

Fundamentos de Bancos de Dados

Aula 1



Prof. Me. Marco Aurelio M. Antunes

Marco Aurelio Migliorini Antunes



Formação:

Curso Técnico: Técnico em Contabilidade

Graduação: Análise de Sistemas – USC

Graduação: Matemática - FAEL

Pós Graduação: Recursos Humanos com Sistemas de Informação – FIB

Mestrado: Tv Digital – FAAC – UNESP

Atuação Profissional:

- Lógica de Programação
- Banco de Dados
- Desenvolvimento de Sistemas
- Desenvolvimento de Sites
- Pós Graduação – FIB
- Professor/Coordenador Curso Técnico em Informática – ABFA/INSSC
- Professor/Coordenador Fibonline/Atividade Complementar - FIB

Contato:

- prof_antunes@outlook.com
- <https://www.linkedin.com/in/marco-aurelio-m-antunes-85826027/>
- <http://lattes.cnpq.br/0148514838383451>

Fundamentos de Bancos de Dados

Ementa:

Conceitos de Sistema Gerenciador de Banco de Dados. Conceitos de Banco de Dados. Conceitos, arquitetura e modelos de dados. Estudar modelos de Entidade-Relacionamento e modelo Relacional. Estudar Normalização e Normalização de Arquivos de Dados. Utilizar ferramentas de administração e segurança de Banco de Dados.

Objetivos Gerais:

Preparar e capacitar os alunos a planejarem e desenvolverem sistemas de gerenciamento de banco de dados de forma interativa e programada. E realizar análise e modelagem de dados na programação contida em um sistema, buscando sua otimização e a prevenção de falhas que podem ocorrer durante a sua utilização.

Objetivos Específicos:

- Proporcionar ao aluno o conhecimento necessário para o criar e gerenciar banco de dados em sistemas completos utilizando scripts e funções.
- Desenvolver a capacidade de realizar a análise, modelagem de dados e otimização de banco de dados, tabelas e relacionamentos.
- Capacitar os alunos com comandos e técnicas de operação, proporcionando assim, condições de gerenciar banco de dados de forma interativa e programada.
- Estudos de exemplos práticos e simulações de recuperação de dados, segurança de banco de dados e backup de dados.

CONTEÚDO PROGRAMÁTICO

Conceitos Básicos

- Terminologia básica de banco de dados
- Dados e Informações
- Definição de Banco de Dados, tabelas e campos
- Gerência de Banco de Dados
- Inserir/Consultar informações

Organização de Arquivos

- Tipos de organização de arquivos
- Montagem de Banco de Dados
- Manutenção de Arquivos
- Recuperação de Dados
- Relacionamento entre tabelas
- Pesquisar/Filtrar informações

Banco de Dados

- Criação do banco de dados
- Ambiente de Trabalho
- Criação de tabelas
- Inclusão de Dados - registros
- Manutenção de Dados
- Operações com arquivos
- Excluir/Alterar informações



Fundamentos de Bancos de Dados

Bibliografia:

MANZANO, J. A. N. G., **Microsoft SQL Server 2014 Express. Guia Prático e Interativo**, 1ª ed. Erica, 2014, 224p.

GONCALVES, R.R., **T-SQL com Microsoft SQL Server 2012 Express na prática**, 1ª ed. Erica, 2013, 120p.

BAPTISTA, L. F., **Linguagem SQL - Guia Prático de Aprendizagem**, 1ª ed., Erica, 2011, 160p.

MANZANO, J. A. N. G., **Microsoft SQL Server 2008 R2 Express - Guia Prático**, 1ª ed., Erica, 2011, 208p.

MANZANO, J. A. N. G., **Microsoft SQL Server 2012 Express - Guia Prático e Interativo**, 1ª ed., Erica, 2012, 208p.

