

Relatório

Bases de Dados – Trabalho I

Realizado por:

Luís Araújo (1190827)

Danilton Lopes (1191240)

Luís Tovar (1181585)

Índice

Modelo Relacional	3
Decisões tomadas ao realizar o modelo relacional.....	4
Consultas desenvolvidas	5
Parte I	5
a).....	5
b)	6
Parte II	8
a).....	8
b)	9
Parte III	12
a).....	12
b)	13

Modelo Relacional

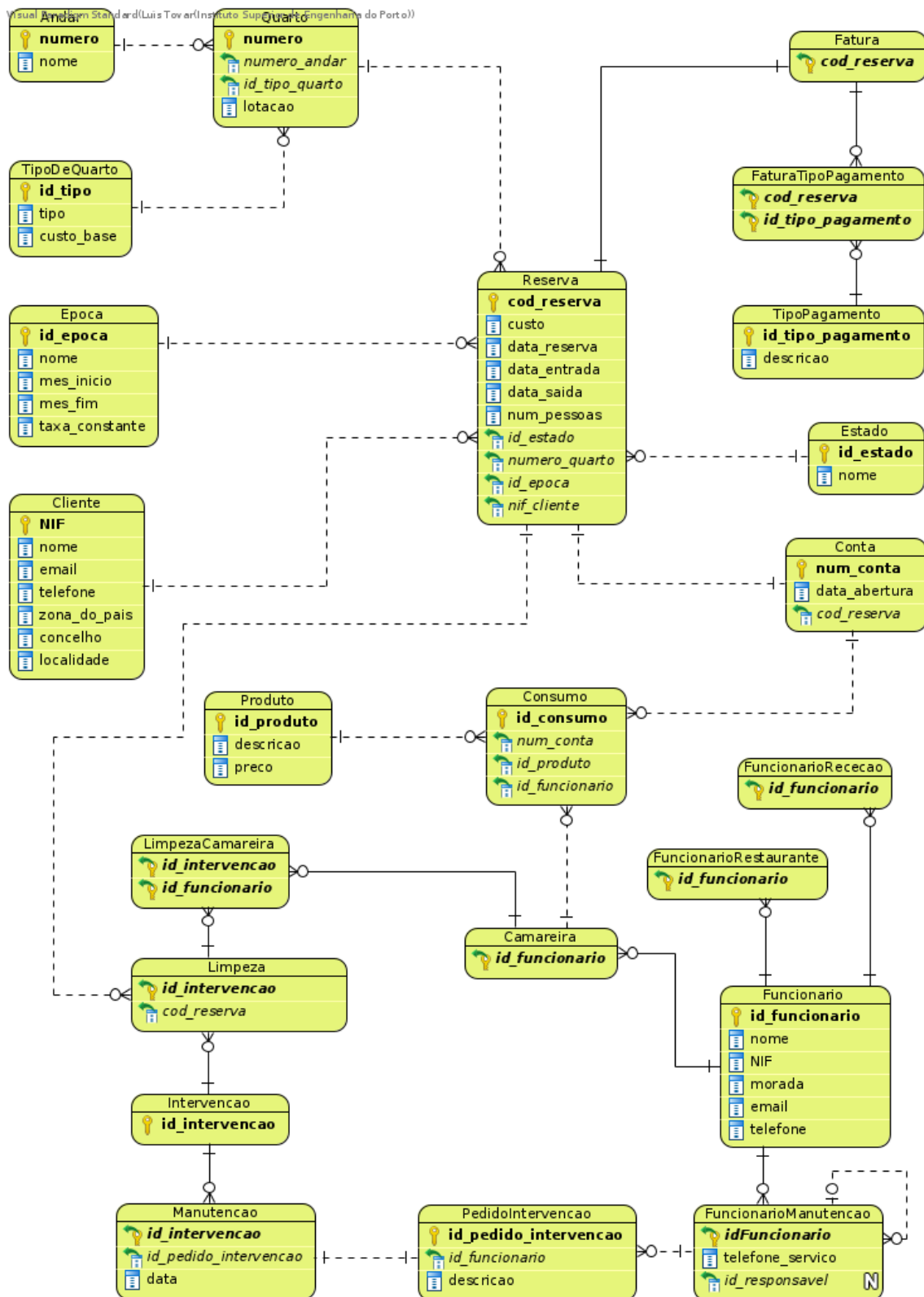


Imagem 1 - Modelo Relacional

Decisões tomadas ao realizar o modelo relacional

1. Criámos a entidade “Andar”, que tem um número. Isto permite que seja possível garantir que o número de cada entrada da entidade “Quarto” tenha o número do andar como primeiro dígito.

2. Cada época tem como atributo uma taxa constante que permite mudar o preço em função da época em que a reserva é feita.

3. Para as relações *many to many*: Fatura-Tipo de Pagamento, Limpeza-Camareira, foram criadas entidades intermédias, respetivamente: FaturaTipoPagamento, LimpezaCamareira. Além disso, pode-se considerar a entidade “Consumo” uma entidade intermédia na relação *many to many* Produto-Conta. Para facilitar a normalização do modelo, decidimos também que todas as chaves primárias seriam simples (geralmente um ID ou código único), exceto para as entidades intermédias.

4. É possível determinar quais são os quartos livres para um determinado período, pois cada quarto tem associadas reservas que contêm informação sobre as datas de início e fim de reserva.

5. A relação entre as entidades “Reserva” e “Fatura” é *one to one*, pelo que decidimos que seria a Fatura a guardar a chave primária da Reserva, já que a Fatura só é gerada no check-out ou no cancelamento da reserva. O mesmo se aplica na entidade “Conta”, pois esta pode nem ser criada para uma reserva.

6. As épocas são limitadas por números que se referem ao mês de início e ao mês de fim da mesma. Seria possível calcular em que época se deu uma reserva extraíndo o mês da data de início e verificando em que intervalo de meses se encaixa, mas acabámos por não dar nenhuma aplicação a estes valores na base de dados. Tentámos fazer uma verificação mas não soubemos fazê-lo sem uma query, e tentámos fazer um default value e tivemos o mesmo problema.

