FMU
CENTRO UNIVERSITÁRIO

1968 • 2018
50 Anos
HISTÓRIA PRA FAZER A SUA

BANCO DE DADOS

AULA – 06: MODELAGEM LÓGICA RELACIONAL

NOTAÇÕES DO DIAGRAMA EM FERRAMENTAS DE MODELAGEM

BANCO DE DADOS

MODELO RELACIONAL

- **OBJETIVOS DE APRENDIZAGEM**
  1. CONSTRUIR DIAGRAMAS COM O AUXÍLIO DE FERRAMENTA COMPUTACIONAL.
  2. EXPERIMENTAR AS POSSIBILIDADES QUE O USO DA FERRAMENTA DE MODELAGEM PROPORCIONA.
  3. IDENTIFICAR A NOTAÇÃO USADA PELA FERRAMENTA PARA REPRESENTAR OS CONCEITOS DE MODELAGEM CONCEITUAL E/OU LÓGICA RELACIONAL.

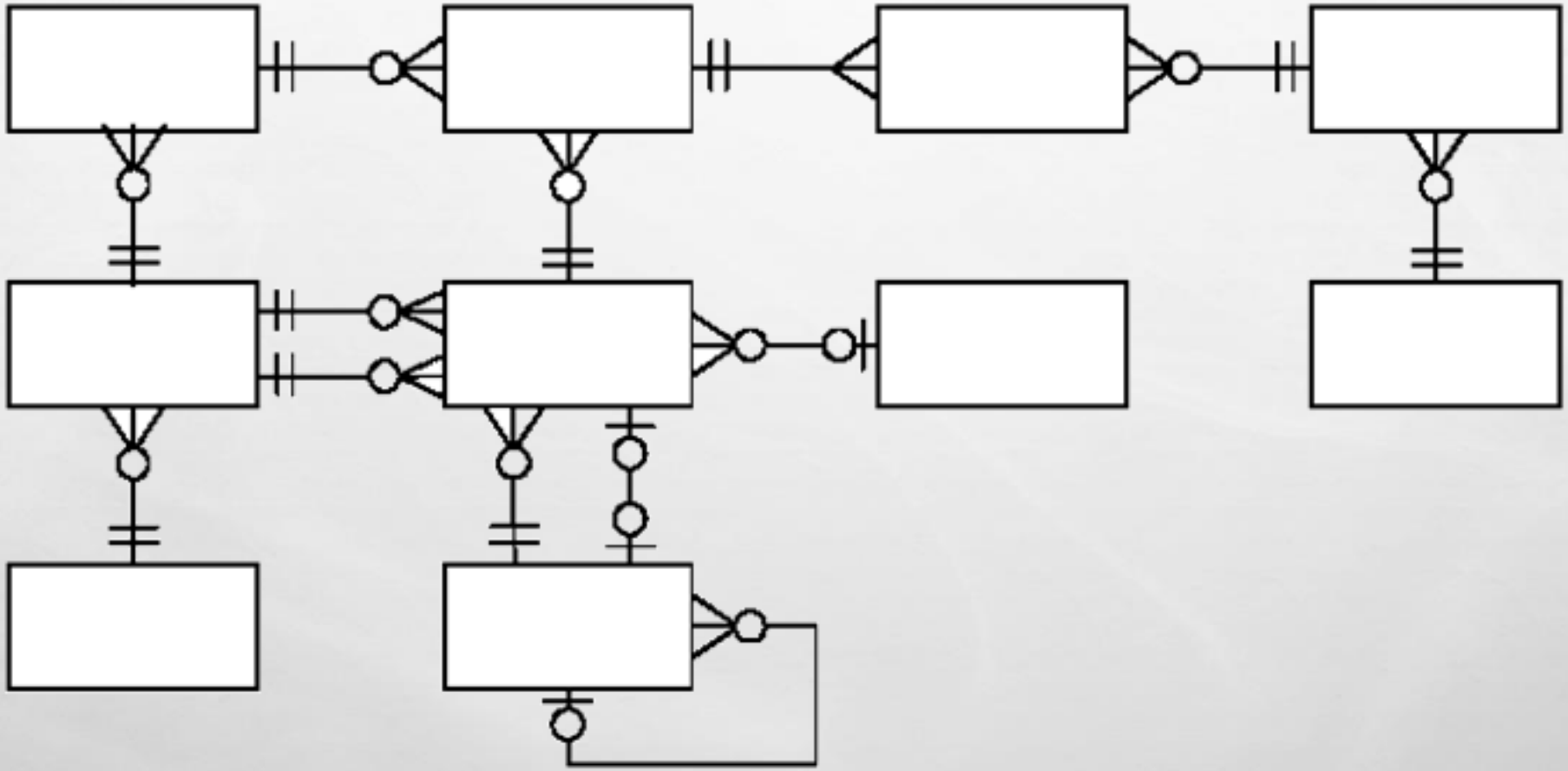
BANCO DE DADOS

MODELO RELACIONAL

- APRESENTAÇÃO SQL
- APRESENTAÇÃO DA DDL; E
- DOMÍNIOS NO SQL-SERVER

SQL - STRUCTURED QUERY LANGUAGE

DDL



STRUCTURED QUERY LANGUAGE - SQL

HISTÓRICO

- **1969** OS AMERICANOS CHEGAM À LUA GRAÇAS AOS MAINFRAMES DA IBM E DO **IMS**, SISTEMA GERENCIADOR DE BANCO DE DADOS QUE UTILIZAVA O MODELO HIERARQUICO. O IMS FOI UTILIZADO PARA CATALOGAR AS 3 MILHÕES DE PEÇAS UTILIZADAS NA CONSTRUÇÃO DO FOGUETE **SATURN V (MOON ROCKET).**
- **1970** EDGAR FRANK CODD PUBLICA "**A RELATIONAL MODEL OF DATA LARGE SHARED BANKS**" COM OS PRINCÍPIOS PARA O MODELO RELACIONAL.

STRUCTURED QUERY LANGUAGE - SQL

HISTÓRICO

- 1974 DONALD CHAMBERLIN E RAYMOND BOYCE DEFINEM A **SEQUEL** (**S**TRUCTURED ENGLISH **Q**UERY **L**ANGUAGE) NO IBM SAN JOSE RESEARCH CENTER".
- 1975 PRIMEIRO PROTÓTIPO IBM (SEQUEL-XRM).
- 1976 REVISÃO DE SEQUEL P/ SEQUEL/2 (POSTERIORMENTE, SQL).
- 1977 **SYSTEM R,** IMPLEMENTANDO SEQUEL/2, TORNA-SE OPERACIONAL.
  - O **SYSTEM R** FOI UM PROJETO DE PESQUISA DA IBM NOS ANOS 1970 QUE RESULTOU NO DESENVOLVIMENTO DE UM SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS RELACIONAL (SGBDR). ELE FOI CRIADO POR UM GRUPO DE PESQUISADORES LIDERADOS POR E. F. CODD

STRUCTURED QUERY LANGUAGE - SQL

HISTÓRICO

- **1975**, TRABALHANDO COMO PROGRAMADOR NA AMPEX, ELISSON É ALOCADO EM UM PROJETO PARA A **CIA - CENTRAL INTELLIGENCE AGENCY**. O PROJETO TINHA POR OBJETIVO O DESENVOLVIMENTO DE UM BANCO DE DADOS, CUJO CODINOME ERA: **ORACLE**.
- **1977 LARRY ELISSON**, FASCINADO PELO DO ARTIGO DE EDGAR F. CODD, INICIA O DESENVOLVIMENTO BASEADO NOS CONCEITOS RELATADOS NO ARTIGO;
- **1977** CIENTE DE QUE TINHA UMA GRANDE OPORTUNIDADE, FUNDA A EMPRESA SOFTWARE DEVELOPMENT LABORATORIES (**SDL**) E INICIA O DESENVOLVIMENTO DE UM SOFTWARE QUE INCORPORASSE CONCEITOS MATEMÁTICOS DESCRITOS NO ARTIGO DE EDGAR F. CODD;

STRUCTURED QUERY LANGUAGE - SQL

HISTÓRICO

- **1978** A SDL CRIA A VERSÃO 1 DO ORACLE. ESCRITA EM ASSEMBLY, LIMITADO A 128 KiB DE MEMÓRIA;
- **1979** A SDL ALTERA O NOME PARA RELATIONAL SOFTWARE, INC (RSI) E LANÇA A VERSÃO 2 DO ORACLE, SENDO ASSIM O PRIMEIRO SGBDR **COMERCIAL** DO NO MERCADO;
- **1982** A RSI ALTERA O NOME PARA ORACLE SYSTEMS CORPORATION;

STRUCTURED QUERY LANGUAGE - SQL

HISTÓRICO

- **1983** IBM LANÇA DB2. OUTROS PRODUTOS RELACIONAIS LANÇADOS NO MERCADO (**SYBASE**, INGRES, ETC). SQL É UM PADRÃO "DE FATO";
- **1986 A SQL** TORNA-SE UM PADRÃO **ANSI – AMERICAN NATIONAL STANDARDS INSTITUTE** PARA LINGUAGEM RELACIONAL;
- **1987** PADRÃO **ANSI** PARA **SQL** É ACEITO PELA **ISO** (SQL/86);
- **1989 SQL** INCORPORA CARACTERÍSTICAS DE REFORÇO DE INTEGRIDADE (SQL/89);

STRUCTURED QUERY LANGUAGE - SQL

HISTÓRICO

- **1992** COMITÊS ISO E ANSI APRESENTAM SQL2 (SQL/92).
    - É INTRODUZIDO FORMALMENTE OS OPERADORES DE JUNÇÃO (JOINS) DE TABELAS COMO O INNER, LEFT OUTER, RIGHT OUTER, FULL OUTER E CROSS.
- **1999** SQL 3 – (ORIENTADO A OBJETO)

STRUCTURED QUERY LANGUAGE - SQL

HISTÓRICO

- O SQL FOI REVISTO NOVAMENTE EM 1999 E 2003 PARA SE TORNAR SQL: 1999 (SQL3) E SQL:2003, RESPECTIVAMENTE.
- O SQL: 1999 USA EXPRESSÕES REGULARES DE EMPARELHAMENTO, QUERIES RECURSIVAS E GATILHOS (TRIGGERS).
- TAMBÉM FOI FEITA UMA ADIÇÃO DE TIPOS NÃO-ESCALADOS E ALGUMAS CARACTERÍSTICAS DE ORIENTAÇÃO A OBJETO.

STRUCTURED QUERY LANGUAGE - SQL

HISTÓRICO

- O SQL:2003 INTRODUZ CARACTERÍSTICAS RELACIONADAS AO XML, SEQUÊNCIAS PADRONIZADAS E COLUNAS COM VALORES DE AUTO GENERALIZAÇÃO (INCLUSIVE COLUNAS-IDENTIDADE).

- SQL: 2008 É A SEXTA REVISÃO DO ISO E ANSI, PADRÃO PARA A SQL . ESSA VERSÃO FOI FORMALMENTE ADOTADA EM JULHO DE 2008.

STRUCTURED QUERY LANGUAGE - SQL

HISTÓRICO

- SQL: 2011 OU ISO / IEC 9075:2011 É A SÉTIMA REVISÃO DA ISO (1987) E ANSI (1986) PADRÃO PARA O SQL. ESSA VERSÃO FOI FORMALMENTE ADOTADA EM DEZEMBRO DE 2011.
  - UMA DAS PRINCIPAIS NOVAS CARACTERÍSTICAS FOI A MELHORIA NO SUPORTE A BANCO DE DADOS TEMPORAL
  - A IBM, NA VERSÃO 10 DO DB2 , DECLARA QUE O DB2 É O PRIMEIRO SGBD EM CONFORMIDADE COM ESSA FUNCIONALIDADE.
  - A ORACLE, NA VERSÃO 10 E SUPERIORES, TEM FUNCIONALIDADE SEMELHANTE.

STRUCTURED QUERY LANGUAGE - SQL

HISTÓRICO

- **SQL: 2019** ÚLTIMA VERSÃO PADRONIZADA PELA ANSI/ISO EM JUL/2019 (9075-15:2019)

STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- ATRAVÉS DE UMA **ARQUITETURA** CHAMADA **ODBC** (OPEN DATA BASE CONNECTIVITY), CRIADA PELO CONSÓRCIO **SQL-ACCESS GROUP** (**HOJE, X/OPEN**) É POSSÍVEL CONVERTER A SINTAXE SQL DE UM PRODUTO PARA OUTRO.
- É UMA LINGUAGEM NÃO PROCEDURAL QUE REQUER DO USUÁRIO QUAL DADO É NECESSÁRIO SEM ESPECIFICAR COMO OBTÊ-LO (**PARADIGMA DECLARATIVO**);

STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- POUPA TEMPO DE PROGRAMAÇÃO MAS, EXIGE TEMPO PARA O PROJETO;
- **LINGUAGEM INTERATIVA DE CONSULTA:**
  - PERMITE CONSULTAS AO BANCO DE DADOS SEM NECESSIDADE DE PROGRAMAS;
- **LINGUAGEM DE PROGRAMAÇÃO:**
  - COMANDOS SQL EMBUTIDOS EM PROGRAMAS DE APLICAÇÃO ATRAVÉS DA UTILIZAÇÃO DE **API'S / DRIVER'S**;

STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- **LINGUAGEM DE ADMINISTRAÇÃO:**
  - TAREFAS DE ADMINISTRAÇÃO DO BANCO DE DADOS PODEM SER FEITAS UTILIZANDO O SQL;
- **LINGUAGEM CLIENTE SERVIDOR:**
  - CLIENTES SE COMUNICANDO COM O SERVIDOR ATRAVÉS DE COMANDOS SQL;
- **LINGUAGEM PARA BANCO DE DADOS DISTRIBUÍDOS:**
  - AUXILIA NA CONVERSÃO DE DIVERSOS PRODUTOS DE BANCO DE DADOS COLOCADOS EM DIFERENTES MÁQUINAS E PLATAFORMAS

STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- **DEFINIÇÃO DE DADOS** (**DDL**):
  - ATRAVÉS DA **DDL** É POSSÍVEL CRIAR E MANIPULAR OS OBJETOS DE DADOS – COMO: O BANCO DE DADOS, TABELAS, ÍNDICES, VIEWS, ETC -, E DEFINIR AS RELAÇÕES EXISTENTE ENTRE ELES.
- **MANIPULAÇÃO DE DADOS** (**DML**):
  - ATRAVÉS DA **DML** É POSSÍVEL A MANIPULAÇÃO DOS DADOS ARMAZENADOS, CONSISTINDO NA RECUPERAÇÃO, INCLUSÃO, EXCLUSÃO E ALTERAÇÃO DOS DADOS;

STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- **CONTROLE DE ACESSO** (**DCL**):
  - PROTEGE OS DADOS DE MANIPULAÇÃO NÃO AUTORIZADAS, ATRAVÉS DE APLICAÇÃO DE POLITICAS DE ACESSO UTILIZANDO COMANDOS ESPECIFICOS;
- **LINGUAGEM DE TRANSAÇÃO DE DADOS** (**DTL**):
  - ATRAVÉS DA **DTL** É POSSÍVEL A DEFINIÇÃO DA ESTRUTURA DE CONTROLE DAS TRANSAÇÕES, UTILIZANDO OS ALGORITMOS E PROTOCOLOS DE CONTROLE DE CONCORRÊNCIA;

STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- **LINGUAGEM DE CONSULTA DE DADOS** (**DQL**)
  - EMBORA TENHA APENAS UM COMANDO, A **DQL** É A PARTE DA SQL MAIS UTILIZADA.
  - O COMANDO **SELECT** PERMITE AO USUÁRIO ESPECIFICAR UMA CONSULTA ("QUERY") COMO UMA DESCRIÇÃO DO RESULTADO DESEJADO.

STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- **COMPARTILHAMENTO DE DADOS**:
  - COORDENA O COMPARTILHAMENTO DOS DADOS ENTRE USUÁRIOS
- **INTEGRIDADE DE DADOS**:
  - DEFINE A INTEGRIDADE DOS DADOS CONTRA CORRUPÇÕES, INCONSISTÊNCIA E FALHAS DO SISTEMA
- **CONTROLE DA TRANSAÇÃO**:
  - INCLUEM COMANDOS QUE CONTROLAM A ESPECIFICAÇÃO DO INÍCIO E FIM DAS TRANSAÇÕES.
- **INDEPENDÊNCIA DE FABRICANTE**:
  - ESTÁ INCORPORADO EM QUASE TODOS OS SGBD EM SEU PADRÃO ANSI/ISO, COM EXTENSÃO PROPRIETÁRIA DE CADA FABRICANTE

STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- **COMPARTILHAMENTO DE DADOS**:
  - COORDENA O COMPARTILHAMENTO DOS DADOS ENTRE USUÁRIOS
- **INTEGRIDADE DE DADOS**:
  - DEFINE A INTEGRIDADE DOS DADOS CONTRA CORRUPÇÕES, INCONSISTÊNCIA E FALHAS DO SISTEMA

STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- **PORTABILIDADE ENTRE COMPUTADORES**:
  - PODE SER USADA DESDE UM PDA/SMARTPHONE ATÉ EM MAINFRAMES
- **REDUÇÃO DE CUSTO NO TREINAMENTO**:
  - AS APLICAÇÕES PODEM MUDAR DE AMBIENTE COM BAIXO CUSTO DE TREINAMENTO

STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- **FACILIDADE NO ENTENDIMENTO**:
  - OFERECE RÁPIDO ENTENDIMENTO COM COMANDO ESCRITO EM UM INGLÊS ESTRUTURADO DE ALTO NÍVEL;
  - BAIXA CURVA DE APRENDIZAGEM;
- **MÚLTIPLAS VISÕES DOS DADOS**:
  - POSSIBILITA LEVAR DIFERENTES VISÕES DOS DADOS A DIFERENTES USUÁRIOS;

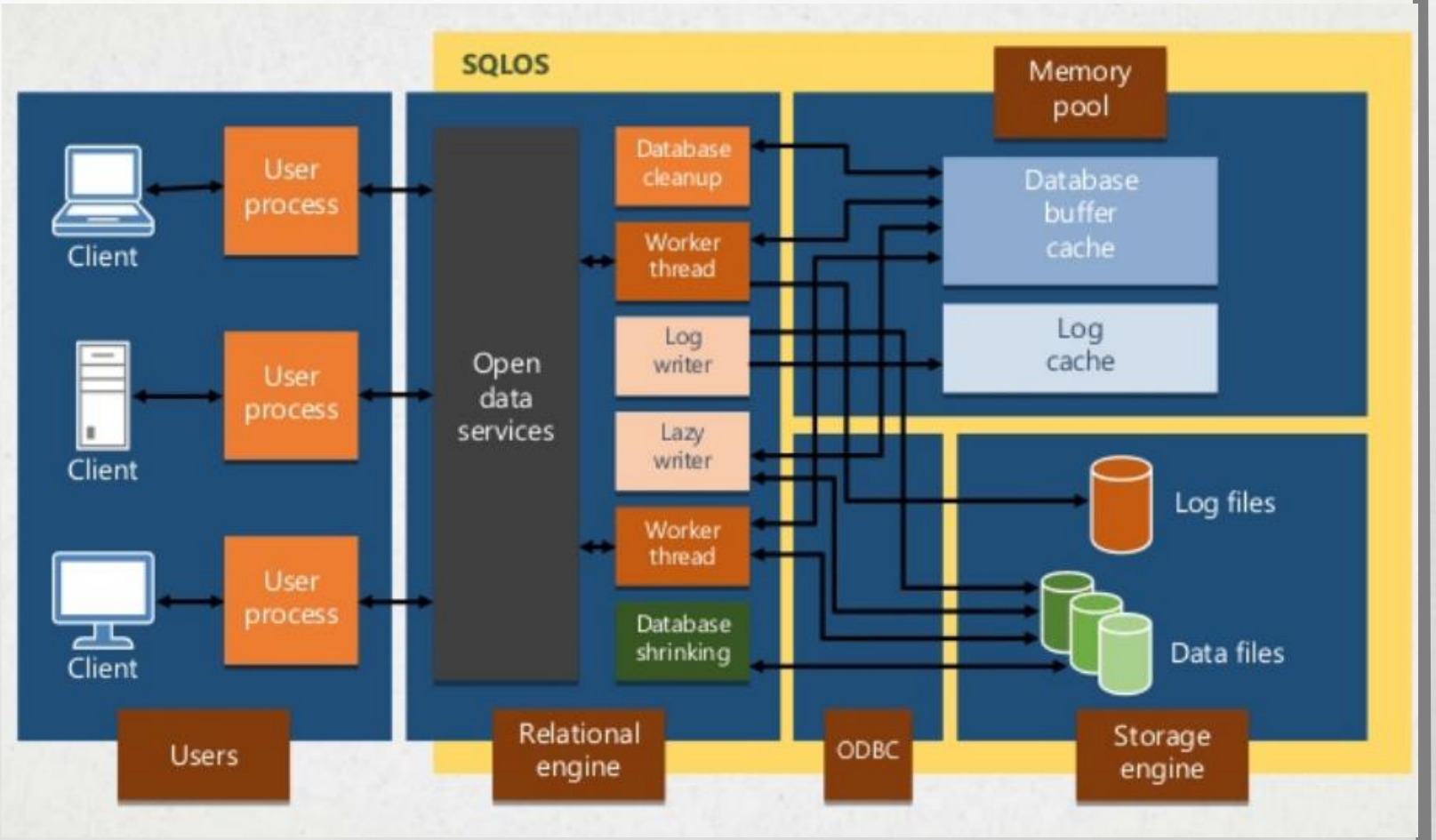
STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- **ARQUITETURA ARMAZENAMENTO DO SQL-SERVER**

SQL-SERVER DATABASE ARCHITECTURE

- ARQUITETURA SIMPLIFICADA DO SQL-SERVER



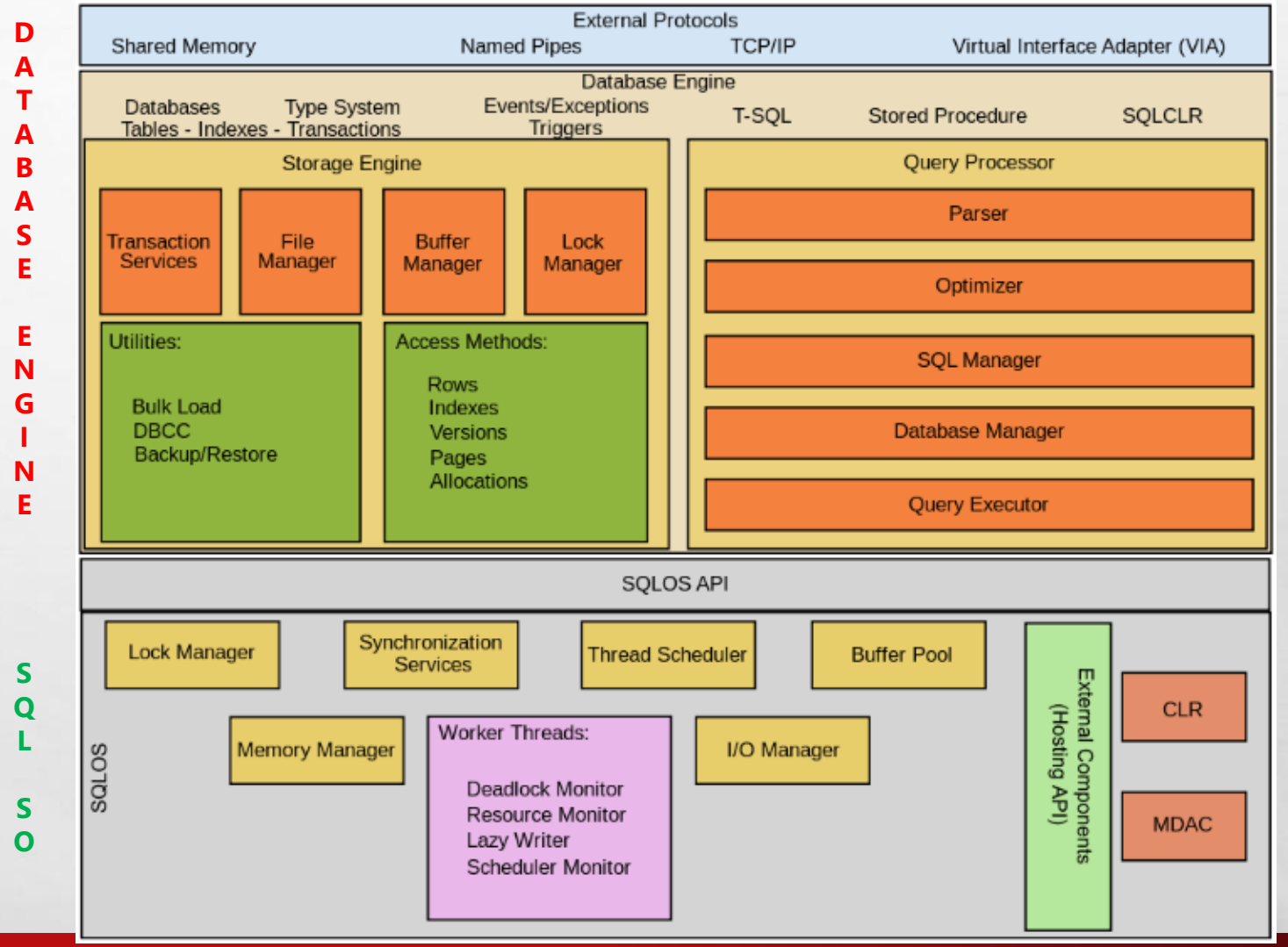
STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- **O QUE É UMA INSTÂNCIA DE BANCO DE DADOS?**
  - UMA **INSTÂNCIA DE BANCO DE DADOS** É UMA CÓPIA AUTÔNOMA E ISOLADA DO MECANISMO DE BANCO DE DADOS DO SQL SERVER, QUE PODE SER INSTALADA EM UM COMPUTADOR E GERENCIADA SEPARADAMENTE DE OUTRAS INSTÂNCIAS DO SQL SERVER.
  - CADA INSTÂNCIA DO SQL SERVER TEM SEU PRÓPRIO CONJUNTO DE BANCOS DE DADOS, CONFIGURAÇÕES E RECURSOS DE SISTEMA, E PODE SER ACESSADA POR APLICATIVOS QUE SE CONECTAM A ELA POR MEIO DE UMA REDE OU LOCALMENTE.

SQL-SERVER DATABASE ARCHITECTURE

- ARQUITETURA SIMPLIFICADA DO SQL-SERVER



SQL-SERVER DATABASE ARCHITECTURE

- **ARQUITETURA SIMPLIFICADA DO SQL-SERVER**
  - **CAMADA DATABASE ENGINE**: É RESPONSÁVEL POR GERENCIAR O ARMAZENAMENTO DE DADOS NO SQL SERVER. ISSO INCLUI O GERENCIAMENTO DE ARQUIVOS DE DADOS E DE LOG, CONTROLE DE TRANSAÇÕES, CONTROLE DE CONCORRÊNCIA, PROCESSAMENTO DE CONSULTAS E ACESSO A DADOS.
  - **A CAMADA SQL OS**: É RESPONSÁVEL POR FORNECER SERVIÇOS DE SISTEMA OPERACIONAL PARA O SQL SERVER. ISSO INCLUI GERENCIAMENTO DE MEMÓRIA, GERENCIAMENTO DE ARQUIVOS E E/S, GERENCIAMENTO DE REDE, GERENCIAMENTO DE THREADS, ENTRE OUTROS. ESSES SERVIÇOS SÃO USADOS PELO DATABASE ENGINE PARA EXECUTAR OPERAÇÕES DE BANCO DE DADOS.