Fundamentos de Bancos de Dados

Planejamento: 1º Bimestre

9 aulas

28/04 - Avaliação Bimestral - Prova Escrita

TODAS as listas de exercício - **1 ponto**

Planejamento: 2º Bimestre

5 aulas

02/06 - Avaliação Bimestral - Prova Escrita - **peso 7**

09/06 - Avaliação Bimestral - Prova Escrita - **peso 3**

TODAS as listas de exercício - **1 ponto**

2ª Chamada

23/06 - 1º Bimestre e/ou 2º Bimestre

Exame Final

30/06 - Matéria do semestre



Informações Importantes

Sistema Acadêmico:

<http://www.fibbauru.br/alunos>

Calendário Acadêmico:

<https://fibbauru.br/custom/561/uploads/calendario-academico/calendario-academico-2022-primeiro-semester.pdf>

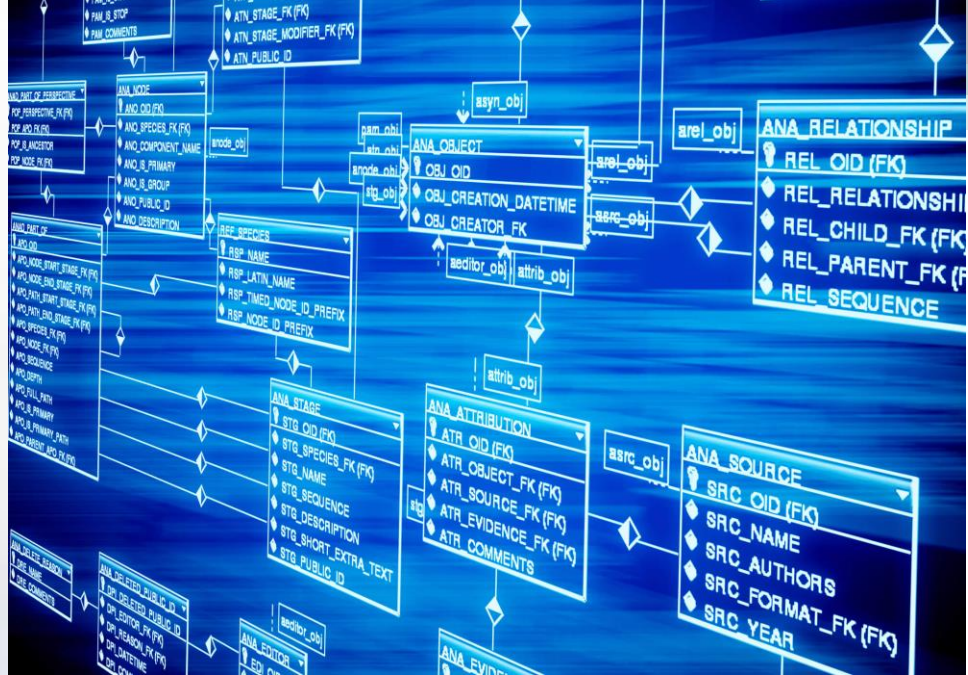
Jornada Científica:

<http://www.fibbauru.br/site/conteudo/174-como-participar.html>

Atividades Complementares:

Atendimento presencial – toda segunda-feira das 19h as 20h30





O que é um Banco de Dados?

São **coleções de informações que se relacionam** de forma que crie um sentido. São de vital importância para empresas, e há duas décadas se tornaram a **principal peça** dos sistemas de informação.

São conjuntos de arquivos relacionados entre si com registros sobre **pessoas, lugares, ocorrências, comportamento, preferências ou tendências**.

São coleções organizadas de dados que **se relacionam de forma a criar algum sentido (Informação)** e dar mais eficiência durante uma pesquisa ou estudo científico.

É um ambiente onde é possível armazenar dados de forma organizada, gerando conjuntos de arquivos. Ele permite o **agrupamento de informações** que, posteriormente, podem ser extraídas para outras funções.