7. O custo da reserva é guardado na entidade “Reserva”, pois isso permite calcular o custo previsto de uma reserva num quarto e numa data sem se precisar de pagar. O custo vai sendo atualizado quando são feitas alterações como a adição de consumos à conta. A Fatura acede posteriormente a este valor quando é gerada. Visto que as reservas canceladas a menos de 15 dias do dia de entrada no hotel terão de ser pagas, decidimos colocar o valor a pagar pelo cliente a 50% do custo do quarto mediante a época em questão.

8. As entidades “Camareira”, “Funcionário Restaurante”, “Funcionário Receção” e “Funcionário Manutenção” são especializações da entidade “Funcionário”, pelo que nenhum dos referidos pode existir sem que exista uma linha correspondente na tabela “Funcionário”. O mesmo se aplica às entidades “Manutenção” e “Limpeza”, que são especializações da entidade “Intervenção”.

9. Um Funcionário de Manutenção pode ter ou não um responsável, por isso o atributo `id_responsavel` na entidade “FuncionarioManutenção” é *nullable*.

10. Cada intervenção de manutenção só pode ser efetuada por um funcionário de manutenção. Cada manutenção é associada a um pedido de intervenção, que por sua vez é associado a um único funcionário de manutenção.

11. Os consumos de cada reserva são registados diariamente pelas camareiras. Para cada reserva, mais que um consumo pode ser realizado por dia, e cada consumo refere-se a um único produto.

Consultas desenvolvidas

Parte I

Realizada por Luís Tovar (1181585).

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.

Inserções (input para a pesquisa)

```
293 -- ## tabela funcionario_Manutencao ##
294 INSERT INTO funcionario_Manutencao (id_Funcionario, telefone, id_Responsavel) VALUES(16, 223572983, 1000);
295 INSERT INTO funcionario_Manutencao (id_Funcionario, telefone, id_Responsavel) VALUES(17, 223572984, 1001);
296 INSERT INTO funcionario_Manutencao (id_Funcionario, telefone, id_Responsavel) VALUES(18, 223572985, 1002);
297 INSERT INTO funcionario_Manutencao (id_Funcionario, telefone, id_Responsavel) VALUES(19, 223572986, 1003);
298 INSERT INTO funcionario_Manutencao (id_Funcionario, telefone, id_Responsavel) VALUES(20, 223572987, 1004);
```

Imagem 3 - Inserts dos funcionários de manutenção

```
393 -- ## tabela pedido_intervencao ##
394 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(1,16,'reparar frigobar');
395 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(2,17,'reparar vasos da varanda');
396 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(3,18,'reparar chuveiro');
397 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(4,19,'reparar frigobar');
398 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(5,20,'reparar vasos da varanda');
399 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(6,16,'reparar chuveiro');
400 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(7,17,'reparar frigobar');
401 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(8,18,'reparar vasos da varanda');
402 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(9,19,'reparar chuveiro');
403 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(10,20,'reparar frigobar');
404 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(11,16,'reparar vasos da varanda');
405 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(12,17,'reparar chuveiro');
406 INSERT INTO pedido_intervencao(id_pedido_intervencao, id_funcionario, descricao) VALUES(13,18,'reparar frigobar');
407
408 -- ## tabela manutencao ##
409 INSERT INTO manutencao(id_intervencao, id_pedido_intervencao, data_Manutencao) VALUES(11,1,TO_DATE('16-11-2020','dd-mm-yyyy'));
410 INSERT INTO manutencao(id_intervencao, id_pedido_intervencao, data_Manutencao) VALUES(12,2,TO_DATE('21-11-2020','dd-mm-yyyy'));
411 INSERT INTO manutencao(id_intervencao, id_pedido_intervencao, data_Manutencao) VALUES(13,3,TO_DATE('16-11-2020','dd-mm-yyyy'));
412 INSERT INTO manutencao(id_intervencao, id_pedido_intervencao, data_Manutencao) VALUES(14,4,TO_DATE('21-11-2020','dd-mm-yyyy'));
413 INSERT INTO manutencao(id_intervencao, id_pedido_intervencao, data_Manutencao) VALUES(15,5,TO_DATE('16-11-2020','dd-mm-yyyy'));
414 INSERT INTO manutencao(id_intervencao, id_pedido_intervencao, data_Manutencao) VALUES(16,6,TO_DATE('21-11-2020','dd-mm-yyyy'));
415 INSERT INTO manutencao(id_intervencao, id_pedido_intervencao, data_Manutencao) VALUES(17,7,TO_DATE('16-11-2020','dd-mm-yyyy'));
416 INSERT INTO manutencao(id_intervencao, id_pedido_intervencao, data_Manutencao) VALUES(18,8,TO_DATE('16-11-2020','dd-mm-yyyy'));
417 INSERT INTO manutencao(id_intervencao, id_pedido_intervencao, data_Manutencao) VALUES(19,9,TO_DATE('21-11-2020','dd-mm-yyyy'));
418 INSERT INTO manutencao(id_intervencao, id_pedido_intervencao, data_Manutencao) VALUES(20,10,TO_DATE('16-11-2020','dd-mm-yyyy'));
```

Imagem 2 - Inserts dos pedidos de manutenção e dos registos de manutenção

Síntese do que é relevante para a pesquisa:

- Existem 5 funcionários de manutenção, que têm os Ids de 16-20.
- Há 13 pedidos de intervenção, com Ids de 1-12.
- Há 10 manutenções, de Id de 11-20.
- Os 3 pedidos de intervenção que não têm manutenção registada têm Ids 11, 12 e 13, e estão alocados respetivamente aos funcionários de manutenção 16, 17 e 18.
- A data considerada atual no momento da pesquisa é 13h de 22/11/2020
- As manutenções feitas há mais de 48h antes dessa data são:
 - id_intervencao=11, id_pedido_intervencao=1 → id_funcionario=16
 - id_intervencao=13, id_pedido_intervencao=3 → id_funcionario=18
 - id_intervencao=15, id_pedido_intervencao=5 → id_funcionario=20
 - id_intervencao=17, id_pedido_intervencao=7 → id_funcionario=17
 - id_intervencao=18, id_pedido_intervencao=8 → id_funcionario=18
 - id_intervencao=20, id_pedido_intervencao=10 → id_funcionario=20
- Os únicos funcionários deste conjunto que estão alocados a um dos 3 pedidos de manutenção sem manutenção registada são os com ID 16 e 17.

