	5	1	10030superior
31	31	13	1	10031suite
32	32	10	1	10032suite
33	33	6	1	10033suite
34	34	10	1	10034suite
35	35	7	1	10035suite
36	36	9	1	10036suite
37	37	6	1	10037suite
38	38	7	1	10038suite
39	39	5	1	10039suite
40	40	9	1	10040suite

Figura 3- Quartos presentes no primeiro andar

A demonstração das tabelas serve única e exclusivamente para comprovar a existência de quartos na base de dados. É de realçar o facto de que apenas foram adicionados 5 tipos de quartos e 2 andares.

Também é possível obter uma tabela de quartos disponíveis/indisponíveis numa determinada data (para o exemplo, será utilizada a data atual 20-11-2020)

NUMEROANDAR	NUMEROQUARTO	DATA CHECK-IN	DATA CHECK-OUT	NOME	NIF CLIENTE	ESTADO
1	2	20005 20.11.15	20.11.24	Joao Felix	133333335	ATIVA
2	2	20020 20.11.18	20.11.26	Filipe Albuquerque	127777776	ATIVA

Figura 4- Quarto, data de check-in e check-out e cliente a ocupar o quarto de reservas ativas

	NUMEROANDAR	NUMEROQUARTO
1	1	10001
2	1	10002
3	1	10003
4	1	10004
5	1	10005
6	1	10006
7	1	10007
8	1	10008
9	1	10009
10	1	10010

Figura 5- Excerto de quartos livres para a data atual

Consultas de Clientes:

	NOME	NIFCLIENTE	EMAIL	TELEFONE
1	Antonio Silva	100000000	tone@gmail.com	(null)
2	Marcos Marques	100000001	(null)	910001111
3	Mariana Moreira	111111111	mary@gmail.com	(null)
4	Tatiana Neves	111111112	Taty@hotmail.com	(null)
5	Fabio Martins	111111113	(null)	911223344
6	Filipa Meireles	111111141	filper@gmail.com	(null)
7	Catarina Romanova	111116141	romanova@gmail.com	(null)
8	Ricardo Couto	111016141	(null)	911111614
9	Joana Alencar	111116147	alencar@gmail.com	(null)
10	Nuno Braga	111816141	FCBraga@gmail.com	(null)
11	Sara Santos	121816141	(null)	911816141
12	Afonso Almeirim	131816141	madeiras@gmail.com	(null)
13	Alice Pereira	131816147	(null)	931816147
14	Paulo Pereira	131716147	(null)	931716147
15	Ana Isabel	111111121	(null)	911111121

Figura 6- Excerto de 15 clientes presentes na base de dados

Ao todo foram inseridos 60 clientes. Na figura apenas estão presentes 15 deles.

Pelo excerto em cima, é possível reparar que se um cliente é caracterizado por um e-mail, ele não terá número de telefone associado, e vice-versa, sendo que cada um só deverá ter ou um ou outro, e não os dois em simultâneo.

Consultas Funcionários:

	NOME	NIFFUNCIONARIO	TIPO_FUNCIONARIO
1	Amelia Rocha	876767676	Rececao
2	Ana Costa	990012132	Manutencao
3	Carla Correira	232345643	Camareira
4	Fernando Medina	411109878	Manutencao
5	Guilherme Marques	199999878	Restauracao
6	Helena Marinho	333888111	Manutencao
7	Inês Martins	777899887	Rececao
8	Joao Tavares	400009878	Manutencao
9	João Costa	555432321	Camareira
10	Maria Sousa	321432321	Camareira
11	Marisa Mota	656543432	Camareira
12	Marta Couto	229989092	Manutencao
13	Martim Manha	333000333	Manutencao
14	Mauricio Castelo	151514140	Manutencao
15	Miguel Rocha	212290921	Manutencao
16	Ricardo Matias	323434432	Camareira
17	Rodrigo Loures	333787787	Restauracao

Figura 7- Funcionários presentes na base de dados, juntamente com a atividade a que pertencem

Para cada funcionário, corresponde uma e uma só atividade, como é possível analisar na figura. O tipo de funcionário será importante no decorrer da explicação do relatório sendo que apenas os funcionários camareiras e manutenção podem realizar intervenções.

Consultas Reservas:

A todas as reservas são necessários um cliente e um quarto, tendo sido necessário explicitar essas tabelas primeiro.

IDRESERVA	ESTADO	NUMEROPESSOAS	DATECHECKIN	DATECHECKOUT	NIFCLIENTE	NUMEROANDAR	NUMEROQUARTO	DATECANCELAMENTO	CUSTOCANCELAMENTO	NOMEEPOCA
1	CANCELADA	1	20.02.17	20.02.20	123456789	1	10001	20.02.15	25	BAIXA
2	DECORRENTE	3	20.11.12	20.11.28	448878760	1	10019	(null)	(null)	BAIXA
3	CANCELADA	4	20.01.17	20.01.25	436778870	1	10028	20.01.09	0	BAIXA
4	RESERVADA	5	20.12.11	20.12.19	119898726	2	20033	(null)	(null)	BAIXA
5	DECORRENTE	1	20.11.17	20.11.25	123456789	2	20004	(null)	(null)	BAIXA
6	FINALIZADA	1	20.04.17	20.04.20	123456789	1	10006	(null)	(null)	MEDIA
7	FINALIZADA	1	20.04.17	20.04.20	217665555	1	10031	(null)	(null)	MEDIA
8	CANCELADA	3	20.05.09	20.05.19	222222222	1	10015	20.05.17	25	MEDIA
9	FINALIZADA	4	20.09.17	20.09.25	299999991	1	10023	(null)	(null)	MEDIA
10	CANCELADA	5	20.10.10	20.10.17	223456789	2	20032	20.10.05	25	MEDIA
11	FINALIZADA	1	20.06.01	20.06.05	223456789	2	20006	(null)	(null)	MEDIA
12	FINALIZADA	1	20.07.12	20.07.20	211111111	1	10003	(null)	(null)	ALTA
13	CANCELADA	3	20.06.19	20.06.25	223456789	1	10021	20.06.09	0	ALTA
14	FINALIZADA	4	20.09.01	20.09.12	100430089	1	10029	(null)	(null)	ALTA

