

INSTITUTO INFNET
Análise e Desenvolvimento de Sistemas

GEOVANI LUIZ DE MOURA

**Projeto de Bloco de Fundamento de Dados
- TP2 -**

Cotia
2024

INTRODUÇÃO

Este trabalho, TP2 - Projeto de Bloco de Fundamento de Dados, tem como objetivo criar uma base de dados com três tabelas: Funcionários, Cargos e Departamentos. As tabelas serão relacionadas e conterão informações como nome, cargo, salário e departamento dos funcionários. Além dos campos obrigatórios, será adicionado um campo extra para cada tabela.

As tabelas foram criadas e relacionadas no SQL Server, usando o DBeaver como ferramenta de execução e consulta. Também foram inseridos registros suficientes para realizar consultas que mostram dados como a descrição do cargo e o nome do departamento, reforçando o uso de chaves estrangeiras.

GITHUB:

https://github.com/geovani-moura/INFNET_TPS_ATS

Navegue até a pasta “Projeto de Bloco Fundamentos de Dados [24E3_5]”

Depois navegue até a pasta “TP2”

Sumário

Sumário	5
1. Listar individualmente a tabelas.....	13
2. Listar todos os departamentos que ficam no quinto andar	15
3. Listar o analista com o salário mais alto.....	16
4. Listar os funcionários com salário maior ou igual ao salário base no departamento de TI.....	17
5. Listar o departamento com o maior número de estagiários.....	18
6. Listar todos os funcionários sem cargo associado	19
7. Listar os funcionários do andar mais alto.....	20
8. Listar cargos com funcionários que ganham entre 3000 e 5000.....	21
9. Listar gerentes que chefiam pelo menos 2 departamentos.....	22
10. Listar o cargo com o salário mais baixo	23
11. Listar o departamento com o salário mais alto	24
12. Listar o andar onde ficam os diretores.....	25
13. Listar funcionários em ordem alfabética, com lógica própria	26
14. Listar cargos em ordem alfabética, com lógica própria.....	27
15. Listar departamentos em ordem alfabética, com lógica própria	28
Considerações Finais	29

1. Listar individualmente a tabelas

Objetivo:

Exibir as tabelas TB_FUNCIONARIO, TB_CARGO e TB_DEPARTAMENTO com os registros organizados alfabeticamente.

Querys:

```
-- Funcionários em ordem alfabética
SELECT * FROM TB_FUNCIONARIO ORDER BY Nome;
```

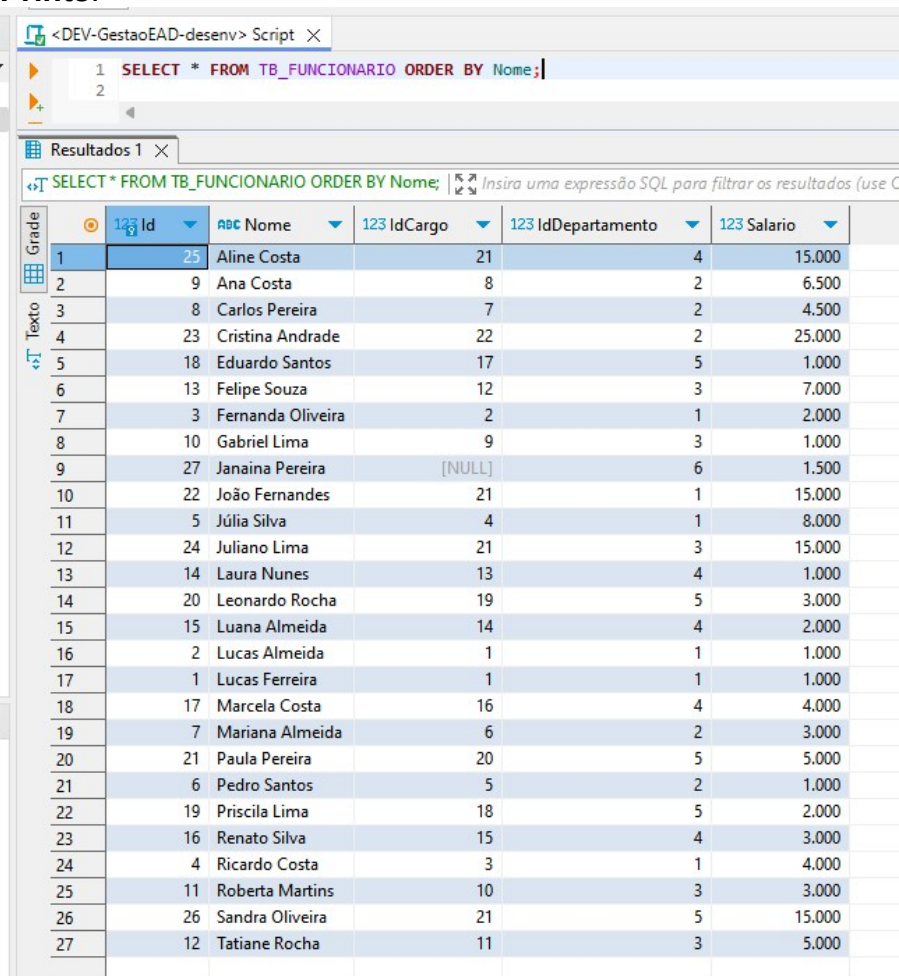
```
-- Cargos em ordem alfabética
SELECT * FROM TB_CARGO ORDER BY Descricao;
```

```
-- Departamentos em ordem alfabética
SELECT * FROM TB_DEPARTAMENTO ORDER BY Nome;
```

Explicação:

Essas consultas organizam os registros de cada tabela em ordem alfabética para facilitar a leitura e análise.

Prints:



The screenshot shows a SQL script editor with the query: `SELECT * FROM TB_FUNCIONARIO ORDER BY Nome;`. Below the script, a result grid titled 'Resultados 1' displays the data. The grid has columns for Grade, Id, Nome, IdCargo, IdDepartamento, and Salario. The data is sorted alphabetically by employee name.