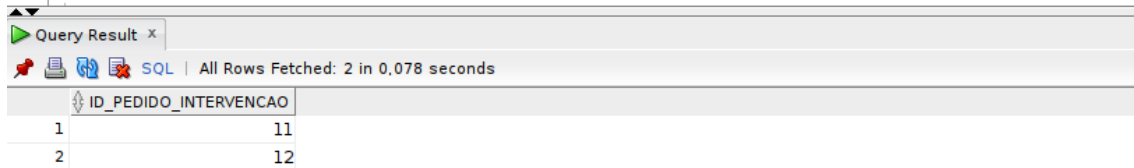
- Os pedidos de manutenção sem manutenção registada com os funcionários 16 e 17 alocados são os com ID 11 e 12, portanto o resultado esperado da pesquisa é {11,12}.

Pesquisa e resultados

```

2 | SELECT id_pedido_intervencao -- todos os pedidos de intervenção alocados a funcionários
3 | FROM Pedido_Intervencao -- que não registaram nenhuma intervenção nas últimas 48h
4 | WHERE id_funcionario IN (
5 |     SELECT fm.id_funcionario -- todos os funcionários que não registaram nenhuma
6 |     FROM Funcionario_Manutencao fm -- intervenção nas últimas 48h
7 |     JOIN Pedido_Intervencao pi ON pi.id_funcionario=fm.id_funcionario
8 |     JOIN Manutencao m ON m.id_pedido_intervencao=pi.id_pedido_intervencao
9 |     WHERE TRUNC (SYSDATE - m.DATA_MANUTENCAO)*24 + TRUNC (MOD ( (SYSDATE - m.DATA_MANUTENCAO) * 24, 24)) < 48
10 | )
11 |
12 | INTERSECT
13 |
14 | SELECT id_pedido_intervencao -- todos os pedidos de intervenção
15 | FROM Pedido_Intervencao -- que não têm uma manutenção registada
16 | WHERE id_pedido_intervencao NOT IN (
17 |     SELECT id_pedido_intervencao
18 |     FROM Manutencao
19 | );

```



The screenshot shows a SQL query window with the following text: "Query Result x", "SQL | All Rows Fetched: 2 in 0,078 seconds". Below this, a table with one column "ID_PEDIDO_INTERVENCAO" is displayed, containing two rows with values 11 and 12.

ID_PEDIDO_INTERVENCAO
11
12

Imagem 4 – Pesquisa 1.a e respetivos resultados

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.

Interpretação de ambiguidades linguísticas

Neste exercício interpretou-se que só serão mostrados clientes - e respetivas datas em que efetuaram a(s) reserva(s) - que não tenham feito reservas no hotel com data de entrada ou data de saída fora de abril ou junho do ano atual.

Inserts (input para a pesquisa)

Para testar a pesquisa, são relevantes os seguintes dados:

- Existe um cliente com uma reserva para junho de 2020 num quarto do tipo suite (NIF 174826492).
- Existe um cliente com uma reserva para abril de 2020 (NIF 183603675).
- Existe um cliente com uma reserva para junho de 2017 (NIF 191511781).
- Existe um cliente com uma reserva para abril de 2020, mas também tem uma para julho de 2020 (NIF 111111411).
- Todos os outros clientes reservam em meses diferentes de abril e junho.

Com estes dados como input, o resultado da pesquisa deve ser os clientes de NIF 174826492 e 183603675, especificando a localidade do cliente de NIF 174826492.

Esta é a pesquisa e os resultados:

```
22 SELECT TO_CHAR(r.data_reserva, 'dd-mm-yyyy hh:mm') "data e hora", c.nome,
23 CASE
24     WHEN q.id_tipo_quarto=4 THEN c.localidade
25     ELSE ' '
26 END AS "Zona do País", r.data_entrada, r.data_saida
27 FROM Cliente c
28 JOIN Reserva r ON r.nif_cliente=c.nif
29 JOIN Quarto q ON r.numero_quarto=q.numero
30 WHERE c.nif NOT IN ( -- todos os clientes que fizeram reservas
31     SELECT c.nif -- fora de Abril ou Junho deste ano
32     FROM Cliente c
33     JOIN Reserva r ON r.nif_cliente=c.nif
34     WHERE NOT TRUNC(sysdate, 'YEAR')=TRUNC(r.data_reserva, 'YEAR') -- não foi este ano
35     OR (
36     NOT (EXTRACT(MONTH FROM r.data_entrada)=4 AND EXTRACT(MONTH FROM r.data_saida)=4) -- não foi em abril
37     AND NOT (EXTRACT(MONTH FROM r.data_entrada)=6 AND EXTRACT(MONTH FROM r.data_saida)=6) -- nem em junho
38     )
39 )
40 -- a = foi este ano = TRUNC(sysdate, 'YEAR')=TRUNC(r.data_reserva, 'YEAR')
41 -- b = foi exclusivamente em abril = (EXTRACT(MONTH FROM r.data_entrada)=4 AND EXTRACT(MONTH FROM r.data_saida)=4)
42 -- c = foi exclusivamente em junho = (EXTRACT(MONTH FROM r.data_entrada)=6 AND EXTRACT(MONTH FROM r.data_saida)=6)
43 -- todas as reservas fora de abril e junho deste ano = !a || (!b && !c)
44 ORDER BY c.nome, "data e hora" DESC;
45
```

Query Result x

SQL | All Rows Fetched: 2 in 0.15 seconds

	data e hora	NOME	Zona do País	DATA_ENTRADA	DATA_SAIDA
1	31-03-2020 12:03	Michelle Obama	Santa Cruz	20.04.10	20.04.17
2	01-02-2020 12:02	Moisés Rebelo		20.06.13	20.06.16

Imagem 5 - Pesquisa e respetivos resultados da parte 1.b

Parte II

Realizada por Luís Araújo (1190827).

