

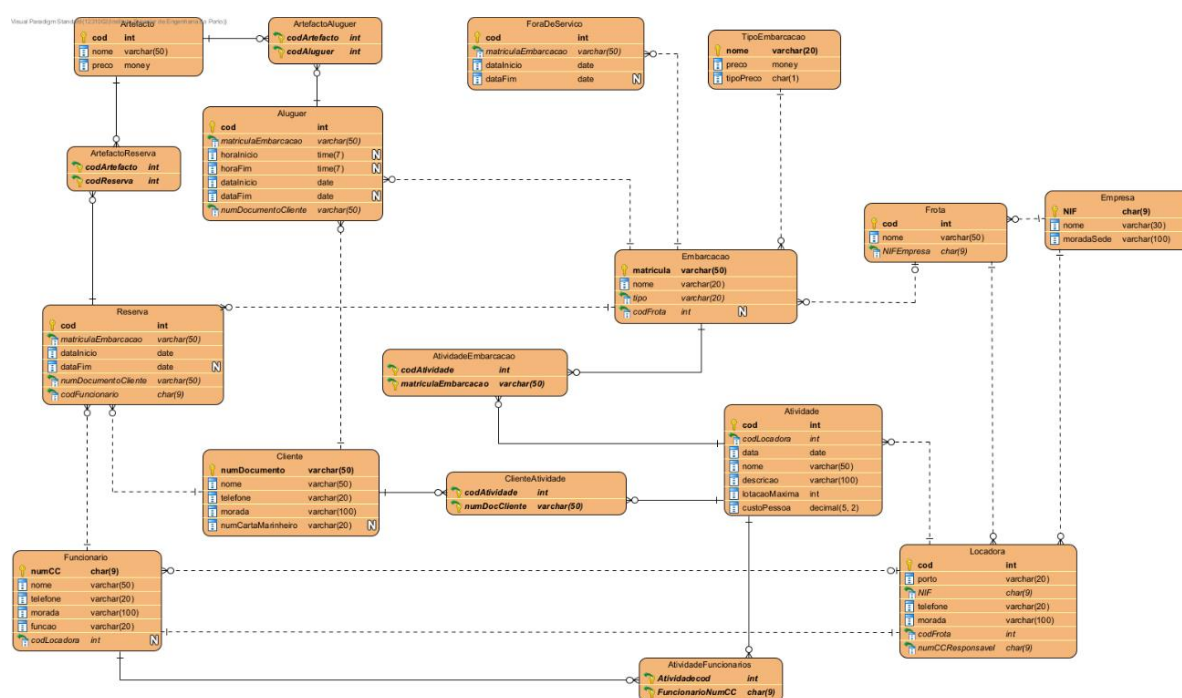
# ISEPBarcos

Ana Lagame, 1221410  
Sónia Castro, 1221710  
Luís Oliveira, 1231002

# 1 Introdução

No âmbito da disciplina Base de Dados e Armazém de Dados da Licenciatura de Engenharia de Telecomunicações e Informática foi-nos proposto a realização de um trabalho sobre uma empresa especializada em aluguer de barcos, a ISEPBarcos. Para isto, foi necessário construir um modelo Entidade-Relação, elaborar *scripts* para criação de tabelas em *SQL Server*, assim como, introdução de dados nestas e consultas.

## 2 Modelo ER



Este modelo foi desenvolvido no âmbito de demonstrar as relações entre entidades. Começou-se por desenvolver a tabela Empresa para guardar os dados, neste caso, da **ISEPBarcos**. A empresa possui locadoras e frotas, existindo uma ligação entre a empresa e frota e também entre frota e locadora, isto porque, é indicado que tanto a empresa como a locadora têm frotas, no entanto, uma frota pode não estar atribuída a uma locadora, mesmo pertencendo à empresa.

Uma locadora tem apenas um funcionário responsável, por isso colocou-se o número de cartão de cidadão do responsável nessa tabela. Uma locadora também engloba vários funcionários, e por essa razão, aparece uma segunda ligação entre estas tabelas. Esta ligação permite deduzir que um funcionário pode ser contratado, mas não ser atribuído a uma locadora imediatamente.

Uma frota, engloba várias embarcações, porém, uma embarcação, no momento do registo, pode ainda não estar alocada, e por isso, o atributo *codFrota* poder ser um valor **NULL**.

Para registar o tipo de embarcação foi criada uma tabela à parte, porque dependendo do tipo, o preço pode ser contabilizado por dia ou por hora.

Uma embarcação pode ter quatro estados: reservado, alugado, fora de serviço ou livre. Esta validação pode ser feita, no momento de pesquisa, verificando se a embarcação se encontra numa das tabelas ForaDeServiço, Aluguer ou Reserva, e se o atributo dataFim desse registo é **NULL**. Caso não se encontre, o seu estado é livre. É importante mencionar, que os atributos horainicio e horaFim da tabela Aluguer, podem ser **NULL**, pois estes campos são referentes a dois tipos de embarcações (Remos e Gaivota), sendo que as duas restantes estão associadas apenas a dataInicio e dataFim.

### 3 Consultas

#### Consultas – Parte II

**2.1 Listar para o cliente com o cartão de cidadão com o número '123456789', o seu nome, a data, as suas reservas, o tipo e nome das embarcações utilizadas.**

```
SELECT
    Cliente.nome AS NomeCliente,
    Reserva.dataInicio AS DataInicioReserva,
    Reserva.dataFim AS DataFimReserva,
    Embarcacao.nome AS NomeEmbarcacao,
    Embarcacao.tipo AS TipoEmbarcacao
FROM
    Cliente
JOIN
    Reserva ON Cliente.numDocumento = Reserva.numDocumentoCliente
JOIN
    Embarcacao ON Reserva.matriculaEmbarcacao = Embarcacao.matricula
WHERE
    Cliente.numDocumento = '123456789';
```

	NomeCliente	DataInicioReserva	DataFimReserva	NomeEmbarcacao	TipoEmbarcacao
1	João Santos	2016-05-24	2016-05-28	Motor1	Motor
2	João Santos	2016-06-01	2016-06-05	Vela2	Vela
3	João Santos	2016-07-22	2016-07-23	Vela2	Vela
4	João Santos	2017-07-26	2017-07-30	Vela5	Vela

Esta *query* utiliza junções (*JOINS*) para combinar informações de três tabelas relacionadas: Cliente, Reserva e Embarcação. Começa por selecionar os campos relevantes de cada tabela, como o nome do cliente, data das reservas, e informações das embarcações. A junção entre **Cliente** e **Reserva** é feita com base no campo numDocumento, enquanto a junção entre **Reserva** e **Embarcacao** utiliza o campo matricula. A cláusula **WHERE** filtra os dados para incluir apenas o cliente com o número de documento '123456789'. Por fim, os resultados mostram os dados combinados para este cliente específico, agrupando informações de todas as tabelas relacionadas.