Figura 8- Excerto de 14 reservas na base de dados

Antes da inserção da tabelas, foi criado um trigger que irá verificar se para uma reserva, existe um quarto já associado a outro cliente na data pretendida de estadia.

```
----- VERIFICAR SE HA RESERVAS PARA O QUARTO NA DATA A SER INSERIDO
CREATE OR REPLACE TRIGGER verificarDataReserva
BEFORE INSERT
ON reserva
FOR EACH ROW
DECLARE
vCount NUMBER;
BEGIN
SELECT count(*) INTO vCount
FROM reserva
WHERE numeroQuarto = :NEW.numeroQuarto AND :NEW.dataCheckIn < dataCheckOut AND :NEW.dataCheckIn <= CURRENT_DATE AND :NEW.dataCheckOut > CURRENT_DATE;

IF vCount > 0 THEN
raise_application_error(-20005, '»»»»Reserva já efetuada na data pretendida!««««');
END IF;
END;
```

Figura 9- Trigger de verificação de quarto reservado

Caso, o quarto já esteja reservado para as determinadas datas, é despoletado um erro, por exemplo, para a seguinte inserção de reservas:

```
INSERT INTO reserva( numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES(1,TO_DATE('2020-11-15 13:31:15', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-11-24 14:55:16', 'yyyy-mm-dd hh24:mi:ss'),133333335,2,20005,'MEDIA');

INSERT INTO reserva( numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES(1,TO_DATE('2020-11-19 13:31:15', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-11-29 14:55:16', 'yyyy-mm-dd hh24:mi:ss'),127788775,2,20005,'MEDIA');
```

Figura 10- Inserção de reservas com uma delas com uma data não pretendida

É despoletado o erro “»»»»Reserva já efetuada na data pretendida!««««”

```
Error starting at line : 895 in command -
INSERT INTO reserva( numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES(1,TO_DATE('2020-11-19 13:31:15', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-11-29 14:55:16', 'yyyy-mm-dd hh24:mi:ss'),127788775,2,20005,'MEDIA')
Error report -
ORA-20005: »»»»Reserva já efetuada na data pretendida!««««
ORA-06512: na "BDDAD_DG2_A.VERIFICARDATARESERVA", linha 9
ORA-04088: erro durante a execução do trigger 'BDDAD_DG2_A.VERIFICARDATARESERVA'
```

Figura 11- Erro obtido por inserção indevida de quarto em data indisponível

A todas as reservas corresponde um estado. Quando uma reserva se encontra cancelada, e a data de cancelamento é 15 dias inferior à data de Check-Out, um valor de 25€ é aplicado ao custo de cancelamento. Revelou-se relevante mostrar esta consulta para se poder indicar alguns comandos de **DML**, que foram os seguintes:

```
----- MUDANCA DO ESTADO DA RESERVA
UPDATE reserva
SET estado = 'FINALIZADA'
WHERE dataCheckOut < CURRENT_DATE AND estado != 'CANCELADA';

----- UPDATE DA RESERVA
UPDATE reserva
SET custoCancelamento = 25, estado = 'CANCELADA'
WHERE dataCancelamento IS NOT NULL AND (dataCheckOut - dataCancelamento) < 15;

UPDATE reserva
SET custoCancelamento = 0, estado = 'CANCELADA'
WHERE dataCancelamento IS NOT NULL AND (dataCheckOut - dataCancelamento) >= 15;

UPDATE reserva
SET estado = 'ATIVA'
WHERE CURRENT_DATE BETWEEN dataCheckIn AND dataCheckOut;
```

Figura 12- Comandos relevantes para a alteração do estado e custo de cancelamento das reservas

Caso a data de check-out se encontre antes da data atual, o estado da reserva será “finalizada”. Se a data atual se encontrar entre a data de check-in e de check-out, o estado da reserva muda para “ativa”, caso contrário, continuará em “reservada”, sendo que, como é dito no enunciado, este deverá ser o seu valor por defeito.

Consultas das informações de uma conta (que pode possuir vários consumos de produtos do Frigo-Bar) e de registo de consumos por funcionários camareiras:

Para a demonstração da consulta das constas, é essencial mostrar qual as linhas de código utilizadas:

```
-- Produtos, Consumo e Conta
SELECT ct.numeroUnico, c.nome, c.nifCliente, count(pr.idProduto) as Nmr_produtos_consumidos, sum(pr.preco) as Preco_total
FROM cliente c
INNER JOIN reserva r ON r.nifCliente = c.nifCliente
INNER JOIN conta ct ON r.idReserva = ct.idReserva
INNER JOIN consumo cons ON cons.numeroUnico = ct.numeroUnico
INNER JOIN produto pr ON cons.idProduto = pr.idProduto
GROUP BY (ct.numeroUnico, c.nome, c.nifCliente)
ORDER BY ct.numeroUnico;
```

Figura 13- Código que permite a listagem de contas

Analisando o código, é possível reparar que existem vários métodos de junção Inner Join, porquê? Uma conta tem um **número único** como chave primária, e uma conta pode possuir **vários consumos**. A cada consumo, está um **número único de uma conta** associado, e a cada consumo podem estar associados **um ou mais produtos**. Só desta maneira seria possível fazer a soma total do preço de todos os produtos num consumo.

O resultado do código em cima descrito é o seguinte:

	NUMEROUNICO	NOME	NIFCLIENTE	NMR_PRODUTOS_CONSUMIDOS	PRECO_TOTAL
1	1	Gabriela Pinto	448878760	3	4
2	2	Jose Silva	123456789	6	10, 05
3	3	Jessica Athaide	299999991	1	1, 65
4	4	Joao Felix	133333335	3	6

Figura 14- Excerto de contas de clientes

Nota: Apenas clientes que fizeram consumos estão a ser listados.

O número de produtos consumidos e o preço total corresponde a todos os produtos do frigo-bar consumidos entre as datas de entrada e de saída de uma reserva (cada reserva tem apenas um número único de uma conta associado).

