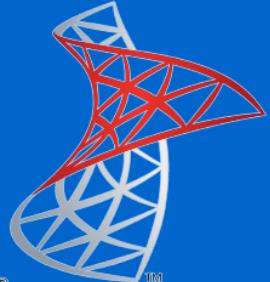


Ângelo Mário Almeida dos Santos

A PORTA DE ENTRADA EM CERTIFICAÇÃO

Fundamentos de Banco de Dados- Exame 98-364

Mais
de 400
págs.
NA
PRÁTICA



Microsoft®
SQL Server®

O Banco de Dados do Banco Central do Brasil
SQL Management Studio
dbForge SQL Studio

MÉTODO DE APRENDIZAGEM PASSO-A-PASSO

Iniciante - Intermediário - Avançado

**Ângelo Mário Almeida
dos Santos**

A PORTA DE ENTRADA

**Micro
SQL's**

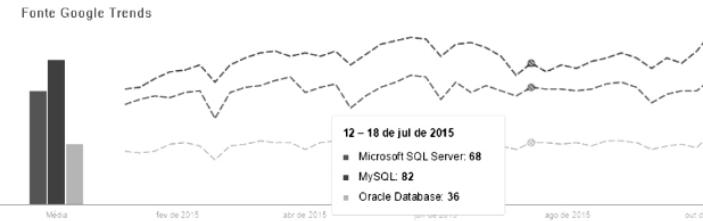
TRADA EM CERTIFICAÇÃO

csoft
Server

Fundamentos de Banco de Dados- Exame 98-364:

1a. Edição 2015

- Método de ensino interativo passo-a-passo;
- Exemplos práticos no aprendizado;
- Introdução a Banco de Dados;
- Introdução a Linguagem SQL;
- Segurança de Dados;
- Administrando Dados com dbForge Studio for SQL Server; ◦ Administrando Dados com Management Studio SQL Server 2014; ◦ Configurando Servidores com SQL Server Configuration Manager; ◦ Conteúdo atende a Certificação Microsoft SQL Server MTA; ◦ Conteúdo Programático Oficial da Microsoft;



dbForge Studio for SQL Server 4.5;



SQL Server 2014
Management Studio;



SQL Server Configuration Manager¹.

¹ <http://img.brothersoft.com/icon/softimage/r/razorsql-340735-1266740261.jpeg>

APRESENTAÇÃO

Microsoft SQL Server é um dos maiores bancos de dados em todo o mundo, robusto.

Vem comigo: Em 1992 ao finalizar o Curso Técnico em Processamento de Dados, que era uma Graduação Pós-Segundo Grau, ou seja, necessitaria o 2º. Grau para estudar o Pós-Técnico com duração de 2 (dois) anos. Um curso comparado ao Curso Tecnólogo Superior de

2 anos, carga horária equivalente. No entanto faltou-me, naquele período, orientação para colocarme mais rápido no mercado de Tecnologia da Informação, por que faltar “foco” em Banco de Dados com conhecimento da Linguagem SQL, eis o porquê detenho a marca da FocoSite.

Pois, depois de 4 (quatro) anos da conclusão do Pós-Técnico que passei a manusear o Visual Basic 6.0 já uma migração de Clipper 5 hoje o xHarbour, e depois me apaixonei com a Programação para Web e como já tinha conhecimento da sintaxe e Linguagem Visual Basic 6 foi um pulo curto para o ASP- Active Server Page. Porém, só usava, até então, o Banco de Dados Microsoft Access por facilitar, demais, a configuração em nosso no computador com o IIS- Information Internet Server (sobre esse assunto no meu livro “Programando na WEB com ASP Clássico, 1a. Edição, São Paulo, Ed.Viena, 2015,

o quê falo, também, do ASP trabalhando com Banco de Dados MySQL, Firebird, SQL Server, Oracle etc).

Em Porta de Entrada para Certificação SQL Server, o leitor ou aprendiz em Banco de Dados vai ter um conteúdo que traz informações, caso queira a Certificação em Microsoft SQL Server estará atualizado com o que irás enfrentar no seu Exame 98-364.

Faz-se necessário a leitura com todo cuidado sem atropelamentos pois é necessário compreender e para isso usaremos de uma metodologia prática com “mapas mentais”, onde contextualizaremos os assuntos a realidade em que vivemos, a começar o mundo é um Banco de Dados, e para administrar e gerenciar esse montante de informações surgiram o SGBD.

O nascimento desse livro foi um upgrade (atualização) com profunda pesquisa na Internet e interação com

Profissionais DBA. Verifiquei o comportamento de quem já alcançou a Certificação MTA – Microsoft Technology Associate em SQL Server. Filtrei como eles conseguiram passar no Exame, bem como, quais conteúdos incluiram no Plano de Estudos.

Aqui aplico, também, um pouco do conhecimento de como reage o cérebro diante de desafios, as vezes, incluindo conteúdo com “mapas mentais” esses diagramas abrem gatilhos para o cérebro encontrar o ponto de compreensão.

Por fim, vamos juntos nessa jornada, não se demotive, aprenda a aprender como se aprende e sempre é com muito esforço.

Ângelo Mário Almeida dos Santos
Programador ASP e DBA-SQL Server e MySQL

DEDICATÓRIA

Ao meu sobrinho Alef que não muito tardar possa apreciar a Tecnologia de Banco de Dados. A você, bem como, tantos outros adolescentes que navegam na Internet, ainda não sabe o quê está por trás desse emaranhado de teias que interligam o mundo digital.

Ao leitor que deseja a Certificação Microsoft SQL Server – Fundamentos em Banco de Dados.

SUMÁRIO

CAPÍTULO 1

1. Certificação Microsoft SQL Server MTA
- 1.1 - Primeiro Exame: Microsoft Technology Associate (MTA), nível entrada.

- 1.1.1 - Conteúdo Programático Oficial da Microsoft:
- 1.1.1.1 - Entendendo os conceitos principais de bancos de dados (20–25%)
 - 1.1.1.2 - Criar objetos do banco de dados (20–25%)
 - 1.1.1.3 - Manipular dados (25–30%)
 - 1.1.1.4 - Entender armazenamento de dados (15–20%)
 - 1.1.1.4.1 - Descrição de Normalização
 - 1.1.1.4.1.1 - Primeira Forma Normalizada
 - 1.1.1.4.1.2 - Segunda Forma Normalizada
 - 1.1.1.4.1.3 - Terceira Forma Normalizada
 - 1.1.1.4.1.5 - Quarta Forma Normalizada
 - 1.1.1.4.1.6 - Quinta Forma Normalizada
 - 1.1.1.4.1.7 – Outros formulários de normalização
 - 1.1.1.4.1.8 - Briefing do Processo de Design do Banco de Dados
 - 1.1.1.4.1.9 - Entender chaves primárias, estrangeiras e compostas
 - 1.1.1.4.1.10 - Entender índices
 - 1.1.1.5 - Entender conceitos de segurança do banco de dados
- 1.2 - Configuração e administração de SQL Server Express 2014, SQL Management Studio e dbForge Studio
- 1.2.1 - Configurando SQL Server Express 2014
 - 2 - Introdução Administração de Banco de Dados
 - 2.1 - Criando Banco de Dados
 - 2.1.1 – SQL Management Studio
 - 2.1.2 - SQL Server Configuration Manager
 - 2.1.3 - Usando dbForge Studio for SQL
 - 2.2 - Uso Comando "CONSTRAINTS" - restrições
 - 2.2.1 - Not Null
 - 2.2.2 - Unique
 - 2.2.3 - Primary Key
 - 2.2.4 - FOREIGN KEY
 - 2.2.5 - CHECK

2.2.6 - DEFAULT

2.3 - Criando Tabelas

2.3.1 - Tipos de dados no SQL Server

2.3.1.1 - Tipos alfanumérico

2.3.1.2 - Tipos de números

2.3.1.3 - Tipos de datas

2.3.1.4 - Outros tipos de dados

2.4 - Alterar e Excluir Tabelas e Colunas

2.5 - Inserindo dados

2.5.1 - Comandos SQL inserindo conteúdo na tabela_autores

2.5.2 - Comandos SQL inserindo conteúdo na tabela_livro

2.5.2.1 - Verificar os dados gravados

2.6 - Consultar os dados

2.6.1 - Especificar a ordem da consulta

2.6.2 - Filtros de registros em uma consulta - WHERE

2.6.3 - Filtros de registros com cadeia de caracteres

correspondentes - LIKE

CAPÍTULO 2

1. Introdução a Banco de Dados

1.1 - Definição do SBGD

1.2 - Sistema de Banco de Dados

1.3 - Conceitos Gerais de Banco de Dados

1.3.1 - Modelagem de Banco de Dados

1.3.1.1 - Estrutura da Modelagem

1.3.1.1.1 - Modelo Relacional

1.3.1.1.2 - Modelo Hierárquico

1.3.1.1.3 - Modelo Rede

1.3.1.1.4 - Revisando Criação de Banco de Dados

1.3.1.1.4.1 - Auditoria de Informação

1.3.1.1.4.1.1 - O que é uma auditoria em BD?

1.3.1.1.4.1.1.a - Como auditar um DB?

1.3.1.1.4.1.1.b - O quê auditar em DB?

1.3.1.1.4.1.1.b.1 - Recursos e tarefas no Banco de Dados

- 1.3.1.1.4.1.1.b.1.a - Nas Tabelas espelhado existente
- 1.3.1.1.4.1.1.b.1.b - Fazer backup e restaurar bancos de dados 2.2 - Instrução RESTORE
- 1.3.1.1.4.1.1.b.1.c - Monitorar e ajustar o desempenho 2.2.A - Restaurando um banco de dados completo
- 1.3.1.1.4.1.1.b.1.d - Índices e chave primárias 2.2.B - Restaurando backups de banco de dados diferenciais e completos
- 1.3.1.1.4.1.1.b.1.e - Notificação de Eventos 2.2.C - Restaurando um banco de dados usando a sintaxe RESTART
- 2 - Instruções BACKUP e RESTORE (Transact-SQL) 2.2.D - Restaurando um banco de dados e movendo arquivos
- 2.1 - Instrução BACKUP 2.2.E - Copiando um banco de dados usando BACKUP e RESTORE
- 2.1.A - Criando um backup completo de arquivos dos grupos de arquivos secundários 2.2.F - Restauração pontual usando STOPAT
- 2.1.B - Criando um backup diferencial de arquivos dos grupos de arquivos secundários 2.2.G - Restaurando o log de transações até uma marca
- 2.1.C - Criando e fazendo backup em um conjunto de mídias espelhado de uma única família 2.2.H - Restaurando com o uso da sintaxe de TAPE
- 2.1.D - Criando e fazendo backup em um conjunto de mídias espelhado de várias famílias 2.2.I - Restaurando usando a sintaxe FILE e FILEGROUP
- 2.1.E - Fazendo backup em um conjunto de mídias

CAPÍTULO 3

1. Estrutura de Dados e Cardinalidade de

Relacionamentos	2.3.1 - Como índices são usados pelo otimizador de consulta
1.1 - Entidade	
1.2 - Relacionamento	
1.2.1 - Tipos de relacionamentos (1:1) um para um	
1.2.2 - Tipos de relacionamentos (1:*) um para muitos	
1.2.3 - Tipos de relacionamentos (*:*) muitos para muitos	
1.3 - Atributo	
2. Conceitos de banco de dados relacionais	
2.1 - Chave Estrangeira	
2.1.1- Criar relações de chaves estrangeiras no SQL Server 2014	
2.1.1.1 - Pemissões	
2.2 - Chave Primária	
2.2.1 - Índice CLUSTED e NOCLUSTERED	
2.2.1.1 - CLUSTED - Índice Agrupados	
2.2.1.2 - NONCLUSTERED – Índices Não Agrupados	
2.3 - Índices e Restrições	
	CAPÍTULO 4
	1. O quê estudar para o Exame 98-64?
	1.1- Algumas questões que caem no Exame 98-64
	1.2 - Perguntas e Respostas do Exame 98-64
	1.2.1 - Parte I
	1.2.2 - Parte II
	1.2.3 - Parte III
	CAPÍTULO 5
	1 - DML SQL
	1.1 - Select
	1.1.1 - Uso de DISTINCT
	1.1.2 - Uso de UNION
	1.1.3 - Uso de BETWEEN
	1.2 - Inserção

1.3 - Atualização

1.4 - Exclusão

2 - DDL SQL

3 - DCL SQL

4 - TCL SQL

CAPÍTULO 6

1 - Gatilhos DML

1.1 - Benefícios do gatilho DML

1.2 - Tipos de gatilhos DML

1.2.1 - Gatilho AFTER

1.2.2 - Gatilho INSTEAD OF

CAPÍTULO 7

1. Stored Procedures Controle de Decisões

1.1 – Criando Stored Procedures

1.2 – Alterando Procedures passando parâmetros

1.3 - Revisando Procedures e Funções

1.4 - Regras e Prática Recomendada com Stored Procedures

CAPÍTULO 8

1.1 - Como obter todas as tabelas com ou sem restrição de chave primária no SQL Server?

1.2 - PRINT / mensagens instrução SELECT dentro de loop while ou lote de declaração não é exibida imediatamente depois que é Sql Server Execution

1.3 - Como encontrar todas as tabelas com nenhum índice em tudo no SQL Server?

1.4 - A função MID()

1.5 - A função LEN()

1.6 - A função Format()

1.7 - A função GROUP BY()

1.8 - Função AVG()

GLOSSÁRIO

ANEXOS

- Anexo I – Erro de conexão com banco de dados
- Anexo II – Erro no SQL Management Studio
- Anexo III – Erro de Instância Parada no Server Configuration Manager
- Anexo IV - Introdução ao API Sistema Windows
- Anexo V – Uso de ALTER TABLE e CREATE TRIGGER

REFERÊNCIAS

AUTOR CAPÍTULO 1

É possível obtermos informações sobre o cenário e interesse por SQL Server no Brasil no Estado de São Paulo, a exemplo? Sim! Temos ferramentas na Internet como Google Trends (www.google.com/trends) que nos dá resultados sobre pesquisa de comparativo em termos escritos na digitados na busca do Google Trends.

Veja, no gráfico da imagem 1.x, os termos “Microsoft SQL Server”, “MySQL” e “Oracle Database” as tecnologias em Banco de Dados que em 2015 no Brasil mais despertou interesse no Estado de São Paulo. O Microsoft SQL Server mantem-se em segundo lugar, superando o Oracle Database, um pouco atrás, somente, para o Banco de Dados MySQL. E o porquê em nosso livro “Programando na WEB com ASP Clássico”, Ed. Viena, 2015 – incluímos um tópico de ASP com MySQL”. No entanto, o mercado de TI em Banco de Dados milhares de Empresas usam SQL Server com ASP.

