



Centro Universitário FACENS

Caderno de Exercícios Banco de Dados

Prof. Marco Montebello

marco.montebello@facens.br

Versão 2.2

Sorocaba

Julho/2025

Sumário

INTRODUÇÃO.....	3
1. Modelo ER	5
2. DDL (Data Definition Language)	7
2.1. Importante.....	7
2.2. Modelo ER Atualizado	8
3. DML (Data Manipulation Language)	10
4. DQL (Data Query Language) – SELECT.....	12
5. Funções de Agregação (Sumarização).....	14
6. JOINS (Junções).....	16
7. Stored Procedures (Procedimentos Armazenados)	18
8. Function (Funções)	20
9. Triggers (Gatilhos)	22
9.1. Modelo ER da Tabela de Logs.....	23
10. Views (Visões).....	25
11. SubConsultas (SubQuery)	27

INTRODUÇÃO

Neste material, você encontrará atividades designadas para serem realizadas após a conclusão de cada assunto abordado. O objetivo dessas atividades é proporcionar prática e reforço dos conceitos abordados durante as aulas.

Caso surjam dúvidas, você pode esclarecê-las enviando um e-mail para marco.montebello@facens.br ou através da plataforma **Canvas**.

Esteja à vontade para buscar ajuda sempre que necessário, e aproveite ao máximo os recursos oferecidos para aprimorar seu aprendizado.

01

Modelo ER

1. Modelo ER

Para realização das atividades será criado um novo banco de dados utilizando o Modelo ER descrito a seguir:



Empregado

IDEmpregado int auto-incremento chave-primária
 Nome varchar(250) not null
 Cidade varchar(250) not null
 IDDepartamento int not null REFERÊNCIA Departamento(IDDepartamento)
 IDCargo int not null REFERÊNCIA Cargo(IDCargo)
 DataContratacao datetime not null

Departamento

IDDepartamento int auto-incremento chave-primária
 Nome varchar(250) not null

Cargo

IDCargo int auto-incremento chave-primária
 Nome varchar(100)
 Nivel varchar(100) (Exemplo de valores: Júnior, Pleno, Sênior)
 SalarioBase numeric(18,2) not null

02

DDL

Data Definition Language

2. DDL (Data Definition Language)

Para realização dessa atividade devem ser empregados os conceitos relacionados aos comandos **DDL (Data Definition Languagem)**, ou seja, Linguagem de Manipulação de Dados, que são:

- **CREATE TABLE**: Criação de Tabelas
- **ALTER TABLE**: Alteração de Tabelas
- **DROP TABLE**: Exclusão de Tabelas

1. Realizar a criação das tabelas do modelo ER indicado.
2. Criar as restrições indicadas (not null, identity, etc.).
3. Criar os relacionamentos necessários (chaves primárias e chaves estrangeiras)
4. Adicionar na tabela Empregado a coluna Estado do tipo varchar(250)
5. Adicionar na tabela Cargo a coluna Descricao do tipo varchar(150)
6. Alterar o campo Nome da tabela Cargo para não permitir valores nulos (not null)
7. Excluir o campo Descricao da tabela Cargo.
8. Caso fosse necessário excluir todas as tabelas deveria ser seguida uma ordem? Se sim qual seria?

2.1. Importante

1. Todas as operações devem ser realizadas via script.
2. Todas as chaves primárias devem ser identity (auto incremento).
3. Atenção para a ordem de criação das tabelas, pois existem dependências entre elas.

2.2. Modelo ER Atualizado

Modelo ER atualizado com base nas operações realizadas no item DDL



03

DML

Data Manipulation Language

3. DML (Data Manipulation Language)

Para realização dessa atividade devem ser empregados os conceitos relacionados aos comandos **DML (Data Manipulation Language)**, ou seja, Linguagem de Manipulação de Dados, que são:

- **INSERT**: Inserção de dados
- **UPDATE**: Atualização de dados
- **DELETE**: Exclusão de dados

1. Inserir em cada uma das tabelas pelo menos 5 (cinco) linhas de dados
 - Atenção para a ordem de inserção das informações nas tabelas
 - Inserir um empregado com o nome “João da Silva”
 - Inserir um departamento com o nome “Judicial”
2. Atualizar o Nível dos cargos para “Júnior” quando o nome for “Tester”
3. Atualizar o Estado dos Empregados para SP, quando a cidade for Sorocaba, São Paulo ou São Roque.
4. Atualizar o Nome do Departamento com nome “Judicial” para “Jurídico”.
5. Excluir empregados com nome “João da Silva”
6. Excluir os cargos com SalarioBase menores que R\$ 500,00
7. Excluir os departamentos com nome “Operacional”.