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;

Inserções (input para a pesquisa)

```
-- ## tabela cliente ##
/** últimos 6 meses **/
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('José Silva', 111111111, 'jose@silva.com', 962994584, 'Norte', 'Alijó', 'Vila Real');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Eduarda Magalhães', 111111112, 'eduarda@magalhaes.com', 932855628, 'Sul', 'Albufeira', 'Algarve');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Telmo Custódio', 111111113, 'telmo@custodio.com', 912774511, 'Centro', 'Mealhada', 'Coimbra');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Marco Baião', 111111114, 'marco@baiao.com', 952918351, 'Norte', 'Ponte de Lima', 'Viana do Castelo');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Márlia Coimbra', 111111115, 'marilia@coimbra.com', 951003517, 'Sul', 'Lagos', 'Algarve');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('André Garrido', 111111116, 'andre@garido.com', 951966152, 'Centro', 'Penacova', 'Coimbra');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Pedro Lopes', 111111117, 'pedro@lopes.com', 961455528, 'Norte', 'Amares', 'Braga');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Dulce Lousã', 111111118, 'dulce@lousa.com', 910000361, 'Sul', 'Loulé', 'Algarve');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Ângela Martinho', 111111119, 'angela@martinho.com', 918882411, 'Centro', 'Amor', 'Leiria');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Piedade Lopes', 111111110, 'piiedade@lopes.com', 951733615, 'Norte', 'Esposende', 'Braga');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('João Neto', 111111411, 'joao@neto.com', 950016221, 'Sul', 'Santiago do Cacém', 'Alentejo');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Álvaro Novais', 111151111, 'alvaro@novais.com', 958133281, 'Centro', 'Milagres', 'Leiria');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Rafael Lagos', 111341111, 'rafael@lagos.com', 950015533, 'Norte', 'Maia', 'Porto');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Adelino Proença', 171111111, 'adelino@proenca.com', 950122350, 'Sul', 'Sines', 'Alentejo');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Matália Domingues', 111115611, 'natalia@domingues.com', 951077341, 'Centro', 'Lardosa', 'Castelo Branco');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Fátima Quintas', 111345111, 'fatima@quintas.com', 960001469, 'Norte', 'Trofa', 'Porto');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Cristina Ferreira', 117811111, 'cristina@ferreira.com', 913477712, 'Sul', 'Arraiolos', 'Alentejo');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Josué Aldeia', 111193111, 'josue@aldeia.com', 960933331, 'Centro', 'Tinalhas', 'Castelo Branco');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Francisco Guedes', 171111511, 'francisco@guedes.com', 961111731, 'Norte', 'Vimioso', 'Bragança');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Maria Leal', 111111781, 'maria@leal.com', 950144442, 'Sul', 'Mouriz', 'Alentejo');

/** fora dos últimos 6 meses **/
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Barack Obama', 174929363, 'barackoficial@obama.com', 961945188, 'Ilhas', 'Santa Cruz', 'Madeira');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Russell Lucas', 184729562, 'russell@lucas.com', 958889153, 'Centro', 'Calde', 'Viseu');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Moisés Rebelo', 174826492, 'moises@rebelo.com', 951119266, 'Sul', 'Portimão', 'Alentejo');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Kevin Felgueiras', 173048744, 'kevin@felgueiras.com', 951777251, 'Norte', 'Mirandela', 'Bragança');
INSERT INTO cliente(nome, nif, email, telefone, zona_Do_Pais, localidade, concelho) VALUES('Michelle Obama', 183603675, 'michelleoficial@obama.com', 955518451, 'Ilhas', 'Santa Cruz', 'Madeira');
```