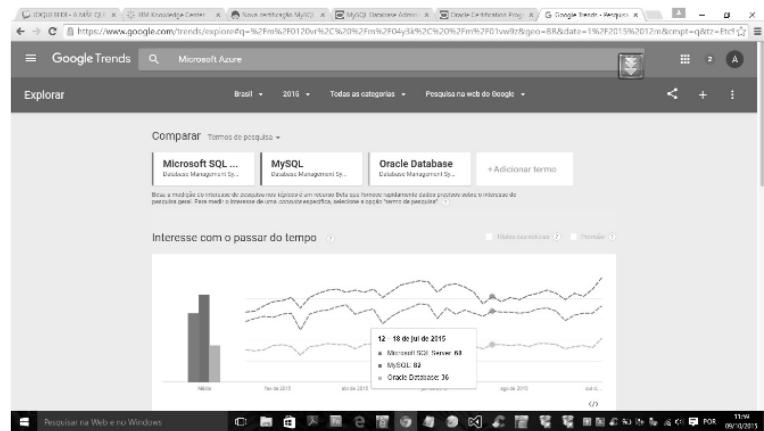
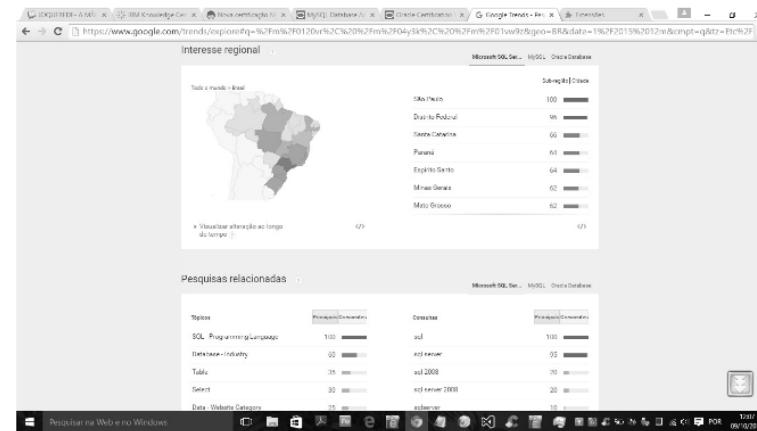


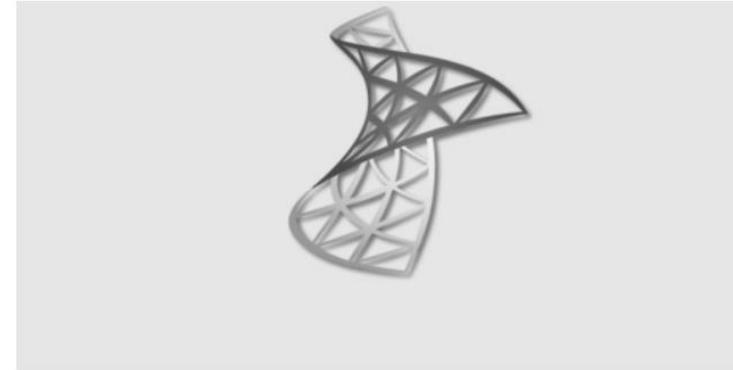
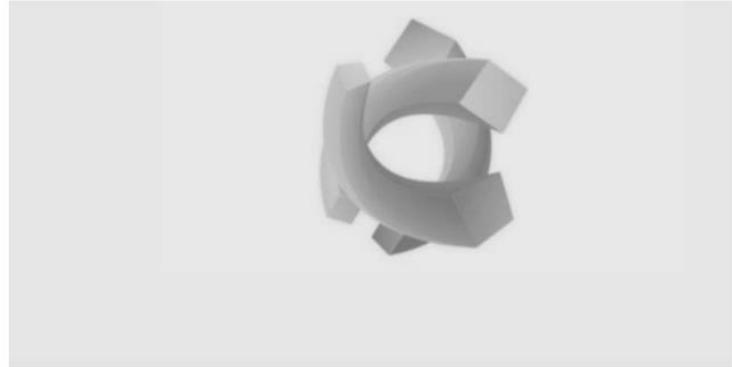
Imagen 1.x



Cenário SQL Server em São Paulo-Brasil - Imagem 2.x

Na imagem 2.x temos o cenário por nome de tecnologia detalhada. Onde a “SQL” vem com 100% a cada 100 usuários na Internet essa pesquisa “SQL” é de interesse em São Paulo é o gráfico no mapa de “Interesses Regionais” na imagem 2.x. Sendo, portanto, o Estado de São Paulo em 2015 que mais se interessa por Microsoft SQL Server.

NOTA:



dbForge Studio Express for SQL Server Visual
Studio SQL Server Studio Management

Perceba que na imagem 2.x o interesse é na tecnologia “SQL – Programming Language”, ou seja, interesse na Programação SQL, exatamente o que propomos nesse livro. Um conhecimento prático de como utilizar a SQL, que funciona tanto no SQL Server, MySQL, Oracle, Sybase, etc. Essa resposta da Google Trends, demonstra uma tendência em interesse crescente ao longo dos últimos anos nos comandos e instruções SQL. Dominando-a você se adaptará, agora, somente a operacionalização dos diversos Softwares que facilitam através do visual a operacionalização da SQL.

Esses Softwares “Visual Studio”, “Studio Management” e “dbForge Studio Express for SQL Server” são administradores de banco de dados; realizam conexão de banco de dados; criam tabelas; inclui, consulta, exclui, altera dados e todos os recursos que SQL favorece ao Administrador em Banco de Dados.

NOTA

Nosso focofoco será incluir você no rol de Profissionais em Administradores de Banco de Dados – DBA, esse Profissional Certificado Microsoft MTA.

1. Certificação Microsoft SQL Server MTA

A documentação Oficial da Microsoft é quem responde essa pregunta:

“Microsoft Technology Associate (MTA) é um introdutório certificação Microsoft para os indivíduos, considerando uma carreira em tecnologia. Certificação MTA aborda um amplo espectro de conceitos técnicos fundamentais, avalia e valida o seu conhecimento técnico do núcleo, e aumenta a sua credibilidade técnica.”²

NOTA:

“Recentemente a Microsoft reformulou o processo de certificação para

iniciantes conhecida como MTA (Microsoft Technology Associate), este é um ponto de entrada recomendado para iniciar suas certificações de TI e preparação para o mercado de trabalho. Passe em apenas um exame e você já vai **Passe em apenas um exame e você já vai** ganhar a certificação, dando o seu primeiro passo para uma carreira em tecnologia. Se você está apenas começando sua carreira de TI ou está querendo melhorar a compreensão dos fundamentos de TI, MTA irá validar o seu conhecimento. Está é uma certificação reconhecida pela indústria e opcional para os que buscam uma carreira em infraestrutura de TI, desenvolvimento e design (criação) de banco de dados e softwares utilizando tecnologias Microsoft.³

Após esse primeiro Exame 98-364, o Profissional de TI, tem opção para três áreas: **1.** Infraestrutura de TI; **2.** Bancos de dados;

² Microsoft Technology Associate, <https://www.microsoft.com/en-us/learning/mta-certification.aspx>,
03/10/2015

³ Rogério Molina, TecNet, Certificações MTA, <http://social.technet.microsoft.com/wiki/ptbr/contents/364.aspx>

[articles/15262.certificacoes-mta.aspx, 04/09/2015](#)

NOTA:

É meio caminho andado, para você leitor, o segmento de Banco de Dados, afinal você irá adquirir conhecimento sobre “Fundamento de Banco de Dados”, com um nível avançado.

3. Desenvolvedor.

NOTA:

Nossa atenção nesse livro é voltada para atender esse primeiro Exame.

1.1 - Primeiro Exame: Microsoft Technology Associate (MTA), nível entrada.

Database Fundamentals:

Fundamentals:

364.aspx

NOTA:

Reserve um tempo para conhecer os vídeos do ANEXO IV.

1.1.1 - Conteúdo Programático Oficial da Microsoft:

O intuito da Microsoft é qualificar o profissional para atender a demanda de Gerenciamento de Banco de Dados sobre a Plataforma SQL Server. Sendo uma grande demanda no mercado de Tecnologia de Informação.

O pode do SQL Server conhecemos nesses artigos:

Cito:

“Banco Central do Brasil otimiza armazenamento de dados com SQL Server 2012, 24/01/2013

A autoridade do país em bancos, o Banco Central do Brasil, está encarregada de acompanhar as finanças de mais de 150 bancos. Para isso, mantém um ambiente de Data Warehouse de missão crítica que armazena as operações de crédito realizadas por esses bancos. O Banco Central também utiliza os dados para desenvolver relatórios de

risco para as instituições financeiras.

A autoridade do país em bancos, o Banco Central do Brasil, está encarregada de acompanhar as finanças de mais de 150 bancos. Para isso, mantém um ambiente de Data Warehouse de missão crítica que armazena as operações de crédito realizadas por esses bancos. O Banco Central também utiliza os dados para desenvolver relatórios de risco para as instituições financeiras.

Com o crescimento do volume de dados que o Banco precisava armazenar, era necessário aumentar o tamanho da capacidade de armazenamento. A instituição espera que seu Warehouse cresça para oito terabytes no próximo ano. “O maior problema para nós é que, quando a quantidade de dados é muito grande, os cálculos de análise de dados se tornam mais complexos”, afirma José Claudio Mendonça de Freitas, do Banco

Central do Brasil. Para os próximos 10 anos, a previsão é que este número seja de 30 terabytes.

Para dar conta de armazenar tantos dados, a instituição precisava melhorar o desempenho de seu Data Warehouse. Em outubro de 2011, para agregar uma solução de tecnologia de alto desempenho, o Banco Central do Brasil decidiu testar uma nova solução baseada em SQL Server 2012. A organização já utiliza outras tecnologias da Microsoft, há muitos anos. “Nossos usuários são muito acostumados a trabalhar com tecnologia Microsoft, aumentando assim os impactos positivos a longo prazo sobre o banco”, comenta Freitas.

A instituição se interessou pelo SQL Server por conta do xVelocity que armazena dados por colunas em vez de linhas. Como a compactação de dados é mais fácil dentro de uma coluna, este novo recurso acelera

drasticamente o tempo de processamento de banco de dados de consulta. “A solução tem um custo acessível e que proporciona grandes benefícios”, acredita.

Após a implantação, o Banco Central do Brasil conseguiu acessar a consulta de dados 30% mais rápida com o SQL Server 2012, o que ajudará a empresa a entregar relatórios mais rapidamente. O Banco também espera melhorar a capacidade de análise de dados e prestar um melhor serviço aos bancos em geral. “Nós vimos uma melhoria de 30% em consultas de data warehouse usando SQL Server 2012 com o recurso xVelocity implementado”, afirma Freitas.

Com o SQL Server, o Banco Central do Brasil vai agregar à sua infraestrutura uma maior capacidade de armazenamento de dados e a solução para analisá-los que irá ajudar a organização a atender melhor seus clientes. Para Claudio, a solução é inovadora. “Qualquer

empresa que usa uma nova tecnologia como essa, com essas vantagens de desempenho, pode ser considerada uma companhia inovadora. Esta solução também irá nos ajudar a oferecer uma melhor análise e serviços para os bancos”, finalizao executivo.”⁴

1.1.1.1 - Entendendo os conceitos principais de bancos de dados (20-25%)

. Entender como os dados são armazenados em tabelas

O ser humano vive em comunidade não é uma ilha isolada, ao contrário vivemos a Era da Conexão e por isso o afã de selfs a fim de se mostrar ao mundo. Cada grupo que vive no mundo é “uma entidade” que têm “atributos específicos”, ou seja, “dono de cachorro” precisamos ter uma “entidade” e essa é conhecida com “tabela” e nela temos que definir “atributos”. Quais os atributos de um dono de cachorro?

- _ **Tabela_proprietarios:** Tabela_proprietarios:
codigo_proprietario, nome_proprietario,
endereco_proprietario, rg, cpf, telefone;
- _ Tabela_caes: codigo_cao, codigo_proprietario,
idade_cao, tipo_vacina, enfermidades_apresentadas,
codigo_raca,codigo_vacina;

⁴ Café Partner Brasil, <https://www.cafepartnerbrasil.com.br/cafe/banco-central-do-brasil-otimizaarmazenamento-de-dados-com-sql-server-2012/>, 15/10/2015

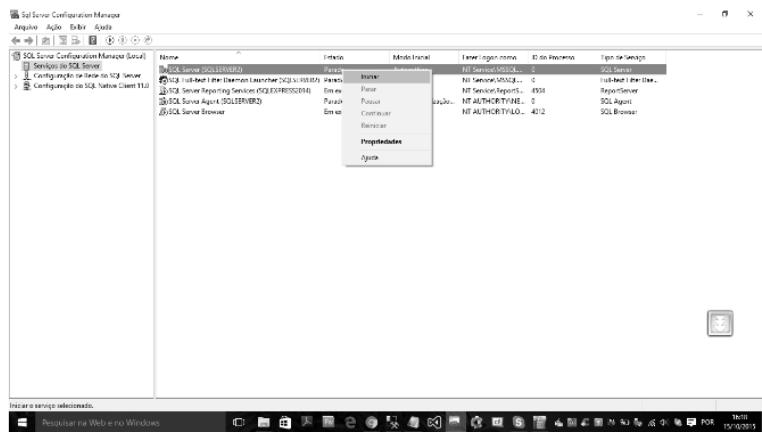
- _ Tabela_vacinas: codigo_vacina, tipo_vacina,
objetivo_da_vacina;
- _ **Tabela_raca:** codigo_raca, nome_raca, idade_minima,
idade_maxima, alimentacao, tipo_vacina

Na prática vamos construir esse banco de dados, agora iremos ter nosso primeiro contato com o SQL Management Studio, construindo as tabelas:

- 1.** Proprietário;
- 2.** Raça;
- 3.** Vacinas;
- 4.** Cães.