04

DQL

Data Query Language

SELECT

4. DQL (Data Query Language) – SELECT

Para realização dessa atividade devem ser empregados os conceitos relacionados aos comandos **DQL (Data Query Language)**, ou seja, Linguagem de Consulta de Dados, que são:

- **SELECT**: Selecionar as informações
- **FROM**: Tabelas que serão consultadas
- **WHERE**: Aplicação de filtro
 - **LIKE**: Utilização da busca por padrão (palavra-chave)
 - **BETWEEN**: Busca por intervalo de valores
 - **IN**: Busca “dentro” de vários valores.
 - **AND + OR**: “E” e “OU”. Inclusão de diversas cláusulas.
- **ORDER BY**: Ordenar informações

1. Selecionar todas as informações da tabela Departamento
2. Selecionar o Nome, Nível e Salário Base dos Cargos
3. Selecionar o Nome, Cidade e Data de Contratação dos Empregados
4. Selecionar os cargos onde o salário está entre R\$ 2.500,00 e R\$ 5.000,00.
5. Selecionar o Nome, Cidade, Estados e Data de Contratação dos Empregados contratos entre os anos de 2020 e 2021
6. Selecionar o Nome e Cidade dos empregados contratos em 2022 da cidade de Sorocaba.
7. Selecionar os departamentos com nome igual a “Marketing” ou “Financeiro”
8. Selecionar o Nome, Nível e Salário Base dos Cargos que possuem a palavra “Tecnologia” em qual parte do nome.
9. Selecionar os Empregados onde as cidades começam com “São” e terminam qual qualquer sequência.
10. Selecionar o Nome, Nível e Salário base dos Cargos com Nível “Pleno” e Salário Base maior ou igual a R\$ 3.000,00
11. Selecionar Nome, Cidade e Data de Contratação dos empregados com sobrenome “Silva” ou “Oliveira”. Nesse caso devemos verificar os empregados que terminado com os textos indicados.
12. Selecionar as cidades distintas dos Empregados.

05

Funções de Agregação (Sumarização)

5. Funções de Agregação (Sumarização)

Para realização dessa atividade devem ser empregados os conceitos de Agrupamento e Totalização, ou seja, sumarização de informações, que são:

- **COUNT**: Retorna o número de linhas afetadas pelo comando.
- **SUM**: Retorna o somatório do valor das colunas especificadas.
- **AVG**: Retorna a média aritmética dos valores das colunas.
- **MIN**: Retorna o menor valor da coluna de um grupo de linhas.
- **MAX**: Retorna o maior valor da coluna de um grupo de linha.
- **GROUP BY**: Agrupamento de informações.
- **HAVING**: Filtrar os dados sumarizados.

1. Exibir a quantidade de cargos cadastrados.
2. Exibir a soma de todos os salários bases dos cargos.
3. Exibir a média geral dos salários bases dos cargos.
4. Exibir o menor e o maior salário base dos cargos.
5. Exibir a soma dos salários base dos Cargos com Nível “Júnior”.
6. Exibir a quantidade de empregados por cidade (exibir a cidade e a respectiva quantidade).
7. Exibir a média salarial (salário base) de cada um dos níveis (exibir o nome do nível e a respectiva média salarial).
8. Exibir a quantidade de contratação por ano (exibir o ano e a respectiva quantidade).
Somente devem ser exibidos os anos com mais de 2 (duas) contratações.
9. Exibir a média salarial (salário base) de cada um dos níveis (exibir o nome do nível e a respectiva média salarial). Somente devem ser exibidos os níveis que a média salarial é inferior a R\$ 2.000,00.
10. Exibir a quantidade de empregados por cidade (exibir a cidade e a respectiva quantidade).
Somente devem ser exibidas as cidades com mais de 3 (três) empregados.

06

JOIN – Junções

6. JOINS (Junções)

Para realização dessa atividade devem ser empregados os conceitos junções (JOINS), que são:

- **INNER JOIN**: Retorna todas as linhas que correspondem ao critério de ligação entre as tabelas A e B.
 - **LEFT JOIN**: Retorna todas as linhas da tabela a esquerda da junção e as respectivas correspondências quando existirem.
 - **RIGHT JOIN**: Retorna todas as linhas da tabela a direita da junção e as respectivas correspondências quando existirem.
 - **FULL JOIN**: Retorna todas as linhas das tabelas a esquerda e direita da junção e as respectivas correspondências quando existirem.
1. Criar uma consulta para exibir o Nome e Cidade do Empregado e o Nome do Departamento.
 2. Criar uma consulta para exibir o Nome e Data de Contratação do Empregado e o Nome e Nível do Cargo.
 3. Criar uma Consulta para exibir a quantidade de empregados por departamento (exibir o nome do departamento e a respectiva quantidade de empregados).
 4. Criar uma consulta para exibir a quantidade de empregados por cargo (exibir o nome do cargo e a respectiva quantidade de empregados).
 5. Criar uma consulta par exibir o Nome e Cidade do Empregado e o Nome do Departamento quando o nome do departamento possuir “Tecnologia” ou “TI”.
 6. Criar uma consulta para exibir o Nome e Data de Contratação dos Empregados e o Nome do Cargo dos empregados contratados entre 2019 e 2022.
 7. Criar uma consulta para exibir o Nome, Cidade e Data de Contratação do Empregado, o Nome do Departamento e o Nome, Nível e Salário Base do Cargo.
 8. Criar uma Consulta para exibir o nome de TODOS os departamentos e a quantidade de empregados de cada um deles (exibir o nome do departamento e a respectiva quantidade de empregados, quando não possuir empregados a quantidade deve ser ZERO).
 9. Criar uma Consulta para exibir o nome de TODOS os cargos e a quantidade de empregados de cada um deles (exibir o nome do cargo e a respectiva quantidade de empregados, quando não possuir empregados a quantidade deve ser ZERO).

07

Stored Procedures

Procedimentos Armazenados

7. Stored Procedures (Procedimentos Armazenados)

Para realização dessa atividade devem ser empregados os conceitos de procedimentos armazenados (**Stored Procedures**).

1. Criar uma Stored Procedure que retorna à quantidade de Cargos cadastrados.
2. Criar uma Stored Procedure que recebe como parâmetro o nome da Cidade e retorno a quantidade de empregados na mesma.
3. Criar uma Stored Procedure que realiza a inserção de um Cargo (todos os campos devem ser informados via parâmetro).
4. Criar uma Stored Procedure que realiza a inserção de um Departamento (somente o nome deve ser informado via parâmetro).
5. Criar uma Stored Procedure que recebe como parâmetro o Nome do Empregado e retorna via parâmetro de OUTPUT o ano e mês da sua contratação.
6. Criar uma Stored Procedure que recebe 3 (três) valores decimais, e retorna via parâmetro de OUTPUT a média aritmética entre eles.
7. Criar uma Stored Procedure que realiza a inserção de um Empregado (somente os campos obrigatórios (not null) devem ser informados via parâmetro). Realizar o tratamento de erros durante o processo.
8. Criar uma Stored Procedure que recebe como parâmetro o ID do Departamento e realiza a sua exclusão. Para esse processo deve ser aplicado o procedimento de tratamento de erros, e quando o ele ocorrer uma mensagem de erro personalizada deve ser exibida.
9. Criar uma Stored Procedure que recebe como parâmetro o nome do Cargo e o novo valor do Salário Base para atualização. Somente deve ser realizada a atualização se o novo valor do salário base for pelo menos 10% maior que o valor atual, caso contrário a atualização não deve ser realizada e uma mensagem de erro personalizada deve ser exibida.

08

Function

Funções

8. Function (Funções)

Para realização dessa atividade devem ser empregados os conceitos funções (**FUNCTIONS**), que são:

- **Scalar Function:** Realiza o retorno de um único valor.
 - **InLine Table Function:** Realiza o retorno de uma tabela (“views parametrizadas”)
 - **MultiStatement Function:** Realiza o retorno de uma tabela (lógica mais elaborada)
1. Criar uma Scalar Function que recebe como parâmetro uma data e retorna a quantidade de anos com base na data atual (diferença entre a data informada e a data atual)
 - Como exemplo, realizar a chamada da função utilizando a data de contratação do empregado.
 2. Criar uma Scalar Function que recebe como parâmetro o ID do Cargo e retorna o Salário Base.
 3. Criar uma Scalar Function que recebe como parâmetro o ID do Empregado e com base no Salário Base retorna se ele está abaixo ou acima da média.
 4. Criar uma InLine Table Function que recebe como parâmetro um ano e retorna o Nome, Cidade, Data de Contratação do Empregado, Nome do Departamento e Nome e Salário Base do Cargo dos empregados contratados no ano indicado.
 5. Criar uma InLine Table Function que recebe como parâmetro um salário e retorna o Nome e Cidade do Empregado e o Nome e salário base do cargo que estão acima do salário informado.
 6. Criar uma MultiStatement Table Function que recebe como parâmetros um intervalo de anos e retorna o Nome e Data de Contratação do Empregado, o Nome do Departamento e o Nome do Cargo.
 7. Criar uma MultiStatement Table Function que recebe como parâmetros um intervalo de anos e retorna o ano e a quantidade de empregados em cada um deles.
 8. Criar uma MultiStatement Table Function que recebe como parâmetro um intervalo de salários e retorna o Nome do Empregado, Nome do Departamento e Nome do Cargo.