Grade	Id	Nome	IdCargo	IdDepartamento	Salario
1	25	Aline Costa	21	4	15.000
2	9	Ana Costa	8	2	6.500
3	8	Carlos Pereira	7	2	4.500
4	23	Cristina Andrade	22	2	25.000
5	18	Eduardo Santos	17	5	1.000
6	13	Felipe Souza	12	3	7.000
7	3	Fernanda Oliveira	2	1	2.000
8	10	Gabriel Lima	9	3	1.000
9	27	Janaina Pereira	[NULL]	6	1.500
10	22	João Fernandes	21	1	15.000
11	5	Júlia Silva	4	1	8.000
12	24	Juliano Lima	21	3	15.000
13	14	Laura Nunes	13	4	1.000
14	20	Leonardo Rocha	19	5	3.000
15	15	Luana Almeida	14	4	2.000
16	2	Lucas Almeida	1	1	1.000
17	1	Lucas Ferreira	1	1	1.000
18	17	Marcela Costa	16	4	4.000
19	7	Mariana Almeida	6	2	3.000
20	21	Paula Pereira	20	5	5.000
21	6	Pedro Santos	5	2	1.000
22	19	Priscila Lima	18	5	2.000
23	16	Renato Silva	15	4	3.000
24	4	Ricardo Costa	3	1	4.000
25	11	Roberta Martins	10	3	3.000
26	26	Sandra Oliveira	21	5	15.000
27	12	Tatiane Rocha	11	3	5.000

<DEV-GestaoEAD-desenv> Script X

```
1 SELECT * FROM TB_CARGO ORDER BY Descricao;
```

Resultados 1 X

SELECT * FROM TB_CARGO ORDER BY Descricao; Insira uma expressão SQL para filtra

Grade	123 Id	123 IdNivel	ABC Descricao	123 SalarioBase
1	13	1	Analista de Marketing	1.000
2	14	2	Analista de Marketing	2.000
3	15	3	Analista de Marketing	3.000
4	16	4	Analista de Marketing	4.000
5	5	1	Analista de RH	1.000
6	6	2	Analista de RH	3.000
7	7	3	Analista de RH	4.500
8	8	4	Analista de RH	6.500
9	1	1	Analista de Sistemas	1.000
10	2	2	Analista de Sistemas	2.000
11	3	3	Analista de Sistemas	4.000
12	4	4	Analista de Sistemas	8.000
13	9	1	Contador	1.000
14	10	2	Contador	3.000
15	11	3	Contador	5.000
16	12	4	Contador	7.000
17	22	6	Diretor	25.000
18	21	5	Gerente	15.000
19	17	1	Vendedor	1.000
20	18	2	Vendedor	2.000
21	19	3	Vendedor	3.000
22	20	4	Vendedor	5.000

<DEV-GestaoEAD-desenv> Script X

```
1 SELECT * FROM TB_DEPARTAMENTO ORDER BY Nome;
```

Resultados 1 X

Insira uma expressão SQL

Grade	123 Id	123 IdFuncionario	ABC Nome	123 Andar
1	3	24	Financeiro	3
2	6	[NULL]	Limpeza	6
3	4	25	Marketing	4
4	2	22	RH	2
5	1	22	TI	1
6	5	26	Vendas	5

2. Listar todos os departamentos que ficam no quinto andar

Objetivo:

Listar todos os departamentos localizados no quinto andar.

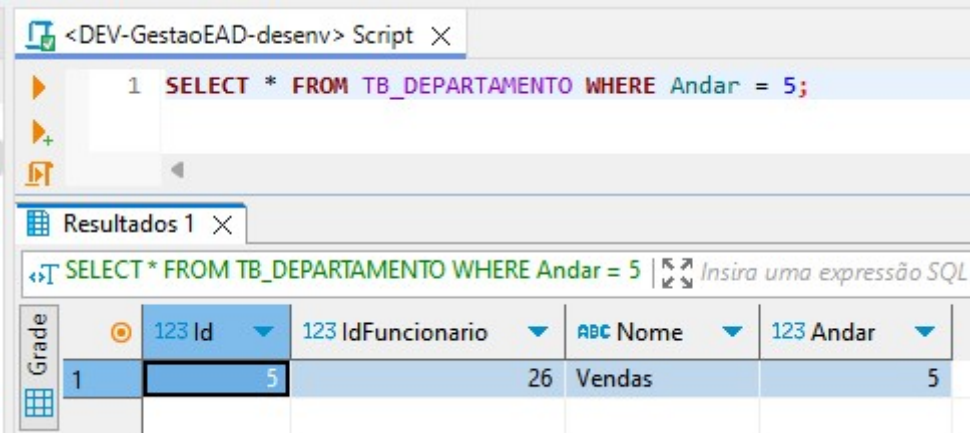
Querys:

```
SELECT * FROM TB_DEPARTAMENTO WHERE Andar = 5;
```

Explicação:

A query filtra os departamentos com o campo Andar igual a 5.

Prints:



The screenshot shows a SQL IDE interface. At the top, a script editor window titled '<DEV-GestaoEAD-desenv> Script' contains the query: `1 SELECT * FROM TB_DEPARTAMENTO WHERE Andar = 5;`. Below the script editor, a results window titled 'Resultados 1' displays the query results in a table. The table has five columns: 'Grade', '123 Id', '123 IdFuncionario', 'ABC Nome', and '123 Andar'. The first row of data shows the values: '1', '5', '26', 'Vendas', and '5'.

Grade	123 Id	123 IdFuncionario	ABC Nome	123 Andar
1	5	26	Vendas	5

3. Listar o analista com o salário mais alto

Objetivo:

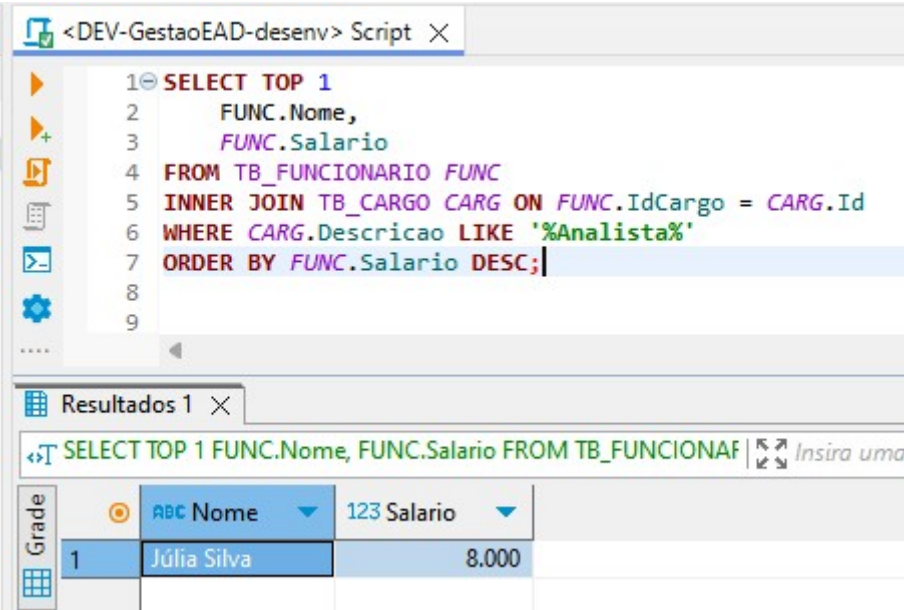
Encontrar o analista com o salário mais alto na organização.