Antes, porém, verifique se o Servidor SQLSERVER2 está ativado. Acione o SQL Server Configuration Manager,

Tela 8.8:



Tela 8.8

- . Selecione a instância do Servidor conforme Tela 8.8 com segundo botão do mouse e clique em “iniciar”;
- . Aguarde inicialização do Servidor;
- . Retorne ao “SQL Management Studio” e clique em “conexão”;
- . Agora clique em “Nova Consulta”;
- . Vamos criar um “novo banco de dados”;
- . Digite o seguinte código fonte SQL dentro de “Nova Consulta”:

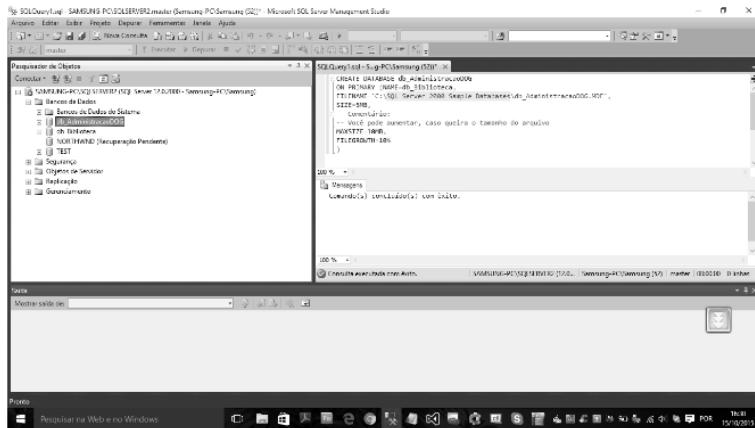
```
CREATE DATABASE db_AdministracaoDOG
ON PRIMARY (
NAME=db_Biblioteca,
FILENAME='C:\SQL Server 2000 Sample Databases
\db_AdministracaoDOG.MDF',
SIZE=5MB,
```

-- Comentário:

-- Você pode aumentar, caso queira o tamanho do arquivo, o valor – mínimo é 5MB
MAXSIZE=10MB,
FILEGROWTH=10%
)

. Cuidado com “SQL Server 2000 Sample Databases” esse é o caminho. Um erro de digitação pode apresentar erro;

. Pronto conforme tela 8.9 já temos criado o nosso novo banco de dados “db_AdministracaoDOB.MDF”;



Tela 8.9

Vamos criar nossas quatro tabelas: Proprietário; Raça; Vacinas; Cães. **1.** Cadastro de proprietário de cachorro: código, nome, endereço, cep,

Cadastro de proprietário de cachorro:

telefone, cidade, idade, cpf, rg;

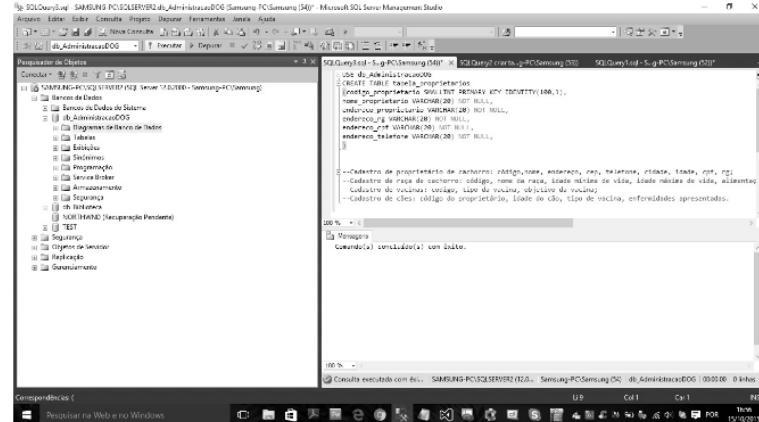
2. Cadastro de raça de cachorro: código, nome da raça,

idade mínima de vida, idade máxima de vida, alimentação, cuidados específicos, data de registro, tipo de vacina;

3. Cadastro de vacinas: código, tipo da vacina, objetivo da vacina; **4.** Cadastro de cães: código do proprietário, idade do cão, tipo de vacina, enfermidades apresentadas.

NOTA:

Iremos retomar esse assunto nos capítulos seguintes, aqui é mais para e conhecer o relacionamento dessas entidades/tabelas através da Ferramenta do SQL Management Studio Diagramas de Banco de Dados.



Tela 8.9 – Tabela Proprietário

. Vamos criar a Tabela Proprietário conforme Tela 8.9, digite o código fonte SQL:

```
USE db_AdministracaoDOG
```

```
CREATE TABLE tabela_proprietarios  
(codigo_proprietario SMALLINT PRIMARY KEY
```

```

IDENTITY(100,1), nome_proprietario VARCHAR(20)
NOT NULL,
endereco_proprietario VARCHAR(20) NOT NULL,
endereco_rg VARCHAR(20) NOT NULL,
endereco_cpf VARCHAR(20) NOT NULL,
endereco_telefone VARCHAR(20) NOT NULL,
)

```

. Agora vamos criar a Tabela_Vacinas. Clique “nova consulta” e digite o código fonte SQL:

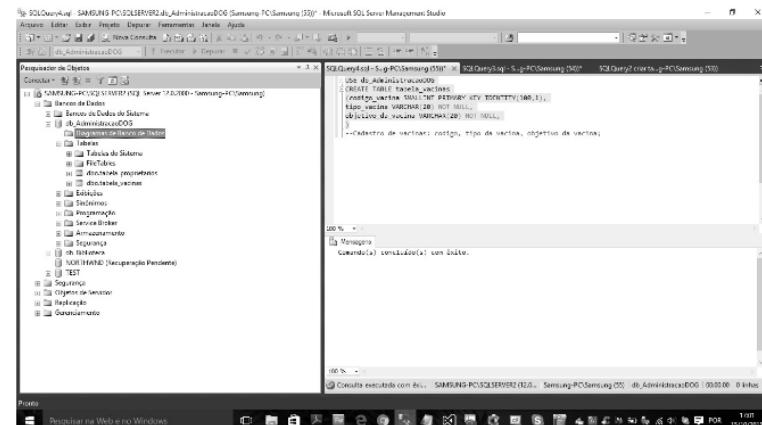
```
USE db_AdministracaoDOG
```

```

CREATE TABLE tabela_vacinas
(codigo_vacina      SMALLINT      PRIMARY      KEY
IDENTITY(100,1), tipo_vacina    VARCHAR(20)  NOT
NULL,
objetivo_da_vacina VARCHAR(20) NOT NULL,
)

```

- Clique em “executar”;
- Os textos em verde são comentários;
- Observe a Tela 9.2.1 que aparecem as tabelas criadas:



Tela 9.2.1

- Clique em “nova consulta” e vamos criar a “tabela_caes”, digite o código fonte SQL. E clique em “executar”:

```

USE db_AdministracaoDOG
CREATE TABLE tabela_caes
(codigo_cao SMALLINT PRIMARY KEY IDENTITY(100,1),
codigo_proprietario VARCHAR(20) NOT NULL,
idade_cao VARCHAR(20) NOT NULL,
tipo_vacina VARCHAR(20) NOT NULL,
enfermidades_apresentadas VARCHAR(20) NOT NULL, )

```

. Vamos criar, também, a última “tabela_raca”, digite o código fonte SQL. E, depois, “executar”:

```

USE db_AdministracaoDOG
CREATE TABLE tabela_raca
(codigo_raca      SMALLINT      PRIMARY      KEY
IDENTITY(100,1), nome_raca VARCHAR(20) NOT NULL,
idade_minima VARCHAR(20) NOT NULL,
idade_maxima VARCHAR(20) NOT NULL,
alimentacao VARCHAR(20) NOT NULL,

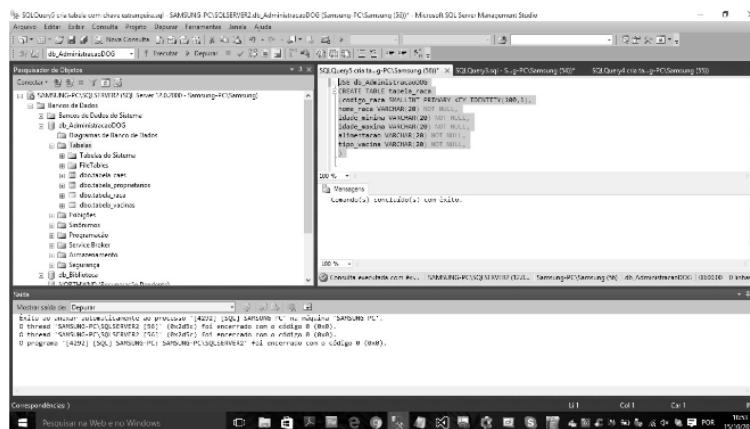
```

```

tipo_vacina VARCHAR(20) NOT NULL,
)

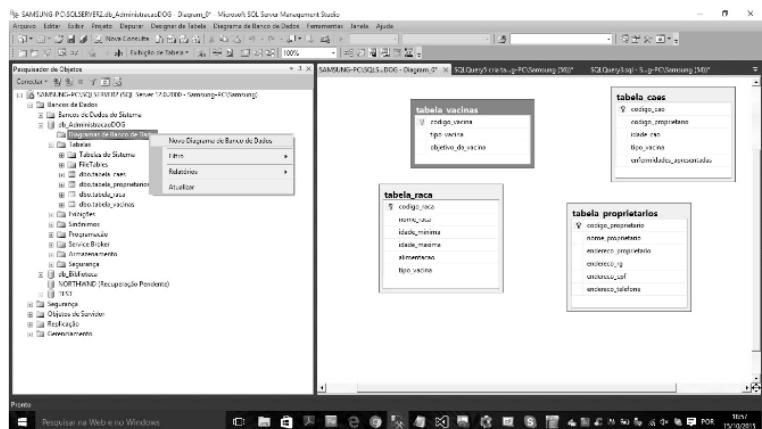
```

. Na tela 9.4 – temos todas as tabelas criadas, para isso, necessário selecionar “tabelas” e com segundo botão do mouse clique em “atualizar”;



Tela 9.4 – Atualização de todas tabelas

Todo esse trabalho para uma finalidade conhecer como as entidades/tabelas se relacionam, veja imagem Tela 9.5



Tela 9.5

NOTA: na Tela 9.5 as entidades não estão se relacionando por que não foram definidos as chaves_estrangeiras que indicam a existência de um campo em outra tabela. Ou

seja, observe que os campos `codigos_raca`, `codigo_vacina`, `codigo_caes`, `codigo_proprietario` – tem uma “chave” cor ouro, isso indica a chave primária da tabela. Iremos nos aprofundar sobre “chave primária” e “chave estrangeira” nos capítulos seguintes. Porém, na Tela 9.5 as entidades/tabelas deveria está em ligação uma com a outra, sendo ligadas por uma linha indicando relacionamento. Vamos, agora fazer acontecer essa situação.

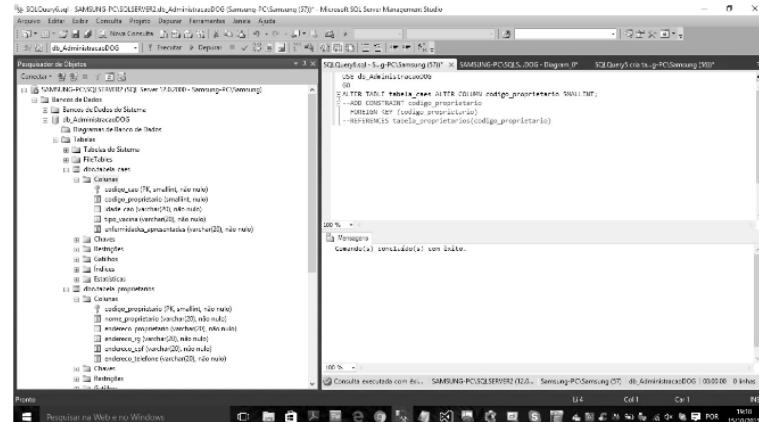
. Vamos concertar alguns erros na hora de criar as tabelas, um desses erros foi o tipo de dados, declaramos que “`codigo_proprietario`” é do tipo “`VARCHAR`” e está errado, por que, na Tabela_Proprietario o campo “`codigo_proprietário`” foi declarado como do tipo `SMALLINT`. Clique “nova consulta” e digite o código fonte SQL:

```

USE db_AdministracaoDOG
GO
ALTER TABLE tabela_caes ALTER COLUMN
codigo_proprietario SMALLINT;

```

- . Clique em executar;
- . Nesse caso alteramos o tipo de dados da coluna codigo_proprietario para SMALLINT. Conforme Tela 9.9 – após atualizar o campo “codigo_proprietario” na Tabela_caes é modificado



Tela 9.9

Código fonte SQL:

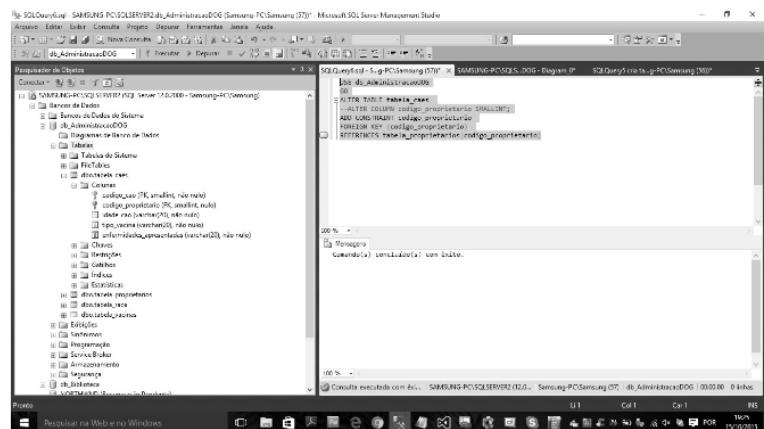
NOTA:

Alterar o campo `codigo_proprietario` na Tabela_caes para ser uma chave estrangeira com referência a sua existência na Tabela_proprietario.

```

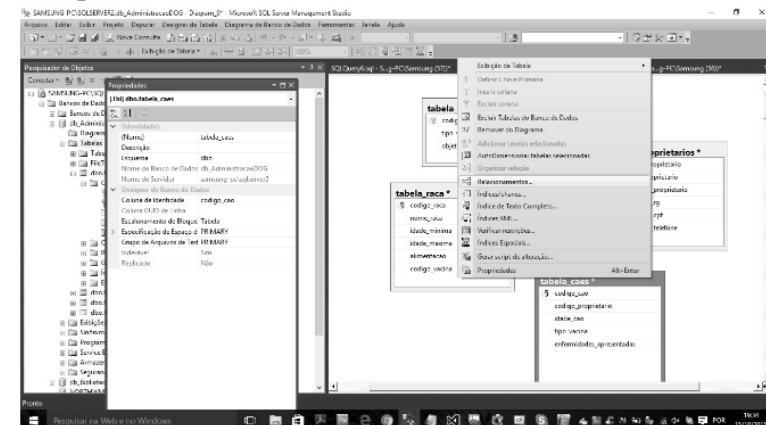
USE db_AdministracaoDOG
GO
ALTER TABLE tabela_caes
ADD CONSTRAINT codigo_proprietario
FOREIGN KEY (codigo_proprietario)
REFERENCES tabela_proprietarios(codigo_proprietario)

```



Tela 10

• Clique em “executar”;