## 2.2 Listar os clientes que realizaram alugueres ou reservas em 2016 e que ainda não efetuaram alugueres ou reservas em 2017.

```

SELECT DISTINCT *
FROM Cliente C
WHERE C.numDocumento IN (
    SELECT A.numDocumentoCliente
    FROM Aluguer A
    WHERE YEAR(A.dataInicio) = 2016

    UNION

    -- Reservas em 2016
    SELECT R.numDocumentoCliente
    FROM Reserva R
    WHERE YEAR(R.dataInicio) = 2016
)
AND C.numDocumento NOT IN (
    SELECT A.numDocumentoCliente
    FROM Aluguer A
    WHERE YEAR(A.dataInicio) = 2017

    UNION

    SELECT R.numDocumentoCliente
    FROM Reserva R
    WHERE YEAR(R.dataInicio) = 2017
);

```

	numDocumento	nome	telefone	morada	numCartaMarinheiro
1	004221182	Mariano Amaral	+351 935879896	Avenida João C Real 120, 3865-214	NULL
2	099339688	Elder Oliveira	+351 965659896	R Palmeira 110, 2495-017	732208878
3	128956789	Alex Mendez	+351 935452130	R Refugo 67, 9760-571	452163598
4	133313027	Vitória Rebelo	+351 915235698	Rua Mouco 116, 2750-802	808862904
5	163915478	Hermínia Guerra	+351 915423215	Avenida Madre Andaluz 77, 2040-103	183448334
6	165699226	Josué Silva	+351 935452789	R Camões 79, 3515-563	NULL
7	207511995	Alexandre de Freitas	+351 935632897	Avenida Júlio Dinis 70, 1200-287	523675984
8	217526761	Inácio Azevedo	+351 964712473	Avenida Francelos 23, 4460-820	402875706
9	239220318	Celso Freire	+351 915896547	R Camões 56, 3520-108	452632879
10	301286967	Mónica Rodrigues	+351 911111111	R Camões 57, 3525-999	NULL
11	318272510	Alexandre Varela	+351 935236489	Travessa Choupelo 32, 4475-128	456235987
12	343781818	Eduarda Vila	+351 965533253	Avenida Praia 28, 3100-634	NULL
13	346409304	Doroteia Freitas	+351 912656598	R Portela 61, 3530-160	953375591
14	350225540	Cristina Saraiva	+351 965685685	R Família Marques 20, 2870-112	NULL
15	359326391	Bernardino Venân...	+351 914785647	Bairro St António 82, 4950-355	NULL
16	378063581	Lara Martins	+351 935986321	Rua Vale Formoso 115, 8100-297	158069503
17	388159795	Bartolomeu Ramos	+351 963775660	Rua Nogueiras 1, 4630-084	NULL
18	460868275	Lisandro do Rosário	+351 968562347	Avenida Doutor Fernando Gomes 1...	541232597
19	513317199	Júlia Varela	+351 915557550	R Luís Sardoeira 6, 4600-720	120076631
20	514394036	Ângelo Silva	+351 917447880	R Camões 19, 3515-672	402589014
21	576749267	Gilda Rocha	+351 915654150	Rua Pero Vaz Caminha 117, 3045-4...	975299747
22	698925581	Sol Medeiros	+351 915652565	R Tapada Marinha 120, 4450-395	581522351
23	711200030	Mónica Cruz	+351 933578751	R Pádua Correia 2, 4430-690	NULL
24	816046347	Iria Varejão	+351 969853265	R 1º Dezembro 50, 4935-706	394380317
25	892654807	Cecília Barros	+351 917447008	Rua Pé Mouro Armz 69, 2740-129	NULL
26	918063606	Olga Alves	+351 969856431	Rua Doutor Luís Bettencourt 85, 95...	015494582

Esta *query* identifica clientes que realizaram alugueres ou reservas em 2016, mas que não fizeram nenhuma das duas em 2017. Primeiro, obtém na tabela Aluguer e Reserva os clientes com nelas inseridos em 2016, unindo os resultados dessas tabelas. Em seguida, verifica na mesma forma os clientes que inseridos em 2017. Por fim, compara os dois conjuntos, retornando apenas os clientes que aparecem na lista de 2016, mas não na de 2017, garantindo que os dados sejam únicos com o uso de **DISTINCT**.

### 2.3 Listar todas as embarcações, apresentando para cada uma, o número de vezes que foi utilizada (alugada ou reservada).

```
SELECT
    E.matricula,
    E.nome,
    COALESCE(R.numReservas, 0) AS totalReservas,
    COALESCE(A.numAlugueres, 0) AS totalAlugueres,
    COALESCE(R.numReservas, 0) + COALESCE(A.numAlugueres, 0) AS
totalUtilizacoes
FROM Embarcacao E
LEFT JOIN(
    SELECT
        matriculaEmbarcacao,
        COUNT(*) AS numReservas
    FROM Reserva
    GROUP BY matriculaEmbarcacao
)R ON E.matricula = R.matriculaEmbarcacao
LEFT JOIN(
    SELECT
        matriculaEmbarcacao,
        COUNT(*) AS numAlugueres
    FROM Aluguer
    GROUP BY matriculaEmbarcacao
)A ON E.matricula = A.matriculaEmbarcacao
ORDER BY totalUtilizacoes DESC, E.nome;
```

	matricula	nome	totalReservas	totalAlugueres	totalUtilizacoes
1	2-VE-001	Vela2	27	36	63
2	10-GA-001	Gaivota10	0	7	7
3	1-MO-001	Motor1	3	1	4
4	5-VE-001	Vela5	1	1	2
5	1-GA-001	Gaivota1	0	1	1
6	14-GA-001	Gaivota14	0	1	1
7	18-GA-001	Gaivota18	0	1	1
8	2-GA-001	Gaivota2	0	1	1
9	20-GA-001	Gaivota20	0	1	1
10	27-GA-001	Gaivota27	0	1	1
11	3-GA-001	Gaivota3	0	1	1
12	4-GA-001	Gaivota4	0	1	1
13	5-GA-001	Gaivota5	0	1	1
14	14-MO-001	Motor14	0	1	1
15	18-MO-001	Motor18	0	1	1
16	2-MO-001	Motor2	0	1	1
17	20-MO-001	Motor20	0	1	1
18	27-MO-001	Motor27	0	1	1
19	3-MO-001	Motor3	0	1	1
20	4-MO-001	Motor4	0	1	1
21	5-MO-001	Motor5	0	1	1
22	1-RE-001	Remos1	0	1	1
23	14-RE-001	Remos14	0	1	1
24	18-RE-001	Remos18	0	1	1
25	2-RE-001	Remos2	0	1	1
26	20-RE-001	Remos20	0	1	1
27	27-RE-001	Remos27	0	1	1
28	3-RE-001	Remos3	0	1	1
29	4-RE-001	Remos4	0	1	1