Um sistema de gerenciamento de banco de dados **(SGBD)** é um software que possui recursos capazes de manipular as informações do banco de dados e interagir com o usuário – SQL Server / Oracle / MySQL.

Para que serve um Banco de Dados?

1. Bancos de dados armazenam e dão acesso rápido às informações

É por meio dos bancos de dados que os analistas de suporte ao cliente/usuário têm acesso às bases de dados de uma empresa para trazer à tona **informações**. Além disso, eles ajudam a evitar os custos e a inconveniência de lidar com registros em papel.

2. Permitem automatizar processos

Os bancos de dados também são muito úteis porque eles podem **automatizar uma série de procedimentos**. Em vez de verificar as transações financeiras manualmente, por exemplo, os responsáveis podem trabalhar com bases de dados e parametrizar (definir) comandos para que elas gerem relatórios automáticos.

3. São fundamentais para nortear planos de ação e tomada de decisão

Os bancos de dados podem determinar várias iniciativas. Da política de promoções personalizada à segmentação de ações de marketing, passando por projetos de expansão dos negócios, entre outras.

DEFINIÇÕES IMPORTANTES

Muitos consideram ***dados e informações*** como palavras sinônimas, mas na verdade não são. Para entender o que é um banco de dados é muito importante saber a diferença entre essas duas palavras.


Dados são registros soltos, aleatórios, sem quaisquer análise, que constituem a matéria prima da informação, ou seja, é a informação não tratada que ainda não apresenta relevância.

Exemplos de dados

Dados do estudante: sobre os formulários de admissão: quando os alunos recebem admissão em uma faculdade. Eles preenchem o formulário de admissão. Este formulário contém fatos em bruto (dados do aluno) como nome, nome do pai, endereço do aluno etc.

Dados dos cidadãos: durante o recenseamento, os dados de todos os cidadãos são coletados.

Dados do levantamento: diferentes empresas coletam dados por pesquisa para conhecer a opinião das pessoas sobre seu produto.



Informação é a ordenação e organização dos dados de forma a transmitir significado e compreensão dentro de um determinado contexto. Seria o conjunto ou consolidação dos dados de forma a fundamentar o conhecimento.

Exemplos de informações

Etiquetas de endereço do estudante/clientes: os dados armazenados dos alunos/clientes podem ser usados para imprimir rótulos de endereços dos alunos/clientes.

Relatório do Censo: os dados do Censo são utilizados para obter relatórios / informações sobre a população total de um país e a taxa de alfabetização, costumes de gastos.

Relatórios de pesquisa e resultados: os dados da pesquisa são resumidos em relatórios / informações para apresentar ao gerenciamento da empresa que buscam obter **CONHECIMENTO**.

Metadados - são dados sobre dados – mais especificamente, informações sobre um determinado conteúdo. São **informações estruturadas que descrevem, explicam, localizam**, ou seja, tornam fácil recuperar, usar ou gerenciar uma fonte de informação. Como exemplo, no contexto de uma biblioteca, onde os dados são o conteúdo dos títulos em estoque, os metadados a respeito de um título normalmente incluem uma descrição do conteúdo, o autor, a data de publicação e sua localização física.

Os tipos mais comuns de metadados são:

- Número de telefones, endereços de e-mail e os nomes das pessoas que usam serviços;
- Dados de Localização: onde está o seu telefone celular;
- Data e hora em que foram feitas as ligações, e-mails, arquivos e fotos;
- Informações do aparelho que você está usando;
- Os títulos (assuntos) de seus e-mails;