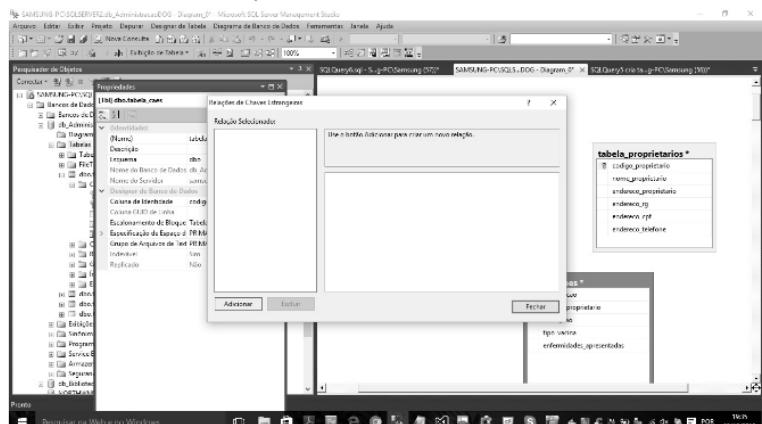


Tela 10.1

• Agora selecione “TABELAS” na aba Pesquisador de Objetos, depois use o segundo botão do mouse e clique em “atualizar”, veja que, agora, clique no sinal de [+] para abrir a cortina e na Tabela_caes observe que tem

uma chave de cor prata no campo `codigo_proprietario`, indicando chave estrangeira;

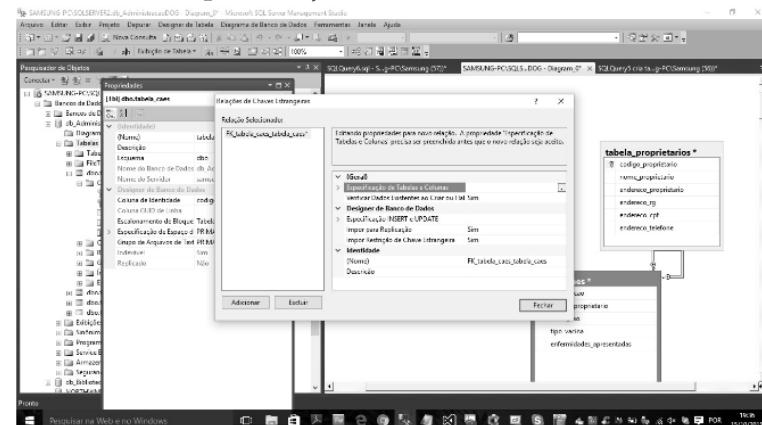
- Selecione a entidade `Tabela_caes`, conforme Tela 10.1;
- Clique no segundo botão do mouse e selecione "relacionamentos";



Tela 10.2

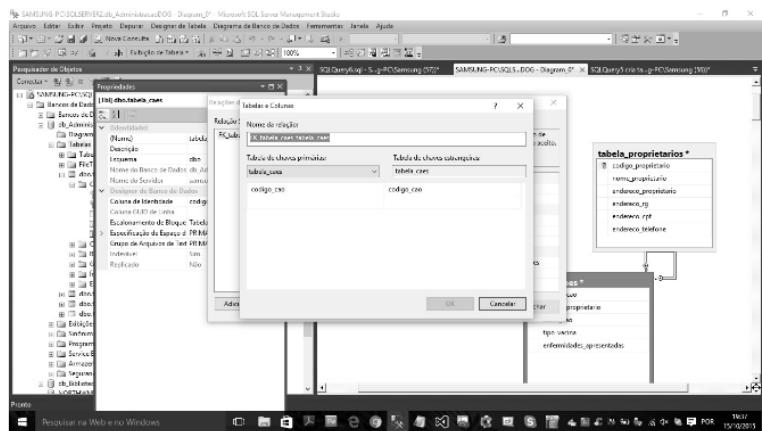
- Clique no botão "adicionar" conforme Tela 10.2;

◦ Selecione "Especificações de Tabelas e Colunas"



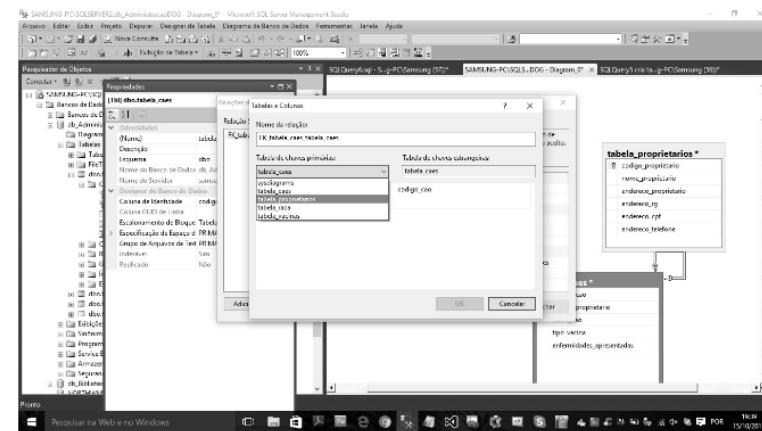
Tela 10.3

◦ Observe Tela 10.4 abre-se opções para seleciona a tabela `caes` com a tabela `proprietario` tendo o campo "codigo_proprietario" como elo de ligação entre as duas entidades/tabelas;



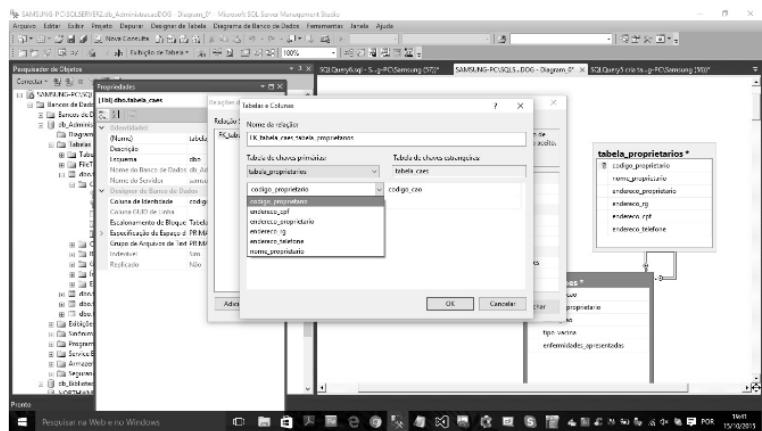
Tela 10.4

. Selecione a Tabela_proprietario conforme a Tela 10.5;



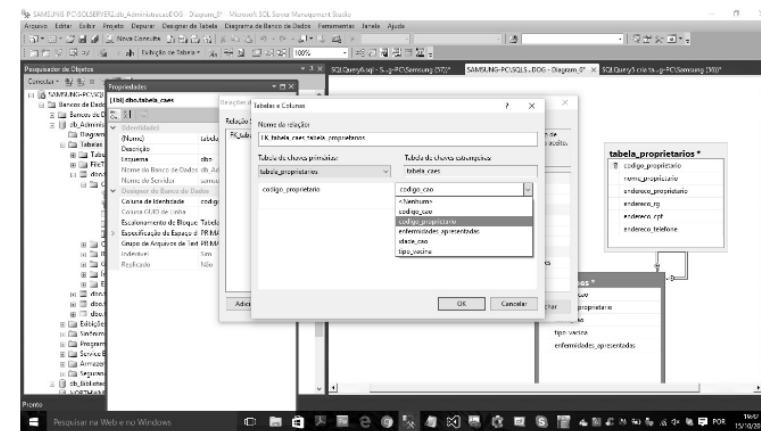
Tela 10.5

. Agora, selecione o campo “codig_proprietario” conforme Tela 10.6

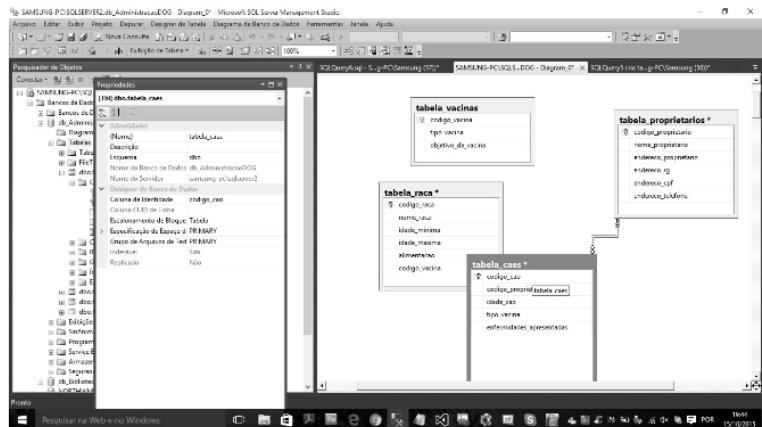


Tela 10.6

- Selecione, também, conforme Tela 10.7 o campo “codigo_proprietario” da Tabela_caes. Clique no botão “ok”, depois clique no botão “fechar” aparece a Tela 10.8 – veja que existe uma linha de relacionamento com chaves indicativas;



Tela 10.7



Tela 10.8

- . A Tela 10.8 indica que foi realizado com sucesso o relacionamento entre as Tabelas “tabela_caes” e “tabela_proprietarios”

NOTA:

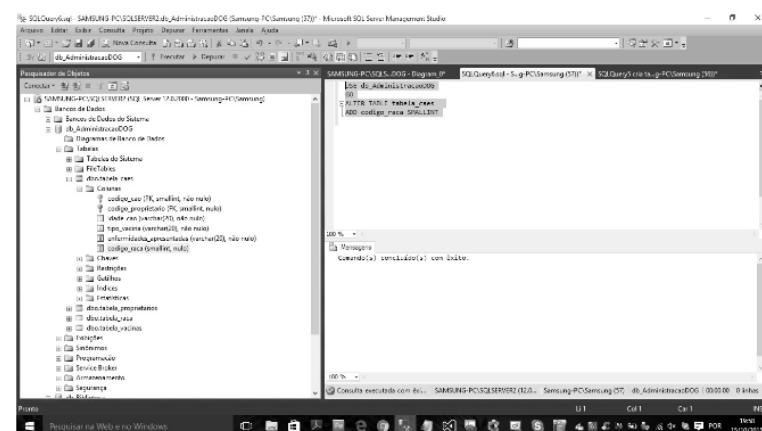
- . Esta faltando na Tabela_caes o campo "codigo_raca" da

Tabela_raca. Clique nova consulta e digite o código fonte

```
SQL: USE db_AdministracaoDOG GO
ALTER TABLE tabela_caes ADD codigo_raca SMALLINT
```

. Clique em executar;

- . Selecione a tabela com segundo botão do mouse e selecione atualiza, veja que aparece "codigo_raca" criado na Tabela_Caes. Tudo certo veja Tela 10.9

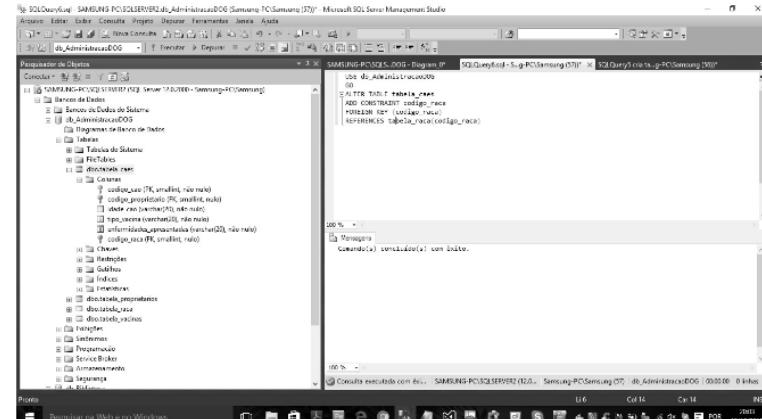


Tela 10.9

Precisamos transforma essa coluna “codigo_raca” em chave estrangeira. Pois, assim a Tabela_caes se relacionará com a Tabela_raca;

Código fonte SQL:

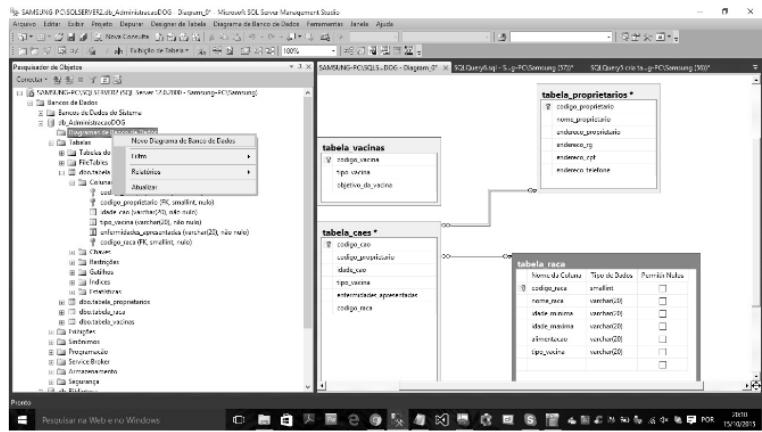
```
USE db_AdministracaoDOG  
GO  
ALTER TABLE tabela_caes  
ADD CONSTRAINT codigo_raca  
FOREIGN KEY (codigo_raca)  
REFERENCES tabela_raca(codigo_raca)
```



Tela 10.10

. Observe que, agora, após atualização da Tabela_caes a coluna “codigo_raca”, também, aparece com a chave prata indicando chave estrangeira com referência a Tabela_raca;

. Pronto selecione Diagramas de Banco de Dados e temo a Tela 10.11



Tela 10.11

- . Adicione todas as tabelas depois de clicar em Novo Diagrama de Banco de

Dados;

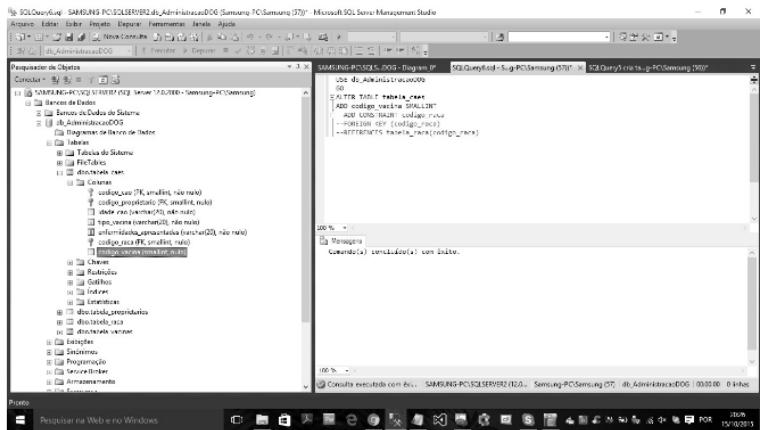
- . Agora, precisamos atribuir o “codigo_vacina” na Tabela_caes, afinal precisamos informar qual vacina foi injetada nesse

animal. Então, faremos os mesmo passos, anteriores onde adicionaremos uma nova coluna na

Tabela_caes;