Querys:

```
SELECT TOP 1
    FUNC.Nome,
    FUNC.Salario
FROM TB_FUNCIONARIO FUNC
INNER JOIN TB_CARGO CARG ON FUNC.IdCargo = CARG.Id
WHERE CARG.Descricao LIKE '%Analista%'
ORDER BY FUNC.Salario DESC;
```

Explicação:

A consulta faz uma junção entre TB_FUNCIONARIO e TB_CARGO, filtra os cargos que contêm "Analista" na descrição e exibe o funcionário com o maior salário.

Prints:

The screenshot displays a SQL query in a script editor and its results in a results window.

Script Editor:

```
1 SELECT TOP 1
2     FUNC.Nome,
3     FUNC.Salario
4 FROM TB_FUNCIONARIO FUNC
5 INNER JOIN TB_CARGO CARG ON FUNC.IdCargo = CARG.Id
6 WHERE CARG.Descricao LIKE '%Analista%'
7 ORDER BY FUNC.Salario DESC;
```

Results Window:

Resultados 1

SELECT TOP 1 FUNC.Nome, FUNC.Salario FROM TB_FUNCIONAF

Grade	Nome	Salario
1	Júlia Silva	8.000

4. Listar os funcionários com salário maior ou igual ao salário base no departamento de TI

Objetivo:

Exibir funcionários do departamento de TI com salário superior ou igual ao salário base do cargo.

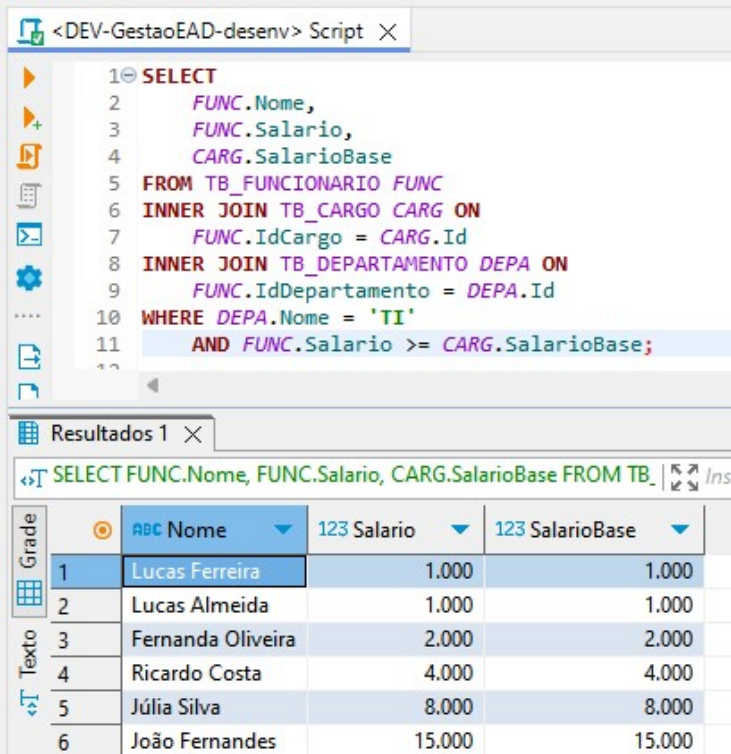
Querys:

```
SELECT
    FUNC.Nome,
    FUNC.Salario,
    CARG.SalarioBase
FROM TB_FUNCIONARIO FUNC
INNER JOIN TB_CARGO CARG ON
    FUNC.IdCargo = CARG.Id
INNER JOIN TB_DEPARTAMENTO DEPA ON
    FUNC.IdDepartamento = DEPA.Id
WHERE DEPA.Nome = 'TI'
    AND FUNC.Salario >= CARG.SalarioBase;
```

Explicação:

A query realiza a junção entre as tabelas de funcionários, cargos e departamentos, filtrando por departamento "TI" e comparando o salário dos funcionários com o salário base dos cargos.

Prints:



The screenshot shows a SQL IDE window titled "<DEV-GestaoEAD-desenv> Script". The query editor contains the following SQL code:

```
1 SELECT
2     FUNC.Nome,
3     FUNC.Salario,
4     CARG.SalarioBase
5 FROM TB_FUNCIONARIO FUNC
6 INNER JOIN TB_CARGO CARG ON
7     FUNC.IdCargo = CARG.Id
8 INNER JOIN TB_DEPARTAMENTO DEPA ON
9     FUNC.IdDepartamento = DEPA.Id
10 WHERE DEPA.Nome = 'TI'
11     AND FUNC.Salario >= CARG.SalarioBase;
```

Below the query editor, the "Resultados 1" tab is active, displaying the query results in a table. The table has 4 columns: "Grade", "ABC Nome", "123 Salario", and "123 SalarioBase". The results are as follows:

Grade	ABC Nome	123 Salario	123 SalarioBase
1	Lucas Ferreira	1.000	1.000
2	Lucas Almeida	1.000	1.000
3	Fernanda Oliveira	2.000	2.000
4	Ricardo Costa	4.000	4.000
5	Júlia Silva	8.000	8.000
6	João Fernandes	15.000	15.000

5. Listar o departamento com o maior número de estagiários

Objetivo:

Identificar o departamento que possui mais estagiários.

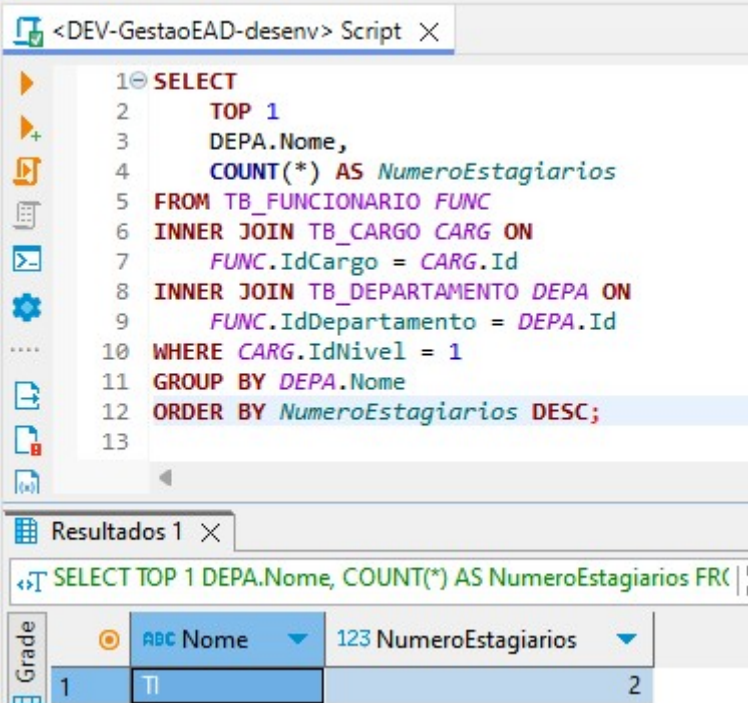
Querys:

```
SELECT
    TOP 1
    DEPA.Nome,
    COUNT(*) AS NumeroEstagiarios
FROM TB_FUNCIONARIO FUNC
INNER JOIN TB_CARGO CARG ON
    FUNC.IdCargo = CARG.Id
INNER JOIN TB_DEPARTAMENTO DEPA ON
    FUNC.IdDepartamento = DEPA.Id
WHERE CARG.IdNivel = 1
GROUP BY DEPA.Nome
ORDER BY NumeroEstagiarios DESC;
```

Explicação:

A query conta o número de estagiários por departamento e exibe o departamento com a maior quantidade.