(**NOTA:** ESTA IMAGEM É APENAS UM RECORTE DOS DADOS OBTIDOS, POIS A TABELA ORIGINAL OBTINHA 120 RESULTADOS)

Esta *query* apresenta uma lista de todas as embarcações, mostrando o seu número total de reservas, alugueres e o total de utilizações (somatório de ambos). A partir da tabela **Embarcação**, realiza-se um LEFT JOIN com subconsultas que agregam o número de reservas e alugueres por matrícula, utilizando as tabelas **Reserva** e **Aluguer**, respectivamente. As funções COALESCE garantem que embarcações sem reservas ou alugueres sejam exibidas com um valor zero nos respectivos campos. O resultado é ordenado em ordem decrescente pelo total de utilizações e, em caso de empate, alfabeticamente pelo nome da embarcação.

**2.4 Liste o nome e tipo da(s) embarcação(ões) que em 2017 esteve(estiveram) mais vezes fora de serviço.**

```
WITH ForaDeServicoCount AS (
    SELECT
        e.nome AS nomeEmbarcacao,
        e.tipo AS tipoEmbarcacao,
```

```

        COUNT(*) AS totalForaDeServico
    FROM
        ForaDeServico fs
    INNER JOIN
        Embarcacao e ON fs.matriculaEmbarcacao = e.matricula
    WHERE
        YEAR(fs.dataInicio) = 2017
    GROUP BY
        e.nome, e.tipo
)
SELECT
    nomeEmbarcacao,
    tipoEmbarcacao
FROM
    ForaDeServicoCount
WHERE
    totalForaDeServico = (SELECT MAX(totalForaDeServico) FROM
    ForaDeServicoCount);

```

	nomeEmbarcacao	tipoEmbarcacao
1	Gaivota24	Gaivota

Esta *query* identifica as embarcações que ficaram mais vezes fora de serviço no ano de 2017. Utilizando uma **Common Table Expression (CTE)** chamada ForaDeServicoCount, ela calcula o total de ocorrências em que cada embarcação esteve fora de serviço, agrupando os resultados pelo nome e tipo da embarcação. Em seguida, na consulta principal, são selecionadas as embarcações com o maior número de ocorrências de indisponibilidade, determinado pelo subquery que obtém o valor máximo de totalForaDeServico na CTE. Este processo permite destacar as embarcações com o maior índice de indisponibilidade.

## 2.5 Liste os clientes que reservaram as mesmas embarcações que o cliente com o cartão de cidadão '123456789' durante o ano de 2016

```

WITH ReservasDoCliente AS(
    SELECT DISTINCT
        matriculaEmbarcacao
    FROM
        Reserva
    WHERE
        numDocumentoCliente = '123456789'
        AND YEAR(dataInicio) = 2016
)
SELECT DISTINCT
    nome AS nomeCliente,
    numDocumento AS documentoCliente,
    telefone,
    morada,
    numCartaMarinheiro
FROM
    Cliente
INNER JOIN
    Reserva ON Cliente.numDocumento = Reserva.numDocumentoCliente

```



```

WHERE
matriculaEmbarcacao IN (SELECT matriculaEmbarcacao FROM
ReservasDoCliente)
AND YEAR(dataInicio) = 2016
AND numDocumento != '123456789';

```

	nomeCliente	documentoCliente	telefone	morada	numCartaMarinheiro
1	Mariano Amaral	004221182	+351 935879896	Avenida João C Real 120, 3865-214	NULL
2	Alex Mendez	128956789	+351 935452130	R Refugo 67, 9760-571	452163598
3	Vitória Rebelo	133313027	+351 915235698	Rua Mouco 116, 2750-802	808862904
4	Josué Silva	165699226	+351 935452789	R Camões 79, 3515-563	NULL
5	Bernardo Ca...	186067311	+351 915447550	Avenida Forças Armadas 90, 1200...	678625819
6	Alexandre de ...	207511995	+351 935632897	Avenida Júlio Dinis 70, 1200-287	523675984
7	Mónica Rodri...	301286967	+351 911111111	R Camões 57, 3525-999	NULL
8	Alexandre Var...	318272510	+351 935236489	Travessa Choupelo 32, 4475-128	456235987
9	Eduarda Vila	343781818	+351 965533253	Avenida Praia 28, 3100-634	NULL
10	Doroteia Freitas	346409304	+351 912656598	R Portela 61, 3530-160	953375591
11	Cristina Saraiva	350225540	+351 965685685	R Família Marques 20, 2870-112	NULL
12	Lara Martins	378063581	+351 935986321	Rua Vale Fomoso 115, 8100-297	158069503
13	Bartolomeu R...	388159795	+351 963775660	Rua Nogueiras 1, 4630-084	NULL
14	Lisandro do R...	460868275	+351 968562347	Avenida Doutor Fernando Gomes ...	541232597
15	Júlia Varela	513317199	+351 915557550	R Luís Sardoeira 6, 4600-720	120076631
16	Ángelo Silva	514394036	+351 917447880	R Camões 19, 3515-672	402589014
17	Frederica Fer...	680244098	+351 965984213	Avenida Madre Andaluz 24, 2025-...	912119066
18	Sol Medeiros	698925581	+351 915652565	R Tapada Marinha 120, 4450-395	581522351
19	Mónica Cruz	711200030	+351 933578751	R Pádua Correia 2, 4430-690	NULL
20	Iria Varejão	816046347	+351 969853265	R 1ª Dezembro 50, 4935-706	394380317
21	Emiliana Couto	930766405	+351 911112222	R Bacelo 109, 4770-367	NULL

Esta *query* lista os clientes que reservaram as mesmas embarcações reservadas pelo cliente identificado pelo número de documento '123456789' durante o ano de 2016, excluindo este cliente da listagem. Primeiro, uma **Common Table Expression (CTE)** chamada **ReservasDoCliente** identifica as embarcações reservadas por esse cliente no período especificado. Em seguida, na consulta principal, as informações dos clientes que realizaram reservas nas mesmas embarcações, no mesmo ano, são recuperadas por meio de uma junção entre as tabelas **Cliente** e **Reserva**. A cláusula **WHERE** filtra as reservas relevantes, assegurando que o cliente de referência não seja incluído nos resultados.