. Digite o código fonte SQL em “nova consulta”:

```
USE db_AdministracaoDOG
GO
ALTER TABLE tabela_caes ADD codigo_vacina
SMALLINT
```



- Clique em “execute” e veja a Tela 10.12
- Alterar, agora, a coluna “codigo_vacina” para chave estrangeira em

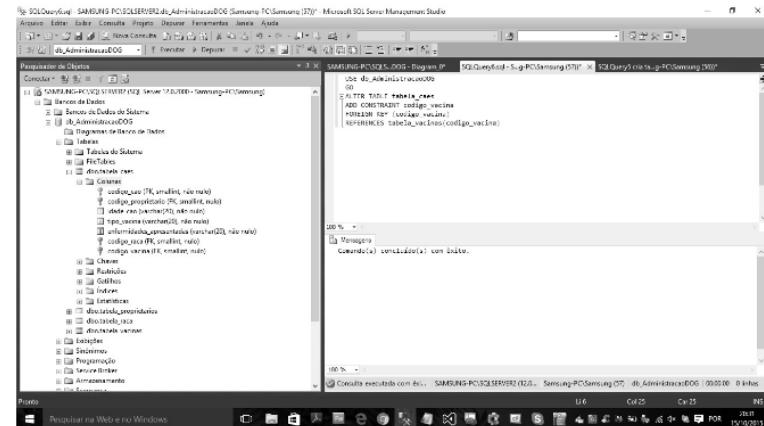
Tabela_caes, clique em “nova consulta” e digite o código fone SQL: USE db_AdministracaoDOG

GO

ALTER TABLE tabela_caes

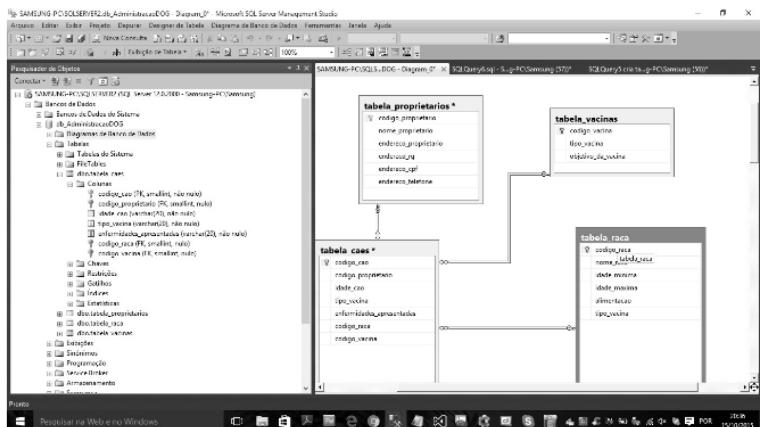
**ADD CONSTRAINT codigo_vacina
FOREIGN KEY (codigo_vacina)
REFERENCES tabela_vacinas(codigo_vacina)**

- Clique no botão “executar”. O resultado está na Tela 10.13



Tela 10.13

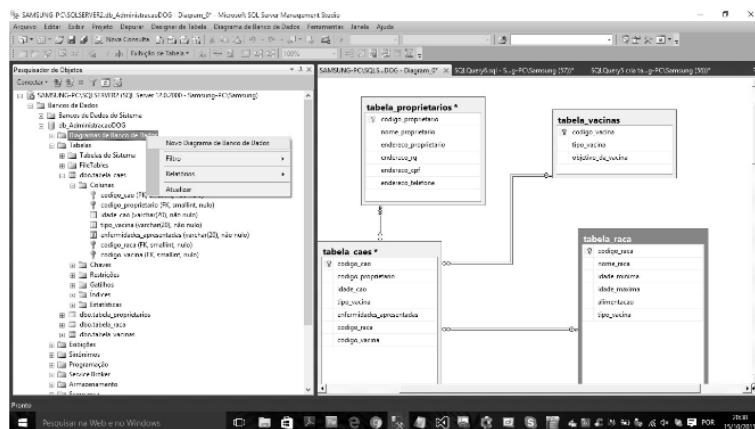
. Observe que a Tabela_caes está se relacionando com todas as outras, e por isso a presença das chaves estrangeiras codigo_vacina, codigo_raca, codigo_proprietario, temos a Tela 10.14;



Tela 10.14

. Lembrando que para ter a Tela 10.14 é necessário

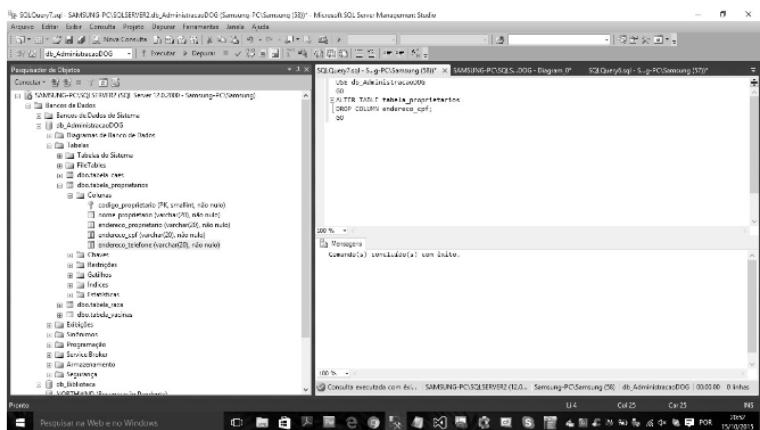
acionar em Diagrama de Banco de Dados o segundo botão do mouse e clicar em “novo diagrama de banco de dados” como está na Tela 10.15



Tela 10.15

. Apure o seu olhômetro veja que o nome das colunas “endereco_rg”, “endereco_cpf”, “endereco_telefone” não

estão corretos. O que deveria ser “rg”, “cpf” e “telefone”, vamos excluir todas essas colunas e demos adiná-las novamente, agora, correto vejta a Tela 10.16



Tela 10.16

• Clique em nova consulta e digite o código fonte SQL:
USE db_AdministracaoDOG

GO

ALTER TABLE tabela_proprietarios DROP COLUMN endereco_telefone; GO

• Substitua o código fonte para excluir a coluna “endereco_cpf” USE db_AdministracaoDOG

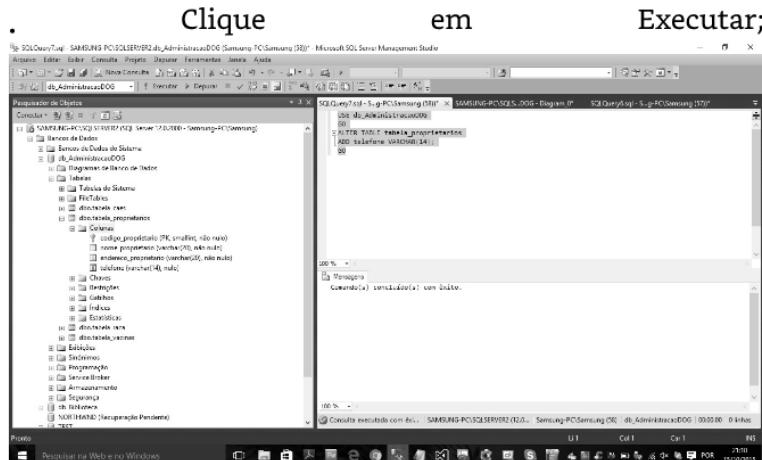
GO

ALTER TABLE tabela_proprietarios
DROP COLUMN endereco_cpf;

GO

• Agora adicionaremos as novas colunas na Tabela_proprietarios. Clique nova consulta e digite:

USE db_AdministracaoDOG GO
ALTER TABLE tabela_proprietarios ADD telefone
VARCHAR(14); GO



Tela 10.17

- . Observe que apareceu a nova coluna “telefone” na Tela 10.17;
- . Vamos adicionar a coluna “rg” porém o tipo de dados é numérico sendo declarado no código fonte SQL “NUMERIC(14)” de tamanho 14;

Execute;

USE db_AdministracaoDOG

GO

```
ALTER TABLE tabela_proprietarios ADD rg
NUMERIC(14);
```

GO

. Entender o que é uma tabela e como ela se relaciona com os dados armazenados no banco de dados; colunas/linhas/registros

NOTA:

Todas alterações até aqui realizadas não podem acontecer, portanto, nos Capítulos a seguir estudaremos melhor a “Estruturação de Dados” a fim de evitar esse desgaste, fazer e refazer. Demonstrando um amadorismo na “Modalidade de Dados”. Para isso o Gerenciador de

DBA – Banco de Dados SQL Server trabalha sempre com um Analista de Sistemas mais experiente e mais vivido na área de Projetos de Desenvolvimento de Dados.

“NOTA

Tabela e Organização Index - Descreve a organização de páginas de tabelas e índices.

Estruturas Heap - Explica como as linhas de dados são armazenados em uma pilha e como eles são acessados e mantidos

Estruturas de índice em cluster - Explica como as linhas de dados são armazenados em um índice de cluster e como eles são acessados e mantidos.

Estruturas de índice Nonclustered - Explica como linhas de índice são armazenados em um índice sem clusters e como eles são acessados e mantidos.”⁵

Exemplos de ALTER TABLE:

-
-- Inclua a coluna DATA_NASCIMENTO na tabela ALUNO do tipo string, de tamanho 10 caracteres

-
ALTER TABLE ALUNO ADD DATA_NASCIMENTO
VARCHAR(10);

-
-- Altere a coluna DATA_NASCIMENTO para
NASCIMENTO e seu tipo de dado para DATE

-
ALTER TABLE ALUNO CHANGE DATA_NASCIMENTO
NASCIMENTO DATE NULL;

-
-- Crie um novo índice na tabela ALUNO, para o campo

ALUNO

-

ALTER TABLE ALUNO ADD INDEX
INDEX_ALUNO(ALUNO);

-- Inclua o campo EMAIL na tabela INSTRUTOR, com
tamanho de 100 caracteres

-

[5 MSDN, Estrutura de dados e índice, https://msdn.microsoft.com/en-us/library/ms180978.aspx, 04/10/2015](https://msdn.microsoft.com/en-us/library/ms180978.aspx)

ALTER TABLE INSTRUTOR ADD EMAIL VARCHAR(100);

-

-- Crie um novo índice na tabela CURSO, para o campo
INSTRUTOR

-

ALTER TABLE CURSO ADD INDEX
INDEX_INSTRUTOR(INSTRUTOR);

-

-- Remova o campo EMAIL da tabela INSTRUTOR

-

ALTER TABLE INSTRUTOR DROP EMAIL;

. Entender conceitos do banco de dados relacional

. Entender o que é um banco de dados relacional,
a necessidade de sistemas de gerenciamento de banco
de dados relacional (RDBMS) e como as relações são
estabelecidas

**. Entender a linguagem de manipulação de dados
(DML)**

. Entender o que é a DML e sua função nos bancos de

dados

. **Entender a linguagem de definição de dados (DDL)**

- . Entender como o T-SQL pode ser usado para criar objetos do banco de dados, como tabelas e exibições

1.1.1.2 - Criar objetos do banco de dados (20-25%)

- . Escolher tipos de dados

- . Entender o que são tipos de dados, por que eles são importantes e como afetam os requisitos de armazenamento

- . Entender tabelas e como criá-las

- . Objetivo das tabelas; criar tabelas em um banco de dados usando a sintaxe adequada de ANSI SQL

- . Criar exibições

- . Entender quando usar exibições e como criar uma exibição usando o T-SQL ou designer gráfico

“Veja o resultado na Figura 1.6 - Tabela Produtos.

Listagem 1: Criando uma view código fonte: CREATE

VIEW vwProdutos

AS

```
SELECT IdProduto AS Código , Nome AS Produto  
, Fabricante  
, Quantidade  
, VlUnitario AS [Valor Unitário] , Tipo
```

FROM

Produtos

Id	Nome	Fabricante	Quantidade	VlUnitario	Tipo
1	Playstation 3	Sony	100.00	1999.00	Console
2	Core 2 Duo 4GB Ram 500GB HD	Dell	200.00	1899.00	Notebook
3	Xbox 360 120GB	Microsoft	350.00	1299.00	Console
4	GT-I6220 Quad Band	Samsung	300.00	499.00	Celular
5	iPhone 4 32GB	Apple	50.00	1499.00	Smartphone
6	Playstation 2	Sony	100.00	399.00	Console
7	Sofá Estofado	Coréia	200.00	499.00	Sofá
8	Armário de Serviço	Aracaju	50.00	129.00	Armário
9	Refrigerador 420L	CCE	200.00	1499.00	Refrigerador
10	Wii 120GB	Nintendo	250.00	999.00	Console

Figura 1.6 - Tabela Produtos

Para consultarmos os dados na view usamos o comando **SELECT**, da mesma forma que se estivéssemos fazendo uma consulta em uma tabela comum.”⁶

6 Wellington B. Camargo, Conceitos e Criação da View - Views no SQL Server - Parte

1 <http://www.devmedia.com.br/conceitos-e-criacao-da-view-views-no-sql-server-parte>

“A Listagem 2Listagem 2 exibe a consulta na view, e a Figura 2.1Figura 2.1Figura 2.1 nos exibe o resultado desta

Listagem 2

consulta.