Imagem 6- Inserts da tabela “cliente”

```
154 -- ## tabela reserva ##
155 -- época baixa
156 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(407, TO_DATE('2017-04-10','yy yy-mm-dd'), TO_DATE('2017-04-17','yyyy-mm-dd'), 222, 1, 1, 111911781, 4, 4, TO_DATE('2017-04-01','yyyy-mm-dd'));
157 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(404, TO_DATE('2018-04-11','yy yy-mm-dd'), TO_DATE('2018-04-14','yyyy-mm-dd'), 222, 4, 1, 911261781, 4, 4, TO_DATE('2018-04-01','yyyy-mm-dd'));
158 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(100, TO_DATE('2020-01-13','yy yy-mm-dd'), TO_DATE('2020-01-16','yyyy-mm-dd'), 230, 8, 1, 111111111, 4, 1, TO_DATE('2020-01-02','yyyy-mm-dd'));
159 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(500, TO_DATE('2020-05-13','yy yy-mm-dd'), TO_DATE('2020-05-16','yyyy-mm-dd'), 230, 9, 1, 184729562, 4, 2, TO_DATE('2020-04-30','yyyy-mm-dd'));
160 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(100, TO_DATE('2020-02-13','yy yy-mm-dd'), TO_DATE('2020-02-16','yyyy-mm-dd'), 115, 10, 1, 174826492, 1, 2, TO_DATE('2020-02-01','yyyy-mm-dd'));
161 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(512, TO_DATE('2020-03-15','yy yy-mm-dd'), TO_DATE('2020-03-18','yyyy-mm-dd'), 220, 11, 1, 173048744, 4, 4, TO_DATE('2020-03-03','yyyy-mm-dd'));
162 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(403.5, TO_DATE('2020-04-10','yy yy-mm-dd'), TO_DATE('2020-04-17','yyyy-mm-dd'), 232, 12, 1, 183603675, 4, 4, TO_DATE('2020-03-31','yyyy-mm-dd'));
163
164 -- época média
165 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(600, TO_DATE('2017-11-04','yy yy-mm-dd'), TO_DATE('2017-11-26','yyyy-mm-dd'), 210, 5, 2, 111710781, 2, 4, TO_DATE('2017-11-01','yyyy-mm-dd'));
166 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(600, TO_DATE('2018-10-01','yy yy-mm-dd'), TO_DATE('2018-10-08','yyyy-mm-dd'), 200, 6, 2, 111741781, 2, 4, TO_DATE('2018-09-20','yyyy-mm-dd'));
167 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(75, TO_DATE('2020-10-07','yy yy-mm-dd'), TO_DATE('2020-10-12','yyyy-mm-dd'), 115, 18, 2, 111111117, 1, 1, TO_DATE('2020-10-01','yyyy-mm-dd'));
168 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(753.5, TO_DATE('2020-10-13','yy yy-mm-dd'), TO_DATE('2020-10-16','yyyy-mm-dd'), 105, 19, 2, 111111116, 4, 2, TO_DATE('2020-10-02','yyyy-mm-dd'));
169 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(305.5, TO_DATE('2020-11-07','yy yy-mm-dd'), TO_DATE('2020-11-12','yyyy-mm-dd'), 135, 20, 2, 111111118, 4, 2, TO_DATE('2020-11-01','yyyy-mm-dd'));
170 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(300, TO_DATE('2020-11-15','yy yy-mm-dd'), TO_DATE('2020-11-21','yyyy-mm-dd'), 138, 21, 2, 111111119, 3, 2, TO_DATE('2020-11-01','yyyy-mm-dd'));
171 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(600, TO_DATE('2018-12-04','yy yy-mm-dd'), TO_DATE('2018-12-26','yyyy-mm-dd'), 200, 22, 2, 111511781, 2, 4, TO_DATE('2018-11-20','yyyy-mm-dd'));
172 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(600, TO_DATE('2020-12-09','yy yy-mm-dd'), TO_DATE('2020-12-16','yyyy-mm-dd'), 200, 23, 2, 111111118, 2, 4, TO_DATE('2020-12-01','yyyy-mm-dd'));
173
174 -- época alta
175 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(1007, TO_DATE('2017-08-01','yy yy-mm-dd'), TO_DATE('2017-08-10','yyyy-mm-dd'), 215, 2, 3, 191511781, 4, 2, TO_DATE('2017-07-20','yyyy-mm-dd'));
176 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(1000, TO_DATE('2017-09-03','yy yy-mm-dd'), TO_DATE('2017-09-11','yyyy-mm-dd'), 215, 3, 3, 111811781, 4, 2, TO_DATE('2017-08-25','yyyy-mm-dd'));
177 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(1006.5, TO_DATE('2019-07-03','yy yy-mm-dd'), TO_DATE('2019-07-11','yyyy-mm-dd'), 230, 7, 3, 117311781, 4, 2, TO_DATE('2019-06-20','yyyy-mm-dd'));
178 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(203.5, TO_DATE('2020-07-13','yy yy-mm-dd'), TO_DATE('2020-07-16','yyyy-mm-dd'), 206, 13, 3, 111111411, 4, 1, TO_DATE('2020-07-01','yyyy-mm-dd'));
179 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(206, TO_DATE('2020-07-16','yy yy-mm-dd'), TO_DATE('2020-07-22','yyyy-mm-dd'), 204, 14, 3, 111511111, 4, 1, TO_DATE('2020-07-01','yyyy-mm-dd'));
180 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(200, TO_DATE('2020-08-17','yy yy-mm-dd'), TO_DATE('2020-08-20','yyyy-mm-dd'), 224, 15, 3, 113411111, 1, 2, TO_DATE('2020-08-01','yyyy-mm-dd'));
181 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(400, TO_DATE('2020-08-21','yy yy-mm-dd'), TO_DATE('2020-08-24','yyyy-mm-dd'), 231, 16, 3, 171111111, 1, 2, TO_DATE('2020-08-01','yyyy-mm-dd'));
182 INSERT INTO reserva(custo, data_Entrada, data_Saida, numero_Quarto, cod_Reserva, id_Epoca, nif_Cliente, id_Estado, num_Pessoas, data_Reserva) VALUES(1006.5, TO_DATE('2020-09-03','yy yy-mm-dd'), TO_DATE('2020-09-10','yyyy-mm-dd'), 235, 17, 3, 111115611, 4, 2, TO_DATE('2020-08-20','yyyy-mm-dd'));
```