Prints:



The screenshot shows a SQL script editor window titled "<DEV-GestaoEAD-desenv> Script". The script contains the following SQL query:

```
1 SELECT
2     TOP 1
3     DEPA.Nome,
4     COUNT(*) AS NumeroEstagiarios
5 FROM TB_FUNCIONARIO FUNC
6 INNER JOIN TB_CARGO CARG ON
7     FUNC.IdCargo = CARG.Id
8 INNER JOIN TB_DEPARTAMENTO DEPA ON
9     FUNC.IdDepartamento = DEPA.Id
10 WHERE CARG.IdNivel = 1
11 GROUP BY DEPA.Nome
12 ORDER BY NumeroEstagiarios DESC;
```

Below the script editor, the "Resultados 1" window displays the query results. The query is partially visible as "SELECT TOP 1 DEPA.Nome, COUNT(*) AS NumeroEstagiarios FR". The results table has two columns: "Nome" and "NumeroEstagiarios". The first row shows the department "TI" with 123 estagiários.

Grade	Nome	NumeroEstagiarios
1	TI	123

6. Listar todos os funcionários sem cargo associado

Objetivo:

Exibir funcionários que não possuem um cargo definido.

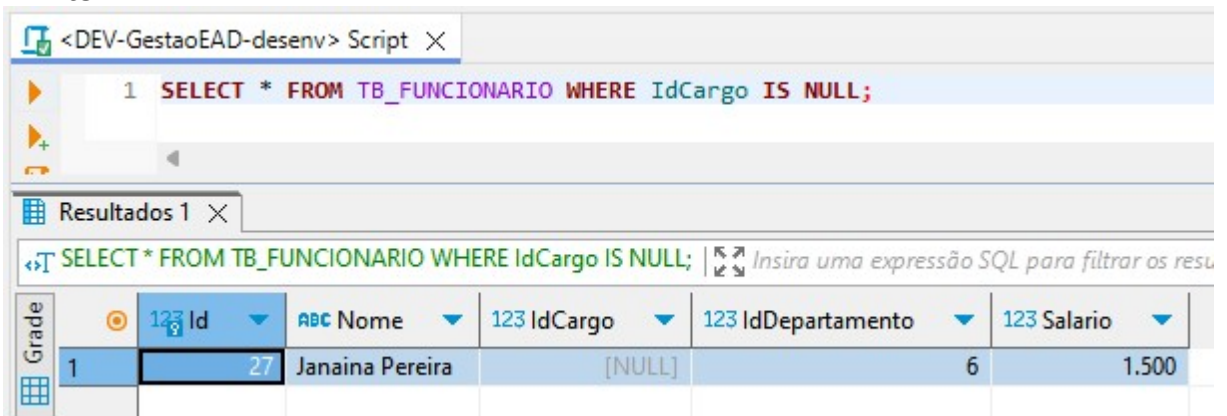
Querys:

```
SELECT * FROM TB_FUNCIONARIO WHERE IdCargo IS NULL;
```

Explicação:

A consulta seleciona todos os registros na tabela de funcionários em que o campo IdCargo é nulo.

Prints:



The screenshot shows a database management interface. At the top, a script editor window titled '<DEV-GestaoEAD-desenv> Script' contains the query: `1 SELECT * FROM TB_FUNCIONARIO WHERE IdCargo IS NULL;`. Below the script editor, a results window titled 'Resultados 1' displays the query results in a table. The table has columns: 'Id', 'Nome', 'IdCargo', 'IdDepartamento', and 'Salario'. The first row shows a record with Id 27, Nome 'Janaina Pereira', IdCargo [NULL], IdDepartamento 6, and Salario 1.500.

Grade	Id	Nome	IdCargo	IdDepartamento	Salario
1	27	Janaina Pereira	[NULL]	6	1.500

7. Listar os funcionários do andar mais alto

Objetivo:

Exibir os funcionários que trabalham no andar mais alto.

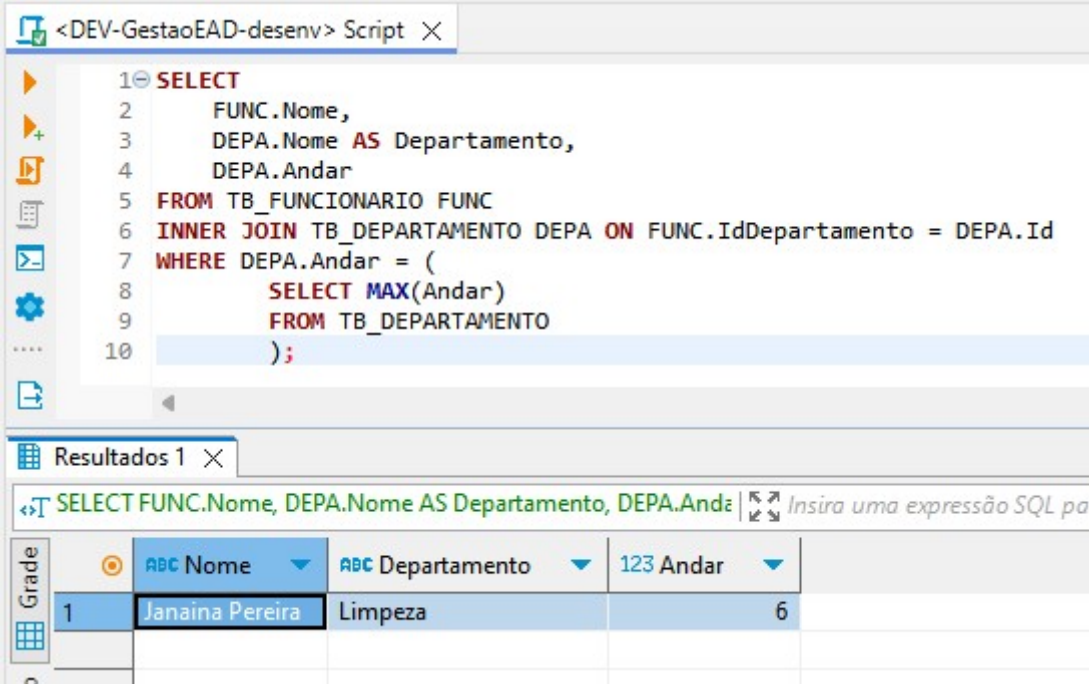
Querys:

```
SELECT
    FUNC.Nome,
    DEPA.Nome AS Departamento,
    DEPA.Andar
FROM TB_FUNCIONARIO FUNC
INNER JOIN TB_DEPARTAMENTO DEPA ON FUNC.IdDepartamento = DEPA.Id
WHERE DEPA.Andar = (
    SELECT MAX(Andar)
    FROM TB_DEPARTAMENTO
);
```

Explicação:

A consulta seleciona os funcionários que estão no andar com o maior valor de Andar.