Listagem 2: Consultando uma View

```
SELECT * FROM vwProdutos
```

Código	Produto	Fabricante	Quantidade	Valor Unitário	Tipo
1	Playstation 3	Sony	100.00	1999.00	Console
2	Core 2 Duo 4GB Ram 500GB HD	Dell	200.00	1899.00	Notebook
3	Xbox 360 120GB	Microsoft	350.00	1299.00	Console
4	GT-I6220 Quad Band	Samsung	300.00	499.00	Celular
5	iPhone 4 32GB	Apple	50.00	1499.00	Smartphone
6	Playstation 2	Sony	100.00	399.00	Console
7	Sofá Estofado	Coréia	200.00	499.00	Sofá
8	Armário de Serviço	Aracaju	50.00	129.00	Armário
9	Refrigerador 420L	CCE	200.00	1499.00	Refrigerador
10	Wii 120GB	Nintendo	250.00	999.00	Console

Figura 2.1

“Outros exemplos de VIEW

-

-- Crie uma visão que traga o título e preço somente dos cursos de programação da FocoSite

-

```
CREATE VIEW cursos_programacao AS SELECT curso,
valor FROM curso INNER JOIN tipo ON curso.tipo
= tipo.codigo WHERE tipo.tipo = 'programação';
```

1/22390#ixzz3oezwzKVC, 15/10/2015”

- -- Crie uma visão que traga os títulos dos cursos, tipo do curso e nome do instrutor

- ```
CREATE VIEW cursos_programacao_completo AS
SELECT curso, valor, instrutor.instrutor from curso
INNER JOIN tipo ON curso.tipo = tipo.codigo INNER JOIN
instrutor ON curso.instrutor = instrutor.codigo WHERE
tipo.tipo = 'programação';
```

- -- Crie uma visão que exiba os pedidos realizados, informando o nome do aluno, data e código do pedido

- ```
CREATE VIEW pedidos_realizados AS
SELECT pedido.codigo, pedido.datahora, aluno.aluno
from pedido INNER JOIN aluno ON
```

pedido.aluno = aluno.codigo;

A. Usando um simples CREATE VIEW

O exemplo a seguir cria uma exibição usando um simples SELECT comunicado. Uma visão simples é útil quando uma combinação de colunas é consultado com freqüência. Os dados à partir dessa visão vem das HumanResources.Employee e Person.Person são tabelas do banco de dados

AdventureWorks2012. Os dados fornecem nome e contratar informações de data para os funcionários da Adventure Works Cycles. A visão poderia ser criado para a pessoa encarregada de aniversários de trabalho de rastreamento, mas sem dar essa pessoa o acesso a todos os dados dessas tabelas.

```
1. USE AdventureWorks2012 ;
2. GO
3. IF OBJECT_ID ('hiredate_view', 'V') IS NOT NULL
4. DROP VIEW hiredate_view ;
5. GO
6. CREATE VIEW hiredate_view
7. AS
8. SELECT p.FirstName, p.LastName, e.BusinessEntityID,
e.HireDate
FROM
HumanResources.EmployeeHumanResources.Employee
AND JOIN Person.PersonPerson.Person AS p ON
e.BusinessEntityID = p.BusinessEntityID ;
9. GO
```

COMENTÁRIO:

A compreensão do script acima mencionado em SQL entende-se como:

1. Abra a conexão com o banco de dados nomeado "AdventureWorks2012";
2. Vá para a próxima linha;
3. Se não é vazio ou nulo (então existe) a VIEW hiredate_view, então elimine DROP VIEW;
- 5.
6. Crie uma nova VIEW, com novos parâmetros, intitulada hiredate_view;
- 7.
8. Seleciona a Tabela HumanResources.EmployeeHumanResources.Employee E a tabela Person.Person, porém retornando os registros na publicação das colunas p.FirstName, p.LastName, e.BusinessEntityID, e.HireDatep.FirstName, p.LastName, e.BusinessEntityID, e.HireDate;

**p.FirstName, p.LastName, e.BusinessEntityID,
e.HireDate**

atendendo a condição de JOIN, ou seja, traga o que existe em ambas as tabelas HumanResources.EmployeeHumanResources.Employee E na tabela Person.Person. SPerson.Person. Se existir igualdade nos campos (e.BusinessEntityID = p.BusinessEntityID) tanto em uma Tabela como na outra igualdade nos campos e.BusinessEntityID = p.BusinessEntityID

NOT NULL

```
DROP VIEW Purchasing.PurchaseOrderReject ;
GO
CREATE VIEW Purchasing.PurchaseOrderReject
WITH ENCRYPTION
AS
SELECT PurchaseOrderID, ReceivedQty, RejectedQty,
RejectedQty / ReceivedQty AS RejectRatio, DueDate
FROM Purchasing.PurchaseOrderDetail
WHERE RejectedQty / ReceivedQty > 0
AND DueDate > CONVERT(DATETIME,'20010630',101) ;
GO
```

B. Usando WITH ENCRYPTION

O exemplo a seguir usa a criptografia com opção e shows de colunas computadas, rebatizado colunas e várias colunas.

Aplica-se : SQL Server 2008 através de SQL Server 2016.

```
USE AdventureWorks2012 ;
GO
IF OBJECT_ID ('Purchasing.PurchaseOrderReject', 'V') IS
```

C. Usando WITH CHECK OPTION

O exemplo a seguir mostra uma vista nomeada SeattleOnly que faz referência a cinco mesas e permite modificações de dados a ser aplicado apenas aos empregados que vivem em Seattle.

```
USE AdventureWorks2012 ;
GO
IF OBJECT_ID ('dbo.SeattleOnly', 'V') IS NOT NULL DROP
VIEW dbo.SeattleOnly;

GO
CREATE VIEW dbo.SeattleOnly
AS
SELECT p.LastName, p.FirstName, e.JobTitle, a.City,
sp.StateProvinceCode FROM HumanResources.Employee
e
INNER JOIN Person.Person p
ON p.BusinessEntityID = e.BusinessEntityID
```

```
INNER JOIN Person.BusinessEntityAddress bea
ON bea.BusinessEntityID = e.BusinessEntityID
INNER JOIN Person.Address a
ON a.AddressID = bea.AddressID
INNER JOIN Person.StateProvince sp
ON sp.StateProvinceID = a.StateProvinceID

WHERE a.City = 'Seattle'
WITH CHECK OPTION ;
GO
```

D. Usando funções internas dentro de uma visão

O exemplo a seguir mostra uma definição de exibição que inclui uma função builtin. Quando você usa funções, você deve especificar um nome de coluna para a coluna derivada.

```
1. USE AdventureWorks2012 ;
2. GO
3. IF OBJECT_ID ('Sales.SalesPersonPerform', 'V') IS NOT
NULL DROP VIEW Sales.SalesPersonPerform ;
4. GO
5. CREATE VIEW Sales.SalesPersonPerform
6. AS
7. SELECT TOP (100) SalesPersonID, SUM(TotalDue) AS
TotalSales FROM Sales.SalesOrderHeader
WHERE OrderDate >
CONVERT(DATETIME,'20001231',101) GROUP BY
SalesPersonID;
GO
```

COMENTÁRIO:

A compreensão do script acima mencionado em SQL entende-se como:

1. Abra a conexão com o banco de dados nomeado "AdventureWorks2012";
2. Vá para a próxima linha;
3. Se não é vazio ou nulo (então existe) a VIEW Sales.SalesPersonPerform, então elimine DROP VIEWI ;
- 4.
5. Crie uma nova VIEW, com novos parâmetros, intitulada Sales.SalesPersonPerform
6. desde que
7. Seleccione a Tabela Sales.SalesOrderHeader retornando 100 reistros e publicando as colunas SalesPersonID; porém, somando SUM(TotalDue). Com filtro de data maior que 31/12/2000, sendo, também, uma publicação organizada por seus grupos com base no campo SalesPersonID

E. Usando dados particionado

O exemplo a seguir usa tabelas nomeadas SUPPLY1, SUPPLY2, SUPPLY3, e SUPPLY4. Estes quadros correspondem às tabelas fornecedor de quatro escritórios, localizados em diferentes países / regiões.

--Create the tables and insert the values.

- Criar as tabelas e insira valores

```
CREATE TABLE dbo.SUPPLY1 (
    supplyID INT PRIMARY KEY CHECK (supplyID
        BETWEEN 1 and 150), supplier CHAR(50)
);
CREATE TABLE dbo.SUPPLY2 (
    supplyID INT PRIMARY KEY CHECK (supplyID
        BETWEEN 151 and 300), supplier CHAR(50)
);
CREATE TABLE dbo.SUPPLY3 (
    supplyID INT PRIMARY KEY CHECK (supplyID
```

```
BETWEEN 301 and 450), supplier CHAR(50)
);
CREATE TABLE dbo.SUPPLY4 (
    supplyID INT PRIMARY KEY CHECK (supplyID
        BETWEEN 451 and 600), supplier CHAR(50)
);
```

GO

```
INSERT dbo.SUPPLY1 VALUES ('1', 'CaliforniaCorp'), ('5',
'BraziliaLtd'), ('231', 'FarEast'), ('280', 'NZ')
, ('321', 'EuroGroup'), ('442', 'UKArchip')
, ('475', 'India'), ('521', 'Afrique');
```

GO

--Create the view that combines all supplier tables.

- Cria a visualização combinando com vários campos da tabelas

```
CREATE VIEW dbo.all_supplier_view WITH SCHEMABINDING
AS
    – Tabela SUPPLY1
    SELECT supplyID, supplier FROM dbo.SUPPLY1

    UNION ALL

    – Tabela SUPPLY2
    SELECT supplyID, supplier FROM dbo.SUPPLY2

    UNION ALL

    – Tabela SUPPLY3
    SELECT supplyID, supplier
    FROM dbo.SUPPLY3
    UNION ALL
    – Tabela SUPPLY4
```

WITH
SCHEMABINDING
AS
SELECT supplyID, supplier
FROM dbo.SUPPLY4;"⁷

. Criar procedimentos armazenados e funções
. Selecionar, inserir, atualizar ou excluir dados
1.1.1.3 - Manipular dados (25–30%)

. Selecionar dados
. Utilizar consultas SELECT para extrair dados de uma tabela, extrair dados usando junções, combinar conjuntos de resultados usando UNION e INTERSECT

“ Exemplos:
SELECT codigo, nome, salario FROM funcionarios
- Nesse caso acima será selecionado, somente, os campos código, nome e salario de todos da tabela funcionários.

7 MSDN, <https://msdn.microsoft.com/en-us/library/ms187956.aspx>,

15/10/2015

```
SELECT * FROM empregado WHERE CODIGO in(SELECT codempregado FROM vendas GROUPY BY codempregado HAVING MAX(valortotal)>50000)
```

_ No código acima retornamos os empregados que venderam um montante maior que R\$ 50.000,00.

-

```
SELECT      *      FROM      empregado      WHERE  
EXISTS(SELECT      *      FROM      pedidos  
WHERE      empregado.codigo=pedidos.codempregado  
GROUP      BY      PEDIDOS.codempregado      HAVING  
MAX(pedidos.valor_total)>50000)
```

- Neste comando são retornados o nome e o departamento dos funcionários cuja soma de vendas

ultrapassaram R\$ 50.000,00

-- Exibir todas as informações de todos os alunos SELECT * FROM ALUNO;

-- Exibir somente o título de cada curso SELECT CURSO FROM CURSO;

-- Exibir somente o título e valor de cada curso cujo preço seja maior que 200 SELECT CURSO, VALOR FROM CURSO WHERE VALOR > 200;

-- Exibir somente o título e valor de cada curso cujo preço seja maior que 200 e menor que 300

1. SELECT CURSO, VALOR FROM CURSO WHERE VALOR > 200 AND VALOR < 300; 2. SELECT CURSO, VALOR FROM CURSO WHERE VALOR BETWEEN 200 AND 300;

-- Exibir as informações da tabela PEDIDOS para os pedidos realizados entre 15/04/2010 e 18/04/2010

15/04/2010 e 18/04/2010
04-19';

-- Outra solução para o exercício seria esta

-- Outra solução para o exercício seria esta
19';

-- Exibir as informações da tabela PEDIDOS para os pedidos realizados na data de 15/04/2010

```
SELECT * FROM PEDIDO WHERE DATE(DATAHORA) =  
'2010-04-15';8
```

. Inserir dados

. Entender como os dados são inseridos em um banco de dados, como usar instruções INSERT

-- Tabela TIPO:

```
INSERT INTO TIPO (CODIGO, TIPO) VALUES (1, 'Banco de dados');
```

```
INSERT INTO TIPO VALUES (2, 'Programação');
```

```
INSERT INTO TIPO VALUES (3, 'Modelagem de dados');
```

Comentário--> na tabela TIPO temos os campos CODIGO, TIPO sendo atribuído valores 1, BANCO DE DAO'S ...

-- Tabela INSTRUTOR:

```
INSERT INTO INSTRUTOR  
(CODIGO, INSTRUTOR, TELEFONE) VALUES (1, 'André Milani', '1111-1111');
```

⁸ Ângelo M. Almeida dos Santos, Programação ASP Clássico, Ed. Viena, 2015, 15/10/2015

```
INSERT INTO INSTRUTOR VALUES (2, 'Carlos Tosin',  
'1212-1212');
```

Comentário--> na tabela INSTRUTOR temos os campos

CODIGO, INSTRUTOR, TELEFONE, sendo atribuído valores 1, 'André Milani', '1111-1111' ...

-- Tabela CURSO:

```
INSERT INTO CURSO (CODIGO, CURSO, TIPO, INSTRUTOR, VALOR) VALUES (1, 'Java Fundamentos', 2, 2, 270);
```

```
INSERT INTO CURSO VALUES (2, 'Java Avançado', 2, 2, 330);
```

```
INSERT INTO CURSO VALUES (3, 'SQL Completo', 1, 1, 170);
```

```
INSERT INTO CURSO VALUES (4, 'Php Básico', 2, 1, 270);
```

Comentário--> na tabela CURSO temos os campos CODIGO, CURSO, TIPO, INSTRUTOR, VALOR, sendo atribuído valores 1, 'Java Fundamentos', 2, 2, 270...

-- Tabela ALUNO:

```
INSERT INTO ALUNO (CODIGO, ALUNO, ENDERECO, EMAIL) VALUES (1, 'José', 'Rua XV de Novembro ', 'jose@focosite.com.br');
```

```
INSERT INTO ALUNO (CODIGO, ALUNO, ENDERECO, EMAIL) VALUES (2, 'Wagner', 'Av. Paulista', 'wagner@focosite.com.br');
```

```
INSERT INTO ALUNO (CODIGO, ALUNO, ENDERECO, EMAIL) VALUES (3, 'Emílio', 'Rua Lajes , ap: 23', 'emilio@focosite.com.br');
```

```
INSERT INTO ALUNO (CODIGO, ALUNO, ENDERECO, EMAIL) VALUES (4, 'Cris', 'Rua Tauney 0999', 'cris@focosite.com.br');
```

```
INSERT INTO ALUNO (CODIGO, ALUNO, ENDERECO, EMAIL) VALUES (5, 'Regina', 'Rua Salles 12305', 'regina@focosite.com.br');
```

```
INSERT INTO ALUNO (CODIGO, ALUNO, ENDERECO, EMAIL) VALUES (6, 'Fernando', 'Av. Central 12430',
```

```
'fernando@focosite.com.br');
```

Comentário--> na tabela ALUNO temos os campos CODIGO, ALUNO, ENDERECO, EMAIL sendo atribuído valores 5, 'Regina', 'Rua Salles 12305', 'regina@focosite.com.br'...

-- Tabela PEDIDO:

```
INSERT INTO PEDIDO (CODIGO, ALUNO, DATAHORA)
VALUES (1, 2, '2010-04-15 11:23:32');
```

```
INSERT INTO PEDIDO VALUES (2, 2, '2010-04-15
14:36:21');
```

```
INSERT INTO PEDIDO VALUES (3, 3, '2010-04-16
11:17:45');
```

```
INSERT INTO PEDIDO VALUES (4, 4, '2010-04-17
14:27:22');
```

```
INSERT INTO PEDIDO VALUES (5, 5, '2010-04-18
11:18:19');
```

```
INSERT INTO PEDIDO VALUES (6, 6, '2010-04-19
13:47:35');
```

```
INSERT INTO PEDIDO VALUES (7, 6, '2010-04-20
18:13:44');
```

Comentário--> na tabela PEDIDO temos os campos CODIGO, ALUNO, DATAHORA sendo atribuído valores 7, 6, '2010-04-20 18:13:44'...