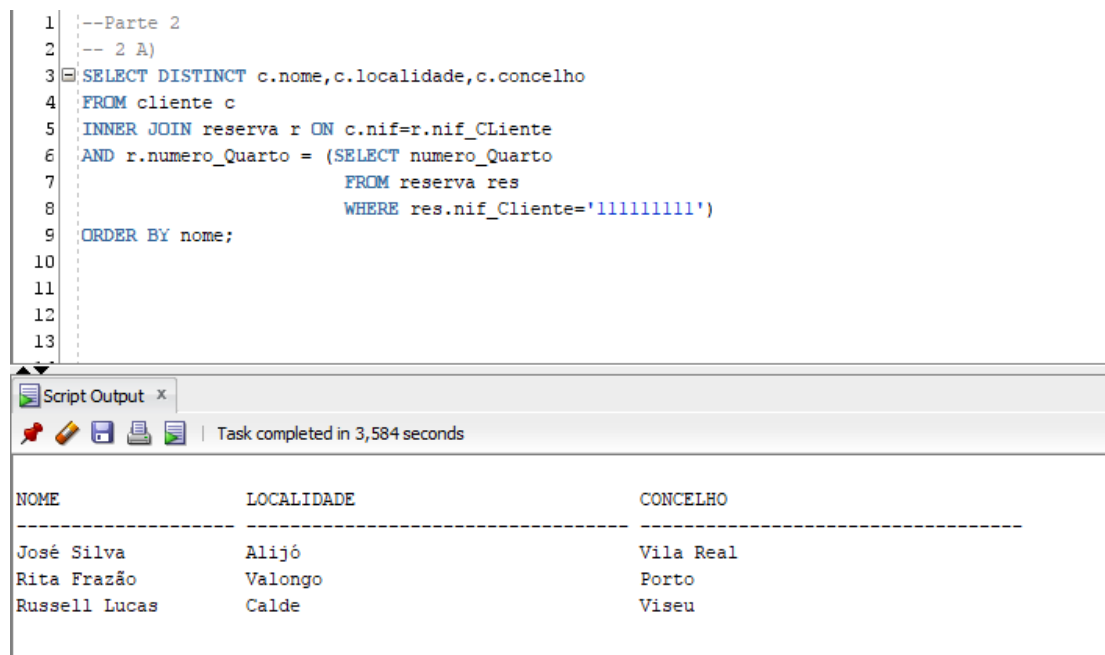
Imagem 7- Inserts da tabela "reserva"

Aspetos a ter em conta:

- Cliente José Silva tem o NIF=”111111111” e reside em Vila Real, Alijó;
- A reserva deste mesmo cliente encontra-se finalizada (id_Estado=4) e esteve no quarto com o número 230.

Nesta pesquisa pretendíamos obter o nome, localidade e concelho (SELECT DISTINCT c.nome, c.localidade, c.concelho FROM cliente c) dos clientes que já estiveram alojados no hotel. Para isso, o nif destes mesmos teria de se encontrar na tabela reserva como forma de garantir que já lá estiveram (INNER JOIN reserva r ON c.nif=r.nif_Cliente). Ainda assim, e como forma de garantir que estes clientes estiveram no mesmo quarto que o José Silva, utilizei uma subquery. Através desta o número do quarto dos outros clientes tinham de ser iguais ao número de quarto da reserva do José Silva, reserva esta que foi encontrada através do seu nif(“11111111”) como visto anteriormente na tabela cliente (AND r.numero_Quarto = (SELECT numero_Quarto FROM reserva res WHERE res.nif_Cliente=’11111111’)). No final ordenei por ordem alfabética (ORDER BY nome).

Pesquisa e resultados



```

1  --Parte 2
2  -- 2 A)
3  SELECT DISTINCT c.nome, c.localidade, c.concelho
4  FROM cliente c
5  INNER JOIN reserva r ON c.nif=r.nif_Cliente
6  AND r.numero_Quarto = (SELECT numero_Quarto
7                        FROM reserva res
8                        WHERE res.nif_Cliente='11111111')
9  ORDER BY nome;

```

Script Output x

Task completed in 3,584 seconds

NOME	LOCALIDADE	CONCELHO
José Silva	Alijó	Vila Real
Rita Frazão	Valongo	Porto
Russell Lucas	Calde	Viseu

Imagem 7- Pesquisa e respetivos resultados

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.

Inserções (input para a pesquisa)

```

247 -- ## tabela funcionario ##
248 -- 5 registos de camareira #acrescentar id_tipo_funcionario
249 INSERT INTO funcionario(id_Funcionario, nome, nif, morada, email, telefone) VALUES(1, 'Maria Barbosa', 183507249, 'Celeiros', 'maria@barbosa.com', 936428463);
250 INSERT INTO funcionario(id_Funcionario, nome, nif, morada, email, telefone) VALUES(2, 'Madalena Pereira', 159284026, 'Aveleda', 'madalena@pereira.com', 961734297);
251 INSERT INTO funcionario(id_Funcionario, nome, nif, morada, email, telefone) VALUES(3, 'Branca Rodrigues', 103273738, 'Sequeira', 'branca@rodrigues.com', 962007144);
252 INSERT INTO funcionario(id_Funcionario, nome, nif, morada, email, telefone) VALUES(4, 'Rute Lopes', 162852015, 'Lamego', 'rute@lopes.com', 936277751);
253 INSERT INTO funcionario(id_Funcionario, nome, nif, morada, email, telefone) VALUES(5, 'Bruna Carvalho', 183628529, 'Figueira da Foz', 'bruna@carvalho.com', 962610063);

```

Imagem 8- Inserts da tabela funcionario

```

371 | -- ## tabela limpeza ##
372 | INSERT INTO limpeza(id_Intervencao, cod_Reserva) VALUES(1, 14);
373 | INSERT INTO limpeza(id_Intervencao, cod_Reserva) VALUES(2, 17);
374 | INSERT INTO limpeza(id_Intervencao, cod_Reserva) VALUES(3, 3);
375 | INSERT INTO limpeza(id_Intervencao, cod_Reserva) VALUES(4, 6);
376 | INSERT INTO limpeza(id_Intervencao, cod_Reserva) VALUES(5, 8);
377 | INSERT INTO limpeza(id_Intervencao, cod_Reserva) VALUES(6, 10);
378 | INSERT INTO limpeza(id_Intervencao, cod_Reserva) VALUES(7, 12);
379 | INSERT INTO limpeza(id_Intervencao, cod_Reserva) VALUES(8, 13);
380 | INSERT INTO limpeza(id_Intervencao, cod_Reserva) VALUES(9, 15);
381 | INSERT INTO limpeza(id_Intervencao, cod_Reserva) VALUES(10, 20);