Prints:



The screenshot shows a SQL IDE interface. The top pane displays a script titled "<DEV-GestaoEAD-desenv> Script" containing the following SQL query:

```
1 SELECT
2     FUNC.Nome,
3     DEPA.Nome AS Departamento,
4     DEPA.Andar
5 FROM TB_FUNCIONARIO FUNC
6 INNER JOIN TB_DEPARTAMENTO DEPA ON FUNC.IdDepartamento = DEPA.Id
7 WHERE DEPA.Andar = (
8     SELECT MAX(Andar)
9     FROM TB_DEPARTAMENTO
10 );
```

The bottom pane, titled "Resultados 1", shows the query results in a table. The table has four columns: "Grade", "Nome", "Departamento", and "Andar". The first row of data shows a functionary named "Janaina Pereira" working in the "Limpeza" department on floor "6".

Grade	Nome	Departamento	Andar
1	Janaina Pereira	Limpeza	6

8. Listar cargos com funcionários que ganham entre 3000 e 5000

Objetivo:

Listar cargos com funcionários cujo salário está entre 3000 e 5000.

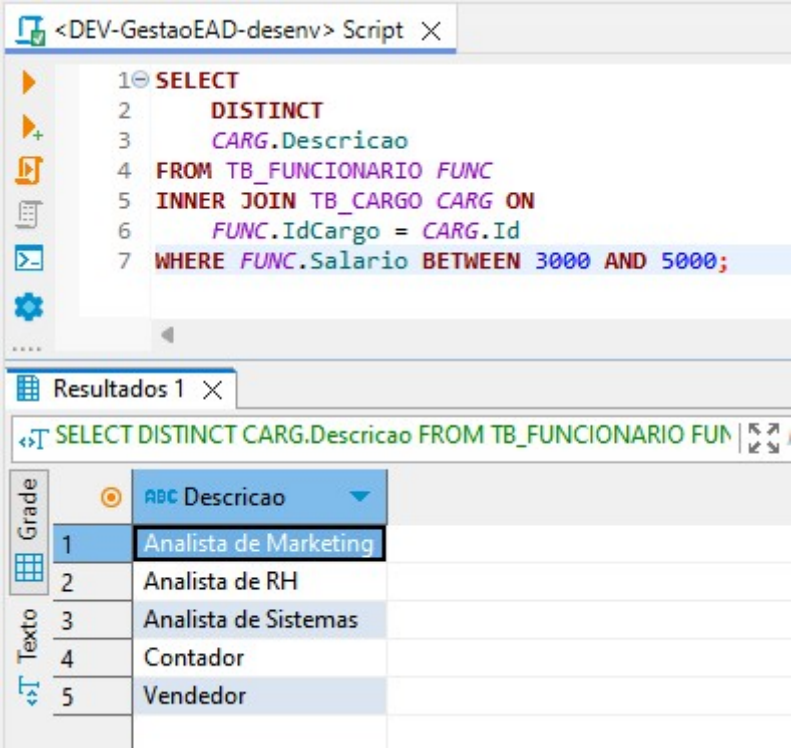
Querys:

```
SELECT
    DISTINCT
    CARG.Descricao
FROM TB_FUNCIONARIO FUNC
INNER JOIN TB_CARGO CARG ON
    FUNC.IdCargo = CARG.Id
WHERE FUNC.Salario BETWEEN 3000 AND 5000;
```

Explicação:

A consulta identifica os cargos com funcionários que têm salários entre os valores especificados.

Prints:



The screenshot shows a database management tool interface. The top pane displays a SQL query script for a project named '<DEV-GestaoEAD-desenv>'. The query is as follows:

```
1 SELECT
2     DISTINCT
3     CARG.Descricao
4 FROM TB_FUNCIONARIO FUNC
5 INNER JOIN TB_CARGO CARG ON
6     FUNC.IdCargo = CARG.Id
7 WHERE FUNC.Salario BETWEEN 3000 AND 5000;
```

The bottom pane, titled 'Resultados 1', shows the results of the query. It displays a table with 5 rows and 1 column, 'Descricao'. The results are:

Grade	Descricao
1	Analista de Marketing
2	Analista de RH
3	Analista de Sistemas
4	Contador
5	Vendedor

9. Listar gerentes que chefiam pelo menos 2 departamentos

Objetivo:

Encontrar gerentes que chefiam dois ou mais departamentos.

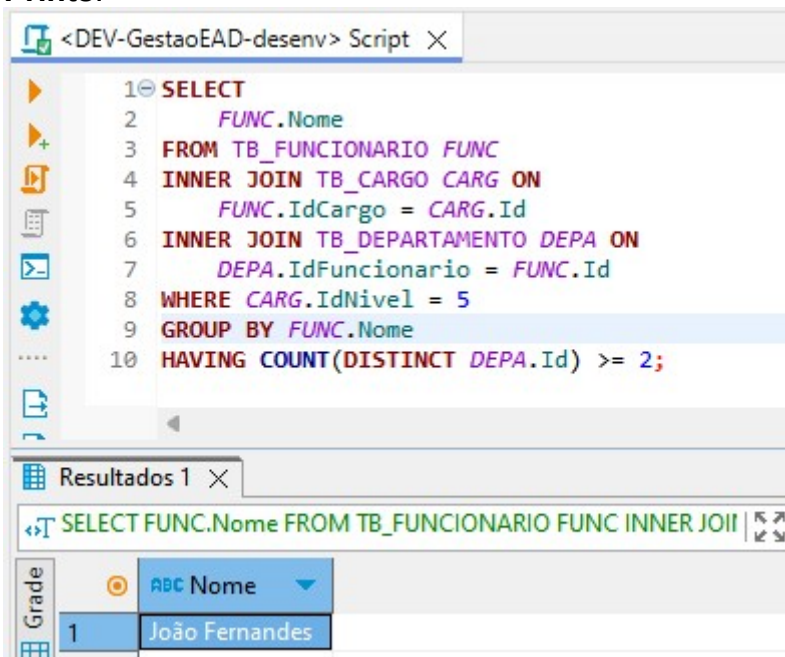
Querys:

```
SELECT
    FUNC.Nome
FROM TB_FUNCIONARIO FUNC
INNER JOIN TB_CARGO CARG ON
    FUNC.IdCargo = CARG.Id
INNER JOIN TB_DEPARTAMENTO DEPA ON
    DEPA.IdFuncionario = FUNC.Id
WHERE CARG.IdNivel = 5
GROUP BY FUNC.Nome
HAVING COUNT(DISTINCT DEPA.Id) >= 2;
```

Explicação:

A consulta filtra os gerentes (nível 5) que lideram dois ou mais departamentos.