-- Tabela PEDIDO_DETALHE:

```
INSERT INTO pedido_detalhe (PEDIDO, CURSO, VALOR)
VALUES (1, 1, 270); INSERT INTO pedido_detalhe
VALUES (1, 2, 330);
```

```
INSERT INTO pedido_detalhe VALUES (2, 1, 270);
```

```
INSERT INTO pedido_detalhe VALUES (2, 2, 330);
```

```
INSERT INTO pedido_detalhe VALUES (2, 3, 170);
```

```
INSERT INTO pedido_detalhe VALUES (3, 4, 270);
```

```
INSERT INTO pedido_detalhe VALUES (4, 2, 330);
```

```
INSERT INTO pedido_detalhe VALUES (4, 4, 270);
INSERT INTO pedido_detalhe VALUES (5, 3, 170);
INSERT INTO pedido_detalhe VALUES (6, 3, 170);
Comentário--> na tabela PEDIDO_DETALHE temos os campos PEDIDO,CODIGO, VALOR sendo atribuído valores 6, 3, 170..."9
```

- . Atualizar dados
- . Entender como os dados são atualizados em um banco de dados e como gravar os dados atualizados no banco de dados usando as instruções UPDATE apropriadas, atualizar usando uma tabela

"Outro exemplo: funcionários de menor salário receberão aumento de 10%

```
UPDATE empregados SET SALARIO=salario *1.1 WHERE SALARIO=(SELECT MIN(salario) FROM EMPREGADOS)
-- Altere o endereço do aluno JOSÉ para 'Av. Brasil 778'
```

```
UPDATE ALUNO SET ENDERECO = 'Av. Brasil 778'
WHERE CODIGO = 1;
-- Altere o e-mail do aluno ANGELO para 'angeloalmeida@focosite.com.br' UPDATE ALUNO SET EMAIL = 'angeloalmeida@focosite.com.br' WHERE CODIGO = 4;
```

```
-- Aumente em 10% o valor dos cursos abaixo de 300
UPDATE CURSO SET VALOR = VALOR * 1.1 WHERE VALOR < 300; -- Altere o nome do curso de ASP Básico para ASP Fundamentos-
```

9 Ângelo M. Almeida dos Santos, Programação ASP Clássico, Ed. Viena, 2015, 15/10/2015

```
UPDATE CURSO SET CURSO = 'ASP Fundamentos'
WHERE CURSO = 'ASP Básico';10
```

- . Excluir dados
- . Excluir dados de uma ou várias tabelas, garantir a

integridade dos dados e referencial usando transações

1.1.1.4 - Entender armazenamento de dados (15–20%)



Fonte da imagem¹¹

Quando se fala em armazenagem de algo é preciso definir onde, como e o quê armazenar. Bem assim, quando uma viagem precisamos saber o ponto de partida, ponto de parada e ponto de chegada, para isso é necessário conhecer o roteiro. Se um cozinheiro ou confeiteiro está trabalhando é preciso saber os ingredientes, mas, também, a quantidade de cada ingrediente. É necessário conhecer para melhor fazer.

10 Ângelo M. Almeida dos Santos, Programação ASP Clássico, Ed. Viena, 2015, 15/10/2015 11 http://www.trssistemas.com.br/uploads/banner_banco.png, 12/10/2015

A realização de qualquer atitude próspera se dá em “normalizar”. Mas, o que é normalização?

CITO:

“Diz-se que uma tabela num banco de dados relacional está numa certa forma normal se satisfaz certas condições. O trabalho original de Edgar F. Codd definiu três dessas formas, mas existem hoje outras formas normais geralmente aceitas. Damos aqui uma curta panorâmica informal das mais comuns. Cada forma normal listada abaixo representa uma condição mais forte que a precede na lista. Para a maioria dos efeitos práticos, considera-se que as bases de dados estão normalizadas se aderirem à terceira forma normal.

- . Primeira Forma Normal (ou 1FN1FN) requer que todos os valores de colunas em uma tabela sejam atômicosatômicos (exemplo: um número é um átomo, enquanto uma lista ou um conjunto não o são). A normalização para a primeira forma normal elimina grupos repetidos, pondo-os cada um em uma tabela separada, conectando-os com uma chave primária ou estrangeira;
- . Segunda Forma Normal (ou 2FN2FN) requer que não haja dependência funcional não-trivial de um atributo que não seja a chave, em parte da chave candidata;
- . Terceira Forma Normal (ou 3FN3FN) requer não haver dependências funcionais não-triviais de atributos que não sejam chave, em qualquer coisa exceto um superconjunto de uma chave candidata;
- . Forma Normal de Boyce-CoddBoyce-Codd (ou(ou BCNF)BCNF) requer que não exista nenhuma dependência funcional não-trivial de atributos em algo mais do que um superconjunto de uma chave candidata. Neste estágio, todos os atributos são dependentes de uma chave, de uma chave inteira e de nada mais que uma chave (excluindo dependências triviais, como $A \rightarrow A$);
- . Quarta Forma Normal (ou 4FN4FN) requer que não exista nenhuma dependência multivalorada não-trivial de conjuntos de atributo em algo mais de que um superconjunto de uma chave candidata;
- . **Quinta Forma Normal** (ou 5FN5FN ou PJ/NFPJ/ NF) requer que não exista dependências de joins (associações) não triviais que não venham de restrições chave;
joins
 - .

Domain-Key Normal Form (ou DK/NFDK/NF) requer que todas as restrições sigam os **DK/NF** domínios e restrições chave.”¹²

- . Entender normalização
- . Entender os motivos para normalização, os cinco níveis mais comuns de normalização, como normalizar um banco de dados para o terceiro formulário normal da chave primária.”

12 Wikipedia, https://pt.wikipedia.org/wiki/Normaliza%C3%A7%C3%A3o_de_dados, 12/10/2015

“NOTA

Princípios do Design de Banco de Dados:

- A Primeira forma normalA Primeira forma normal (**1NF**) estabelece algumas regras básicas para um

A Primeira forma normal

banco de dados: eliminar colunas duplicativas na mesma tabela.

- A Segunda forma normalA Segunda forma normal (**2NF**) = **1NF** + remove subconjuntos de dados que se aplicam a múltiplas linhas de uma tabela e coloca-as em tabelas separadas.
- A Terceira forma normalA Terceira forma normal (**3NF**) = **2NF** + remove colunas que não são dependentes”¹³

A Equipe de Suporte da Microsoft disponibilizou um artigo “Descrição dos conceitos básicos de normalização de banco de dados”. Haja vista que o objetivo principal desse livro é o EXAME 98-364 – Fundamentos de Banco de Dados, então, precisamos conhecer o que a Microsoft juntamente com seus Especialista vão cobrar nesse

EXAME. Sendo assim, precisamos nos familiarizar com a linguagem expositiva vindo da Microsoft.

CITO:

“1.1.1.4.1 Descrição de Normalização”

Normalização é o processo de organizar os dados em um banco de dados. Isso inclui a criação de tabelas e estabelecer relacionamentos entre essas tabelas de acordo com as regras criadas para proteger os dados e para tornar o banco de dados mais

13 MSDN, Princípios do Design de Banco de Dados, <https://msdn.microsoft.com/en-us/library/aa200276>, 03/10/2015

flexíveis, eliminando a redundância e dependência inconsistente.

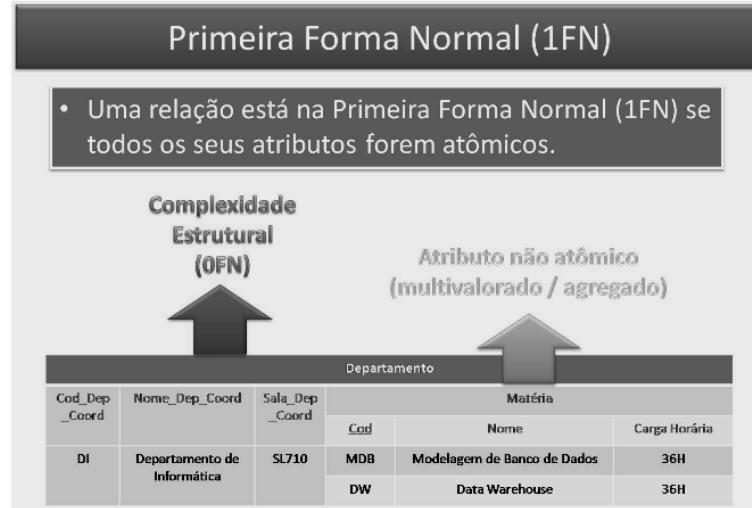
Dados redundantes desperdiça espaço em disco e cria

problemas de manutenção. Se os dados que existem em mais de um local devem ser alterados, os dados devem ser alterados da mesma maneira em todos os locais. Uma alteração de endereço de cliente é muito mais fácil de implementar se esses dados são armazenados somente na tabela cliente e lugar no banco de dados.

O que é uma “dependência inconsistente?” Embora seja intuitivo para um usuário examinar a tabela de clientes para o endereço de um cliente específico, pode não fazer sentido procurar lá para o salário do funcionário que chama no cliente. Salário do funcionário está relacionado, ou dependentes, o funcionário e, portanto, deve ser movido para a tabela Funcionários. Dependências inconsistentes podem dificultar dados para acesso porque o caminho para localizar os dados pode estar ausente ou danificado.

Existem algumas regras de normalização de banco de dados. Cada regra é chamada de um “formulário normal”. Se a primeira regra é observada, o banco e dados é considerado na “primeira forma normalizada”. Se as primeiras três regras forem observadas, o banco de dados é considerado na “terceira forma normalizada”. Embora outros níveis de normalização são possíveis, a terceira forma normalizada é considerada o mais alto nível...

1.1.1.4.1.1 - Primeira Forma Normalizada - 1FN



Fonte imagem¹⁴

- . Elimine grupos de repetição em tabelas individuais;
- . Crie uma tabela separada para cada conjunto de dados relacionados;

- . Identifica cada conjunto de dados relacionados com uma chave primária.

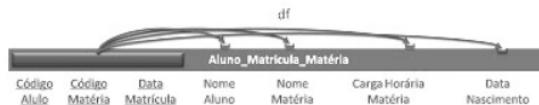
1414
h/
Banco_de_Dados_Normalizacao_Forma_Normal_Primeira_1FN_01.jp
g, 12/10/2015

Não use vários campos em uma única tabela para armazenar dados semelhantes. Por exemplo, para rastrear um item de estoque pode vir de duas fontes possíveis, um registro de estoque pode conter campos para código do fornecedor 1 e 2 do código e fornecedor.

O que acontece quando você adiciona um terceiro fornecedor? Adicionar um campo não é a resposta; ela requer modificações de programa e a tabela não acomoda suavemente um número dinâmico de fornecedores. Em vez disso, coloque todas as informações de fornecedor em uma tabela separada denominada fornecedores e estoque de vínculo aos fornecedores com uma tecla de número de item, ou fornecedores de estoque com uma chave de código de fornecedor.

1.1.1.4.1.2 - Segunda Forma Normalizada - 2FN

- Uma relação está na Segunda Forma Normal (2FN) se ela estiver na 1FN e se ela não apresentar dependências funcionais parciais.



Fonte imagem¹⁵

- . Crie tabelas separadas para conjuntos de valores que se aplicam a vários registros;
- . Relacione essas tabelas com uma chave externa.

Registros não dependeria de algo que não seja a chave primária da tabela (uma chave composta, se necessário). Por exemplo, considere o endereço do cliente em um sistema contábil. O endereço é necessário, a tabela clientes, mas também pelas

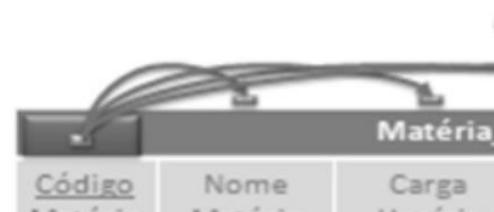
15 http://bp1.blogspot.com/_aFYkUPKyX3M/R9MdX87xCyI/AAAAAAAEErc/iRstlvMlbY/s400/

tabelas pedidos, remessa, faturas, contas à receber. Em vez de armazenar o endereço do cliente uma entrada separada em cada uma dessas tabelas, armazená-lo em um lugar, na tabela de clientes ou em uma tabela endereços separada.

1.1.1.4.1.3 - Terceira Forma Normalizada - 3FN

Terceira Forma Normalizada

- Uma relação está na Terceira Forma Normal (3FN) se ela estiver na 2FN e se ela não apresentar dependências funcionais transitivas.



a Normal (3FN)

ira
o

if

Departamento	
Código	Nome
100	Administrador

Materia	Materia	Materia
Departamento Coordenador	Departamento Coordenador	

Fonte da imagem¹⁶

. Elimine campos que não dependem da chave.

Valores em um registro que não são parte da chave do registro não fazem parte da tabela. Em geral, sempre que o conteúdo de um grupo de campos pode se aplicar a mais de um único registro na tabela, considere colocar esses campos em uma tabela separada.

Por exemplo: em uma tabela e recrutamento de funcionários, nome da Universidade e o endereço do candidato pode ser incluídos. Mas você precisa de uma lista completa de universidades para correspondências de grupo. Se informações da

16 http://lh5.ggpht.com/franciscogpneto/SNDDB2uNOqI/AAAAAAAAGyI/dOMKfg9uwLM/image_thumb%5B6%5D.png?imgmax=800, 12/10/2015

Universidade são armazenadas na tabela de candidato, não é possível para Universidades uma lista com candidatos não atuais. Criar uma tabela separada de Universidades e vinculá-lo à tabela de candidatos com uma chave de código de Universidade.

“1.1.1.4.1.5 - Quarta Forma Normalizada

Definição

Uma tabela está na 4FN, se e somente se, estiver na 3FN e não existirem dependências multivaloradas.

Exemplo (base de dados sobre livros)

Relação não normalizada: Livros(nrol, (autor), título, (assunto), editora, cid_edit, ano_public)

1FN:1FN: Livros(nrol, autor, assunto, título, editora, cid_edit, ano_public)

2FN: Livros(nrol, título, editora, cid_edit, ano_public) AutAssLiv(nrol, autor, assunto)

3FN: Livros(nrol, título, editora, ano_public) Editoras(editora, cid_edit)

AutAssLiv(nrol, autor, assunto)

Na 3FN, a base de dados ainda apresenta os seguintes problemas:

Redundância para representar todas as informações;

. Representação não-uniforme (repete alguns elementos ou posições nulas).

Passagem à 4FN

. Geração de novas tabelas, eliminando dependências multivaloradas;

Resultado

Livros(nrol, título, editora, ano_public) Editoras(editora, cid-edit)

AutLiv (nrol, autor)

AssLiv(nrol, assunto) ”¹⁷

“1.1.1.4.1.6 - Quinta Forma Normalizada

Está ligada à noção de dependência de junção.