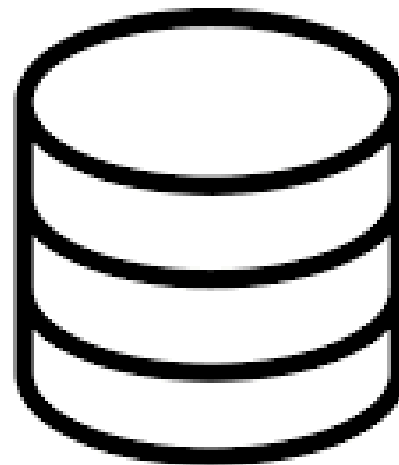
A gestão dos bancos de dados

A fim de poder controlar os dados bem como os usuários, é necessário um sistema de gestão. A gestão do banco de dados faz-se graças a um sistema chamado **SGBD (Sistema de Gestão de Bancos de Dados)**. O SGBD é um conjunto de serviços (aplicações software) que permitem gerenciar os bancos de dados, quer dizer:

- permitir o acesso aos dados de maneira simples;
- autorizar um acesso às informações a múltiplos usuários;
- manipular os dados presentes no banco de dados (inserção, supressão, modificação).

A solid green rectangular box containing the text "BANCO DE DADOS" in bold white uppercase letters, with "Conceitos Fundamentais" in smaller white lowercase letters below it.

BANCO DE DADOS
Conceitos Fundamentais





PostgreSQL



<http://db-engines.com/en/ranking>

383 systems in ranking, February 2022

Rank			DBMS	Database Model	Score		
Feb 2022	Jan 2022	Feb 2021			Feb 2022	Jan 2022	Feb 2021
1.	1.	1.	Oracle	Relational, Multi-model	1256.83	-10.05	-59.84
2.	2.	2.	MySQL	Relational, Multi-model	1214.68	+8.63	-28.69
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	949.05	+4.24	-73.88
4.	4.	4.	PostgreSQL	Relational, Multi-model	609.38	+2.83	+58.42
5.	5.	5.	MongoDB	Document, Multi-model	488.64	+0.07	+29.69
6.	6.	7.	Redis	Key-value, Multi-model	175.80	-2.18	+23.23
7.	7.	6.	IBM Db2	Relational, Multi-model	162.88	-1.32	+5.26
8.	8.	8.	Elasticsearch	Search engine, Multi-model	162.29	+1.54	+11.29
9.	9.	11.	Microsoft Access	Relational	131.26	+2.31	+17.09
10.	10.	9.	SQLite	Relational	128.37	+0.94	+5.20
11.	11.	10.	Cassandra	Wide column	123.98	+0.43	+9.36
12.	12.	12.	MariaDB	Relational, Multi-model	107.11	+0.69	+13.22
13.	13.	13.	Splunk	Search engine	90.82	+0.37	+2.28
14.	14.	15.	Microsoft Azure SQL Database	Relational, Multi-model	84.95	-1.37	+13.67
15.	17.	35.	Snowflake	Relational	83.18	+6.36	+64.96
16.	15.	14.	Hive	Relational	81.88	-1.57	+9.56
17.	16.	17.	Amazon DynamoDB	Multi-model	80.36	+0.50	+11.21
18.	18.	16.	Teradata	Relational, Multi-model	68.57	-0.56	-2.33
19.	19.	20.	Solr	Search engine, Multi-model	58.53	+0.00	+7.84
20.	20.	19.	Neo4j	Graph	58.25	+0.21	+6.08
21.	21.	21.	SAP HANA	Relational, Multi-model	56.31	-0.61	+6.09
22.	22.	22.	FileMaker	Relational	54.14	-1.72	+7.94
23.	23.	18.	SAP Adaptive Server	Relational, Multi-model	49.54	-1.52	-2.71
24.	24.	24.	Google BigQuery	Relational	45.10	-0.52	+9.21
25.	25.	23.	HBase	Wide column	43.62	-0.37	-0.87
26.	26.	25.	Microsoft Azure Cosmos DB	Multi-model	39.95	-0.09	+8.29

Microsoft SQL Server

Um Sistema Gerenciador de Banco de Dados(SGDB) necessita de uma linguagem que permita aos usuários acesso aos dados.