Prints:



The screenshot shows a SQL IDE interface. The top pane, titled "<DEV-GestaoEAD-desenv> Script", contains the following SQL query:

```
1 SELECT
2     FUNC.Nome
3 FROM TB_FUNCIONARIO FUNC
4 INNER JOIN TB_CARGO CARG ON
5     FUNC.IdCargo = CARG.Id
6 INNER JOIN TB_DEPARTAMENTO DEPA ON
7     DEPA.IdFuncionario = FUNC.Id
8 WHERE CARG.IdNivel = 5
9 GROUP BY FUNC.Nome
10 HAVING COUNT(DISTINCT DEPA.Id) >= 2;
```

The bottom pane, titled "Resultados 1", shows the results of the query. It includes a toolbar with a search icon, a dropdown menu set to "ABC Nome", and a table with one row:

Grade	Nome
1	João Fernandes

10. Listar o cargo com o salário mais baixo

Objetivo:

Identificar o cargo com o menor salário base.

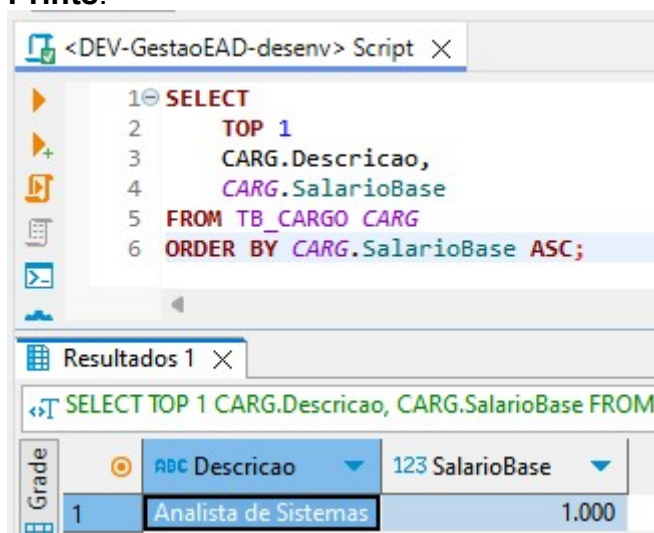
Querys:

```
SELECT
    TOP 1
    CARG.Descricao,
    CARG.SalarioBase
FROM TB_CARGO CARG
ORDER BY CARG.SalarioBase ASC;
```

Explicação:

A consulta seleciona o cargo com o menor valor no campo SalarioBase.

Prints:



The screenshot displays a SQL query in a script editor and its results in a results window. The query is as follows:

```
SELECT
    TOP 1
    CARG.Descricao,
    CARG.SalarioBase
FROM TB_CARGO CARG
ORDER BY CARG.SalarioBase ASC;
```

The results window shows the query executed and the results table with columns Grade, Descricao, and SalarioBase. The result shows Grade 1, Descricao 'Analista de Sistemas', and SalarioBase 1.000.

Grade	Descricao	SalarioBase
1	Analista de Sistemas	1.000

11. Listar o departamento com o salário mais alto

Objetivo:

Exibir o departamento onde o salário mais alto está localizado.

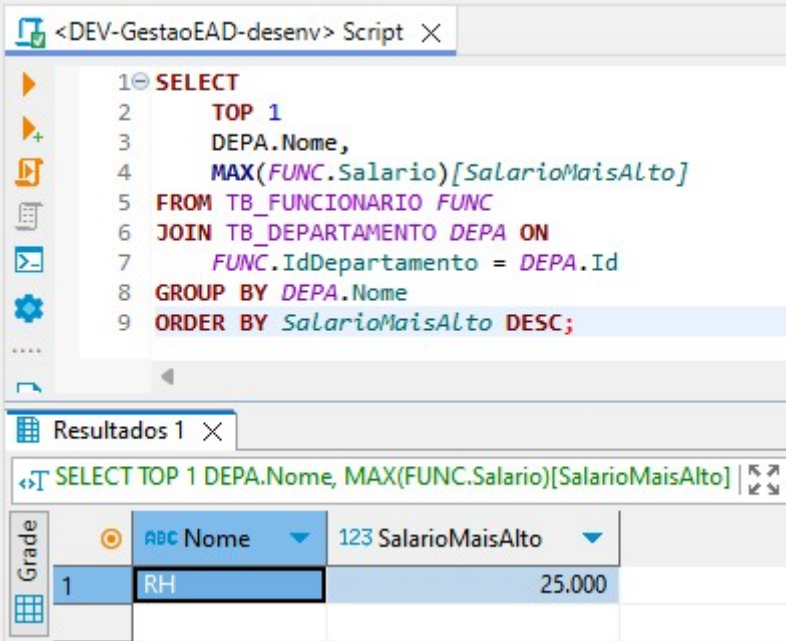
Querys:

```
SELECT
    TOP 1
    DEPA.Nome,
    MAX(FUNC.Salario)[SalarioMaisAlto]
FROM TB_FUNCIONARIO FUNC
JOIN TB_DEPARTAMENTO DEPA ON
    FUNC.IdDepartamento = DEPA.Id
GROUP BY DEPA.Nome
ORDER BY SalarioMaisAlto DESC;
```

Explicação:

A consulta exibe o departamento com o maior salário entre os funcionários.

Prints:



The screenshot shows a SQL development environment with a script editor and a results pane. The script editor contains the following query:

```
1 SELECT
2     TOP 1
3     DEPA.Nome,
4     MAX(FUNC.Salario)[SalarioMaisAlto]
5 FROM TB_FUNCIONARIO FUNC
6 JOIN TB_DEPARTAMENTO DEPA ON
7     FUNC.IdDepartamento = DEPA.Id
8 GROUP BY DEPA.Nome
9 ORDER BY SalarioMaisAlto DESC;
```

The results pane, titled "Resultados 1", displays the output of the query. It shows a single row with the department name "RH" and the maximum salary "25.000".