## 2.6 Apresente o NIF e o nome do(s) cliente(s) com mais alugueres do que os clientes que fizeram alugueres na locadora de código '123'.

```

SELECT nome, numDocumento, COUNT(Aluguer.cod) AS nrAluguer
FROM Cliente
LEFT JOIN Aluguer ON Cliente.numDocumento =
Aluguer.numDocumentoCliente
GROUP BY nome, numDocumento
HAVING COUNT(DISTINCT Aluguer.cod) > (
    SELECT MAX(countAluguer) AS maximo
    FROM (
        SELECT clientes123.nome, COUNT(DISTINCT Aluguer.cod) AS
        countAluguer
        FROM (
            SELECT Cliente.nome, Cliente.numDocumento
            FROM Cliente

```



```

LEFT JOIN Aluguer ON Cliente.numDocumento =
    Aluguer.numDocumentoCliente
LEFT JOIN Embarcacao ON Embarcacao.matricula =
    Aluguer.matriculaEmbarcacao
LEFT JOIN Frota ON Embarcacao.codFrota =
Frota.cod
LEFT JOIN Locadora ON Locadora.codFrota =
Frota.cod
WHERE Locadora.cod LIKE '123'
) clientes123
LEFT JOIN Aluguer ON clientes123.numDocumento =
    Aluguer.numDocumentoCliente
GROUP BY clientes123.nome
) nrAlugueres)

```

	nome	numDocumento	nrAluguer
1	Bernardo Campos	186067311	11

Nesta *query* temos uma primeira *subquery* “clientes123” que apanha todos os clientes que fizeram aluguer em embarcações que pertencem a uma frota da locadora “123”; apanhando estes clientes é feito um *SELECT* do nome que retorna da “clientes123” e *COUNT* dos códigos de aluguer, que para ter acesso ao código foi necessário *JOIN* com a tabela **Aluguer**; seguidamente é feito o *MAX()* do *COUNT* obtido na *subquery* “nrAlugueres” para obtermos o valor máximo de alugueres feitos; assim, na cláusula *HAVING* filtramos (com um *COUNT* dos códigos de alugueres) os totais maiores que os totais de alugueres dos clientes que fizeram aluguer na locadora “123”, demonstrando na tabela o nome, o número de documento e o *COUNT* dos alugueres.

**2.7 Apresente a lista das locadoras (porto e morada) incluindo, se existirem, as atividades nelas desenvolvidas no 3.º trimestre de 2016. Das atividades, apresente apenas o nome e a descrição. A lista deve vir ordenada por nome da locadora (1.º critério) e nome da atividade (2.º critério).**

```

SELECT
Locadora.porto,
Locadora.morada,
Atividade.nome AS nomeAtividade,
Atividade.descricao
FROM
Locadora
LEFT JOIN
Atividade ON Locadora.cod = Atividade.codLocadora AND data BETWEEN
'2016-07-01' AND '2016-09-30'
ORDER BY

```

Locadora.porto, Atividade.nome

	porto	morada	nomeAtividade	descricao
1	Angra do Heroísmo	Rua das Baleias 30, 9700-120	NULL	NULL
2	Aveiro	Largo da Ria 3, 3800-300	NULL	NULL
3	Canical	Travessa do Farol 17, 9225-180	NULL	NULL
4	Cascais	Rua do Porto 6, 2750-700	NULL	NULL
5	Faro	Praça das Ondas 5, 8000-040	NULL	NULL
6	Figueira da Foz	Praça do Forte 7, 3080-400	NULL	NULL
7	Funchal	Praça do Atlântico 11, 9000-0...	Aventura Ná...	Explora...
8	Horta	Avenida do Canal 25, 9900-110	NULL	NULL
9	Lagos	Avenida da Marina 14, 8600-0...	NULL	NULL
10	Lajes do Pico	Travessa das Marés 11, 9930...	NULL	NULL
11	Leixões	Praça do Mar 37, 3876-200	Passeio Edu...	Explora...
12	Lisboa	Avenida das Naus 10, 1200-8...	NULL	NULL
13	Machico	Praça do Cabo 22, 9200-170	NULL	NULL
14	Nazaré	Rua do Farol 25, 2450-500	NULL	NULL
15	Olhão	Travessa do Porto 9, 8700-050	NULL	NULL
16	Peniche	Travessa da Lota 18, 2520-600	Comida de B...	Competi...
17	Ponta Delgada	Travessa do Oceano 3, 9500-...	NULL	NULL
18	Portimão	Rua da Enseada 22, 8500-030	NULL	NULL
19	Porto	Avenida da Ribeira 20, 4050-...	Sessão de P...	Pesca ...
20	Porto Santo	Rua do Leste 18, 9400-090	NULL	NULL
21	Rabo de Peixe	Rua da Pescaria 20, 9600-160	NULL	NULL
22	Ribeira Brava	Rua do Sol 12, 9350-190	NULL	NULL
23	São Vicente	Avenida das Palmeiras 9, 924...	Curso de Sal...	Treina...
24	Sesimbra	Rua da Praia 11, 4900-100	NULL	NULL
25	Setúbal	Praça do Mar 12, 2900-900	NULL	NULL
26	Sines	Rua do Horizonte 8, 7520-010	NULL	NULL
27	St Cruz das Flores	Avenida do Porto 18, 9970-150	NULL	NULL
28	Tavira	Avenida das Gaivotas 13, 880...	NULL	NULL
29	Vila do Porto	Praça dos Marinheiros 7, 958...	NULL	NULL
30	VR de St António	Rua do Golfo 15, 8900-070	Comida de Je...	Competi...