SQL(Structured Query Language) – Linguagem Estruturada de Consultas é a linguagem utilizada por muitos sistemas de banco de dados relacionais.



Principais Características

- Compartilhamento de dados;
- Recuperação de dados;
- Inserir, atualizar e excluir registros de uma tabela;
- Criar, modificar e excluir objetos do banco de dados;
- Controlar acesso ao banco de dados;
- Garantir consistência de dados;
- Cópias segurança.
- Evitar redundância de dados.



Banco de Dados

Após a instalação

SQL Server Management Studio Express New Query – Nova Consulta

Criando um Banco de Dados

Um banco de dados é um conjunto coerente e lógico de dados relacionados. Esses dados representam aspectos do mundo real e devem ser mantidos para atender aos requisitos da empresa e estão dispostos em uma ordem pré-definida para atender a determinadas necessidades dos usuários.

Para iniciar o trabalho de criação do banco de dados, deve-se inicialmente transformar o modelo lógico, representado pelo modelo de Entidade-Relacionamento, no modelo físico que será implementado.

O processo básico consiste em atribuir tipo de dado e tamanho para cada um dos atributos identificados.

Sintaxe

CREATE DATABASE nome_do_banco_de_dados;

REGRAS: Nome_do_banco_de_dados

- Deve ser único.
- Não pode usar nome composto.
- Não pode começar com número.
- Não pode usar acentos, cedilhas e símbolos e caracteres especiais.

Obs. O terminador de comando específico utilizado pelo SQL é ponto e vírgula (;) e os comentários de sintaxe é utilizado dois traços (--).

RULES:

- 1.
- 2.
- 3.



IMPORTE!!!
aos bons
programação!

Fique atento
costumes da

Exemplo

-- criando banco de dados FIB

create database fib;

Obs. Mensagem esperada

Command(s) completed successfully.