Para demonstrar que funcionário camareira registou um determinado consumo, é também essencial mostrar o código da consulta:

```
SELECT r.idReserva, c.nome as Nome_Cliente, r.numeroAndar, r.numeroQuarto, f.nome as Nome_Camareira, f.nifFuncionario as Nif_Camareira,
TO_CHAR(cons.dataRegistoConsumo, 'yyyy-mm-dd hh24:mi:ss') as Data_Registo
FROM consumo cons
INNER JOIN conta ct ON cons.numeroUnico = ct.numeroUnico
INNER JOIN funcionario f ON cons.nifFuncionario = f.nifFuncionario
INNER JOIN reserva r ON r.idReserva = ct.idReserva
INNER JOIN cliente c ON c.nifCliente = r.nifCliente
ORDER BY cons.idConsumo;
```

Figura 15- Código da listagem de funcionários que registaram consumos

Mais uma vez, existem vários métodos de junção Inner Join, sendo que uma **conta** possui um **número único** como chave primária, e um **consumo** tem uma **conta** associada. Os consumos possuem um nifFuncionario, pois os funcionários camareira registam o que foi consumido.

O resultado do código descrito é:

IDRESERVA	NOME_CLIENTE	NUMEROANDAR	NUMEROQUARTO	NOME_CAMAREIRA	NIF_CAMAREIRA	DATA_REGISTO
1	2 Gabriela Pinto	1	10019	Ricardo Matias	323434432	2020-11-13 15:03:31
2	2 Gabriela Pinto	1	10019	Ricardo Matias	323434432	2020-11-14 14:05:24
3	2 Gabriela Pinto	1	10019	Ricardo Matias	323434432	2020-11-15 14:31:22
4	6 Jose Silva	1	10006	Carla Correira	232345643	2020-04-17 14:31:22
5	6 Jose Silva	1	10006	Ricardo Matias	323434432	2020-04-17 14:31:22
6	6 Jose Silva	1	10006	Marisa Mota	656543432	2020-04-18 11:31:22
7	6 Jose Silva	1	10006	Marisa Mota	656543432	2020-04-18 11:31:22
8	6 Jose Silva	1	10006	Ricardo Matias	323434432	2020-04-19 09:20:45
9	6 Jose Silva	1	10006	Marisa Mota	656543432	2020-04-20 08:43:09
10	9 Jessica Athaide	1	10023	Carla Correira	232345643	2020-09-20 08:43:09
11	28 Joao Felix	2	20004	João Costa	555432321	2020-09-18 08:43:09
12	28 Joao Felix	2	20004	Marisa Mota	656543432	2020-09-19 10:55:11
13	28 Joao Felix	2	20004	João Costa	555432321	2020-09-20 11:16:15

Figura 16- idReserva, Nome do cliente e número do quarto a que o funcionário camareira registou o consumo de produtos

Consultas das intervenções realizadas:

IDINTERVENCAO	DESIGNACAO	NOME	NIFFUNCIONARIO
1	1Avaria no frigobar	Fernando Medina	411109878
2	3Avaria no ar condicionado	Joao Tavares	400009878
3	5Avaria no ar condicionado	Ana Costa	990012132
4	11Avaria no microondas	Martim Manha	333000333
5	12Limpeza e Arrumação	Maria Sousa	321432321

Figura 17- Excerto das intervenções realizadas

É relevante mostrar todas as intervenções (já realizadas), na base de dados. A cada intervenção está associado um funcionário, ou do tipo camareira ou do tipo manutenção. Para ser possível chegar a um **funcionário manutenção/camareira**, é necessário partir da **intervenção**, passar para **intervenção limpeza/manutenção** recorrendo ao **id da intervenção**, e só depois é possível atingir o **funcionário** pois cada intervenção dos dois diferentes tipos **tem um NIF funcionário** associado, como demonstra o código a baixo indicado.

```
SELECT im.idIntervencao, i.designacao, f.nome, im.nifFuncionario
FROM intervencao_manutencao im
INNER JOIN intervencao i ON im.idIntervencao = i.idIntervencao
INNER JOIN funcionario f ON f.nifFuncionario = im.nifFuncionario
WHERE registoAtividade IS NOT NULL
UNION ALL
SELECT il.idIntervencao, i.designacao, f.nome, il.nifFuncionario
FROM intervencao_limpeza il
INNER JOIN intervencao i ON il.idIntervencao = i.idIntervencao
INNER JOIN funcionario f ON f.nifFuncionario = il.nifFuncionario
```

Figura 18- Código da listagem das intervenções

Consulta de faturas e pagamentos efetuados:

É também possível visualizar as faturas de cada reserva finalizada, com as necessárias informações dos clientes. Sabendo que uma reserva também tem que ser paga, é possível verificar os pagamentos a partir da fatura.

IDFATURA	NOME	NIFCLIENTE	VALORTOTAL	IDPAGAMENTO	VALORTOTAL_1	TIPOPAGAMENTO
1	15 Filipa Meireles	111111141	6700	15	6700	cartão de crédito
2	47 Afonso Almeirim	131816141	2000	47	2000	cartão de crédito
3	33 Alice Pereira	131816147	1780	33	1780	cartão de crédito
4	21 Paulo Pereira	131716147	987	21	987	cartão de crédito
5	7 Ivete Sangalo	217665555	1425	7	1425	cartão de crédito
6	17 Ivete Sangalo	217665555	7400	17	7400	cartão de crédito
7	30 Ivete Sangalo	217665555	2700	30	2700	cartão de crédito
8	24 Alexandre Pires	217665554	5670	24	5670	cartão de crédito
9	44 Alexandre Pires	217665554	3150	44	3150	cartão de crédito
10	19 Manuel Azevedo	123456769	4560	19	4560	cartão de crédito

Figura 19- Excerto de faturas e pagamentos efetuados por cliente

Parte 1- Guilherme Daniel (1181743)

Alínea a - Apresentar todos os pedidos de intervenção em aberto (intervenção ainda não registada) alocados a funcionários de manutenção que não fizeram nenhuma intervenção (registo) nas últimas 48 horas.