Esta *query* exibe informações sobre as atividades realizadas por cada locadora no período de julho a setembro de 2016, incluindo as locadoras que não realizaram atividades nesse intervalo. A consulta utiliza um LEFT JOIN para combinar as tabelas **Locadora** e **Atividade**, garantindo que todas as locadoras sejam exibidas, mesmo aquelas sem atividades no período especificado. Os dados incluem o porto e a morada de cada locadora, além do nome e descrição das atividades associadas (quando disponíveis). Por fim, os resultados são organizados alfabeticamente pelo nome do porto e, em seguida, pelo nome das atividades.

**2.8 Apresente as embarcações da(s) locadora(s) situadas no porto de “Sesimbra” que tiveram reservas ou alugueres todos os dias entre ‘01-07-2016’ e ‘31-08-2016’. Apresente o número de alugueres e reservas.**

```

SELECT matricula, embarcacao.nome, tipo, COUNT(DISTINCT Aluguer.cod)
AS numeroAlugueres, COUNT(DISTINCT Reserva.cod) AS numeroReservas
FROM Embarcacao
LEFT JOIN frota ON Embarcacao.codFrota = Frota.cod
LEFT JOIN locadora ON frota.cod = locadora.codFrota
LEFT JOIN Aluguer ON embarcacao.matricula =
aluguer.matriculaEmbarcacao AND Aluguer.dataInicio BETWEEN '2016-07-
01' AND '2016-08-31'
LEFT JOIN Reserva ON Embarcacao.matricula =
Reserva.matriculaEmbarcacao AND Reserva.dataInicio BETWEEN '2016-07-
01' AND '2016-08-31'
WHERE porto LIKE 'Sesimbra' AND Reserva.dataInicio IS NOT NULL AND
Aluguer.dataInicio IS NOT NULL
GROUP BY matricula, embarcacao.nome, tipo

```

	matricula	nome	tipo	numeroAlugueres	numeroReservas
1	2-VE-001	Vela2	Vela	35	26

Esta *query* apresenta-nos a matrícula da embarcação, o seu nome, o tipo e dá-nos quanto alugueres e reservas existem para esta, apenas retornando um (1) resultado. É feita a junção das tabelas **Embarcacao**, **Frota**, **Aluguer** e **Reserva** para ser possível mostrar dados relativos a tabelas diferentes; o filtro das datas é feito na cláusula ON.

Na cláusula *WHERE* temos o filtro do porto para apenas apanharmos resultados do porto de Sesimbra e também foi feito filtro para retirar todos os valores a NULL nas datas de início pois não podemos contar esses valores.

É feito agrupamento pela matrícula, nome e tipo.

**2.9 Para embarcações da categoria gaivotas ou a remos, obter uma lista contendo o nome da embarcação e número médio de horas em que esteve alugada, por dia. Devem ser excluídas as embarcações que tenham um tempo médio inferior a 2 horas. A lista produzida deve vir ordenada de forma decrescente pelo segundo atributo (número médio de horas).**

```

SELECT nome, ABS(AVG(DATEDIFF(HOUR, Aluguer.horaInicio,
Aluguer.horaFim))) AS numeroMedioHoras
FROM Embarcacao
LEFT JOIN Aluguer ON Embarcacao.matricula =
Aluguer.matriculaEmbarcacao
WHERE tipo IN('gaivota', 'remos')
GROUP BY nome
HAVING AVG(DATEDIFF(HOUR, Aluguer.horaInicio, Aluguer.horaFim)) >= 2
ORDER BY numeroMedioHoras DESC

```

	nome	numeroMedioHoras
1	Remos27	14
2	Gaivota18	12
3	Remos3	8
4	Remos4	8
5	Remos2	8
6	Remos20	6
7	Gaivota20	5
8	Gaivota27	5
9	Gaivota14	4
10	Remos14	4
11	Remos5	4
12	Remos1	3
13	Remos18	3
14	Gaivota1	3
15	Gaivota5	2

Esta *query* mostra-nos o nome, o valor absoluto da média da diferença de horas entre a horaInicio e a horaFim. São feitos os *joins* entre as tabelas **Embarcacao** e **Aluguer**, filtrando o tipo de barcos na cláusula *WHERE*; na cláusula *HAVING* é onde filtramos para excluir embarcações com tempo médio inferior a duas (2) horas e para a lista ser ordenado por ordem decrescente fazemo-lo na cláusula *ORDER BY*.

**2.10 Liste as atividades de 2016, cujo número de participantes é igual à lotação máxima da atividade, e onde estejam envolvidos mais do que um monitor.**

SELECT

```

    A.cod AS CodigoAtividade,
    A.nome AS NomeAtividade,
    A.data AS DataAtividade,
    A.descricao AS Descricao,
    A.lotacaoMaxima AS LotacaoMaxima,
    COUNT(DISTINCT AF.numCCFuncionario) AS NumeroDeMonitores,
    COUNT(DISTINCT CA.numDocCliente) AS NumeroDeParticipantes

```

FROM

```

    Atividade A

```

JOIN

```

    AtividadeFuncionarios AF ON A.cod = AF.codAtividade

```

JOIN

```

    Funcionario F ON AF.numCCFuncionario = F.numCC

```

JOIN

```

    ClienteAtividade CA ON A.cod = CA.codAatividade

```

WHERE

```

    F.funcao = 'Monitor'

```

```

AND YEAR(A.data) = 2016
GROUP BY
    A.cod, A.nome, A.data, A.descricao, A.lotacaoMaxima
HAVING
    COUNT(DISTINCT CA.numDocCliente) = A.lotacaoMaxima
    AND COUNT(DISTINCT AF.numCCFuncionario) > 1;

```

	CodigoAtividade	NomeAtividade	DataAtividade	Descricao	LotacaoMaxima	NumeroDeMonitores	NumeroDeParticipantes
1	11	Caça ao Tesouro Marítima	2016-12-07	Uma experiência divertida de caça ao tesouro em ...	25	2	25

Esta *query* lista informações sobre atividades realizadas em 2016 que atingiram a lotação máxima de participantes e contaram com mais de um monitor. Ela recupera o código, nome, data, descrição, lotação máxima, número de monitores e número de participantes para cada atividade. A partir das tabelas relacionadas, como **AtividadeFuncionarios** e **ClienteAtividade**, verifica-se o número de monitores associados (com função "Monitor") e o número de clientes participantes. O filtro no HAVING garante que apenas atividades com lotação completa e mais de um monitor sejam exibidas. Os resultados são agrupados por identificadores únicos das atividades.