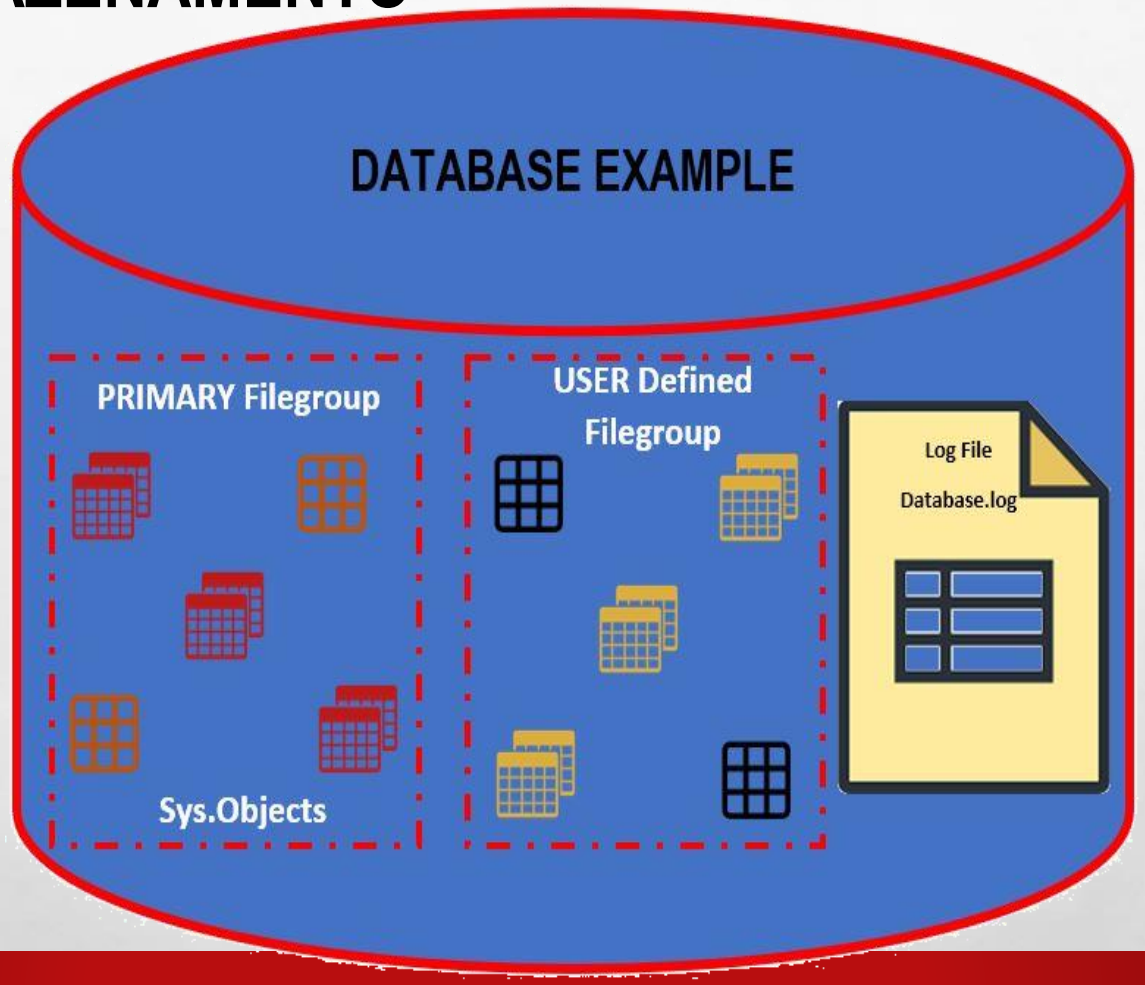
STRUCTURED QUERY LANGUAGE - SQL

CARACTERÍSTICAS

- **ESTRUTURA DE ARMAZENAMENTO DO SQL-SERVER**

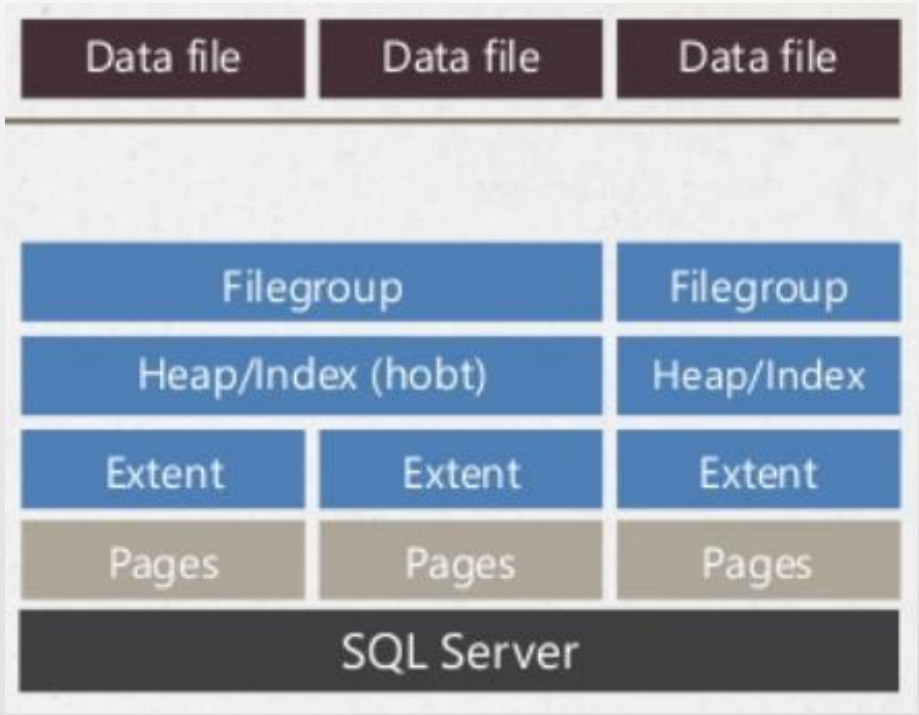
SQL-SERVER DATABASE ARCHITECTURE

- ESTRUTURA DE ARMAZENAMENTO



SQL-SERVER DATABASE ARCHITECTURE

- ESTRUTURA DE ARMAZENAMENTO



SQL-SERVER DATABASE ARCHITECTURE

- **ESTRUTURA DE ARMAZENAMENTO DO SQL-SERVER**

1. **FILEGROUP**: UM FILEGROUP É UM CONJUNTO LÓGICO DE ARQUIVOS DE DADOS QUE ARMAZENAM OBJETOS RELACIONAIS DO BANCO DE DADOS.

   O SQL SERVER PERMITE QUE OS DBAS CRIEM VÁRIOS FILEGROUPS EM UM BANCO DE DADOS, PARA MELHOR GERENCIAMENTO DO ARMAZENAMENTO DE DADOS. CADA TABELA E ÍNDICE PODE SER ATRIBUÍDO A UM FILEGROUP DIFERENTE.

SQL-SERVER DATABASE ARCHITECTURE

- **ESTRUTURA DE ARMAZENAMENTO DO SQL-SERVER**

2. **HEAP E ÍNDICES**: UMA TABELA PODE SER ARMAZENADA COMO UM HEAP, OU SEJA, SEM UM ÍNDICE. ALTERNATIVAMENTE, PODE TER UM OU MAIS ÍNDICES CRIADOS.

UM ÍNDICE É UMA ESTRUTURA DE DADOS QUE ACELERA A CONSULTA DE DADOS NA TABELA, INDEXANDO UMA OU MAIS COLUNAS.

OS ÍNDICES PODEM SER CRIADOS NO MESMO FILEGROUP OU EM FILEGROUPS DIFERENTES, PERMITINDO UMA MELHOR ALOCAÇÃO DE RECURSOS.

SQL-SERVER DATABASE ARCHITECTURE

- **ESTRUTURA DE ARMAZENAMENTO DO SQL-SERVER**

3. **EXTENT**: UMA EXTENT É UM GRUPO CONTÍGUO DE OITO PÁGINAS DE DADOS OU DE LOG.

   AS EXTENSÕES SÃO USADAS PARA ALOCAR ESPAÇO PARA OS OBJETOS RELACIONAIS NO BANCO DE DADOS.

SQL-SERVER DATABASE ARCHITECTURE

- **ESTRUTURA DE ARMAZENAMENTO DO SQL-SERVER**

4. **PÁGINAS**: AS PÁGINAS SÃO A MENOR UNIDADE DE ARMAZENAMENTO NO SQL SERVER, CADA UMA COM UM TAMANHO FIXO DE 8 KB.

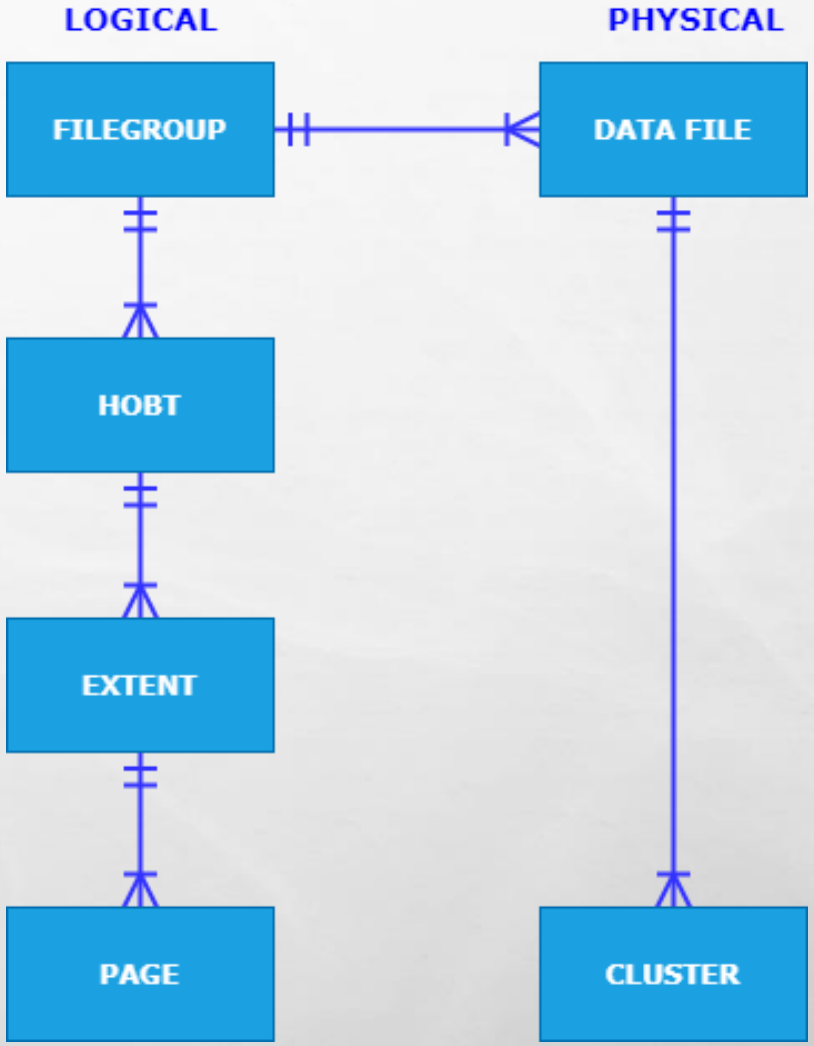
   AS PÁGINAS SÃO USADAS PARA ARMAZENAR DADOS E ÍNDICES EM UM HEAP OU ÍNDICE.

   CADA PÁGINA É IDENTIFICADA POR UM NÚMERO DE PÁGINA LÓGICO E UMA EXTENSÃO FÍSICA.

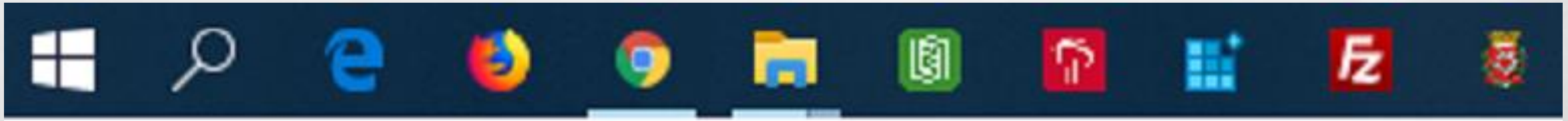
   AS PÁGINAS SÃO ARMAZENADAS NOS ARQUIVOS DE DADOS DO SQL SERVER.