Para o desenvolvimento desta alínea, foram criadas 11 intervenções do tipo manutenção, para ser possível validar as restrições impostas no exercício. Para existir um registo, o **nifFuncionario** e a **dataIntervencao** têm que ser inseridas, caso contrário considera-se que a intervenção do tipo manutenção não foi registada.

```
INSERT INTO intervencao_manutencao(idIntervencao, registoAtividade, nifFuncionario, dataIntervencao)
VALUES (1, 'reparação de frigobar', 411109878, TO_DATE('2020-11-16 09:30:00', 'yyyy-mm-dd hh24:mi:ss'));

INSERT INTO intervencao_manutencao(idIntervencao, nifFuncionario)
VALUES (2, 411109878);

INSERT INTO intervencao_manutencao(idIntervencao, registoAtividade, nifFuncionario,dataIntervencao)
VALUES (3, 'reparação de ar condicionado', 400009878, TO_DATE('2020-11-16 08:20:00', 'yyyy-mm-dd hh24:mi:ss'));

INSERT INTO intervencao_manutencao(idIntervencao, nifFuncionario)
VALUES (4, 400009878);

INSERT INTO intervencao_manutencao(idIntervencao, registoAtividade, nifFuncionario,dataIntervencao)
VALUES (5, 'reparação de ar condicionado', 990012132, TO_DATE('2020-11-10 08:20:00', 'yyyy-mm-dd hh24:mi:ss'));

INSERT INTO intervencao_manutencao(idIntervencao, nifFuncionario)
VALUES (6, 990012132);

INSERT INTO intervencao_manutencao(idIntervencao, nifFuncionario)
VALUES (7,229989092);

INSERT INTO intervencao_manutencao(idIntervencao, nifFuncionario)
VALUES (8,333888111);

INSERT INTO intervencao_manutencao(idIntervencao, nifFuncionario)
VALUES (9,151514140);

INSERT INTO intervencao_manutencao(idIntervencao, nifFuncionario)
VALUES (10,333000333);

INSERT INTO intervencao_manutencao(idIntervencao, registoAtividade, nifFuncionario,dataIntervencao)
VALUES (11,'reparação do microondas',333000333, TO_DATE('2020-11-08 15:35:00', 'yyyy-mm-dd hh24:mi:ss'));
```

Figura 20-Inserção de intervenções do tipo manutenção

Ao analisar os dados a ser inseridos, prevê-se que as intervenções não registadas referentes aos nifFuncionario **411109878** e **400009878** não sejam mostradas na listagem da alínea, pois têm um registo nas últimas 48 horas (é de destacar o facto de que a data em que está a ser escrita esta parte do relatório é **17/11/2020**). Todas as outras intervenções não registadas deverão ser listadas, incluindo a intervenção não registada do nifFuncionario **333000333**, pois a intervenção registada desse funcionário aconteceu há mais de 48 horas.

```
-- alínea a
SELECT DISTINCT f.nome, f.nifFuncionario, inter.idIntervencao, inter.numeroAndar, inter.numeroQuarto
FROM manutencao m
INNER JOIN funcionario f on f.nifFuncionario = m.nifFuncionario
INNER JOIN intervencao_manutencao inM on inM.nifFuncionario = m.nifFuncionario
INNER JOIN intervencao inter on inter.idIntervencao = inM.idIntervencao
WHERE m.nifFuncionario NOT IN(
    SELECT m.nifFuncionario
    FROM manutencao m
    INNER JOIN intervencao_manutencao inM on inM.nifFuncionario = m.nifFuncionario
    WHERE 24 * (CURRENT_DATE - inM.dataIntervencao) <= 48 AND inM.registoAtividade IS NOT NULL) AND inM.registoAtividade IS NULL;
```

Figura 21- Comandos SQL para a listagem obedecer às diferentes restrições

	NOME	NIFFUNCIONARIO	DESIGNACAO	IDINTERVENCAO	NUMEROANDAR	NUMEROQUARTO
1	Marta Couto	229989092	Avaria nas colunas	7	2	20015
2	Martim Manha	333000333	Avaria no forno	10	1	10008
3	Mauricio Castelo	151514140	Avaria no forno	9	2	20005
4	Ana Costa	990012132	Avaria na banheira	6	2	20006
5	Helena Marinho	333888111	Avaria no radio	8	2	20020

Figura 22- Listagem do resultado dos comandos

Aos pedidos de intervenção em aberto, foi adicionado o nome e o nif do funcionário a que a intervenção está alocada, para facilitar a compreensão dos dados, bem como a verificação das restrições da alínea. Analisando os dados, confirma-se que apenas estão a ser listadas todas as intervenções não registadas correspondentes aos funcionários de manutenção, sendo que os dois funcionários com nif **411109878** e **400009878** não tiveram as suas intervenções não registadas listadas, sabendo que realizaram intervenções nas últimas 48 horas. Confirma-se assim a execução da sequência de comandos.

Alínea b - Apresentar a data, a hora e o nome dos clientes que reservaram quartos somente durante o mês de Abril e Junho deste ano. No caso de algum cliente ter reservado um quarto do tipo suite, deverá apresentar a localidade desse cliente numa coluna intitulada “Zona do País”. O resultado deverá ser apresentado por ordem alfabética do nome de cliente e por ordem descendente da data e hora da reserva.

Para além de diversas reservas criadas (44 reservas), que não obedecem às restrições acima indicadas, foram criadas 12 para verificar se a linha da comandos criada funcionará ou não.

Para todos os clientes que as reservas representam, não haviam sido, até ao momento, adicionadas quaisquer outras reservas.

```
-- VALORES QUE OBEDECEM A CLIENTES QUE FIZERAM RESERVAS APENAS EM ABRIL OU JUNHO

-- EM QUARTOS SUITE (QUARTOS SINGLE ESTAO ENTRE O NUMERO 10031 E 10040 E ENTRE 20031 E 20040)
INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',1,TO_DATE('2020-06-01 12:11:12', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-06-09 12:12:11', 'yyyy-mm-dd hh24:mi:ss'),100000001,1,10031,'MEDIA');

INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',2,TO_DATE('2020-06-02 13:21:22', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-06-10 13:22:22', 'yyyy-mm-dd hh24:mi:ss'),120036769,1,10032,'MEDIA');

INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',3,TO_DATE('2020-06-02 14:31:32', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-06-11 14:32:33', 'yyyy-mm-dd hh24:mi:ss'),121816141,1,10033,'MEDIA');

INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',1,TO_DATE('2020-04-04 15:41:42', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-06-12 15:42:44', 'yyyy-mm-dd hh24:mi:ss'),144477444,2,20031,'MEDIA');

INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',2,TO_DATE('2020-04-05 11:51:52', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-06-13 11:52:55', 'yyyy-mm-dd hh24:mi:ss'),109456769,2,20032,'MEDIA');

INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',3,TO_DATE('2020-04-06 14:41:12', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-06-14 14:12:16', 'yyyy-mm-dd hh24:mi:ss'),111117171,2,20033,'MEDIA');
```