**2.11 Liste os valores pagos pelos clientes para as reservas que já terminaram. Deve ser possível visualizar, para além do valor pago, quem foi o cliente, em que locadora fez a reserva e a data da reserva.**

```

SELECT
    r.cod AS Codigo_Reserva,
    c.nome AS Nome_Cliente,
    l.porto AS Porto_Locadora,
    r.dataInicio AS Data_Reserva,
    te.preco AS Valor_Embarcacao
FROM
    Reserva r
JOIN
    Cliente c ON r.numDocumentoCliente = c.numDocumento
JOIN
    Embarcacao e ON r.matriculaEmbarcacao = e.matricula
JOIN
    TipoEmbarcacao te ON e.tipo = te.nome
JOIN

```

```

Locadora 1 ON e.codFrota = 1.codFrota

WHERE

r.dataFim IS NOT NULL

ORDER BY

r.cod;

```

	Codigo_Reserva	Nome_Cliente	Porto_Locadora	Data_Reserva	Valor_Embarcacao
1	1	João Santos	Leixões	2016-05-24	150,00
2	2	Bartolomeu Ramos	Leixões	2016-06-01	150,00
3	3	Bernardo Campos	Leixões	2016-06-13	150,00
4	4	João Santos	Sesimbra	2016-06-01	120,00
5	5	Alexandre de Freitas	Sesimbra	2016-07-01	120,00
6	6	Mónica Rodrigues	Sesimbra	2016-07-02	120,00
7	7	Mariano Amaral	Sesimbra	2016-07-03	120,00
8	8	Vitória Rebelo	Sesimbra	2016-07-04	120,00
9	9	Iria Varejão	Sesimbra	2016-07-05	120,00
10	10	Eduarda Vila	Sesimbra	2016-07-06	120,00
11	11	Mónica Cruz	Sesimbra	2016-07-07	120,00
12	12	Lara Martins	Sesimbra	2016-07-08	120,00
13	13	Josué Silva	Sesimbra	2016-07-09	120,00
14	14	Frederica Fernand...	Sesimbra	2016-07-10	120,00
15	15	Ángelo Silva	Sesimbra	2016-07-11	120,00
16	16	Doroteia Freitas	Sesimbra	2016-07-12	120,00
17	17	Emiliana Couto	Sesimbra	2016-07-13	120,00
18	18	Lisandro do Rosário	Sesimbra	2016-07-14	120,00
19	19	Sol Medeiros	Sesimbra	2016-07-15	120,00
20	20	Vitória Rebelo	Sesimbra	2016-07-16	120,00
21	21	Alex Mendez	Sesimbra	2016-07-17	120,00
22	22	Mariano Amaral	Sesimbra	2016-07-18	120,00
23	23	Cristina Saraiva	Sesimbra	2016-07-19	120,00
24	24	Alexandre Varela	Sesimbra	2016-07-20	120,00
25	25	Cristina Saraiva	Sesimbra	2016-07-21	120,00
26	26	João Santos	Sesimbra	2016-07-22	120,00
27	27	Júlia Varela	Sesimbra	2016-07-23	120,00
28	28	Lara Martins	Sesimbra	2016-07-24	120,00
29	29	Sol Medeiros	Sesimbra	2016-07-25	120,00
30	30	Vitória Rebelo	Sesimbra	2016-07-26	120,00
31	31	João Santos	Figueira da Foz	2017-07-26	120,00

Esta *query* tem como objetivo fornecer detalhes sobre as reservas realizadas, incluindo o código da reserva, o nome do cliente, o porto da locadora, a data de início da reserva e o valor da embarcação associada. Para isso, a *query* realiza junções entre as tabelas **Reserva**, **Cliente**, **Embarcacao**, **TipoEmbarcacao** e **Locadora**, permitindo recuperar informações relevantes de diferentes tabelas. A junção entre as tabelas **Reserva** e **Cliente** é feita pelo número do documento do cliente, permitindo identificar quem fez a reserva. A junção com a tabela **Embarcacao** é feita pela matrícula da embarcação, para associar cada reserva a uma embarcação específica. Em seguida, a tabela **TipoEmbarcacao** é utilizada para recuperar o preço da embarcação com base no tipo de embarcação, e a tabela **Locadora** é usada para obter o porto da locadora associada à embarcação.

Na cláusula **WHERE**, a *query* faz um filtro para garantir que apenas reservas com a data de fim definida sejam consideradas, ou seja, excluindo reservas que ainda não terminaram. Finalmente, os resultados são ordenados pela coluna **cod** da tabela **Reserva**, permitindo que os dados sejam apresentados de forma sequencial de acordo com o código da reserva.

O resultado será uma lista de reservas concluídas, incluindo o cliente, o porto da locadora, a data da reserva e o valor da embarcação associada, organizados pelo código da reserva.

**2.12 Liste todas as atividades que cada cliente requereu, assim como o número de participantes nelas envolvidos.**

```
SELECT
    ca.numDocCliente AS Cliente_Número_Documento,
    c.nome AS Cliente_Nome,
    a.cod AS Atividade_Codigo,
    a.nome AS Atividade_Nome,
    participantes.Numero_De_Participantes
FROM
    Atividade a
JOIN
    ClienteAtividade ca ON a.cod = ca.codAtividade
JOIN
    Cliente c ON ca.numDocCliente = c.numDocumento -- Assumindo que a
tabela Cliente tenha 'numDocumento' e 'nome'
JOIN
    (SELECT codAtividade, COUNT(numDocCliente) AS
Numero_De_Participantes
    FROM ClienteAtividade
    GROUP BY codAtividade) AS participantes
ON a.cod = participantes.codAtividade
ORDER BY
    ca.numDocCliente, a.cod;
```