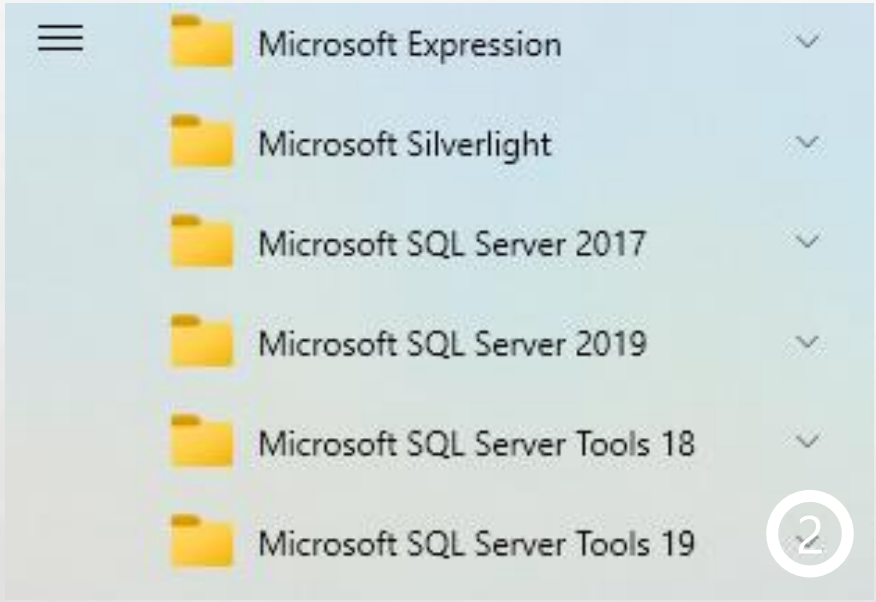
SQL-SERVER DATABASE ARCHITECTURE

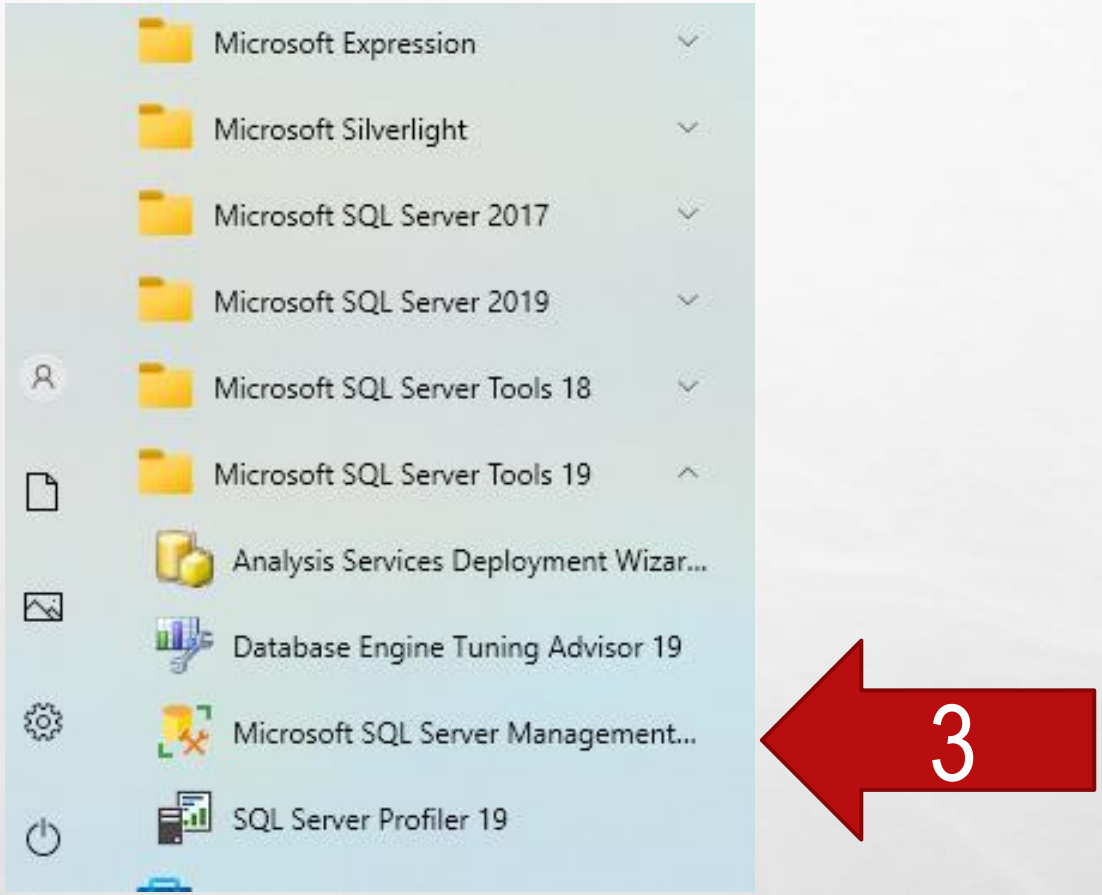
- ESTRUTURAS DE ARMAZENAMENTO LÓGICO



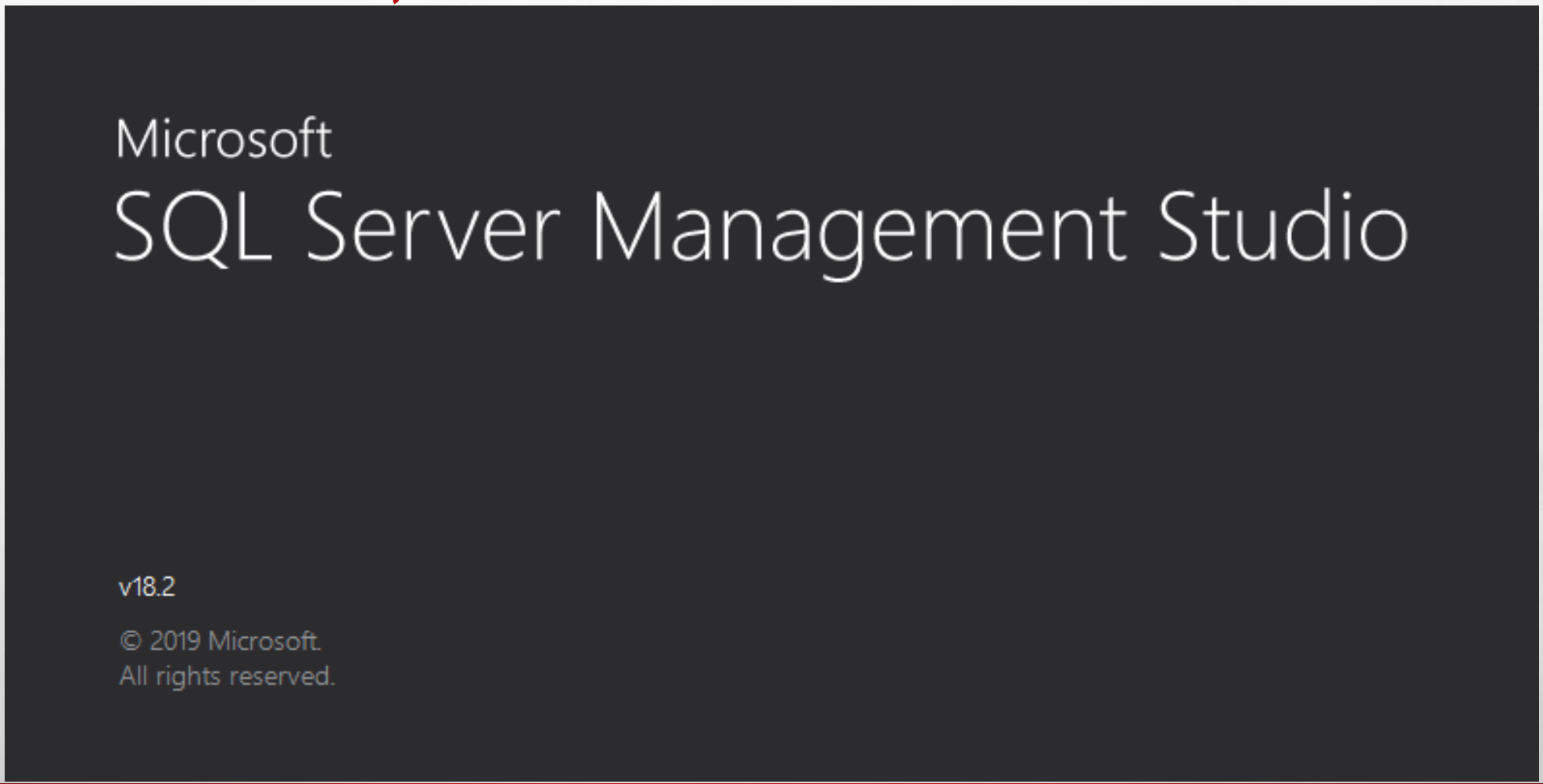
1



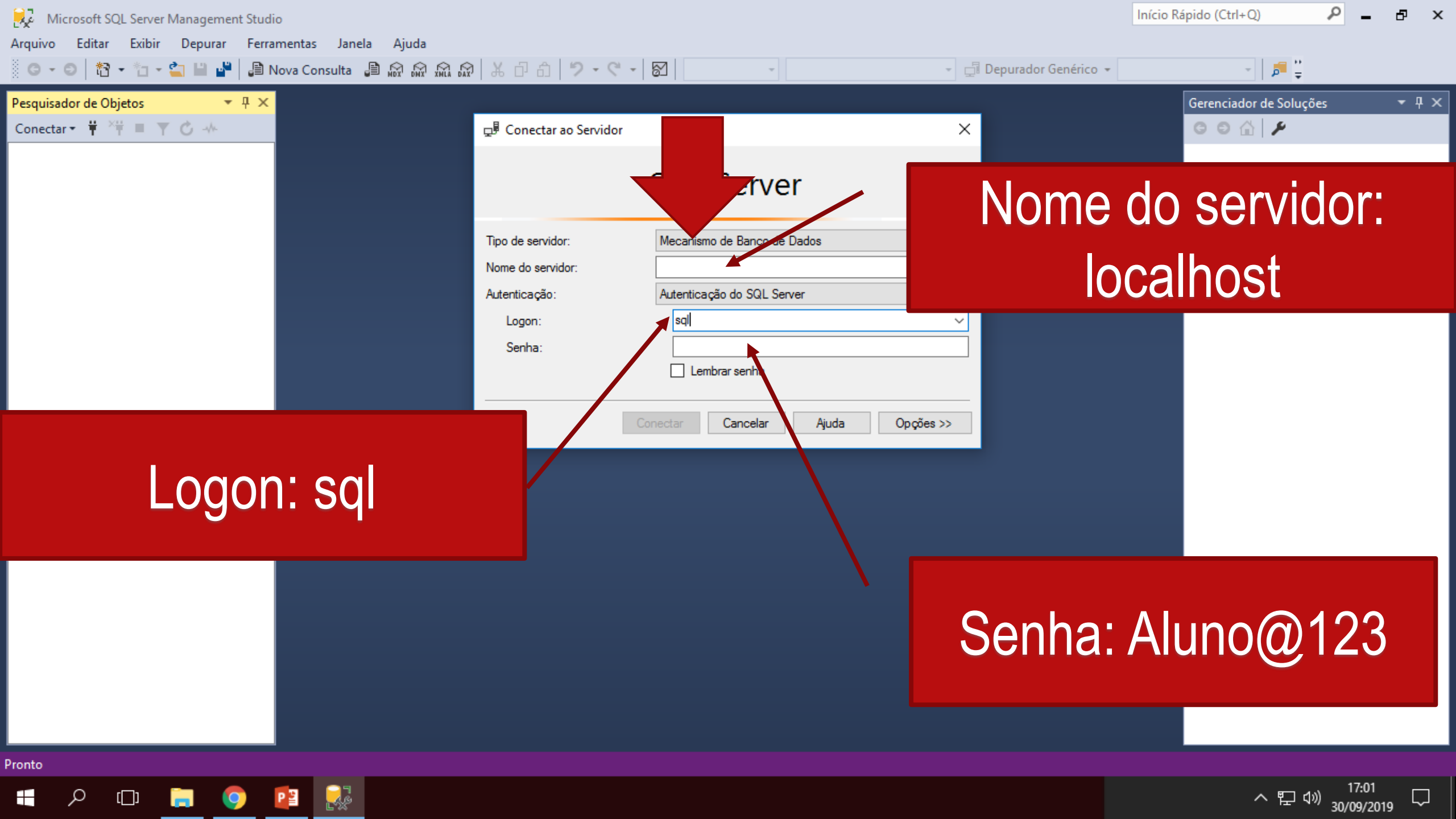

Microsoft Expression
Microsoft Silverlight
Microsoft SQL Server 2017
Microsoft SQL Server 2019
Microsoft SQL Server Tools 18
Microsoft SQL Server Tools 19
2
2


Microsoft Expression
Microsoft Silverlight
Microsoft SQL Server 2017
Microsoft SQL Server 2019
Microsoft SQL Server Tools 18
Microsoft SQL Server Tools 19
Analysis Services Deployment Wizar...
Database Engine Tuning Advisor 19
Microsoft SQL Server Management...
SQL Server Profiler 19
3

4) DURANTE ALGUNS SEGUNDOS A TELA ABAIXO SERA APRESENTADA..;



5) A JANELA LOGIN DO MANAGEMENT STUDIO 17 ou 18 SERÁ APRESENTADA;


Microsoft SQL Server Management Studio
Início Rápido (Ctrl+Q)
Arquivo Editar Exibir Depurar Ferramentas Janela Ajuda
Nova Consulta
Depurador Genérico
Pesquisador de Objetos
Conectar
Gerenciador de Soluções
Conectar ao Servidor
Tipo de servidor:
Mecanismo de Banco de Dados
Nome do servidor:
Autenticação:
Autenticação do SQL Server
Logon:
sql
Senha:
Lembrar senha
Conectar
Cancelar
Ajuda
Opções >>
Nome do servidor: localhost
Logon: sql
Senha: Aluno@123
Pronto
17:01
30/09/2019

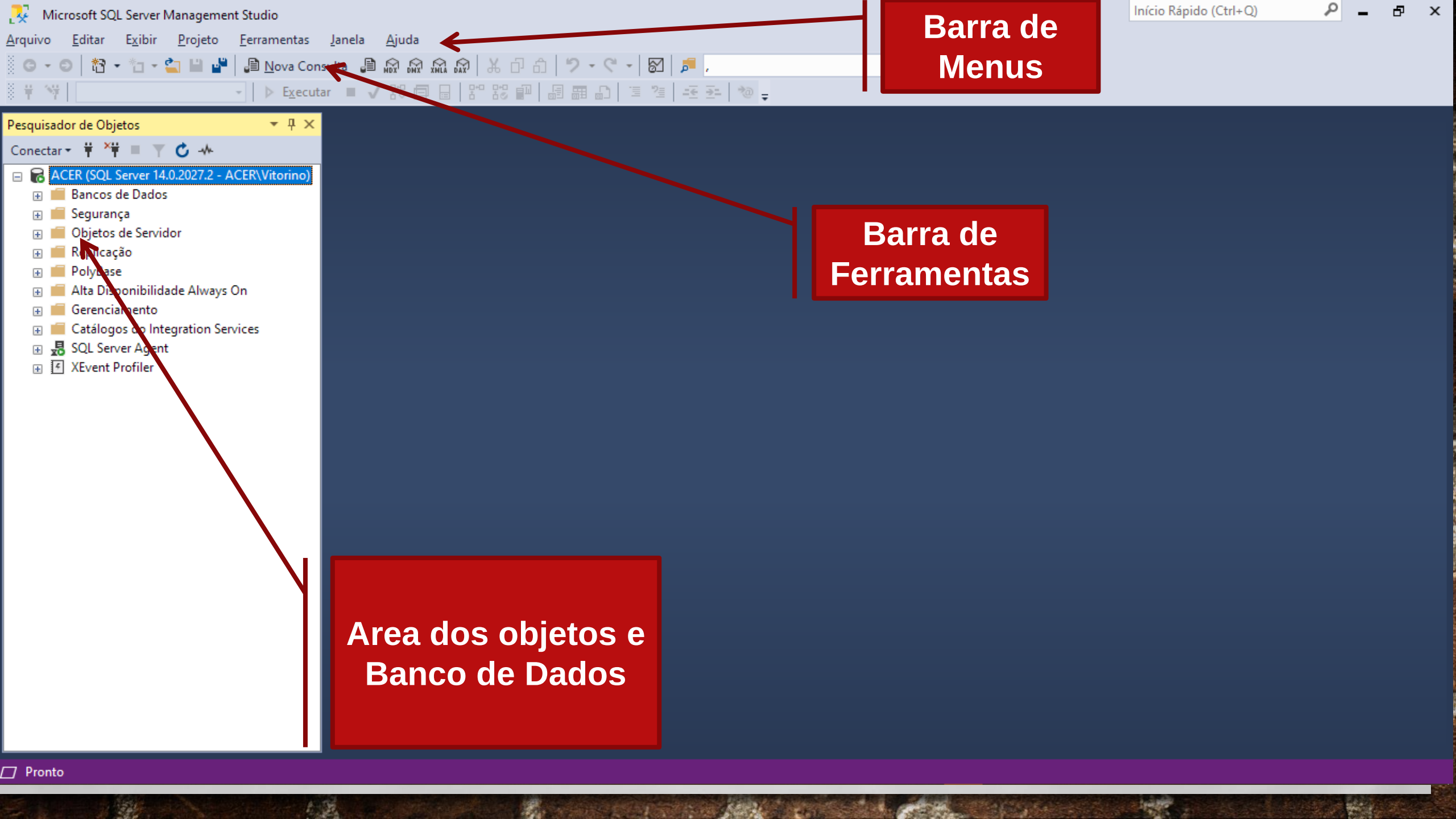
TRANSAÇÕES NO SQL-SERVER 2017

- ROTEIRO

6) Na janela de logo o Tipo de Servidor é:
   a) MECANISMO DE BANCO DE DADOS
7) Nome do Servidor: localhost
8) Login: sql
9) Senha: Aluno@123