09

Triggers Gatilhos

9. Triggers (Gatilhos)

Para realização dessa atividade devem ser empregados os conceitos de gatilhos (**TRIGGERS**):

- **TRIGGERS:** criação de gatilhos
 - Tabelas **INSERTED** e **DELETED**: Tabelas virtuais com as informações que serão inseridas e ou atualizadas.
1. Criar a tabela de Logs conforme modelo ER descrito.
 2. Criar um gatilho (trigger) que permita a inclusão e/ou alteração de empregados somente entre os dias 1 e 10 de cada mês. Quando o processo estiver fora do intervalo indicado uma mensagem de erro personalizada deve ser exibida e ele deve ser cancelado.
 3. Criar um gatilho (trigger) que permita a exclusão de departamentos somente após o dia 25 de cada mês. Quando o processo estiver fora do intervalo indicado uma mensagem de erro personalizada deve ser exibida e ele deve ser cancelado.
 4. Criar um gatilho (trigger) que verifique no momento da inclusão de um novo empregado se a data de contratação é maior que a data atual. Quando isso ocorrer a inclusão deve ser cancelada e uma mensagem de erro personalizada deve ser exibida.
-
5. Criar um gatilho (trigger) que realize a gravação na tabela de logs todas as inclusões realizadas na tabela de departamentos. Armazenar o nome do departamento e a data e hora da inclusão. A trigger criada deve ser *after*.
 - Lembre-se das tabelas INSERTED e DELETED.
 6. Criar um gatilho (trigger) que realize a gravação na tabela de logs todas as alterações realizadas na tabela de empregados. Armazenar o Nome, Cidade e Data de Contratação atuais e novos. A trigger criada deve ser *after*.
 - Lembre-se das tabelas INSERTED e DELETED.
 7. Criar um gatilho (trigger) que realize a gravação na tabela de logs todas as exclusões realizadas na tabela de Cargos. Armazenar o Nome, Nível e Salário Base. A trigger criada deve ser *after*.
 - Lembre-se das tabelas INSERTED e DELETED.

Realizar a gravação da data e hora do processo e qual tabela realizou a ação.

9.1. Modelo ER da Tabela de Logs

LogData	
IDLogData	int
Tabela	varchar(255)
Operacao	varchar(255)
Detalhes	varchar(1000)
DataEvento	datetime

LogData

IDLogData int chave primária auto incremento not null
Tabela varchar(255) not null
Operacao varchar(255) not null
Detalhes varchar(1000) not null
DataEvento datetime not null

10

Views

Visões

10. Views (Visões)

Para realização dessa atividade devem ser empregados os conceitos de visões (**Views**).

1. Criar uma visão (view) para exibir a quantidade de empregados por departamento (exibir o nome do departamento e a respectiva quantidade de empregados).
2. Criar uma visão (view) para exibir a quantidade de empregados por cargo (exibir o nome do cargo e a respectiva quantidade de empregados).
3. Criar uma visão (view) para exibir o valor médio do salário base por cidade (exibir o nome da cidade e o valor médio do salário).
4. Criar uma visão (view) para exibir o Nome, Cidade e Data de Contratação do Empregado, o Nome do Departamento e o Nome, Nível e Salário Base do Cargo.

11

SubConsultas

SubQuery

11. SubConsultas (SubQuery)

Para realização dessa atividade devem ser empregados os conceitos de **subconsultas** ou **subquerys**.

1. Criar uma SubConsulta para exibir o Nome, Cidade e Salário Base dos empregados que estão acima da média salarial.
2. Criar uma SubConsulta para exibir o Nome, Salário Base e Data de Contratação dos empregados que estão abaixo da média salarial.
3. Criar uma SubConsulta para exibir o Nome e Nível dos Cargos que estão sendo utilizados pelos empregados.
4. Criar uma SubConsulta para exibir o Nome dos Departamentos que não estão sendo utilizados pelos empregados.

Obrigado!!!

Marco Montebello

marco.montebello@facens.br

@montebellojr



*“O único lugar onde o sucesso
vem antes do trabalho é no
dicionário.”*

– Albert Einstein