```

Imagem 9- Inserts da tabela limpeza

```

383 | -- ## tabela limpeza_Camareira ##
384 | INSERT INTO limpeza_Camareira(id_Intervencao, id_Funcionario) VALUES(1, 1);
385 | INSERT INTO limpeza_Camareira(id_Intervencao, id_Funcionario) VALUES(2, 2);
386 | INSERT INTO limpeza_Camareira(id_Intervencao, id_Funcionario) VALUES(3, 2);
387 | INSERT INTO limpeza_Camareira(id_Intervencao, id_Funcionario) VALUES(4, 3);
388 | INSERT INTO limpeza_Camareira(id_Intervencao, id_Funcionario) VALUES(5, 4);
389 | INSERT INTO limpeza_Camareira(id_Intervencao, id_Funcionario) VALUES(6, 4);
390 | INSERT INTO limpeza_Camareira(id_Intervencao, id_Funcionario) VALUES(7, 5);
391 | INSERT INTO limpeza_Camareira(id_Intervencao, id_Funcionario) VALUES(8, 5);
392 | INSERT INTO limpeza_Camareira(id_Intervencao, id_Funcionario) VALUES(9, 5);
393 | INSERT INTO limpeza_Camareira(id_Intervencao, id_Funcionario) VALUES(10, 3);

```

Imagem 10- Inserts da tabela limpeza_Camareira

```

11 | -- ## tabela quarto ##
12 | /** 1 andar */
13 | --single
14 | INSERT INTO quarto(lotacao, numero, numero_Andar, id_Tipo_Quarto) VALUES(1, 100, 1, 1);
15 | INSERT INTO quarto(lotacao, numero, numero_Andar, id_Tipo_Quarto) VALUES(1, 101, 1, 1);
16 | INSERT INTO quarto(lotacao, numero, numero_Andar, id_Tipo_Quarto) VALUES(1, 102, 1, 1);
17 | INSERT INTO quarto(lotacao, numero, numero_Andar, id_Tipo_Quarto) VALUES(1, 103, 1, 1);
18 | INSERT INTO quarto(lotacao, numero, numero_Andar, id_Tipo_Quarto) VALUES(1, 104, 1, 1);
19 | INSERT INTO quarto(lotacao, numero, numero_Andar, id_Tipo_Quarto) VALUES(1, 105, 1, 1);
20 | INSERT INTO quarto(lotacao, numero, numero_Andar, id_Tipo_Quarto) VALUES(1, 106, 1, 1);
21 | INSERT INTO quarto(lotacao, numero, numero_Andar, id_Tipo_Quarto) VALUES(1, 107, 1, 1);
22 | INSERT INTO quarto(lotacao, numero, numero_Andar, id_Tipo_Quarto) VALUES(1, 108, 1, 1);
23 | INSERT INTO quarto(lotacao, numero, numero_Andar, id_Tipo_Quarto) VALUES(1, 109, 1, 1);

```

Imagem 11- Inserts da tabela quarto

Aspetos a ter em conta:

- O “cod_Reserva” na tabela “limpeza” constava na tabela “reserva” (apresentada no exercício 2 a));
- O “id_Funcionario” na tabela “limpeza_Camareira” constava na tabela “funcionário” relativamente às camareiras, ou seja, os “id_Funcionario” só variavam entre 1 (inclusive) e 5 (inclusive);
- O “id_Intervencao” na tabela “limpeza_Camareira” constava na tabela “limpeza” e ambos variavam entre os valores 1 (inclusive) e 10 (inclusive);
- Os números de quartos (“numero”) da tabela “quarto” (parte dos inserts apresentados acima) constavam na tabela reserva (apresentada no exercício 2 a)).

Nesta pesquisa pretendíamos obter as camareiras que realizaram mais intervenções desde a data atual até há 6 meses atrás em quartos cujas durações de estadias foram superiores à média de estadias por tipo de quartos.

Para uma mais fácil realização deste exercício separei esta mesma pergunta em partes. Comecei por obter os quartos (SELECT DISTINCT numero_Quarto FROM reserva res) cuja duração de estadia era maior do que a média de estadias por tipo de quarto. Assim, e para obter a primeira parte deste problema apenas subtraí a data de saída pela data de entrada para calcular o tempo no hotel (WHERE ((res.data_Saida) – (res.data_Entrada))). Após isso calculei o tempo médio de estadia por tipo de quarto (SELECT (AVG(res.data_Saida – res.data_Entrada) “tipo_quarto” FROM reserva res) afirmando que o número de quarto na tabela “reserva” era o mesmo que na tabela “quarto” e que o tipo de quarto na tabela “quarto” era o mesmo que na tabela “tipo_De_Quarto” (INNER JOIN quarto q ON res.numero_Quarto=q.numero INNER JOIN tipo_De_Quarto tq ON q.id_Tipo_Quarto=tq.id_Tipo).

Após isso apresentei o id das camareiras juntamente com o seu nome mostrando que o id de funcionário da tabela “camareira” era o mesmo da tabela “funcionário” (SELECT c.id_Funcionario, f.nome FROM camareira c JOIN funcionario f ON f.id_Funcionario=c.id_Funcionario). Ainda assim comprovei que o id de funcionário da tabela “funcionário” era o mesmo que na tabela “limpeza_Camareira” (JOIN limpeza_Camareira lc ON lc.id_Funcionario=c.id_Funcionario) bem como o id de intervenção nesta tabela “limpeza_Camareira” era o mesmo que na tabela “limpeza” (JOIN limpeza l ON l.id_Intervencao=lc.id_Intervencao).

Depois, e para garantir que estes dados apenas se limitavam ao intervalo da data de hoje (22 de novembro) até há 6 meses atrás comprovei que o código da reserva na tabela “reserva” era o mesmo da tabela “limpeza” e utilizei uma função do Oracle (add_months) que retorna a data mais (ou menos) um certo número de meses. Para isso afirmei que a data de hoje teria de ser maior ou igual que a data de hoje menos 6 meses atrás (sysdate>=add_months(sysdate, -6)).