Figura 23- Valores que obedecem aos critérios da alínea (para quartos do tipo suite que estão entre os números 31 e 40 em ambos os andares)

```
-- EM QUARTO NAO SUITE
INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',1,TO_DATE('2020-06-01 12:11:12', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-06-09 12:12:11', 'yyyy-mm-dd hh24:mi:ss'),109456769,1,10020,'MEDIA');

INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',2,TO_DATE('2020-06-02 13:21:22', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-06-10 13:22:22', 'yyyy-mm-dd hh24:mi:ss'),211111777,1,10021,'MEDIA');

INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',3,TO_DATE('2020-06-03 14:31:32', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-06-11 14:32:33', 'yyyy-mm-dd hh24:mi:ss'),100000000,1,10022,'MEDIA');

INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',3,TO_DATE('2020-06-06 14:41:12', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-08-14 13:12:16', 'yyyy-mm-dd hh24:mi:ss'),100000000,2,20022,'MEDIA');

INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',1,TO_DATE('2020-04-04 15:41:42', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-04-12 15:42:44', 'yyyy-mm-dd hh24:mi:ss'),111111112,2,20020,'MEDIA');

INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',2,TO_DATE('2020-04-05 11:51:52', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-04-13 11:52:55', 'yyyy-mm-dd hh24:mi:ss'),120056769,2,20021,'MEDIA');

INSERT INTO reserva(estado, numeroPessoas, dataCheckIn, dataCheckOut, nifCliente, numeroAndar, numeroQuarto, nomeepoca)
VALUES ('Finalizada',3,TO_DATE('2020-04-06 14:41:12', 'yyyy-mm-dd hh24:mi:ss'), TO_DATE('2020-04-14 13:12:16', 'yyyy-mm-dd hh24:mi:ss'),111117171,2,20022,'MEDIA');
```

Figura 24- Valores que obedecem aos critérios da alínea (para quartos do tipo não suite)

É de ter em conta quais os *nifCliente* indicados na inserção de valores de reserva, para ser possível identificar na listagem se os valores estão de acordo com a inserção de dados (para ser possível comprovar que os valores inseridos estão corretos, para além da data, hora e nome do cliente, foi também inserida na listagem a coluna com o nif do cliente). O *nif*s dos clientes **111117171** e **109456769** foram inseridos em duas reservas para ser possível demonstrar a ordenação pela data e hora de reserva. O nif do cliente **100000000** foi atribuído a uma reserva que obedece às restrições de data, mas também a uma reserva que está fora dos meses pretendidos, pelo que a informação desse cliente já não deverá ser listada. Os quartos do tipo **suite** são quartos que estão entre os números **100031** e **100040** e **200031** e **200040**.

```
SELECT r.dataCheckIn as DATA, TO_CHAR(r.dataCheckIn,'HH24:MI:SS') as HORA, c.nome, c.nifCliente, CASE q.designacaoTipo WHEN 'suite' THEN e.localidade ELSE ' ' END as Zona_do_pais
FROM cliente c
INNER JOIN reserva r on r.nifCliente = c.nifCliente
INNER JOIN endereco e on e.nifCliente = c.nifCliente
INNER JOIN quarto q on r.numeroQuarto = q.numeroQuarto
WHERE c.nifCliente NOT IN (
    SELECT c.nifCliente
    FROM cliente c
    INNER JOIN reserva r on r.nifCliente = c.nifCliente
    WHERE EXTRACT(MONTH FROM r.dataCheckIn) != 4 AND EXTRACT(MONTH FROM r.dataCheckIn) != 6)
ORDER BY c.nome, r.dataCheckIn DESC;
```

Figura 25- Comandos SQL para a listagem obedecer às diferentes restrições

	DATA	HORA	NOME	NIFCLIENTE	ZONA_DO_PAIS
1	20.06.02	13:21:22	Ana Pereira	211111777	
2	20.06.02	13:21:22	Carla Ferreira	120036769	Madeira
3	20.04.04	15:41:42	Filomena Cautela	144477444	Evora
4	20.06.01	12:11:12	Marcos Marques	100000001	Açores
5	20.04.06	14:41:12	Mariana Mortagua	111117171	
6	20.04.06	12:41:12	Mariana Mortagua	111117171	Guarda
7	20.04.05	11:51:52	Micaela Contante	120056769	
8	20.06.01	12:11:12	Nelly Furtado	109456769	
9	20.04.05	11:51:52	Nelly Furtado	109456769	Madeira
10	20.06.03	14:31:32	Sara Santos	121816141	Braga
11	20.04.04	15:41:42	Tatiana Neves	111111112	

Figura 26- Listagem do resultado dos comandos

Ao analisar os resultados, comparando o **NIFCLIENTE** com os nifCliente da inserção de dados, verifica-se que todos os dados estão de acordo com o esperado. Os dados do nif **100000000** não aparecem na listagem. Todos os dados estão ordenados por ordem alfabética, e quando se verifica que uma reserva corresponde ao mesmo nif, como no exemplo dos nif's **111117171** e **109456769**, os dados estão ordenados pela data e hora, de maneira decrescente. Verifica-se também que para os nif's a que está associado um quarto do tipo **suite**, são mostradas as localidades na coluna "ZONA_DO_PAIS".

Parte 2- Lucas Sousa (1171589)

Alínea a - Apresentar o nome, a localidade e o nome do concelho dos clientes que já estiveram alojados nos quartos já reservados pelo cliente cujo nome é José Silva, que é do concelho de Vila Real. Considere só as reservas "finalizadas" do cliente José Silva

Para o desenvolvimento desta alínea, foi criado o cliente "José Silva", que reside no concelho de "Vila Real".

```
INSERT INTO Cliente(nome,nifCliente,email)
VALUES('Jose Silva', 123456789, 'josesilva@gmail.com');

INSERT INTO Endereco(nifCliente, rua, concelho, localidade, codPostal, pais)
VALUES(123456789, 'Estrada Nacional', 'Vila Real', 'Vila Real', '5000-188', 'Portugal');
```

Figura 27- Valor correspondente ao pedido no enunciado

Criamos 4 reservas para este cliente, nos quartos "10001", "20004", "10006" e "20015" e atribuímos o estado "Cancelada", "Ativa", "Finalizada" e "Finalizada" assim, sucessivamente.