Grade	ABC Nome	123 SalarioMaisAlto
1	RH	25.000

12. Listar o andar onde ficam os diretores

Objetivo:

Exibir o andar em que os diretores estão localizados.

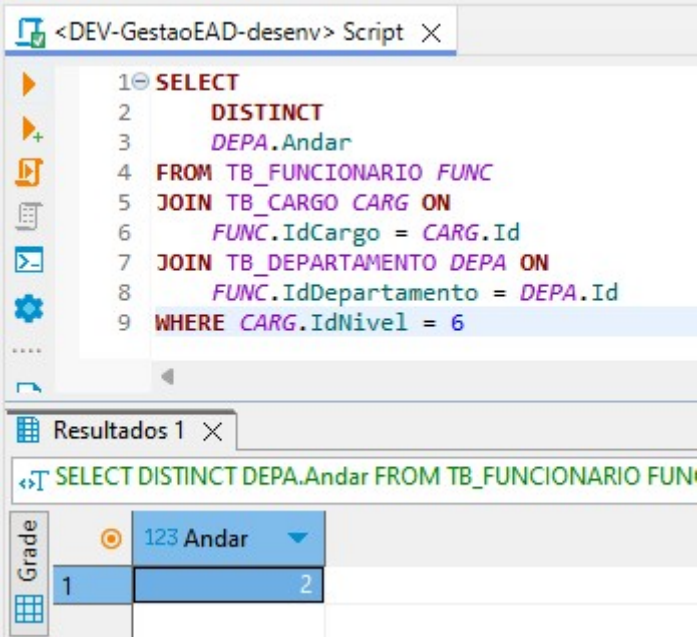
Querys:

```
SELECT
    DISTINCT
    DEPA.Andar
FROM TB_FUNCIONARIO FUNC
JOIN TB_CARGO CARG ON
    FUNC.IdCargo = CARG.Id
JOIN TB_DEPARTAMENTO DEPA ON
    FUNC.IdDepartamento = DEPA.Id
WHERE CARG.IdNivel = 6
```

Explicação:

A consulta filtra os andares onde estão alocados os diretores (nível 6).

Prints:



The screenshot shows a SQL IDE interface. The top pane, titled "<DEV-GestaoEAD-desenv> Script", contains the following SQL query:

```
1 SELECT
2     DISTINCT
3     DEPA.Andar
4 FROM TB_FUNCIONARIO FUNC
5 JOIN TB_CARGO CARG ON
6     FUNC.IdCargo = CARG.Id
7 JOIN TB_DEPARTAMENTO DEPA ON
8     FUNC.IdDepartamento = DEPA.Id
9 WHERE CARG.IdNivel = 6
```

The bottom pane, titled "Resultados 1", shows the query results. It includes a SQL statement icon, the query text "SELECT DISTINCT DEPA.Andar FROM TB_FUNCIONARIO FUN", and a table with the following data:

Grade	123 Andar
1	2

13. Listar funcionários em ordem alfabética, com lógica própria

Objetivo:

Lista funcionários cujo salário está acima da média e cujos nomes começam com "A".

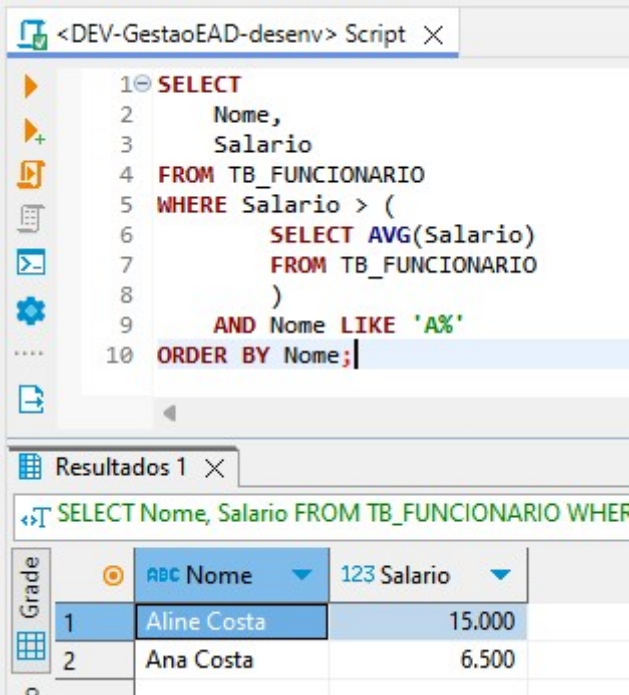
Querys:

```
SELECT
    Nome,
    Salario
FROM TB_FUNCIONARIO
WHERE Salario > (
    SELECT AVG(Salario)
    FROM TB_FUNCIONARIO
)
AND Nome LIKE 'A%'
ORDER BY Nome;
```

Explicação:

- **Filtra Funcionários:** Selecciona funcionários cujo salário é maior do que a média salarial de todos os funcionários.
- **Nome Começa com "A":** Adicionalmente, só são seleccionados funcionários cujo nome começa com a letra "A".
- **Ordena por Nome:** Os resultados são organizados em ordem alfabética pelos nomes dos funcionários.

Prints:



The screenshot shows a SQL IDE window titled "<DEV-GestaoEAD-desenv> Script". The query editor contains the following SQL code:

```
1 SELECT
2     Nome,
3     Salario
4 FROM TB_FUNCIONARIO
5 WHERE Salario > (
6     SELECT AVG(Salario)
7     FROM TB_FUNCIONARIO
8 )
9 AND Nome LIKE 'A%'
10 ORDER BY Nome;
```

Below the query editor, the "Resultados 1" window displays the results of the query. The results are shown in a table with two columns: "Nome" and "Salario". The table is sorted alphabetically by name. The results are as follows:

Grade	Nome	Salario
1	Aline Costa	15.000
2	Ana Costa	6.500

14. Listar cargos em ordem alfabética, com lógica própria

Objetivo:

Listar cargos que contenham "Analista" e possuem salário base abaixo da média.