Obs. Tentativa de duplicação do banco de dados

Msg 1801, Level 16, State 3, Line 2

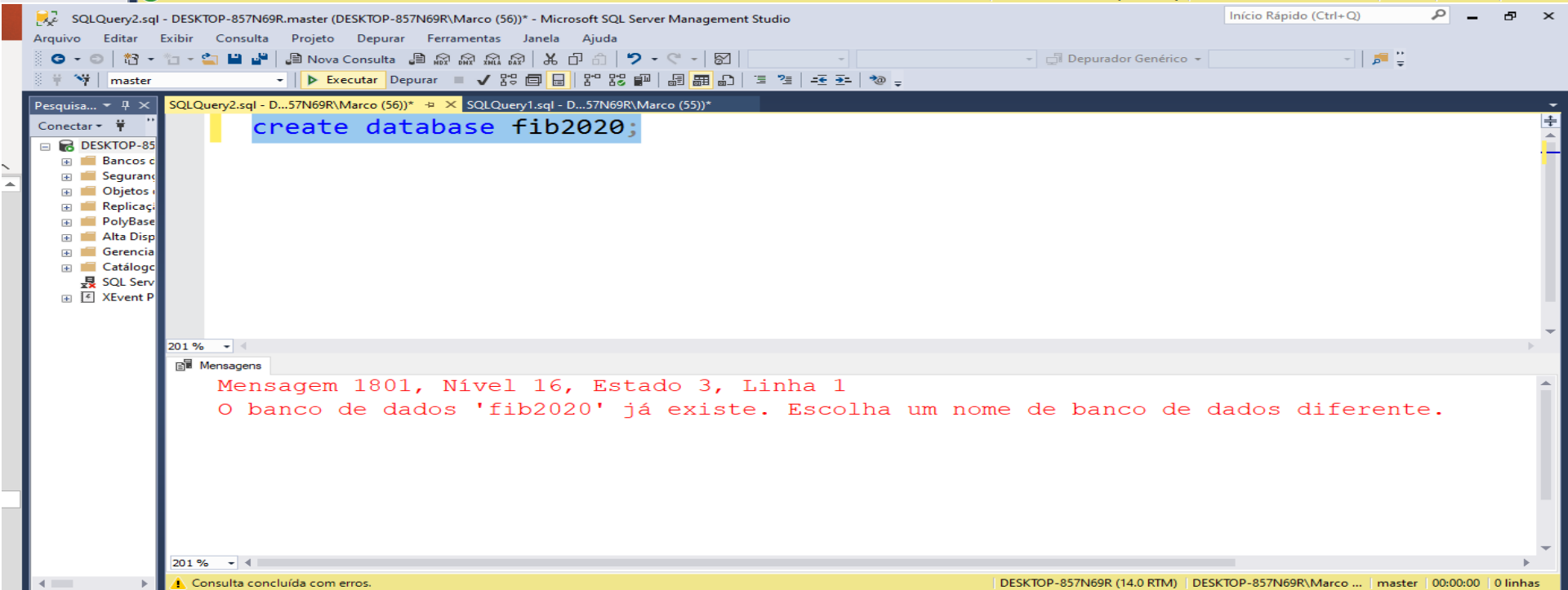
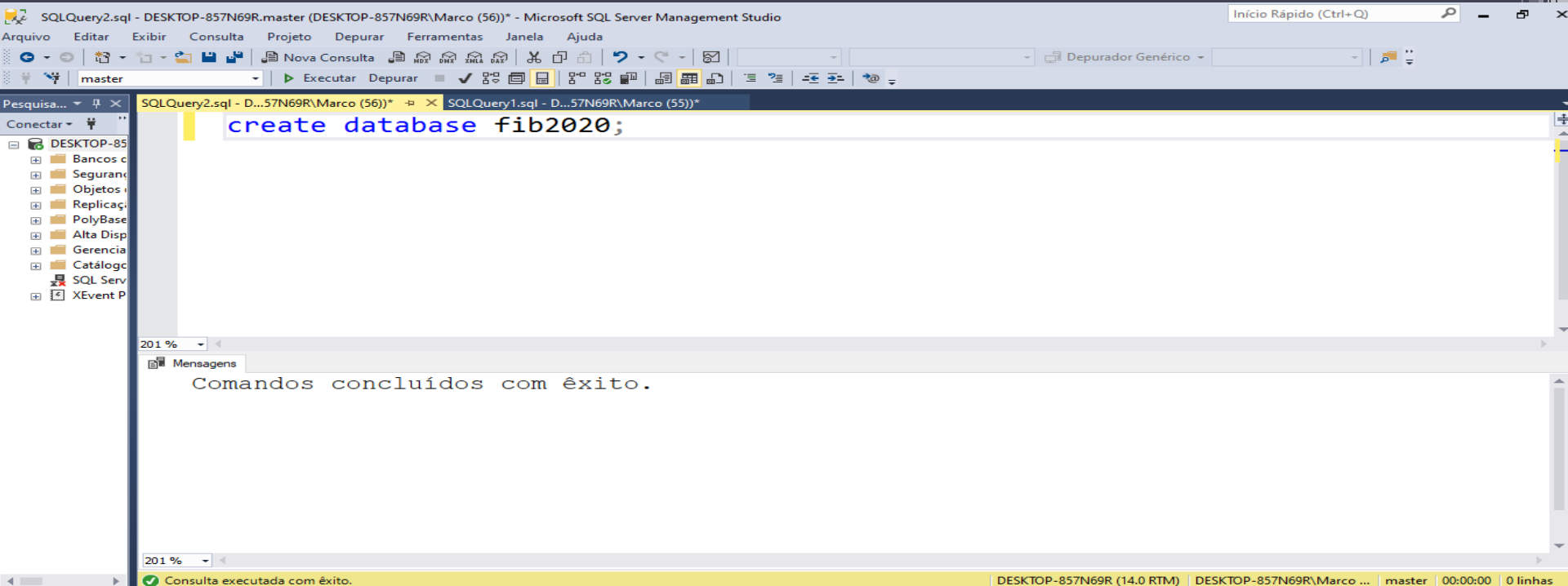
O banco de dados 'fib' já existe. Selecione um nome de banco de dados diferente.