Por fim, comprovei que o número de quarto da tabela “quarto” era o mesmo da tabela “reserva” e afirmei que este mesmo teria de estar na “subquerie” que tratei no primeiro ponto.

```

11 -- 2 B)
12 SELECT c.id_Funcionario, f.nome
13 FROM camareira c
14 JOIN funcionario f ON f.id_Funcionario=c.id_Funcionario
15 JOIN limpeza_Camareira lc ON lc.id_funcionario=c.id_funcionario
16 JOIN limpeza l ON l.id_intervencao=lc.id_intervencao
17 JOIN reserva r ON l.cod_reserva=r.cod_reserva AND
18 sysdate>=add_months(sysdate, -6)
19 JOIN Quarto q ON r.numero_quarto=q.numero
20 WHERE q.numero IN (
21 SELECT DISTINCT numero_Quarto
22 FROM reserva res
23 WHERE ((res.data_Saida) – (res.data_Entrada)) > (SELECT (AVG(res.data_Saida – res.data_Entrada)) "tipo_quarto"
24 FROM reserva res
25 INNER JOIN quarto q ON res.numero_Quarto=q.numero
26 INNER JOIN tipo_De_Quarto tq ON q.id_Tipo_Quarto=tq.id_Tipo)
27 )
28

```

Query Result	
All Rows Fetched: 6 in 0,195 seconds	
ID_FUNCIONARIO	NOME
1	2 Madalena Pereira
2	5 Bruna Carvalho
3	3 Branca Rodrigues
4	1 Maria Barbosa
5	4 Rute Lopes
6	2 Madalena Pereira

Imagem 12- Pesquisa e respetivos resultados

Parte III

Realizado por Danilton Lopes (1191240)

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.

Síntese do que é relevante para a pesquisa:

- Existem 4 quartos com 2 ou mais reservas: Estas que são 219, 222, 223 e 200;
- Mas o quarto 200 é single, logo não deverá aparecer na pesquisa;
- As tabelas estão relacionadas de forma direta ou indireta o que facilita com que tenhamos informações que precisamos para encontrar os dados necessários para a pesquisa;
- Das tabelas anteriores a 200 tem uma reservada cancelado, mas como é single não conta.

```
SELECT numero, (  
SELECT tipo FROM tipo_De_Quarto  
WHERE quarto.id_Tipo_Quarto=id_Tipo) AS "Tipo De Quarto"  
FROM quarto  
WHERE id_Tipo_Quarto!=1 AND numero IN (  
SELECT numero_Quarto FROM reserva  
WHERE id_Estado!=1  
GROUP BY numero_quarto  
HAVING count(numero_Quarto)>1);
```

Imagem 13 - SELECT para Parte III a)

Para esta pesquisa utilizamos uma subquerie para obter o tipo de quarto à tabela “tipo_De_Quarto” através do “id_Tipo_De_Quarto” da tabela “quarto”.

Depois começamos a definir as condições do SELECT, começando por definir que o tipo de quarto seja diferente de “single”, em que o id_Tipo foi definida como 1, e também que todos os quartos tenham pelo menos 2 reservas, excluindo os que tenham estado “cancelada”, que são os mesmos que tem o id_Estado igual a 1.

E o resultado obtido foi:

	NUMERO	Tipo De Quarto
1	219	twin
2	222	superior
3	230	suite

Imagem 14 - Resultado para Parte III a)

Que foram os dados dos quartos que tem 2 ou mais reservas registadas e que o estado é diferente de cancelada.

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.

Síntese do que é relevante para a pesquisa:

- Existem muitos clientes que consumiram os dois produtos mais consumidos;
- Existem 20 quartos do tipo suite
- A única cliente que consumiu os 2 produtos mais consumidos e fez a reserva de uma suite foi a Natália Domingues

```

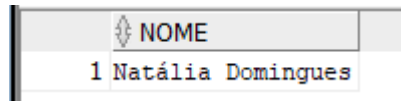
SELECT nome FROM cliente
WHERE nif IN (
  SELECT nif_Cliente FROM reserva
  WHERE numero_Quarto IN (SELECT numero FROM quarto
  WHERE id_Tipo_Quarto=4))
AND
nif IN (
  SELECT nif_Cliente FROM reserva
  WHERE data_Entrada BETWEEN (SELECT data_Inicio FROM epoca WHERE id_Epoca=3) AND (SELECT data_Fim FROM epoca WHERE id_Epoca=3))
AND
nif IN (
  SELECT nif_Cliente FROM reserva
  WHERE data_Entrada>=TO_DATE('2018','yyyy'))
AND
nif IN (
  SELECT nif_Cliente FROM reserva
  WHERE cod_Reserva IN (
    SELECT cod_Reserva FROM conta
    WHERE num_Conta IN (
      SELECT num_Conta FROM conta
      WHERE num_Conta IN (
        SELECT num_Conta FROM consumo
        WHERE id_Produto IN (
          SELECT id_Produto FROM consumo
          GROUP BY id_Produto
          ORDER BY count(*) DESC
          FETCH FIRST ROW ONLY))))))
AND
nif IN (
  SELECT nif_Cliente FROM reserva
  WHERE cod_Reserva IN (
    SELECT cod_Reserva FROM conta
    WHERE num_Conta IN (
      SELECT num_Conta FROM conta
      WHERE num_Conta IN (
        SELECT num_Conta FROM consumo
        WHERE id_Produto IN (
          SELECT id_Produto FROM (
            SELECT id_Produto, ROWNUM rnum FROM (
              SELECT id_Produto FROM consumo
              GROUP BY id_Produto
              ORDER BY count(*) DESC)
            WHERE ROWNUM <=2
          )
          WHERE rnum = 2))))))

```

Imagem 15 - SELECT PARTE III b

Esta pesquisa foi mais exigente pois exigiu muitas subquerie para podermos obter dados como a segunda linha dos produtos mais consumidos, o número da conta que está relacionado ao código de reserva e ainda ordenar pela ordem decrescente do total do consumo.

E o resultado foi o nome da única cliente que havia consumido os 2 produtos mais consumidos e respeitava todos os requisitos:



	NOME
1	Natália Domingues

Imagem 16 - Output do Select III b