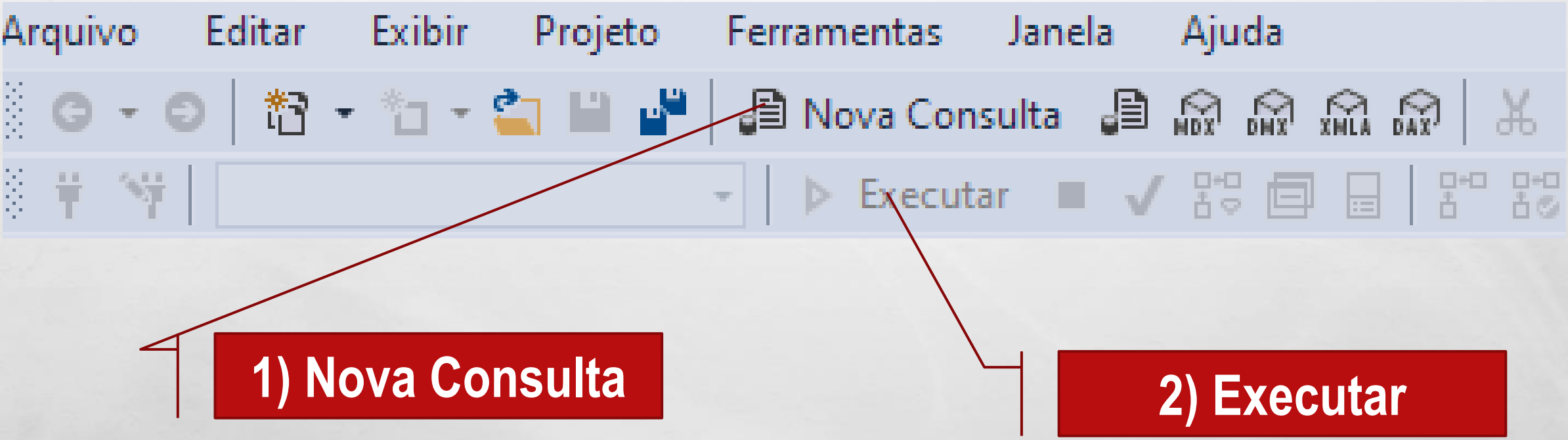
TRANSAÇÕES NO SQL-SERVER 2017

10) COM OS PARÂMETROS CORRETOS DE LOGIN A MANAGEMENT STUDIO 17,18 ou 19 SERÁ APRESENTADA;


Microsoft SQL Server Management Studio
Início Rápido (Ctrl+Q)
Arquivo Editar Exibir Projeto Ferramentas Janela Ajuda
Nova Consulta
Executar
Barra de Menus
Barra de Ferramentas
Pesquisador de Objetos
Conectar
ACER (SQL Server 14.0.2027.2 - ACER\Vitorino)
Bancos de Dados
Segurança
Objetos de Servidor
Replicação
PolyBase
Alta Disponibilidade Always On
Gerenciamento
Catálogos do Integration Services
SQL Server Agent
XEvent Profiler
Area dos objetos e Banco de Dados
Pronto

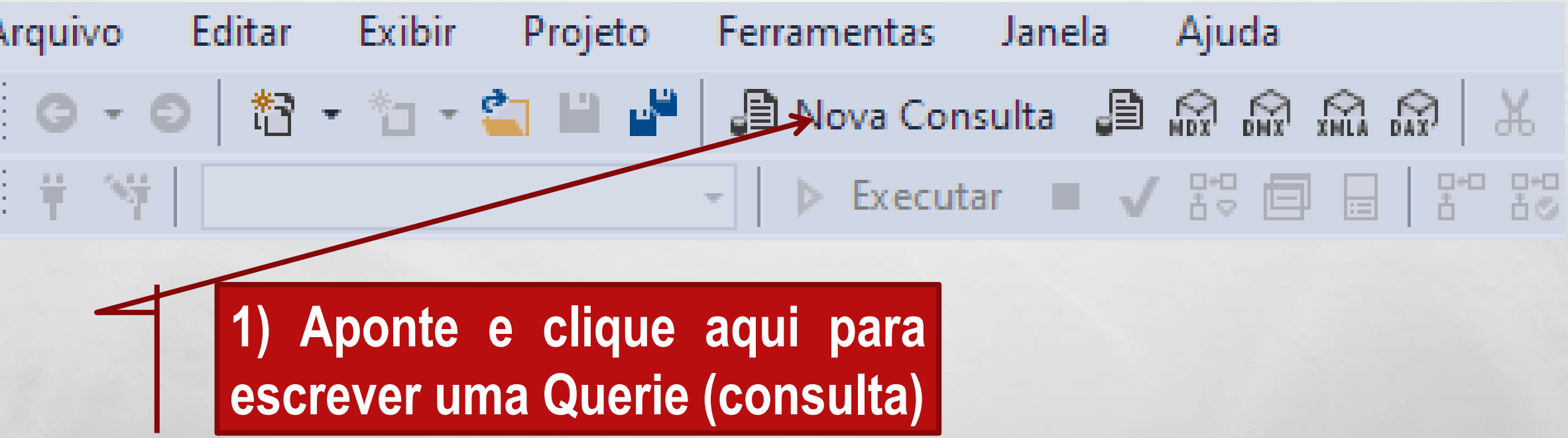
TRANSAÇÕES NO SQL-SERVER 2017

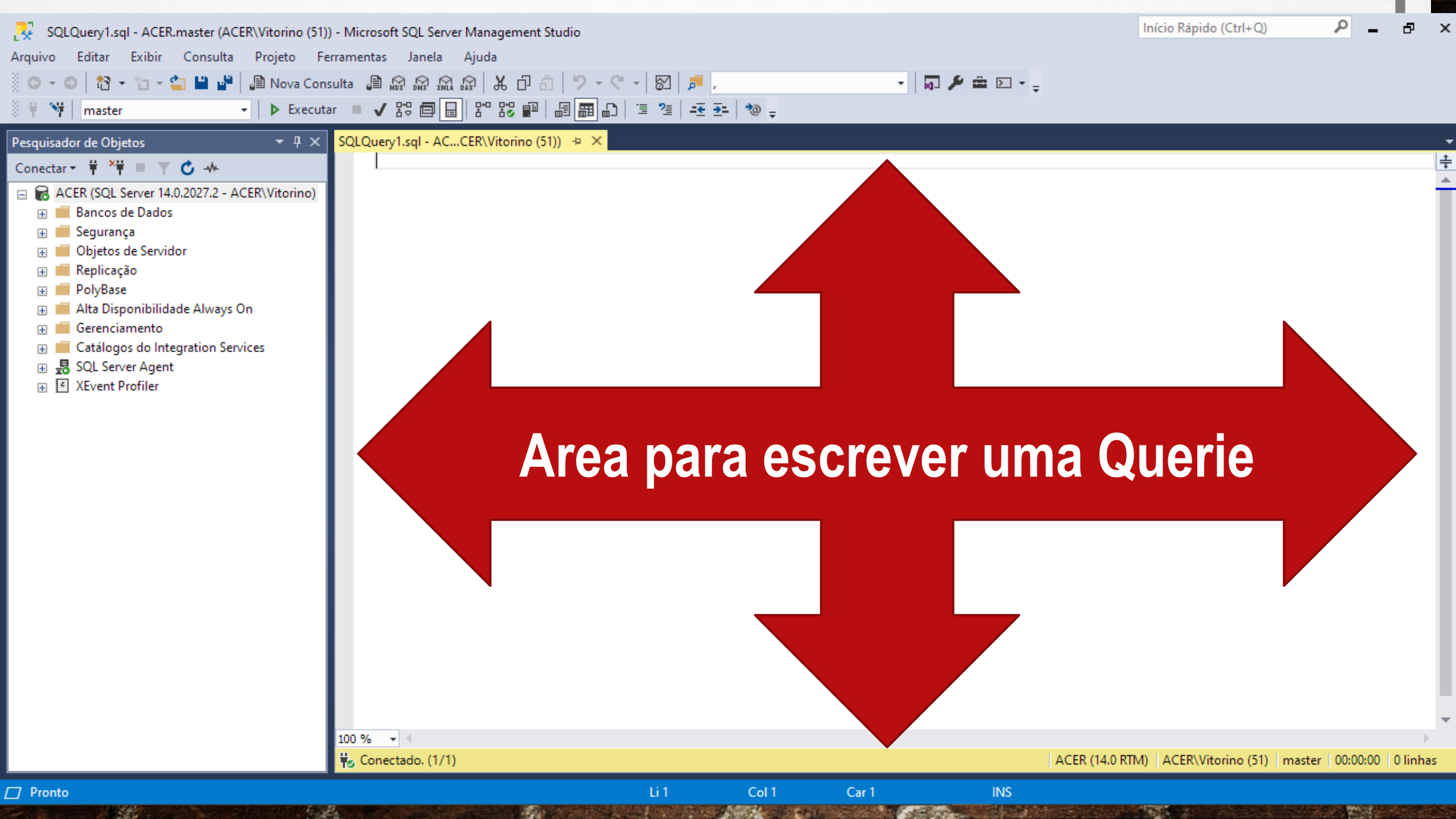
11) NESTA AULA UTILIZAREMOS, BASICAMENTE, DUAS OPÇÕES DA BARRA DE FERRAMENTAS;

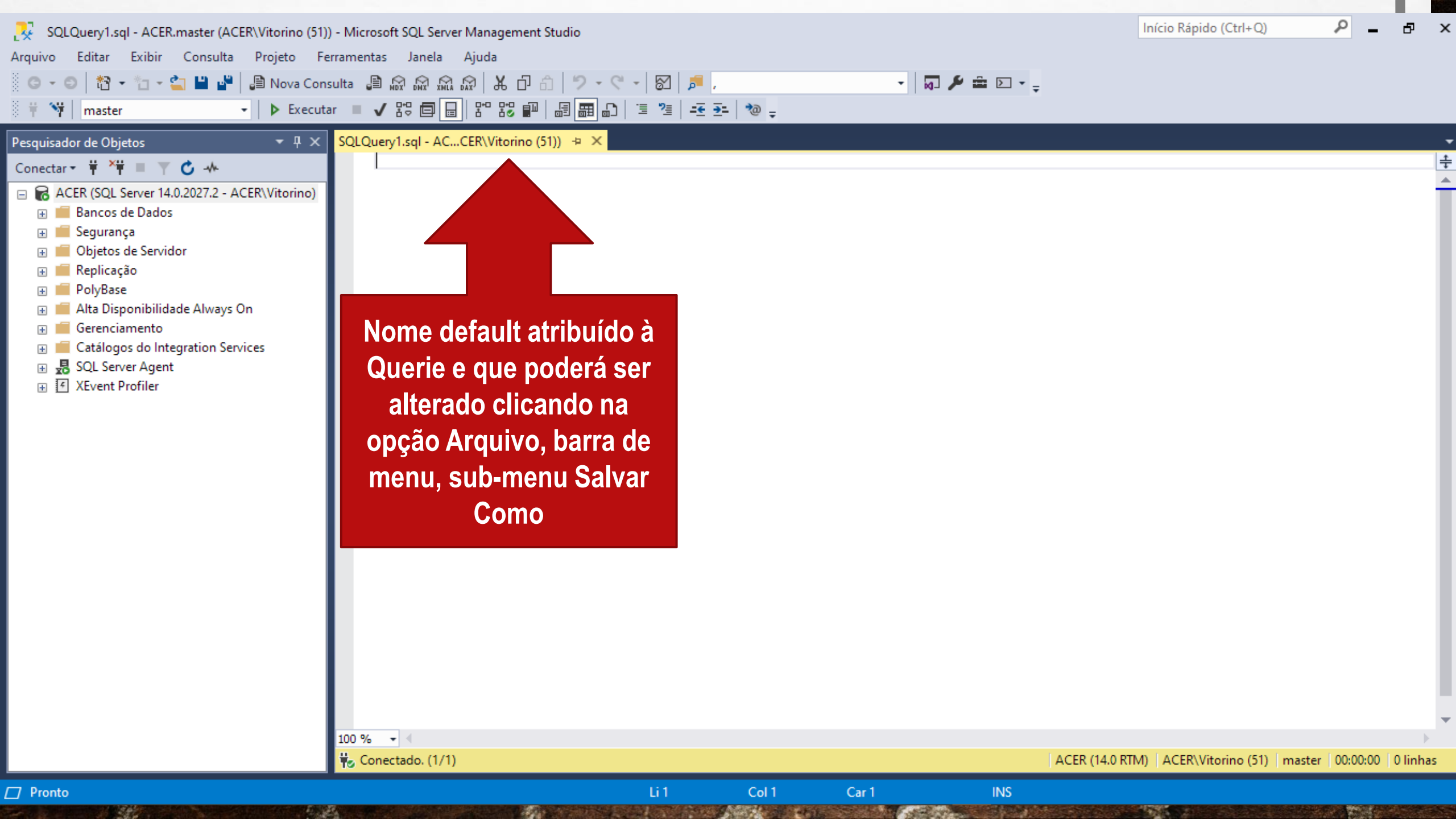


TRANSAÇÕES NO SQL-SERVER 2017

12) Para escrevermos uma querie (consulta), basta apontarmos e clicarmos sobre a opção; Nova Consulta




SQLQuery1.sql - ACER.master (ACER\Vitorino (51)) - Microsoft SQL Server Management Studio
Início Rápido (Ctrl+Q)
Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda
Nova Consulta
master
Executar
Pesquisador de Objetos
Conectar
ACER (SQL Server 14.0.2027.2 - ACER\Vitorino)
Bancos de Dados
Segurança
Objetos de Servidor
Replicação
PolyBase
Alta Disponibilidade Always On
Gerenciamento
Catálogos do Integration Services
SQL Server Agent
XEvent Profiler
SQLQuery1.sql - AC...CER\Vitorino (51))
Area para escrever uma Querie
100 %
Conectado. (1/1)
ACER (14.0 RTM)
ACER\Vitorino (51)
master
00:00:00
0 linhas
Pronto
Li 1
Col 1
Car 1
INS


SQLQuery1.sql - ACER.master (ACER\Vitorino (51)) - Microsoft SQL Server Management Studio
Início Rápido (Ctrl+Q)
Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda
Nova Consulta
master
Executar
Pesquisador de Objetos
Conectar
ACER (SQL Server 14.0.2027.2 - ACER\Vitorino)
Bancos de Dados
Segurança
Objetos de Servidor
Replicação
PolyBase
Alta Disponibilidade Always On
Gerenciamento
Catálogos do Integration Services
SQL Server Agent
XEvent Profiler
SQLQuery1.sql - AC...CER\Vitorino (51))
Nome default atribuído à Querie e que poderá ser alterado clicando na opção Arquivo, barra de menu, sub-menu Salvar Como
100 %
Conectado. (1/1)
ACER (14.0 RTM)
ACER\Vitorino (51)
master
00:00:00
0 linhas
Pronto
Li 1
Col 1
Car 1
INS

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS (DDL):

- **ELEMENTOS DO SQL-SERVER (BANCO DE DADOS DO SISTEMA)**

1. **MASTER DB**: O BANCO DE DADOS MASTER CONTÉM AS INFORMAÇÕES DE CONFIGURAÇÃO DE NÍVEL DE SERVIDOR, COMO LOGINS, ENDPOINTS E SERVIDORES VINCULADOS. O DBA PODE CONFIGURAR E GERENCIAR ESSAS INFORMAÇÕES, ALÉM DE MODIFICAR OUTRAS CONFIGURAÇÕES DE NÍVEL DE SERVIDOR.
2. **MODEL DB**: O BANCO DE DADOS MODEL É USADO COMO UM MODELO PARA CRIAR NOVOS BANCOS DE DADOS NO SERVIDOR. O DBA PODE MODIFICAR AS CONFIGURAÇÕES PADRÃO DO MODEL DB PARA REFLETIR AS NECESSIDADES ESPECÍFICAS DA ORGANIZAÇÃO.