Selecionando banco de dados

Para conectar-se a um Banco de dados, utilize:

USE nome_do_banco_de_dados;

use fib;



master

Executar Depurar

SQLQuery3.sql - D...57N69R\Marco (57))*

SQLQuery2.sql - D...57N69R\Marco (56))

```
use fib2020;
```

fib2020

SQLQuery3.sql - D...57N69R\Marco (57))*

SQLQuery2.sql - D...57N69R\Marco (56))*

```
use fib2020;
```

201 %

Mensagens

Comandos concluídos com êxito.

TIPOS DE DADOS

VARCHAR(N): Armazena N caracteres (até 8.000). Se a quantidade de caracteres armazenada no campo for menor que o tamanho total especificado em N, **o resto do campo não é preenchido.**

DECIMAL(I,D): Armazena valores numéricos inteiros com casas decimais utilizando precisão. I deve ser substituído pela quantidade de dígitos total do número e D deve ser substituído pela quantidade de dígitos da parte decimal (após a vírgula).

OBS. Lembrando sempre que o SQL Server internamente armazena o separador decimal como ponto (.) e o separador de milhar como vírgula (,).

INT: Valores numéricos inteiros variando de -2.147.483.648 até 2.147.483.647

DATETIME: Armazena hora e data variando de 1 de janeiro de 1753 até 31 de Dezembro de 9999. A precisão de hora é armazenada até os centésimos de segundos.

Integridade e Criação de Tabelas

Integridade Referencial – Constraints

Constraints são regras agregadas a colunas ou tabelas. Assim, pode-se definir como obrigatório o preenchimento de uma coluna que tenha um valor-padrão quando uma linha for incluída na tabela ou quando aceitar apenas alguns valores preenchidos. No caso de regras aplicadas a tabelas, tem-se a definição de chaves primárias e estrangeiras.

Criação de Tabelas

Tabelas são as estruturas mais importantes no banco de dados. Nas tabelas estará o conteúdo que representa cada objeto do mundo real, cuja importância para o funcionamento do sistema justifica a sua criação.

Entende-se por tabela como um conjunto de linhas e colunas que qualificam cada elemento com informações relacionadas ao objeto.

Utilizando esses conceitos, é possível armazenar dados em uma ou várias tabelas, dependendo do que e como desejamos as informações.

Sintaxe básica:

```
CREATE TABLE nome_da_tabela  
(nome_coluna1 tipo_dado_coluna1 coluna1_constraints,  
(nome_coluna2 tipo_dado_coluna2 coluna2_constraints,  
(nome_colunaN tipo_dado_colunaN colunaN_constraints,  
Constraint_da_tabela);
```

REGRAS: Nome_da_tabela

- Deve ser única no banco de dados.
- Não pode usar nome composto.
- Não pode começar com número.
- Não pode usar acentos, cedilhas, símbolos e caracteres especiais.

REGRAS: Nome_coluna

- Deve ser única na tabela.
- Não pode usar nome composto.
- Não pode começar com número.
- Não utilizar comandos como nome de campo.
- Não pode usar acentos, cedilhas, símbolos e caracteres especiais.

Tipo_dado_coluna - Utilizar tipo e tamanho de dado conforme tabela de tipo de dados.

Coluna_constraints - Regras agregadas a coluna.

Constraints_da_tabela - Regras agregadas a tabela inteira.

RULES:

- 1.
- 2.
- 3.



IMPORTE!!! Fique atento aos bons costumes da programação!