	NUMEROQUARTO	ESTADO
1	10001	CANCELADA
2	20004	ATIVA
3	10006	FINALIZADA
4	20015	FINALIZADA

Figura 28- Listagem dos quartos reservados pelo "José Silva" e o seu estado

Criamos reservas para esses quartos em diferentes estados para credibilizar a resposta a esta alínea.

	NUMEROQUARTO	NOME	ESTADO
1	20015	Afonso Almeirim	FINALIZADA
2	20015	Neyo Santos	FINALIZADA
3	10006	Nelly Furtado	FINALIZADA
4	20004	Joao Felix	FINALIZADA
5	10001	Leonor Costa	FINALIZADA
6	10001	Jose Silva	CANCELADA
7	20004	Jose Silva	ATIVA
8	10006	Jose Silva	FINALIZADA
9	20015	Jose Silva	FINALIZADA
10	10001	Ana Rocha	FINALIZADA
11	20015	Gabriela Pinto	FINALIZADA

Figura 29- Listagem dos quartos reservados pelo “José Silva” assim como os clientes que já os ocuparam e o seu estado

Recorremos a um query que vai encontrar todos os números de quartos já ocupados pelo cliente “José Silva” do concelho de “Vila Real” e que estejam no estado de “Finalizada”, como é pedido no enunciado. Depois com a query inicial vamos verificar se quarto pretendido esta na query falada no início, caso esteja e o seu estado seja “Finalizada”, uma vez que queremos todos os clientes que já estiveram alojados, para tal o estado das suas reservas tem de ser o referido e finalmente retirando o cliente “José Silva” do concelho de “Vila Real”.

```
SELECT DISTINCT c.nome, e.localidade, e.concelho
FROM cliente c
INNER JOIN endereco e on c.nifCliente = e.nifCliente
INNER JOIN reserva r on c.nifCliente = r.nifCliente
WHERE r.numeroquarto IN
(
  SELECT q.numeroquarto
  FROM cliente c
  INNER JOIN reserva r on r.nifCliente = c.nifCliente
  INNER JOIN quarto q on q.numeroquarto = r.numeroquarto
  INNER JOIN endereco e on c.nifCliente = e.nifCliente
  WHERE c.nome LIKE '%Jose Silva%'
        AND e.concelho LIKE '%Vila Real%'
        AND r.estado LIKE '%FINALIZADA%')
AND r.estado LIKE '%FINALIZADA%'
AND (c.nome NOT LIKE '%Jose Silva%' and e.concelho NOT LIKE '%Vila Real%');
```

Figura 30- Query utilizada para responder a alínea a

	NOME	LOCALIDADE	CONCELHO
1	Ana Rocha	Viseu	Lamego
2	Joao Felix	Portalegre	Portalegre
3	Leonor Costa	Santarem	Santarem
4	Gabriela Pinto	Viseu	Lamego
5	Neyo Santos	Madeira	Camara de Lobos
6	Afonso Almeirim	Braga	Braga
7	Nelly Furtado	Madeira	Funchal

Figura 31- Resultado da alínea a

Alínea b - Apresentar por cada mês, apenas para os últimos 6 meses anteriores à data atual, qual a camareira que mais intervenções fez em quartos cuja duração de estadia foi superior à média das estadias agrupadas por tipo de quarto.

Para a elaboração desta alínea foram criadas 184 intervenções de limpeza, para cada quarto reservado, durante o período de estadia todos os dias são feitas intervenções de limpeza no quarto em questão.

```
WITH
  lista as
    (SELECT q.numeroQuarto as nQuarto, q.designacaoTipo as tipo, (r.dataCheckOut - r.dataCheckIn) as diff
     FROM quarto q
     INNER JOIN reserva r ON r.numeroQuarto = q.numeroQuarto),
  media as
    (SELECT AVG(r.dataCheckOut - r.dataCheckIn) m, designacaoTipo as tipo2
     FROM quarto q
     INNER JOIN reserva r on q.numeroQuarto = r.numeroQuarto
     GROUP BY (q.designacaoTipo)),
  intervencoes as
    (SELECT mes, tipo as TipoQuarto, funNome as Camareira, max(fun) as maximo
     FROM
       SELECT distinct EXTRACT (MONTH FROM il.dataIntervencao) mes, nQuarto, tipo, diff, i.idIntervencao idI, f.nome funNome, il.nifFuncionario iNif, count(il.idIntervencao) fun
       FROM lista
       INNER JOIN intervencao i ON i.numeroQuarto = nQuarto
       INNER JOIN intervencao_limpeza il ON i.idIntervencao = il.idIntervencao
       INNER JOIN funcionario f ON f.nifFuncionario = il.nifFuncionario
       WHERE diff > (SELECT m FROM media WHERE tipo = tipo2)
       and il.dataIntervencao between add_months(sysdate,-6) AND sysdate
       GROUP BY (EXTRACT (MONTH FROM il.dataIntervencao), nQuarto, tipo, diff, i.idIntervencao, f.nome, il.nifFuncionario)
       ORDER BY mes,nQuarto, fun DESC)
  group by ( mes, tipo, funNome)
  ORDER BY mes , max(fun) ),
  maximos as
    (SELECT mes, TipoQuarto, max(maximo) as Mcamareira
     FROM intervencoes
     GROUP BY (mes, TipoQuarto)
     ORDER BY mes, TipoQuarto)
SELECT intervencoes.mes , intervencoes.TipoQuarto, Camareira
FROM intervencoes, maximos
where intervencoes.mes = maximos.mes
and intervencoes.TipoQuarto = maximos.TipoQuarto
and intervencoes.maximo = maximos.Mcamareira;
```

Figura 32- Query desenvolvida para responder a alínea b

Para conseguirmos responder a esta alínea recorreremos a uma clausula whit, com 5 query's, “lista”, “media”, “intervencoes”, “maximo” e a final que interliga estas 4. Com a “lista” vamos buscar todos os quartos que já foram reservados. A “media” vai buscar a media de duração de estadia de cada tipo de quarto. A “intervencoes” interliga a “lista” com a “media”, o que nos permite retirar todas as camareiras que tiveram intervenções em quartos com ocupação superior à média de ocupação por tipo. O “maximo” utiliza o “intervencoes” para retirar o máximo de intervenções de cada camareira. Finalmente na última query vamos comparar o mês o tipo de quarto e as intervenções de “maximo” e “intervencoes” para termos o output esperado.