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS (DDL):

- **ELEMENTOS DO SQL-SERVER (BANCO DE DADOS DO SISTEMA)**

1. **MSDB**: O BANCO DE DADOS MSDB É USADO PELO SQL SERVER AGENT PARA AGENDAR TRABALHOS, MANTER O HISTÓRICO DE TRABALHOS E ARMAZENAR ALERTAS E OPERADORES. O DBA PODE CONFIGURAR E GERENCIAR TRABALHOS, ALERTAS E OPERADORES USANDO O SQL SERVER MANAGEMENT STUDIO OU O TRANSACT-SQL.
2. **RESOURCE DB**: O BANCO DE DADOS RESOURCE É SOMENTE LEITURA E CONTÉM OBJETOS DO SISTEMA NECESSÁRIOS PELO SQL SERVER. NÃO PODE SER MODIFICADO PELO DBA.

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS (DDL):

- **ELEMENTOS DO SQL-SERVER (BANCO DE DADOS DO SISTEMA)**

1. **TEMPDB**: O BANCO DE DADOS TEMPDB É USADO PARA ARMAZENAR OBJETOS TEMPORÁRIOS, COMO TABELAS TEMPORÁRIAS E PROCEDIMENTOS ARMAZENADOS TEMPORÁRIOS.

   O DBA PODE CONFIGURAR O TAMANHO E AS OPÇÕES DE INICIALIZAÇÃO DO TEMPDB PARA OTIMIZAR O DESEMPENHO DO SQL SERVER.

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS (DDL):

- **DDL** – DATA DEFINITION LANGUAGE (LINGUAGEM DE DEFINIÇÃO DE DADOS)

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS (DDL):

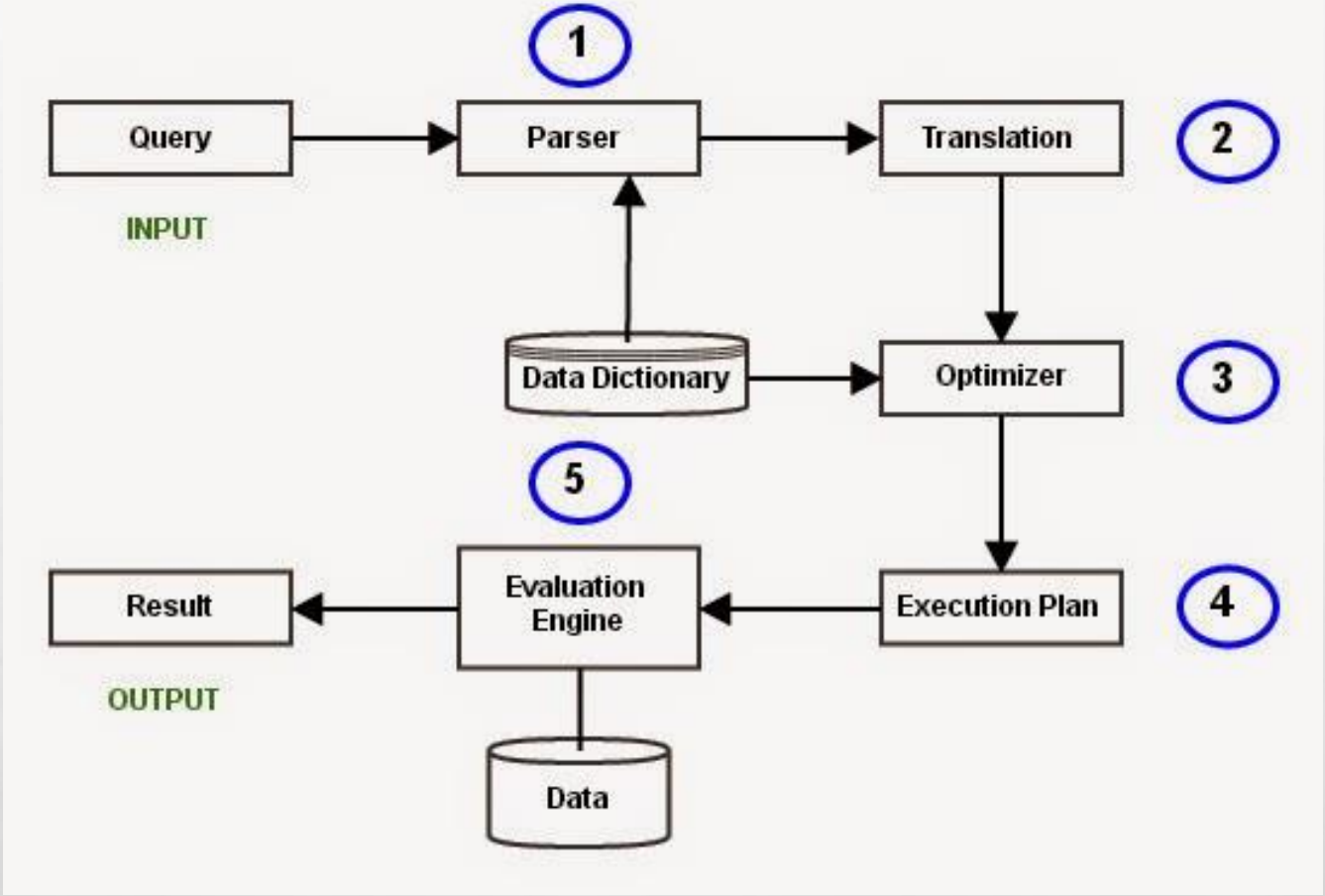
- **CONCEITOS**
  - SCHEMA
  - CATALOG
- **COMANDOS**
  - CREATE
  - ALTER
  - DROP
  - TRUNCATE

CATÁLOGO E/OU DICIONÁRIO DE DADOS

- **IMPORTÂNCIA PARA A VIDA DO DBA**
  - O CORAÇÃO DE QUALQUER SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS RELACIONAL É O **CATÁLOGO DO SISTEMA** QUE DOCUMENTA OS OBJETOS DE BANCO DE DADOS E AS CONFIGURAÇÕES DO SISTEMA QUE ESTÃO SENDO USADOS;
  - O CATÁLOGO DO SISTEMA OFERECE VÁRIAS INFORMAÇÕES SOBRE O SEU SGBD. VOCÊ PODE PENSAR NISSO COMO A BASE DE CONHECIMENTO DE TODOS OS DADOS CONHECIDOS PELO SISTEMA. POR ESSE MOTIVO, É IMPORTANTE QUE OS **DBAS** ENTENDAM O QUE ESTÁ NO CATÁLOGO DO SISTEMA, ALÉM DE COMO ACESSAR E MANIPULAR AS INFORMAÇÕES NELE CONTIDAS.

CATÁLOGO E/OU DICIONÁRIO DE DADOS

- IMPORTÂNCIA PARA A VIDA DO DBA



STRUCTURED QUERY LANGUAGE - SQL

CRIAÇÃO DO BANCO DE DADOS (DATABASE):

- **SINTAXE**

  - **CREATE DATABASE [NOME DO BANCO DE DADOS]**

  - **CREATE DATABASE BancoDadosFMU20212**

SQL-SERVER DATABASE ARCHITECTURE

- **ESTRUTURAS DE ARMAZENAMENTO FÍSICO**
  - **SCHEMA (ESQUEMA) SQL-SERVER**
    - UM SCHEMA (ESQUEMA) DE BANCO DE DADOS É UMA MANEIRA DE AGRUPAR OBJETOS LOGICAMENTE, COMO TABELAS, VIEWS, STORED PROCEDURE, ETC. PENSE EM UM ESQUEMA COMO UM CONTÊINER DE OBJETOS
    - NO SQL SERVER, UM SCHEMA (ESQUEMA) DE BANCO DE DADOS PERMITE A SEGREGAÇÃO DE FUNÇÕES E FACILITA O GERENCIAMENTO DE SEGURANÇA, AUXILIANDO NA DEFINIÇÃO DE QUEM PODE ACESSAR CADA OBJETO DE BANCO DE DADOS.

SQL-SERVER DATABASE ARCHITECTURE

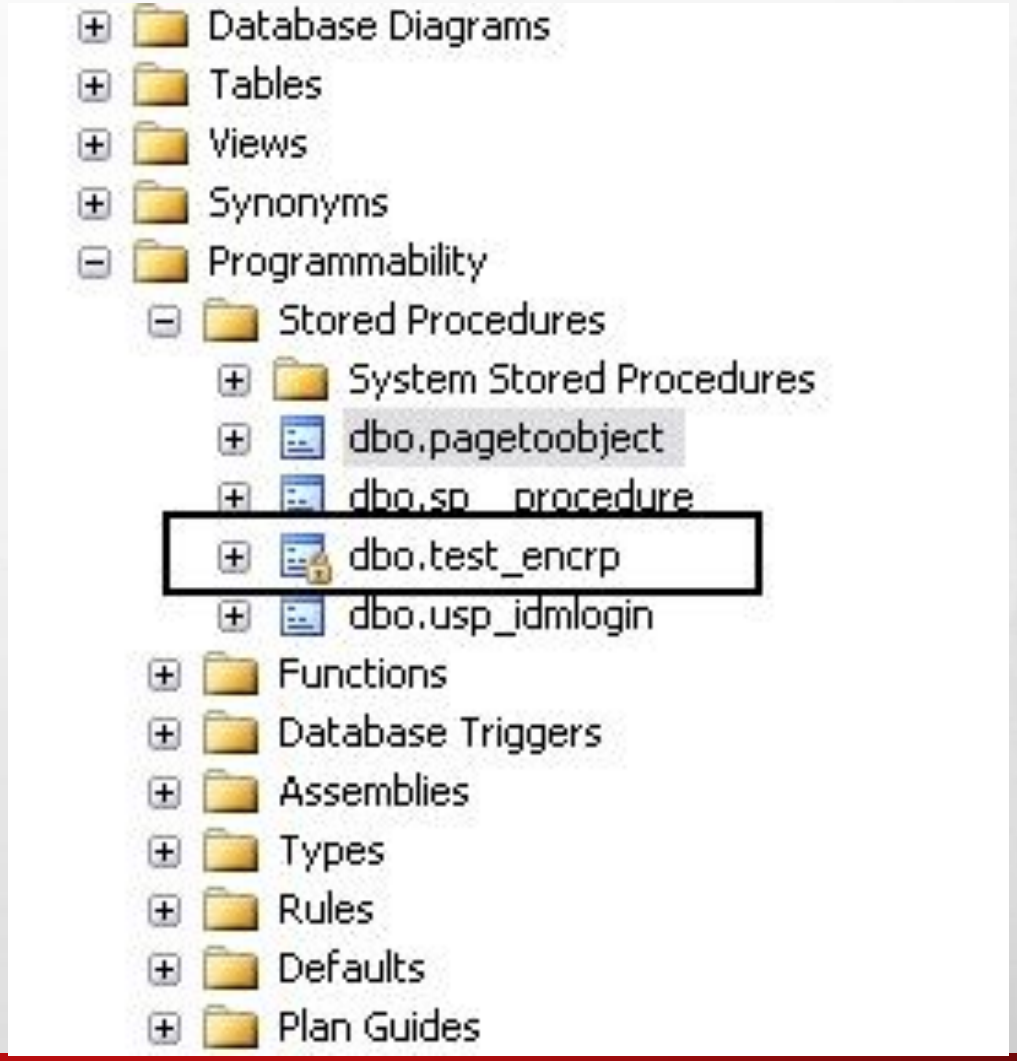
- **ESTRUTURAS DE ARMAZENAMENTO FÍSICO**
  - **SCHEMA (ESQUEMA) SQL-SERVER**
    - VOCÊ PODE ATRIBUIR PERMISSÕES DE LOGIN DE USUÁRIO A UM ÚNICO SCHEMA (ESQUEMA) PARA QUE O USUÁRIO POSSA ACESSAR APENAS OS OBJETOS QUE ESTÃO AUTORIZADOS A ACESSAR.
    - OS SCHEMAS (ESQUEMAS) PODEM SER CRIADOS E ALTERADOS EM UM BANCO DE DADOS, E OS USUÁRIOS PODEM RECEBER ACESSO A UM SCHEMA(ESQUEMA). UM SCHEMA(ESQUEMA) PODE SER DE PROPRIEDADE DE QUALQUER USUÁRIO, E A PROPRIEDADE DO SCHEMA (ESQUEMA) É TRANSFERÍVEL.

SQL-SERVER DATABASE ARCHITECTURE

- **ESTRUTURAS DE ARMAZENAMENTO FÍSICO**
  - **SCHEMA (ESQUEMA) SQL-SERVER**
    - UM SCHEMA (ESQUEMA) DE BANCO DE DADOS TAMBÉM PODE ATUAR COMO UM NAMESPACE. ISSO EVITA CONFRONTOS DE NOMES DE OBJETOS DE DIFERENTES SCHEMAS (ESQUEMAS).

SQL-SERVER DATABASE ARCHITECTURE

- ESTRUTURAS DE ARMAZENAMENTO FÍSICO
  - ESQUEMA SQL-SERVER
  - EXEMPLO:



STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS - (SCHEMA):

- **SINTAXE:**
  - **CREATE SCHEMA [nome do esquema] [AUTHORIZATION] [nome do usuário]**
  - **ONDE:**
    - **NOME DO ESQUEMA** É O NOME QUE SERÁ DADO AO SCHEMA
    - **NOME DO USUÁRIO** É O NOME DO PROPRIETÁRIO DO BANCO DE DADOS (DBA OU **USUÁRIO AVANÇADO**)

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS - (SCHEMA):

- **SINTAXE:**

1. **SELECIONAR, ALTERAR O CONTEXTO, DAR FOCO O/NO BANCO DE DADOS QUE QUEREMOS UTILIZAR.**

   USE [NOME DO BANCO DE DADOS]

2. **CRIAR O SCHEMA**

   CREATE SCHEMA [NOME DO SCHEMA]

STRUCTURED QUERY LANGUAGE - SQL

DDL – LINGUAGEM DE DEFINIÇÃO DE DADOS

- **É UTILIZADO PARA:**
  - **ESPECIFICAR UMA NOVA RELAÇÃO,**
  - **ATRIBUIR NOME A NOVA RELAÇÃO**
  - **ESPECIFICAR E QUALIFICAR:**
    - ATRIBUTOS
    - DOMÍNIO DOS ATRIBUTOS
    - CHAVES (PRIMÁRIA E ESTRANGEIRA)
    - RESTRIÇÕES DE INTEGRIDADE REFERENCIAL

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS - (CREATE TABLE):

- **SINTAXE:**

- **CREATE TABLE [nome do esquema].[nome da tabela]**

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS - (CREATE TABLE):

- ANTES UMA PEQUENA DISCUSSÃO SOBRE OS TIPOS DE DADOS PREVISTOS NO SQL-SERVER

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **DOCUMENTAÇÃO SOBRE TIPOS DE DADOS NO MS SQL-SERVER**

- https://docs.microsoft.com/pt-br/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver15

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2017)

CATEGORIAS DE TIPO DE DADOS / DOMÍNIO

OS TIPOS DE DADOS EM SQL SERVER SÃO ORGANIZADOS NAS SEGUINTES CATEGORIAS:

- NUMÉRICOS EXATOS
- NUMÉRICOS APROXIMADOS
- DATA E HORA
- CADEIA DE CARACTERES
- CADEIA DE CARACTERES
- CADEIA DE CARACTERES BINÁRIA
- OUTROS TIPOS DE DADOS

STRUCTURED QUERY LANGUAGE - SQL

TIPOS DE DADOS E DOMÍNIOS (SQL2017)

CATEGORIAS DE TIPO DE DADOS / DOMÍNIO

NUMÉRICOS EXATOS

- TIPOS DE DADOS NUMÉRICOS EXATOS QUE USAM DADOS INTEIROS. PARA ECONOMIZAR ESPAÇO NO BANCO DE DADOS,
- USE O MENOR TIPO DE DADOS QUE PODE CONTER TODOS OS VALORES POSSÍVEIS DE MANEIRA CONFIÁVEL.
- POR EXEMPLO, **TINYINT** É SUFICIENTE PARA A IDADE DE UMA PESSOA PORQUE NÃO EXISTE NINGUÉM QUE VIVA POR MAIS DE 255 ANOS. MAS **TINYINT** NÃO É SUFICIENTE PARA A IDADE DE UM EDIFÍCIO, PORQUE UM EDIFÍCIO PODE TER MAIS DE 255 ANOS.