	Cliente_Numero_Documento	Cliente_Nome	Atividade_Codigo	Atividade_Nome	Numero_De_Participantes
1	004221182	Mariano Amaral	2	Passeio Turístico	20
2	004221182	Mariano Amaral	3	Mergulho Recreativo	5
3	004221182	Mariano Amaral	5	Corrida de Barcos	12
4	004221182	Mariano Amaral	7	Festa de Réveillon	25
5	004221182	Mariano Amaral	9	Treinamento de So...	8
6	004221182	Mariano Amaral	11	Caça ao Tesouro ...	25
7	004221182	Mariano Amaral	12	Workshop de Velas	7
8	004221182	Mariano Amaral	13	Passeio Gastronô...	16
9	004221182	Mariano Amaral	14	Navegação Noturna	7
10	004221182	Mariano Amaral	15	Expedição Marinha	7
11	004221182	Mariano Amaral	16	Curso de Primeiros...	10
12	004221182	Mariano Amaral	17	Canoagem Aventura	10
13	004221182	Mariano Amaral	20	Passeio Cultural M...	20
14	004221182	Mariano Amaral	22	Workshop de Man...	12
15	004221182	Mariano Amaral	25	Piquenique Náutico	15
16	004221182	Mariano Amaral	26	Passeio ao Pôr do ...	11
17	004221182	Mariano Amaral	27	Regata Amadora	12
18	004221182	Mariano Amaral	28	Exploração de Ilhas	15
19	004221182	Mariano Amaral	30	Atividade de Mergu...	11
20	004221182	Mariano Amaral	31	Canoagem Recrea...	4
21	004221182	Mariano Amaral	33	Caça Submarina	6
22	004221182	Mariano Amaral	34	Workshop de Fotog...	8
23	004221182	Mariano Amaral	36	Pesca Esportiva No...	8
24	004221182	Mariano Amaral	37	Cruzeiro Gastronô...	25
25	004221182	Mariano Amaral	44	Passeio Educativo	15
26	004221182	Mariano Amaral	45	Aventura Náutica	6
27	004221182	Mariano Amaral	47	Passeio com Obse...	18
28	099339688	Elder Oliveira	4	Aula de Navegação	9
29	099339688	Elder Oliveira	7	Festa de Réveillon	25
30	099339688	Elder Oliveira	8	Observação de Gol...	15
31	099339688	Elder Oliveira	11	Caça ao Tesouro ...	25
32	099339688	Elder Oliveira	37	Cruzeiro Gastronô...	25
33	099339688	Elder Oliveira	38	Trilha Marítima	6
34	099339688	Elder Oliveira	40	Passeio de Natal n...	10
35	099339688	Elder Oliveira	41	Curso de Navegaç...	4
36	099339688	Elder Oliveira	46	Curso de Salvagem	7

Esta *query* tem como objetivo recuperar informações sobre as atividades realizadas pelos clientes, incluindo o número de participantes em cada atividade. Para isso, a *query* faz junções entre as tabelas **Atividade**, **ClienteAtividade** e **Cliente**, permitindo acessar os dados relacionados a cada cliente e a sua participação nas atividades.

Primeiramente, a tabela **Atividade** é unida à tabela **ClienteAtividade** por meio do código da atividade, o que possibilita saber quais clientes participaram em cada atividade. Em seguida, a tabela **Cliente** é vinculada à **ClienteAtividade** através do número de documento do cliente (campo **numDocCliente**), permitindo recuperar o nome do cliente associado a cada participação.

Além disso, a *query* utiliza uma subconsulta, que agrupa os dados da tabela **ClienteAtividade** pelo código da atividade (**codAtividade**) e calcula o número de participantes em cada atividade. Essa subconsulta é renomeada como **participantes** e é unida à tabela **Atividade** com base no código da atividade, permitindo obter o número total de participantes para cada atividade.

Por fim, a consulta é ordenada pelo número de documento do cliente (**ca.numDocCliente**) e pelo código da atividade (**a.cod**), de modo que os dados sejam apresentados de forma organizada por cliente e, em seguida, por atividade.

O resultado final fornecerá uma lista de clientes com os detalhes das atividades nas quais participaram, incluindo o número de participantes em cada atividade, organizados pelo número de documento do cliente e o código da atividade.

### Consultas – Parte III

### – Funcionários que trabalharam em Atividades no 3º trimestre de 2016

```

SELECT
    f.nome AS funcionario_nome,
    f.funcao,
    a.nome AS atividade_nome,
    a.descricao
FROM
    Funcionario f
JOIN
    AtividadeFuncionarios af ON f.numCC = af.numCCFuncionario
JOIN
    Atividade a ON af.codAtividade = a.cod
WHERE
    a.data BETWEEN '2016-07-01' AND '2016-09-30'
    AND EXISTS (SELECT 1
                FROM Embarcacao e
                JOIN AtividadeEmbarcacao ae ON e.matricula =
ae.matriculaEmbarcacao
                WHERE ae.codAtividade = a.cod)
ORDER BY
    f.nome, a.nome;

```

	funcionario_nome	funcao	atividade_nome	descricao
1	Bruno Costa	Trabalhador de doca	Comida de Jet Ski	Competição em jet ski para participantes habilita...
2	Carla Silva	Monitor	Aventura Náutica	Exploração com desafios no mar
3	Cláudia Reis	Monitor	Comida de Barcos à Vela	Competição recreativa de barcos à vela
4	Fernando Borges	Trabalhador de doca	Comida de Barcos à Vela	Competição recreativa de barcos à vela
5	Fernando Ribeiro	Trabalhador de doca	Aventura Náutica	Exploração com desafios no mar
6	Joana Nogueira	Monitor	Curso de Salvamento Aquático	Treinamento para resgate em ambientes aquáticos
7	Joana Oliveira	Monitor	Passeio Educativo	Exploração marítima com foco educacional
8	Manuel Augusto	Trabalhador de doca	Passeio Educativo	Exploração marítima com foco educacional
9	Rafael Barros	Trabalhador de doca	Curso de Salvamento Aquático	Treinamento para resgate em ambientes aquáticos
10	Ricardo Teixeira	Trabalhador de doca	Sessão de Pesca com Guia	Pesca acompanhada por especialistas
11	Sofia Rodrigues	Monitor	Sessão de Pesca com Guia	Pesca acompanhada por especialistas
12	Vanessa Ferreira	Monitor	Comida de Jet Ski	Competição em jet ski para participantes habilita...

Esta *query* lista os funcionários que participaram de atividades no **3º trimestre de 2016** (entre **1º de julho e 30 de setembro de 2016**). Ela combina as tabelas

**Funcionario, AtividadeFuncionarios, e Atividade** para identificar as atividades realizadas nesse período e os funcionários que estiveram envolvidos. O resultado exibe o nome e a função do funcionário, assim como o nome e a descrição da atividade, sendo os dados ordenados pelo nome do funcionário e pela atividade.