	MES	TIPOQUARTO	CAMAREIRA
1	6	single	Ricardo Matias
2	6	superior	Carla Correira
3	6	twin	Maria Sousa
4	7	single	Maria Sousa
5	7	superior	Maria Sousa
6	8	single	Ricardo Matias
7	8	suite	Ricardo Matias
8	9	single	Marisa Mota
9	9	superior	Ricardo Matias
10	10	suite	Carla Correira
11	11	single	Maria Sousa

Figura 33- Listagem de resposta a alínea b

Parte 3- Kevin Sousa (1180853)

Alínea a - Apresentar por andar, o quarto e o tipo de quarto, que teve o maior número de reservas. Deverão ser excluídos todos os quartos em que o número de reservas é inferior a 2 e são do tipo “single”. Não incluir reservas canceladas.

```
select r.numeroQuarto,r.numeroandar,q.designacaotipo
from Reserva r
inner join Quarto q on q.numeroquarto = r.numeroquarto
where r.dataCancelamento is null and r.numeroandar = 1 and q.designacaotipo not like 'single'
group by r.numeroquarto,r.numeroandar,q.designacaotipo
having count(*)>2 and count(*) = (SELECT MAX(COUNT(*))FROM reserva GROUP BY numeroquarto)
union
select r.numeroQuarto,r.numeroandar,q.designacaotipo
from Reserva r
inner join Quarto q on q.numeroquarto = r.numeroquarto
where r.dataCancelamento is null and r.numeroandar = 2 and q.designacaotipo not like 'single'
group by r.numeroquarto,r.numeroandar,q.designacaotipo
having count(*)>2 and count(*) = (SELECT MAX(COUNT(*))FROM reserva GROUP BY numeroquarto);
```

Figura 34- Query utilizada para realizar alínea A

Neste exercício, são selecionados os números dos quartos e números dos andares da **tabela ‘Reserva’** e o tipo do quarto da **tabela ‘Quarto’**. Estes dados são relacionados através de INNER JOIN. Posteriormente, na condição **where**, define-se que a **dataCancelamento** tem de ser null , em que andar estamos e que o tipo de quarto não pode ser single , depois na condição **having** define-se que o quarto tem que ter mais que dois reservasões e tem que ter o maior numero de reservasões nesse andar e a condição **union** o que faz é juntar o resultado do primer andar com o resultado obtido do segundo andar. Caso o quarto cumpra com todas as condições pedidas , então o seu numero, andar e tipo de quarto são impressos.

Consulta de Andares, Tipos de Quarto e Quartos:

	NUMEROQUARTO	NUMEROANDAR	Tipo de quarto	Numero de reservas por Quarto
1	20005	2	single	2
2	20008	2	single	1
3	20020	2	twin	1
4	20032	2	suite	1
5	20033	2	suite	2
6	20015	2	twin	2
7	20030	2	superior	2
8	20002	2	single	1
9	20006	2	single	2
10	20004	2	single	2
11	20010	2	single	1
12	20037	2	suite	1
13	20013	2	twin	1
14	20038	2	suite	1

Figura 34 – Listagem dos quartos do segundo andar

	NUMEROQUARTO	NUMEROANDAR	Tipo de quarto	Numero de reservas por Quarto
1	10001	1	single	1
2	10006	1	single	1
3	10019	1	twin	1
4	10030	1	superior	1
5	10031	1	suite	2
6	10015	1	twin	1
7	10023	1	superior	1
8	10029	1	superior	4
9	10028	1	superior	1
10	10027	1	superior	1
11	10017	1	twin	1
12	10005	1	single	2
13	10003	1	single	1
14	10021	1	superior	3

Figura 35 – Listagem dos quarto do primer andar

Agora como podemos confirmar nas figuras 34 e 35 , o único quarto que cumpre com essas condições é o quarto 10029 no primeiro andar. No segundo andar, como não há nenhum quarto que tenha mais do que duas reservas, não cumpre com uma condição pedida na **alínea A** por isso só é apresentado o quarto 10029 do primeiro andar.

	NUMEROQUARTO	NUMEROANDAR	DESIGNACAOTIPO
1	10029	1	superior

Figura 36 - Resultado da alínea A

Alínea b - Apresentar os clientes que ocuparam quartos do tipo suite na última época alta e consumiram os dois produtos com maior consumo nos últimos dois anos. O resultado deve ser ordenado por ordem decrescente do valor total do consumo.

Consulta dos produtos mais consumidos :

IDPRODUTO	NOME	COUNT(*)
1	1 coca-cola	2
2	3 7up	2
3	10 maltesers	1
4	9 mars	1
5	4 fanta	1
6	6 tango	1
7	5 agua com gas	1
8	8 twix	1
9	7 pepsi	1
10	2 agua	1
11	11 skittles	1

Figura 37- Listagem dos produtos mais consumidos

Consulta de informação relacionada com um cliente que tem o tipo de quarto suite e foi reservado em época alta :

NOME	IDRESERVA	preco produto	tipo quarto	nome epoca	id produto	nome produto	data check in	data abertura conta
1 Neyo Santos	29	2,3 suite	ALTA		11 skittles		20.08.03	20.03.12
2 Danilo Pereira	34	2 suite	ALTA		6 tango		20.08.16	20.03.12
3 Danilo Pereira	34	2 suite	ALTA		8 twix		20.08.16	20.03.12
4 Neyo Santos	29	1,75 suite	ALTA		9 mars		20.08.03	20.03.12
5 Guilherme Sao	24	1,7 suite	ALTA		1 coca-cola		20.08.11	19.03.12
6 Danilo Pereira	34	1,65 suite	ALTA		7 pepsi		20.08.16	20.03.12
7 Guilherme Sao	24	1,65 suite	ALTA		4 fanta		20.08.11	19.03.12
8 Guilherme Sao	24	1,6 suite	ALTA		3 7up		20.08.11	19.03.12
9 Neyo Santos	29	1,6 suite	ALTA		10 maltesers		20.08.03	20.03.12
10 Guilherme Sao	24	1,1 suite	ALTA		5 agua com gas		20.08.11	19.03.12