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2017)

CATEGORIAS DE TIPO DE DADOS / DOMÍNIO

NUMÉRICOS EXATOS

- BIGINT
- BIT
- DECIMAL
- INT
- MONEY
- NUMERIC
- CADEIA DE CARACTERES BINÁRIA
- OUTROS TIPOS DE DADOS

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **NUMÉRICOS EXATOS:**

Tipos de dados	Intervalo	Armazenamento
bigint	**-2^63 (-9.223.372.036.854.775.808) a 2^63-1 (9.223.372.036.854.775.807)**	8 bytes
int	**-2^31 (-2.147.483.648) a 2^31-1 (2.147.483.647)**	4 bytes
smallint	**-2^15 (-32.768) a 2^15-1 (32.767)**	2 bytes
tinyint	**0 a 255**	1 byte

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **NUMÉRICOS EXATOS:**
  - **bit**
    - TIPO DE DADOS INTEIRO QUE PODE ACEITAR UM VALOR **1**, **0** OU **NULL**.
      - O MECANISMO DE BANCO DE DADOS DO SQL SERVER OTIMIZA O ARMAZENAMENTO DE COLUNAS **BIT**.
      - SE HOUVER 8 OU MENOS COLUNAS **BIT** EM UMA TABELA, AS COLUNAS SERÃO ARMAZENADAS COMO 1 BYTE.
      - SE HOUVER DE 9 A 16 COLUNAS **BIT**, AS COLUNAS SERÃO ARMAZENADAS COMO 2 BYTES, E ASSIM POR DIANTE.

STRUCTURED QUERY LANGUAGE - SQL

TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **NUMÉRICOS EXATOS:**
  - **decimal (p,s) ou numeric(p,s)**
    - O PARÂMETRO **p** INDICA O NÚMERO TOTAL MÁXIMO DE DÍGITOS QUE PODEM SER ARMAZENADOS (AMBOS À ESQUERDA E À DIREITA DO PONTO DECIMAL). **p** DEVE SER UM VALOR DE 1 A 38. O PADRÃO É 18.
    - O PARÂMETRO **s** INDICA O NÚMERO MÁXIMO DE DÍGITOS ARMAZENADOS À DIREITA DO PONTO DECIMAL. **s** DEVE SER UM VALOR DE 0 A **p**. O VALOR PADRÃO É 0.
    - PRECISÃO DE NÚMERO FLUTUANTE E NÚMERO DE ESCALA. PERMITE NÚMERO DE -10^38 +1 A 10^38 –1.
    - ARMAZENAMENTO DE 7 A 17 BYTES

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **NUMÉRICOS EXATOS:**
  - **decimal (p,s) ou numeric(p,s)**
    - EXEMPLOS:
      - 3,141516 → **p**=8 e **s**=6
      - 2,7182818285 → **p**=12 e **s**=10
      - 98,517 → **p**=6 e **s** = 3
      - 100572,18 → **p**=9 e **s** =2

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **NUMÉRICOS EXATOS:**
  - **money e smallmoney**
    - TIPOS DE DADOS QUE REPRESENTAM VALORES MONETÁRIOS OU DE MOEDA.

Tipos de dados	Intervalo	Armazenamento
money	**-922.337.203.685.477,5808 a 922.337.203.685.477,5807 (-922.337.203.685.477,58 a 922.337.203.685.477,58**	8 bytes
smallmoney	**-214.748,3648 a 214.748,3647**	4 bytes

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **NUMÉRICOS APROXIMADOS:**

  float [(n)]

  - TIPOS DE DADOS NUMÉRICOS APROXIMADOS PARA USO COM DADOS NUMÉRICOS DE PONTO FLUTUANTE.
  - OS DADOS DE PONTO FLUTUANTE SÃO APROXIMADOS; PORTANTO, NEM TODOS OS VALORES NO INTERVALO DE TIPO DE DADOS PODEM SER REPRESENTADOS DE MANEIRA EXATA.

Valor de n	Precisão	Armazenamento
1-24	**7 dígitos**	4 bytes
25-53	**15 dígitos**	8 bytes

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2017)

CATEGORIAS DE TIPO DE DADOS / DOMÍNIO

DATA E HORA

- date
- datetime2
- datetime
- datetimeoffset
- smalldatetime
- time

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **DATA**
  - **FORMATO AAAA-MM-DD, ONDE:**
    - AAAA – REPRESENTA O ANO;
    - MM – REPRESENTA O MÊS E;
    - DD – REPRESENTA O DIA
- **HORA**
  - **FORMATO: HH:MM:SS, ONDE:**
    - HH - REPRESENTA HORAS;
    - MM - REPRESENTA MINUTOS;
    - SS - REPRESENTA SEGUNDOS

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **DATA:**
  - **TIPOS DE DADOS QUE REPRESENTAM DATAS E HORAS.**

Tipos de dados	Intervalo	Armazenamento
date	0001-01-01 a 9999-12-31	**3 bytes**
datetime	Jan 1, 1753, a dez 31, 9999, precisão de 3,33 milissegundos	**8 bytes**
datetime2	0001-01-01 a 9999-12-31, precisão de 10 nanosegundos	**6 a 8 bytes**
datetimeoffset	0001-01-01 a 9999-12-31, reconhecimento de fuso horário	**8 a 10 bytes**
smalldatetime	1 de jan de 1900 a 6 de jun de 2079 com precisão de 1 minuto	**4 bytes**
time	00:00:00.0000000 a 23:59:59.9999999, precisão de 1 nanoseg	**3 a 5 byes**

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **CADEIAS DE CARACTERES:**
  - **TIPOS DE DADOS DE CARACTERES.**

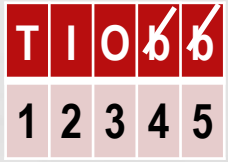
Tipos de dados	Intervalo	Tamanho Máximo (Caracteres)	Armazenamento Em Bytes
char(n)	Tamanho fixo, completado com espaços em brancos	8,000	Tamanho Definido
varchar(n)	Tamanho variável com limite	8,000	2 B + núm. caracteres
varchar(max)	Tamanho variável com limite	1,073,741,824	2 B + núm. caracteres
text	Tamanho variável	2 GiB de dados (texto)	4 B + núm. caracteres

STRUCTURED QUERY LANGUAGE - SQL

TIPOS DE DADOS E DOMÍNIOS

- **CHARACTER-STRING (CADEIA DE CARACTERES):**
  - **TAMANHO FIXO**
    - CHAR(n) OU CHARACTER(N) ONDE n É O NÚMERO DE CARACTERES;
      - VAMOS SUPOR QUE DEFINIMOS n=5, MAS, O DADO QUE QUEREMOS ARMAZENAR É

      - "TIO" QUE TEM APENAS 3 CARACTERES
      - O BANCO DE DADOS ARMAZENARÁ OS 3 CARACTERES SEGUIDO DE DOIS ESPAÇO EM BRANCO, ISTO É,



      - ONDE ø REPRESENTA 1 ESPAÇO EM BRANCO

STRUCTURED QUERY LANGUAGE - SQL

TIPOS DE DADOS E DOMÍNIOS

- **CHARACTER-STRING (CADEIA DE CARACTERES):**
  - **TAMANHO VARIÁVEL**
    - VARCHAR(**n**) OU CHAR VARYING(N) ONDE **n** É O NUMERO MÁXIMO DE CARACTERES
      - SUPONHA QUE **n** É O TAMANHO DEFINIDO PARA SUA COLUNA E **m** É O TAMANHO QUE DESEJA ARMAZENAR.
      - SE **m** = **n**, O ESPAÇO A SER OCUPADO É **n** + 2 BYTES DE CONTROLE
      - SE **m** < **n**, O ESPAÇO A SER OCUPADO É **m** + 2 BYTES DE CONTROLE
      - SE **m** > **n**, O SQL EMITIRÁ UMA MENSAGEM DE ERRO “STRING OU BINARY DATA WOULD BE TRUNCATED”

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS

- **CHARACTER-STRING (CADEIA DE CARACTERES):**
  - **EXEMPLOS DA APLICAÇÃO DO VARCHAR**
    - SEJA nome VARCHAR(40) NOT NULL
    - SE nome = “FACULDADES METROPOLITANAS UNIDAS”
    - n=40 e m=32. COMO m<n, O ESPAÇO OCUPADO SERÁ DE m + 2 BYTES, 32+2 = 34 BYTES
    - SE nome = “FACULDADES ASSOCIADAS RIBEIRÃO DAS ÁGUAS”
    - n=40 e m=40. COMO m=n, O ESPAÇO OCUPADO SERÁ DE n + 2 BYTES, 40 + 2 = 42 BYTES
    - SE nome = “FACULDADES ASSOCIADAS RIBEIRÃO DAS ÁGUAS”
    - n=40 e m=41. COMO m>n, O SQL EMITIRÁ UMA MENSAGEM DE ERRO “String ou binary data would be truncated”

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **CADEIAS DE CARACTERES UNICODE:**
  - **TIPOS DE DADOS DE CARACTERES UNICODE PODE ARMAZENAR QUALQUER TIPO DE DADO DEFINIDO PELO PADRÃO UNICODE.**

Tipos de dados	Intervalo	Tamanho Máximo (Caracteres)	Armazenamento
nchar[(n)]	Tamanho fixo com espaços em brancos	4,000	Tamanho definido x 2
nvarchar[(n)]	Tamanho variável	4,000	2 vezes n bytes + 2 bytes
nvarchar[(max)]	Tamanho variável	4,000	2 vezes n bytes + 2 bytes
ntext	Tamanho variável	2GB de texto	

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2003)

- **CADEIAS DE CARACTERES BINÁRIO:**
  - **TIPOS DE DADOS BINÁRIOS UTILIZADOS PARA ARMAZENAR IMAGENS, ARQUIVOS DE EDITORES DE TEXTO E PROCESSADORES DE TEXTO.**

Tipos de dados	Intervalo	Tamanho Máximo (Caracteres)	Armazenamento
binary(n)	Tamanho fixo (binário)	8,000 bytes	*n* bytes
varbinary(n)	Tamanho variável (binário)	8,000 bytes	*n* + 2 bytes
varbinary(max)	Tamanho variável (binário)	2GB	*n* + 2 bytes
image	Tamanho variável (binário)	2GB	

STRUCTURED QUERY LANGUAGE - SQL
TIPOS DE DADOS E DOMÍNIOS (SQL2017)

CATEGORIAS DE TIPO DE DADOS / DOMÍNIO

OUTROS TIPOS DE DADOS

- CURSOR
- HIERARCHYID
- SQL_VARIANT
- TIPOS DE GEOMETRIA ESPACIAL
- TABLE
- ROWVERSION
- UNIQUEIDENTIFIER
- XML
- TIPOS DE GEOGRAFIA ESPACIAL

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS

DEFINIÇÃO DE DADOS - EXEMPLOS

Nome do schema

```
CREATE TABLE grp01.CURSO
(
        codcurso        char(03)        not null,
        nomecurso       varchar(50)     not null,
        integralizacao  smallint        not null,
        autorizacao     date            not null,
        reconhecimento  date,
        CONSTRAINT PK_CURSO_codcurso PRIMARY KEY (codcurso)
);
```

Nome da tabela

Definição da estrutura da tabela

Definição da chave primária

```
CREATE TABLE grp01.DEPARTAMENTO
(
    coddepto        char(03)        not null,
    nomedepto       varchar(50)     not null,
    codfaculdade    char(03)        not null,
    codcoordenador  char(03)        not null,
    CONSTRAINT PK_DEPARTAMENTO_coddepto PRIMARY KEY (coddepto)
);
```

```
CREATE TABLE grp01.DISCIPLINA
(
        coddisciplina   char(03)        not null,
        sigladisciplina char(05)        not null,
        nomedisciplinachar(50)          not null,
        coddepto        char(03)        not null,
        codcurso        char(03)        not null,
        CONSTRAINT PK_DISCIPLINA_coddisciplina  PRIMARY KEY (coddisciplina),
        CONSTRAINT FK_DISCIPLINA_codcurso FOREIGN KEY (codcurso) REFERENCES
        grp01.CURSO(codcurso),
        CONSTRAINT FK_DISCIPLINA_coddepto FOREIGN KEY (coddepto) REFERENCES
        grp01.DEPARTAMENTO(coddepto)
);
```

STRUCTURED QUERY LANGUAGE - SQL

INTEGRIDADE REFERENCIAL

- A INTEGRIDADE REFERENCIAL É UM RECURSO IMPORTANTE DE BANCO DE DADOS QUE AJUDA A GARANTIR QUE AS RELAÇÕES ENTRE TABELAS SEJAM MANTIDAS CONSISTENTES.
- ISSO SIGNIFICA QUE, QUANDO HÁ UMA RELAÇÃO ENTRE DUAS TABELAS, A INTEGRIDADE REFERENCIAL GARANTE QUE OS VALORES DA TABELA REFERENCIADA (A TABELA "PAI") SEJAM SEMPRE VÁLIDOS E QUE A TABELA REFERENCIADORA (A TABELA "FILHA") SÓ POSSA ARMAZENAR VALORES QUE EXISTAM NA TABELA PAI.