– **Cientes que fizeram mais reservas e alugueres que a média de alugueres e reservas por cliente.**

```
SELECT DISTINCT nome, numDocumento, (COALESCE(COUNT(DISTINCT
Aluguer.cod),0) + COALESCE(COUNT(DISTINCT Reserva.cod),0)) AS totalidade
FROM Cliente
LEFT JOIN Aluguer ON Cliente.numDocumento = Aluguer.numDocumentoCliente
LEFT JOIN Reserva ON Cliente.numDocumento = Reserva.numDocumentoCliente
GROUP BY nome, numDocumento
HAVING (
    COALESCE(COUNT(DISTINCT Aluguer.cod),0) +
    COALESCE(COUNT(DISTINCT Reserva.cod),0)) > (
    SELECT DISTINCT (COALESCE(subQuery.avgAluguer,0) +
    COALESCE(subQuery.avgReserva,0)) AS Total
    FROM (
        SELECT AVG(countAluguer) AS avgAluguer, AVG(countReserva) AS
        avgReserva
        FROM (
            SELECT DISTINCT Cliente.nome, Cliente.numDocumento,
            COUNT(DISTINCT Aluguer.cod) AS countAluguer,
            COUNT(DISTINCT Reserva.cod) AS countReserva
            FROM Cliente
            LEFT JOIN Aluguer ON Cliente.numDocumento =
            Aluguer.numDocumentoCliente
            LEFT JOIN Reserva ON Cliente.numDocumento =
            Reserva.numDocumentoCliente
            GROUP BY Cliente.nome, Cliente.numDocumento
        ) clientesQuery
    ) subQuery
)
ORDER BY totalidade DESC
```

	nome	numDocumento	totalidade
1	Bernardo Campos	186067311	12
2	Frederica Fernandes	680244098	10
3	João Santos	123456789	8
4	Vitória Rebelo	133313027	6
5	Mónica Rodrigues	301286967	5
6	Alexandre Varela	318272510	4
7	Cristina Saraiva	350225540	4
8	Lara Martins	378063581	4
9	Sol Medeiros	698925581	4
10	Mariano Amaral	004221182	4

Esta *query* inicia-se na *subquery* “clientesQuery” onde obtemos o nome, o número de documento e a quantidade de alugueres e quantidade de reservas que cada cliente tem; a *subquery* externa a esta, “subQuery” recebe os totais e calcula a média através da função *AVG* de cada um destes; a última *subquery* soma os totais que recebeu obtendo-se assim a média dos alugueres e reservas por cliente; por fim, no último *SELECT* mostra-se o nome, numDocumento e o total (soma) de reservas e alugueres

por cliente e através da cláusula *HAVING* filtramos para apenas nos mostre os clientes com um total de reservas e alugueres superior à média. Os LEFT JOIN são sempre efetuados para conseguirmos obter dados de diferentes tabelas, nomeadamente **Cliente**, **Reserva** e **Aluguer**.

– Clientes que participaram de mais atividades e o total gasto

```
SELECT c.nome AS Nome_Cliente, Total_Atividades, Gasto_Total
FROM Cliente c
JOIN (
    SELECT ca.numDocCliente, COUNT(*) AS Total_Atividades, SUM(a.custoPessoa)
    AS Gasto_Total
    FROM ClienteAtividade ca
    JOIN Atividade a ON ca.codAtividade = a.cod
    GROUP BY ca.numDocCliente
) SubAtividades ON c.numDocumento = SubAtividades.numDocCliente
WHERE SubAtividades.Total_Atividades > 2
ORDER BY Gasto_Total DESC;
```

	Nome_Cliente	Total_Atividades	Gasto_Total
1	Celso Freire	33	2515.00
2	Alex Mendez	29	2130.00
3	Júlia Varela	28	2120.00
4	Mariano Amaral	27	2070.00
5	Alexandre Varela	24	1815.00
6	Gilda Rocha	24	1785.00
7	Josué Silva	22	1700.00
8	Olga Alves	22	1630.00
9	Lara Martins	22	1620.00
10	Lisandro do Rosário	21	1615.00
11	Vitória Rebelo	21	1550.00
12	Iria Varejão	20	1510.00
13	Bernardino Venâncio	20	1485.00
14	Herminia Guerra	17	1265.00
15	Frederica Fernandes	17	1240.00
16	Sol Medeiros	16	1130.00
17	Alexandre de Freitas	15	1100.00
18	Bernardo Campos	14	1000.00
19	Bartolomeu Ramos	13	960.00
20	João Santos	13	880.00
21	Cecília Barros	10	870.00
22	Ângelo Silva	9	820.00
23	Elder Oliveira	9	820.00
24	Inácio Azevedo	8	780.00
25	Doroteia Freitas	8	730.00

Esta *query* tem como objetivo recuperar informações sobre os clientes que participaram de atividades, incluindo o total de atividades realizadas e o gasto total em cada uma delas. A *query* faz uso de uma subconsulta para calcular esses valores agregados e, em seguida, filtra e ordena os resultados de acordo com o gasto total.

Primeiramente, a subconsulta dentro da cláusula **FROM** é responsável por calcular, para cada cliente, o total de atividades que ele participou e o valor total gasto nessas atividades. A subconsulta faz uma junção entre a tabela **ClienteAtividade** (que relaciona os clientes com as atividades) e a tabela **Atividade** (que contém o custo por pessoa de cada atividade). A subconsulta agrupa os dados por **numDocCliente** (número do documento do cliente) e calcula, usando a função **COUNT** o número total

de atividades em que o cliente participou e a função **SUM** para somar o custo de cada atividade multiplicado pelo número de clientes. O resultado dessa subconsulta é uma tabela com o número do cliente, o total de atividades e o gasto total para cada cliente.

Após a execução da subconsulta, a consulta principal faz uma junção entre a tabela **Cliente** e a subconsulta, utilizando o **numDocumento** de cada cliente para conectar os dados. Dessa forma, podemos acessar as informações do cliente, como o nome, e também incluir os valores calculados pela subconsulta.

Na cláusula **WHERE**, há um filtro para selecionar apenas os clientes que participaram de mais de 2 atividades, ou seja, só serão considerados os clientes cujo **Total\_Atividades** seja superior a 2. Por fim, a *query* ordena os resultados pelo **Gasto\_Total** de forma decrescente (**DESC**), garantindo que os clientes com maior gasto sejam listados primeiro.

O resultado é uma lista de clientes que participaram de mais de 2 atividades, incluindo o nome do cliente, o total de atividades realizadas e o valor total gasto em todas as atividades, ordenados do maior para o menor gasto.

## 4 Conclusão

Com este trabalho foi possível aprofundar os nossos conhecimentos de linguagem SQL, perceber a importância de uma boa análise e planejamento ao criar uma base de dados e que, estrategicamente, uma boa implementação inicial poderá evitar vários problemas posteriores dos dados. Foram perceptíveis algumas dificuldades em criar um modelo ER que correspondesse a todos os critérios pedidos.