Figura 38 - Listagem da informação dos clientes com o tipo de quarto suite e foi reservado em época alta

Agora vamos a dar inicio a solução da alínea b :

Neste exercício, faltou uma restrição que é listar só os clientes que tivessem consumido os dois produtos mais consumidos. Como não encontrei a forma de verificar se o cliente tinha consumido os dois produtos mais consumidos, o que fiz foi simplesmente ver se o cliente tivesse consumido qualquer um dos dois produtos mais consumidos e mostrava a informação desse cliente.

```

select cli.* from cliente cli
inner join Reserva r on r.nifcliente = cli.nifcliente
inner join Quarto q on r.numeroquarto = q.numeroquarto
inner join Conta cont on cont.idreserva = r.idreserva
inner join Consumo consumo on consumo.numerounico = cont.numerounico
inner join produto p on p.idproduto = consumo.idproduto
where upper(r.nomeepoca) like 'ALTA' and upper(q.designacaotipo) like 'SUITE'
and datacheckin between ( select * from (select datainicio from EpocaAno where upper(nomeepoca) like 'ALTA' order by datainicio desc) where rownum = 1)
and ( select * from (select datafim from EpocaAno where upper(nomeepoca) like 'ALTA' order by datafim asc) where rownum = 1)
and (extract(year from cont.dataabertura) = (select (extract(year from sysdate) - 2) from dual) or
extract(year from cont.dataabertura) = (select (extract(year from sysdate)- 1) from dual))
and consumo.idproduto in (select * from (select idproduto from consumo group by idproduto order by count(*) desc ) where rownum <= 2)
group by cli.nome,cli.nifcliente,cli.email,cli.telefone
order by sum(p.preco) desc;

```

Figura 39 - Query alínea b

Neste exercício, para conseguir ver os dados do cliente com as restrições exigidas na alínea b, o que se fez primeiro foi ver qual é o nif do cliente, e depois utilizar o inner join para ir a ver todos os dados desse respetivo cliente nas outras tabelas que precise para fazer o que é pedido na alínea b. Depois, na condição where, vamos a verificar que o cliente reservou em época alta, que o tipo de habitação foi suite, que a data do check in foi na ultima temporada alta, ver que esse produto foi consumido nos ultimos dois anos e, por último, verificar se algum dos dois produtos mais consumidos (**rownum <=2 só mostra os dois primeiros resultados da consulta**) tenha sido consumido por esse cliente. No final, com o order by, ordenamos por ordem decrescente do valor total consumido.

Como podemos observar pelas figuras 37 e 38, **guilherme** é o único a cumprir com a condição de ter, pelo menos, ter consumido um dos produtos mais consumidos e também é o único cliente que fez isso nos últimos dois anos como pode verificar-se na data de abertura da conta.

NOME	NIFCLIENTE	EMAIL	TELEFONE
1 Guilherme Sao	2999999999	sao@hotmail.com	(null)

Figura 40 - Resultado

Agora vou explicar como tentei fazer o exercício da alínea b de maneira a cumprir com todas as condições (**Na figura 39 só verifico que consumi-o um dos dois produtos mais consumidos**). Esta query que vou apresentar a seguir não está funcional, por isso apresentei a query de cima (figura 39) para que pelo menos pudéssemos observar que as condições aplicadas na clausula group by, where e no inner join estão a funcionar.

```
select cli.* from cliente cli
inner join Reserva r on r.nifcliente = cli.nifcliente
inner join Quarto q on r.numeroquarto = q.numeroquarto
inner join Conta cont on cont.idreserva = r.idreserva
inner join Consumo consumo on consumo.numerounico = cont.numerounico
inner join produto p on p.idproduto = consumo.idproduto
where upper(r.nomeepoca) like 'ALTA' and upper(q.designacao tipo) like 'SUITE'
and datacheckin between ( select * from (select datainicio from EpocaAno where upper(nomeepoca) like 'ALTA' order by datainicio desc) where rownum = 1)
and ( select * from (select datafim from EpocaAno where upper(nomeepoca) like 'ALTA' order by datafim asc) where rownum = 1)
and (extract(year from cont.dataabertura) = (select (extract(year from sysdate) - 2) from dual) or
extract(year from cont.dataabertura) = (select (extract(year from sysdate)- 1) from dual))
and consumo.idproduto = (select * from (select idproduto from consumo group by idproduto order by count(*) desc ) where rownum = 1)
and consumo.idproduto = (select * from (select idproduto from consumo group by idproduto order by count(*) desc ) where rownum =2)
group by cli.nome,cli.nifcliente,cli.email,cli.telefone
order by sum(p.preco) desc;
```

Figura 41 - Tentativa de cumprir com todas as restrições

Como podemos observar nesta query tento verificar se o cliente consumi-o os dois produtos mais consumido, a única mudança respeito a figura 39 é que na linha que comparamos os dois produtos mais consumidos tentamos fazer isso com o mesmo comando, só mudando o rownum que neste caso definimos o rownum = 1, para que nos de só o primeiro produto e depois fazemos o mesmo comando e definimos o rownum = 2 para que nos de só o segundo produto mas o rownum = 1 retorna o resultado esperado enquanto que o rownum = 2 não retorna nada e não encontrei outra forma de fazer essa restrição de verificar que o cliente tinha consumido os dois produtos mais consumidos.

Conclusão:

Para a elaboração do trabalho foi necessário, inicialmente, a elaboração do modelo de dados relacional, e para tal, obedecer às regras de normalização.

A inserção de dados tornou-se bastante trabalhosa, sendo que era necessário adicionar valores que obedecessem ao modelo relacional elaborado. Recorremos ao uso de um trigger para validação de valores a ser inseridos, usando diversas matérias abordadas na disciplina.

No percorrer do trabalho, consideramos todas as alíneas da Parte 3 bem-sucedidas,

Para o bom funcionamento das queries, foram usadas, maioritariamente, subqueries, para efetuar consultas que de outra forma seriam bastante mais complicadas. Recorremos também a clausulas de associação de tabelas, como Inner Join, para converter dados encontrados em duas ou mais tabelas.