Constraints

NOT NULL

Indica que o conteúdo de uma coluna não poderá ser nulo. Dessa forma, uma coluna com a restrição Not Null, retornará uma mensagem de erro e não incluirá a linha que não apresente valor.

NULL

Indica que o conteúdo de uma coluna poderá ser nulo. Dessa forma, uma coluna com a restrição null, incluirá uma linha que não apresente valor e não retornará mensagem de erro.

Chave primária

Chave primária é a coluna ou grupo de colunas, que permite identificar um único registro na tabela. Para especificar que uma coluna ou grupo de colunas representa a chave primária de uma tabela, deve-se acrescentar a palavra-chave PRIMARY-KEY seguida do nome da(s) coluna(s).

Exemplo 1

-- Criando tabela de clientes utilizando constraint not null /primary key

```
create table clientes(  
  idcli      int          not null,  
  nomecli    varchar(40)  not null,  
  fonecli    varchar(18)  null  
  primary key(id_cli)  
);
```

-- Inserindo dados na tabela clientes

```
insert into clientes values (1, 'Marco', '014 3262-0000');
```

-- verificando os dados cadastrados

```
select * from clientes;
```

Obs. Possibilidade de erro

- Tentar criar tabela já criada

Msg 2714, Level 16, State 6, Line 2

Já existe um objeto com nome 'CLIENTES' no banco de dados.

- Inserir registro com mesma chave primária

Msg 2627, Level 14, State 1, Line 1

Violação da restrição PRIMARY KEY 'PK__CLIENTES__4830B400'. Não é possível inserir a chave duplicada no objeto 'dbo.CLIENTES'.

A instrução foi finalizada.

Exemplo 2

-- Criando tabela de clientes2 com auto-incremento

```
create table clientes2(  
codcli      int          identity not null,  
cliente     varchar(60)      not null,  
Dtnasc      datetime        not null  
primary key(cliente)  
);
```

-- inserindo dados na tabela de clientes2

```
insert into clientes2 values ('Marco', '11/03/1967');
```

-- verificando dados cadastrados

```
select * from clientes2;
```

Obs. Como o campo CODIGO_CLI é auto-incremento, ele NÃO deve ser preenchido na inclusão do registro.

Exemplo 3

-- Criando tabela de clientes3 SEM chave primária e SEM auto-incremento

```
create table clientes3(  
  idcli      int          not null,  
  cliente    varchar(60)  not null,  
  dtnasc     datetime     null  
);
```

-- inserindo dados na tabela de clientes3

```
insert into clientes3 values (1, 'Marco', '11/03/1967');  
insert into clientes3 values (1, 'Marco', '11/03/1967');
```

-- verificando dados cadastrados

```
select * from clientes3;
```

Obs. Nesse caso a redundância de dados é permitida.



Exercícios - Aula 1

Leia com ATENÇÃO !!! – TODOS os exercícios devem ser desenvolvidos em um único banco de dados e em um único script.

1- Criar o banco de dados fib2022.

2- Criar uma tabela agenda com pelo menos 5 campos com dados de seus amigos e inserir 3 registros. **Utilizar SOMENTE chave primária**

3- Acessar um site de compras online, verificar os campos necessários para criar um cadastro de clientes e criar uma tabela, inserir 3 registros. **Utilizar chave primária e auto incremento**

4- Acessar um site de e-mails, verificar os campos necessários para fazer o cadastro e criar uma tabela, inserir 3 registros. **Utilizar somente auto incremento**

5- Um professor pretende fazer um levantamento de dados dos alunos da sala de aula com informações pessoais, profissionais e acadêmicas. Analise os campos necessários e crie uma tabela com pelo menos 10 campos. Inserir 3 registros. **NÃO utilizar chave primária – NÃO utilizar auto incremento.**

6 – Criar um tabela de produtos com pelo menos 6 campos e utilizando as constraints – regras que você julgar necessária para cada campo. Inserir 3 registros.

Postar lista1.sql