STRUCTURED QUERY LANGUAGE - SQL

INTEGRIDADE REFERENCIAL

- UM EXEMPLO COMUM DE INTEGRIDADE REFERENCIAL É UMA TABELA DE PEDIDOS E OUTRA TABELA DE CLIENTES.
- CADA PEDIDO ESTÁ ASSOCIADO A UM CLIENTE POR MEIO DE **UMA CHAVE ESTRANGEIRA** QUE APONTA PARA UMA **CHAVE PRIMÁRIA** NA TABELA DE CLIENTES.
- A INTEGRIDADE REFERENCIAL IMPEDE QUE UM PEDIDO SEJA ASSOCIADO A UM CLIENTE QUE NÃO EXISTA NA TABELA DE CLIENTES.

STRUCTURED QUERY LANGUAGE - SQL

INTEGRIDADE REFERENCIAL

- É UM CONJUNTO DE REGRAS DE TODO SGBD QUE ASSEGURA QUE O RELACIONAMENTO ENTRE AS TUPLAS (LINHAS) SEJAM VÁLIDOS E QUE **NÃO SEJA POSSÍVEL EXCLUIR ACIDENTALMENTE OU PROPOSITALMENTE DADOS RELACIONADOS.**

INTEGRIDADE REFERENCIAL

- **REGRAS**
  - OS DADOS DE UMA TABELA RELACIONADA SÓ SERÃO ACEITOS SE EXISTIREM NA TABELA PRINCIPAL;
  - OS DADOS NA TABELA PRINCIPAL SÓ PODERÃO SER EXCLUÍDOS OU MODIFICADOS SE NÃO EXISTIREM DADOS RELACIONADOS A ESTA TUPLA(LINHA) NA TABELA RELACIONADA.

STRUCTURED QUERY LANGUAGE - SQL

INTEGRIDADE REFERENCIAL

- EXEMPLO:

```
CREATE TABLE ALUNO
(...,
    integralização        int   not null       default 4,
    CONSTRAINT DELCURSO
    FOREIGN KEY (cod-curso) REFERENCES ALUNO(cod-curso)
    ON DELETE SET NULL ON UPDATE CASCADE
);
```

STRUCTURED QUERY LANGUAGE - SQL

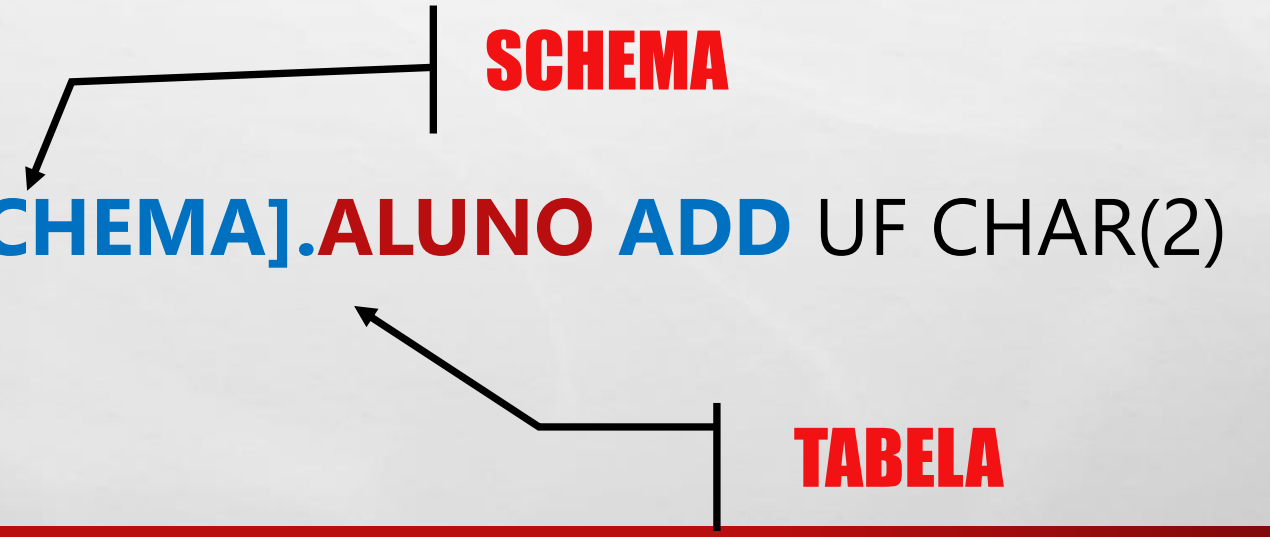
DEFINIÇÃO DE DADOS - ALTER:

- **É O COMANDO QUE PERMITE ALTERAR AS DEFINIÇÕES DE UMA TABELA:**
  - INCLUINDO NOVAS COLUNAS;
  - EXCLUINDO COLUNAS EXISTENTES;
  - ALTERANDO CARACTERISTICA DE UMA COLUNA

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS - ALTER:

- **INCLUINDO NOVAS COLUNAS:**
  - **EXEMPLO:**
    - **ALTER TABLE [SCHEMA].ALUNO ADD** UF CHAR(2)



STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS - ALTER:

- **ALTERANDO TIPO DE DADO:**
  - **EXEMPLO:**

  - **ALTER TABLE** grp01.**ALUNO ALTER COLUMN** sexo **smallint** not null;

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS - ALTER:

- EXCLUINDO, INCLUINDO COLUNAS E ALTERANDO TIPO DE DADOS:
- EXEMPLO:
  - ALTER TABLE grp01. ALUNO DROP COLUMN idade;
  - ALTER TABLE grp01. ALUNO ADD datanasc DATE not null;

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS - ALTER:

- INCLUINDO COLUNA:
- EXEMPLO:

```
ALTER TABLE grp01.BOLETIM ADD radis char (10) not null;
```

STRUCTURED QUERY LANGUAGE - SQL

DEFINIÇÃO DE DADOS - ALTER:

- REDEFININDO CHAVE PRIMARIA:
- EXEMPLO:
  - ALTER TABLE grp01.BOLETIM DROP CONSTRAINT PK_BOLETIM_ra;
  - ALTER TABLE grp01.BOLETIM ADD CONSTRAINT PK_ALUNO_radis PRIMARY KEY (radis);

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS (DML):

- COMANDOS PARA MODIFICAR O CONTEÚDO DE BANCO DE DADOS:

  - INSERT
  - DELETE
  - UPDATE

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - INSERT:

- **É UTILIZADO PARA ADICIONAR UMA LINHA A UMA TABELA**
- **REGRAS:**
  - É NECESSÁRIO ESPECIFICAR O NOME DA TABELA E A LISTA DE VALORES PARA A LINHA;
  - OS VALORES DEVERÃO ESTAR NA MESMA ORDEM DAS COLUNAS DA TABELA.

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - INSERT:

- **EXEMPLOS:**

```
-- USE FmuBSI_Quinta
SET LANGUAGE ENGLISH;
INSERT INTO grp01.CURSO (codcurso, nomecurso, integralizacao,
  autorizacao, reconhecimento)
VALUES
('121','CST em ANÁLISE e DESENVOLVIMENTO de SISTEMAS','6','02/01/2009','07/21/2011'),

('126','CST em BANCO de DADOS','6','02/01/2010','07/20/2012');
```

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - INSERT:

- **EXEMPLOS:**

```
USE FmuBSI_Quinta

INSERT INTO grp01.DEPARTAMENTO (coddepto, nomedepto, codfaculdade,
  codcoordenador)

VALUES

('211','matemática','311','411'),


('212','sistemas da informação','312','412');
```

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - INSERT:

- EXEMPLOS:

  CONTINUANDO...

  BAIXAR OS SCRIPTS DO GOOGLE DRIVE

  Descompactar o arquivo

  Executar todas os Scripts (Queries) de 01 a 10

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - INSERT:

- EXEMPLOS:

```
USE FmuBSI_Quinta
CREATE TABLE grp01.HISTORICO0219
(  ra                 char(07)          not null,
   nomealuno          varchar (70)      not null,
   nomedisciplina     varchar(70)       not null,
   mediabimestral     decimal(5,2)      not null, );
INSERT INTO           grp01.HISTORICO0219 (ra, nomealuno, nomedisciplina, mediabimestral)
SELECT                A.ra, A.nomealuno, D.nomedisciplina, (0.30*B.notaavcont + 0.70*B.notaprvreg)
FROM                  grp01.ALUNO A, grp01.BOLETIM B, grp01.CURSO C, grp01.DISCIPLINA D
WHERE                 (B.ra = A.ra)  AND (B.coddisciplina = D.coddisciplina) AND and (A.codcurso =
                                C.codcurso)  AND A.codcurso ='120'
ORDER BY              D.nomedisciplina;
```

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - DELETE:

- **PERMITE A EXCLUSÃO DE UMA LINHA EM UMA TABELA:**
- **REGRAS:**
  - PERMITE EXCLUSÃO DE LINHAS DE UMA **TABELA** POR VEZ. PORÉM A DELEÇÃO PODE SE PROPAGAR PARA OUTRAS **TABELAS** DESDE QUE UMA AÇÃO QUE TRATE **INTEGRIDADE REFERENCIAL** SEJA DISPARADA.
  - EXIGE A CLÁUSULA **WHERE**.

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - DELETE:

- **EXEMPLOS:**

```
DELETE FROM grp01.ALUNO
WHERE ra = '001359';

DELETE FROM grp01.PROFESSOR
WHERE codprof = '100';
```

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - DELETE:

- **EXEMPLOS:**

```
DELETE FROM grp01.BOLETIM
WHERE ra IN (
                SELECT    ra
                FROM      grp01. ALUNO
                 WHERE codcurso='123'
            );
```

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - UPDATE:

- **PERMITE MODIFICAR VALORES DE UMA OU MAIS LINHAS EM UMA TABELA**
- **REGRAS:**
  - EXIGE A CLÁUSULA **WHERE**;
  - ATUALIZAÇÃO EM UMA **CHAVE PRIMÁRIA** PODE SE PROPAGAR PARA UMA **CHAVE ESTRANGEIRA** EM OUTRA **TABELA** SE TIVER SIDO ESTABELECIDO REGRAS DE **INTEGRIDADE REFERENCIAL**

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - UPDATE:

- **SINTAXE**
  - **UPDATE** **NOME-DA-TABELA**
  - **SET** nome-da-coluna$_1$ = valor$_1$, nome-da-coluna$_2$ = valor$_2$,...,nome-coluna$_n$ = valor$_n$
  - **WHERE** **CONDIÇÃO**

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - UPDATE:

- EXEMPLOS:

```
USE BancoDadosFMU20212
UPDATE   grp01.BOLETIM
SET      notaprvreg = 10,0
WHERE    ra='11022' and coddisciplina='504';
```

STRUCTURED QUERY LANGUAGE - SQL

MANIPULAÇÃO DE DADOS - UPDATE:

- **EXEMPLOS:**

```
USE BancoDadosFMU20212
UPDATE  grp01.BOLETIM
SET     notaprvreg = notaprvreg + 3.25
WHERE   ra='11022' and coddisciplina='799';
```

STRUCTURED QUERY LANGUAGE - SQL

VIEWS (TABELAS VIRTUAIS)

- NA TERMINOLOGIA SQL, **VIEW** É UMA **TABELA** SIMPLES DERIVADA DE OUTRAS TABELAS. ESTAS OUTRAS **TABELAS** PODEM SER OBJETOS DO BANCO DE DADOS OU OUTRAS **VIEWS**.

1. UMA **VIEW** NÃO – NECESSARIAMENTE - EXISTE NA FORMA FÍSICA;
2. ELA É CONSIDERADA UMA **TABELA VIRTUAL**;
3. É PASSÍVEL DE ATUALIZAÇÃO;

STRUCTURED QUERY LANGUAGE - SQL

CRIANDO VIEWS

- EXEMPLOS:

```
CREATE VIEW    grp01.vwHISTORICO
AS  SELECT     A.ra AS "Registro do Aluno",
               A.nomealuno AS "Nome do Aluno",
               D.nomedisciplina AS Nome da Disciplina",
               B.notaavcont, B.notaprvreg AS "Média Final"
FROM   grp01. ALUNO AS  A, grp01. BOLETIM  AS  B, grp01. CURSO  AS  C,
       grp01. DISCIPLINA  AS  D
WHERE  (B. ra=A.ra) AND (B.coddisciplina=D.coddisciplina) AND (A.codcurso=C.codcurso)
                AND C.codcurso ='120' AND A.ra='610491';
```

STRUCTURED QUERY LANGUAGE - SQL

CRIANDO VIEWS

- EXEMPLOS CRIAR VIEW DE OUTRA (S) VIEW(S):

```
CREATE VIEW    grp01.[HISTORICO SEGUNDO SEMESTRE]
AS SELECT              H.[Registro do Aluno], H.[Nome do aluno],
H.[Nome da Disciplina], H.[Média Final]
FROM           grp01.vwHISTORICO AS H
WHERE          H.[Média Final]>=7.0
```

STRUCTURED QUERY LANGUAGE - SQL

EXCLUINDO VIEWS

- EXEMPLOS:

```
DROP VIEW grp01.HISTORICO;
```

OBSERVAÇÕES

- **OBS:**
  - No laboratório, utilizaremos outros exemplos e exploraremos as novas características das novas versões do SQL.
  - A queries são aquelas disponibilizadas na área da rede do laboratório que ora utilizamos:

GLOSSÁRIO

- **BANCO DE DADOS TEMPORAIS**:
  - UM BANCO DE DADOS TEMPORAL É UM BANCO DE DADOS COM **SUPORTE INTEGRADO PARA LIDAR COM DADOS SENSÍVEIS AO TEMPO** . NORMALMENTE, OS BANCOS DE DADOS ARMAZENAM INFORMAÇÕES APENAS SOBRE O ESTADO ATUAL E NÃO SOBRE OS ESTADOS ANTERIORES.
  - EXEMPLO, UM SISTEMA DE CONTROLE DE PREÇOS DE UM DETERMINADO PRODUTO. SE A ALTERAÇÃO CONTEMPLAR APENAS O VALOR, ISTO É O VALOR ANTERIOR E SOBREPOSTO PELO VALOR ATUAL, NÃO TEREMOS COMO SABER QUAIS ERAM OS VALORES ANTERIORES A UMA DETERMINADA DATA.

STRUCTURED QUERY LANGUAGE – SQL

REFERÊNCIAS

- Silberschatz, Abraham; Korth, Henry F.; Sudarshan, S.; SISTEMA DE BANCO DE DADOS; Pearson Education do Brasil;1999; cap. 4
- Elmasri, Ramez; Navathe Shamkant B.; FUNDAMENTALS OF DATABASE SYSTEMS; Addison Wesley; 2000;cap. 8
- Lewis, Philip M.; Bernstein, Arthur; Kifer, Michael; DATABASE AND TRANSACTION PROCESSING – Na Application-Oriented Aproach; Addison Wesley;2002; cap. 10
- Kroenke, David M., Banco de Dados: Fundamentos, Projeto e Implementação; LTC – Livros Técnicos e Científicos Editora S/A: Rio de Janeiro, 1999, cap 3