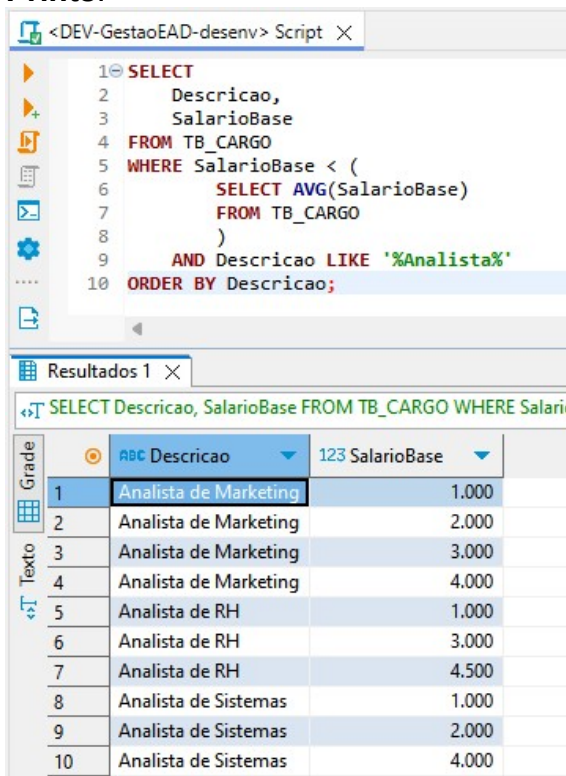
Querys:

```
SELECT
    Descricao,
    SalarioBase
FROM TB_CARGO
WHERE SalarioBase < (
    SELECT AVG(SalarioBase)
    FROM TB_CARGO
)
AND Descricao LIKE '%Analista%'
ORDER BY Descricao;
```

Explicação:

- **Filtra Cargos:** Selecciona cargos que contêm a palavra "Analista" em sua descrição.
- **Salário Base Abaixo da Média:** Só inclui aqueles cujo salário base é menor do que a média salarial base de todos os cargos.
- **Ordena por Descrição:** Os resultados são organizados em ordem alfabética pela descrição dos cargos.

Prints:



The screenshot shows a SQL IDE window with a script editor and a results pane. The script editor contains the following SQL query:

```
1 SELECT
2     Descricao,
3     SalarioBase
4 FROM TB_CARGO
5 WHERE SalarioBase < (
6     SELECT AVG(SalarioBase)
7     FROM TB_CARGO
8 )
9 AND Descricao LIKE '%Analista%'
10 ORDER BY Descricao;
```

The results pane, titled "Resultados 1", displays the query results in a table. The table has two columns: "Descricao" and "SalarioBase". The results are sorted alphabetically by description. The table contains 10 rows of data:

Grade	Descricao	SalarioBase
1	Analista de Marketing	1.000
2	Analista de Marketing	2.000
3	Analista de Marketing	3.000
4	Analista de Marketing	4.000
5	Analista de RH	1.000
6	Analista de RH	3.000
7	Analista de RH	4.500
8	Analista de Sistemas	1.000
9	Analista de Sistemas	2.000
10	Analista de Sistemas	4.000

15. Listar departamentos em ordem alfabética, com lógica própria

Objetivo:

Listar departamentos localizados em andares ímpares e cujo nome contém "E".

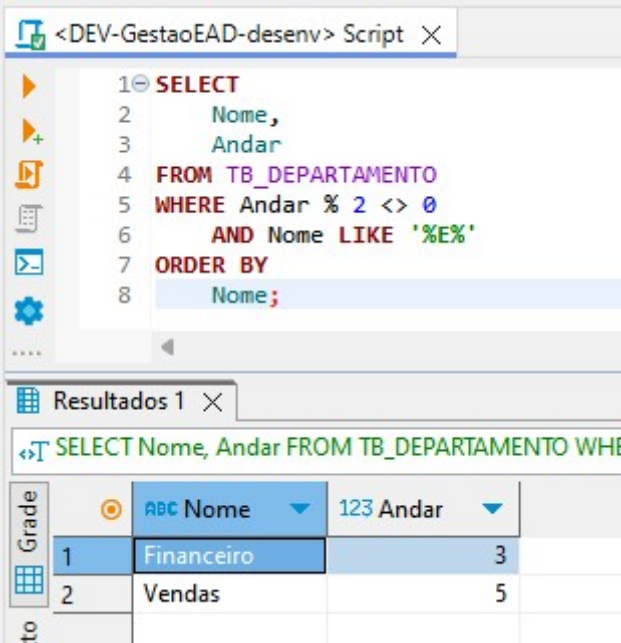
Querys:

```
SELECT
    Nome,
    Andar
FROM TB_DEPARTAMENTO
WHERE Andar % 2 <> 0
    AND Nome LIKE '%E%'
ORDER BY
    Nome;
```

Explicação:

- **Filtra Departamentos:** Seleciona departamentos que estão localizados em andares ímpares (isto é, andares que não são divisíveis por 2).
- **Nome Contém "E":** Só inclui aqueles cujo nome contém a letra "E".
- **Ordena por Nome:** Os resultados são organizados em ordem alfabética pelos nomes dos departamentos.

Prints:



The screenshot shows a SQL IDE window titled "<DEV-GestaoEAD-desenv> Script". The query editor contains the following SQL code:

```
1 SELECT
2     Nome,
3     Andar
4 FROM TB_DEPARTAMENTO
5 WHERE Andar % 2 <> 0
6     AND Nome LIKE '%E%'
7 ORDER BY
8     Nome;
```

Below the query editor, the "Resultados 1" window displays the results of the query. The results are shown in a table with columns "Grade", "Nome", and "Andar". The table has two rows of data:

Grade	Nome	Andar
1	Financeiro	3
2	Vendas	5

Considerações Finais

Neste trabalho, foram realizadas diversas consultas SQL para manipulação e análise de dados em um banco de dados com tabelas relacionadas a funcionários, cargos e departamentos. As consultas foram projetadas para atender a requisitos específicos e fornecer insights úteis sobre a estrutura e o desempenho da organização.

- **Consultas de Listagem:**
 - **Funcionários, Cargos e Departamentos:** As consultas permitiram listar os dados de funcionários, cargos e departamentos em ordem alfabética, facilitando a visualização e a análise de informações básicas.
- **Consultas de Filtro e Análise:**
 - **Departamentos e Salários:** Foi possível identificar departamentos específicos, como aqueles localizados no quinto andar, e analisar salários de funcionários para encontrar padrões e outliers.
 - **Analistas e Estagiários:** As consultas ajudaram a encontrar o analista com o maior salário e a identificar o departamento com o maior número de estagiários, fornecendo informações valiosas sobre a distribuição e a hierarquia dentro da empresa.
- **Consultas Específicas e Criativas:**
 - **Funcionários e Cargos:** As consultas criativas permitiram listar funcionários com salários acima da média e cargos com salários base abaixo da média, além de filtrar departamentos com características específicas, como andares ímpares e nomes contendo uma letra específica.

Em resumo, as consultas SQL realizadas foram eficazes em extrair e analisar informações valiosas do banco de dados. Elas forneceram uma visão clara sobre a estrutura organizacional e ajudaram a identificar áreas de interesse e oportunidades para melhorias. A documentação e as explicações fornecidas servem como uma base sólida para entender e utilizar os dados de forma eficiente.