Se uma relação é decomposta em várias relações e a reconstrução não é possível,

pela junção das outras relações, dizemos que existe uma dependência de junção.

. Existem tabelas na 4FN que não podem ser divididas em duas relações sem que se ¹⁷ [https://pt.wikipedia.org/wiki/Normaliza%C3%A7%C3%A3o_de_dados, 12/10/2015](https://pt.wikipedia.org/wiki/Normaliza%C3%A7%C3%A3o_de_dados)

altere os dados originais.

Exemplo: Sejam as relações R1(CodEmp, CodPrj) e R2(CodEmp, Papel) a decomposição da relação ProjetoRecurso(CodEmp, CodPrj, Papel).

Exemplo:

. Da 4FN para a 5NF;

. Explanação de que a última forma normal pode ser alcançada com projeções”¹⁸

1.1.1.4.1.7 – Outros formulários de normalização

Quarto formulário normais, também chamado Boyce Codd Normal Formulário (BCNF) e quinto normal existe, mas raramente são considerados no Design Prático. Desconsiderando essas regras pode resultar em Design de Banco de Dados muito legível, mas não deve afetar a funcionalidade.”¹⁹

“Definição: uma tabela está na BCNF se e somente se estiver na 3FN e todo atributo não chave depender funcionalmente diretamente da chave primária, ou seja, não há dependências entre atributos não chave.”²⁰

18 https://pt.wikipedia.org/wiki/Normaliza%C3%A7%C3%A3o_de_dados,
12/10/2015

19 Suporte Microsoft, <https://support.microsoft.com/pr-br/kb/283878, 12/10/2015>

20 https://pt.wikipedia.org/wiki/Normaliza%C3%A7%C3%A3o_de_dados,
12/10/2015

1.1.1.4.1.8 - "Brifieng do Processo de Design do Banco de Dados

- . Determinar a finalidade do seu banco de dados;
- . Localizar e organizar as informações necessárias;
- . Dividir as informações em tabelas;
- . Transformar informações em colunas;
- . Especificar as chaves primárias e estrangeiras;
- . Configurar as relações de tabelas;
- . Refinar o design e analisar resultados esperados das tabelas;
- . Aplicar as regras de normalização.”²¹

1.1.1.4.1.9 Entender chaves primárias, estrangeiras e compostas

- . Entender o motivo da existência das chaves em um banco de dados, escolher as chaves primárias apropriadas, selecionar o tipo de dados apropriado para chaves, selecionar campos apropriados para chaves compostas, entender a relação entre chaves estrangeiras e chaves primárias

1.1.1.4.1.10 Entender índices

Entender índices

- . Entender índices em cluster e sem cluster em sua finalidade em um

banco de [21 Suporte Office, https://support.office.com/pt-br/article/Fundamentos-do-design-de-banco-de-dadosb2159cf-1e30-401a-8084-bd4f9c9ca1f5, 13/10/2015](https://support.office.com/pt-br/article/Fundamentos-do-design-de-banco-de-dadosb2159cf-1e30-401a-8084-bd4f9c9ca1f5)
dados

“NOTA

Estrutura de dados e índice

Um índiceíndice contém palavras-chave e dados associados que apontam para a localização de informações mais abrangentes, tais como arquivos e registros em um disco ou chaves de registro em um banco de dados.”²²

1.1.1.5 - Entender conceitos de segurança do banco de dados

. Entender a necessidade de proteger um banco de dados, quais objetos podem ser protegidos, quais devem ser protegidos, contas de usuário e funções

NOTA

“**Qual das opções abaixo** não é um exemplo de um ataque de segurança? a. atribuir funções para conceder acesso
b. escalonamento privilegiado

c. injeção de SQL

Respost: Atribuir funções para conceder acesso não é um exemplo de ataque, os demais exemplos sim.”²³

[22 MSDN, Esturuta de dados e índice, http://msdn.microsoft.com/en-us/library/ms180978.aspx,03/10/2015](http://msdn.microsoft.com/en-us/library/ms180978.aspx) 23 Certiport, Documento Oficial Exame 98-364 - Apostila Preparação para Certificação do

“Segurança do SQL Server (ADO.NET)²⁴

SQL Server tem muitas características que suportam a criação de aplicações de banco de dados seguros. Os recursos de segurança evoluir e se fortalecido a cada nova versão do SQL Server, portanto, não são recursos de segurança reforçada no SQL Server 2005 e 2008 que não existem para SQL Server 2000.

Considerações de segurança comuns, tais como o roubo de dados ou vandalismo, aplicável independentemente da versão do SQL Server que você está usando. A integridade dos dados também deve ser considerada como um problema de segurança. Se os dados não está protegido, é possível que ele poderia se tornar inútil se a manipulação de dados ad hoc é permitido e os dados são inadvertidamente ou maliciosamente modificado com valores incorretos ou suprimido

inteiramente. Além disso, muitas vezes há exigências legais que devem ser respeitados, como o armazenamento correto de informações confidenciais. Armazenar alguns tipos de dados pessoais é proscrito inteiramente, dependendo das leis que se aplicam em uma determinada jurisdição.

Cada versão do SQL Server tem diferentes recursos de segurança, como faz cada versão do Windows, com versões posteriores ter funcionalidade aprimorada ao longo dos anteriores. É importante compreender que recursos de segurança por si só não pode garantir a aplicação de banco de dados seguro. Cada aplicativo de banco de dados é único em suas necessidades, ambiente de execução, modelo de implantação, localização física, e população de usuários. Algumas aplicações que são de âmbito local pode precisar de apenas um mínimo de segurança enquanto que outras aplicações locais ou aplicativos implantados através da Internet pode exigir medidas de segurança rigorosas e de monitorização e avaliação contínua.

MTA,http://certiport.com/Marketing/Mta/Localization/PTB/MTA_SSG_DbAdmin_individual_PTB.pdf, 03/10/2015
24 MSDN, Segurança de SQL Server, <https://msdn.microsoft.com/>

[library/bb669074\(v=vs.90\).aspx](#), 09/10/2015

Os requisitos de um aplicativo de banco de dados SQL Server de segurança devem ser considerados no momento da concepção, e não como uma reflexão tardia. Avaliando ameaças no início do ciclo de desenvolvimento dá-lhe a oportunidade de mitigar os danos potenciais, sempre que uma vulnerabilidade é detectada.

Mesmo se o projeto inicial de um aplicativo é som, novas ameaças podem emergir como o sistema evolui. Ao criar várias linhas de defesa em torno de seu banco de dados, você pode minimizar os danos causados por uma violação de segurança. Sua primeira linha de defesa é reduzir a área de superfície de ataque, para nunca mais conceder mais permissões do que são absolutamente necessárias.”

- . Entender os backups e restauração do banco de dados
- . Entender os vários tipos de backup, como completo e incremental, a importância de backups, como restaurar um banco de dados **NOTA:** Atualmente o Exame, também, é realizado no idioma Português-Brazil.

“ Na nossa área, Tecnologia da informação, em algum momento os profissionais se deparam com um grande desafio: como ser um profissional mais requisitado e competitivo para atender às exigências do mercado de trabalho? Aqui, começamos a perceber que nossos concorrentes sempre tem algo a mais, geralmente, uma certificação.

Primeiro gatilho mental: Não pense que por assistir os vídeos indicados você já está preparado, mantenha-se com modestia, é o primeiro passo para aprender como se aprende. Lembrando que esses vídeos são introdutórios para se familiarizar com o Sistema Operacional do Servidor Windows e pouco se fala de SQL Server. Prestou atenção? Porém Backup, DNS , Serviços, entre outros desses assuntos iremos necessitar conhecer.

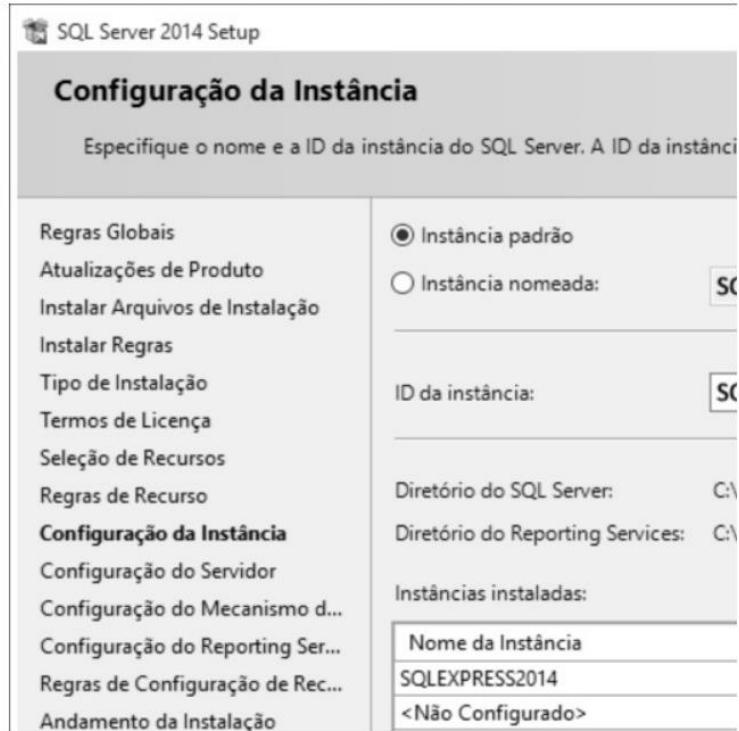
1.2 – Configuração e administração de SQL Server Express 2014, SQL Management Studio e dbForge Studio

Precisamos de dois softwares para estudar o SQL: o SQL Server e Gerenciador de Banco de Dados.

. Download Link <https://www.microsoft.com/pt-br/download/details.aspx?id=42299> . Primeiro instalar: selecione o arquivo “SQLEXPRAADV_x86_PTB.exe”, para instalar em 32bits ou 64bits. Esse

arquivo é grande e completo 1GB. Nele está contido
SQL Server Express 2014 e o Management Studio ;
. Usando Sistema Operacional Windows 10 - 32 bits;
. Selecionei 32 bits no meu caso selecionei. Veja se o seu computador é 64bits.

1.2.1 - Configurando SQL Server Express 2014



a se torna parte do caminho de instalação.

SQLExpress2

SQLExpress2

Program Files\Microsoft SQL Server\MSSQL12. **SQLExpress2**

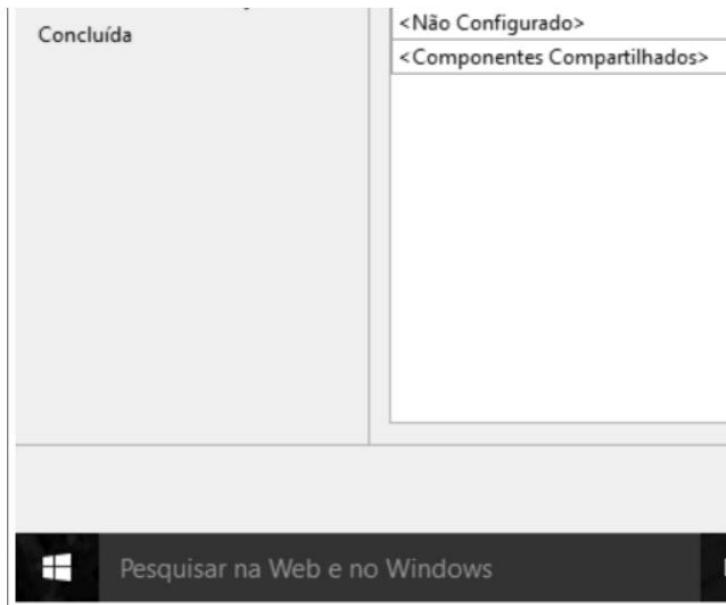
Program Files\Microsoft SQL Server\MSRS12. **SQLExpress2**

ID da Instância	Recursos
MSSQL12.SQLEXPRESS2014,MSRS12.SQLEXPRESS2014	SQLEngine,SQLBrowser,ReportServer,ReportServerT...
SQLExpress2	SQLEngine,SQLBrowser

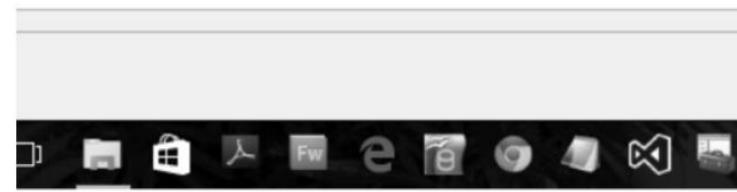


	Edição
Engine\Replication,SQLEngin...	Express
Engine\Replication	

Versão
12.0.2269.0
12.0.2269.0



	SQLEXPRESS2014_1	SQLEngine, SQLEr SSMS, Adv_SSMS
--	------------------	------------------------------------



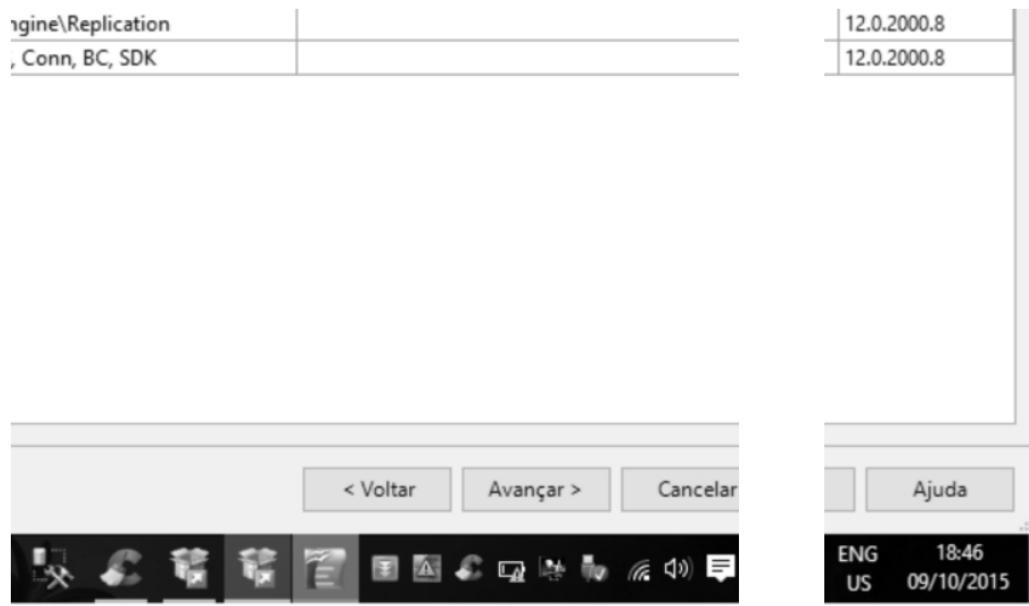
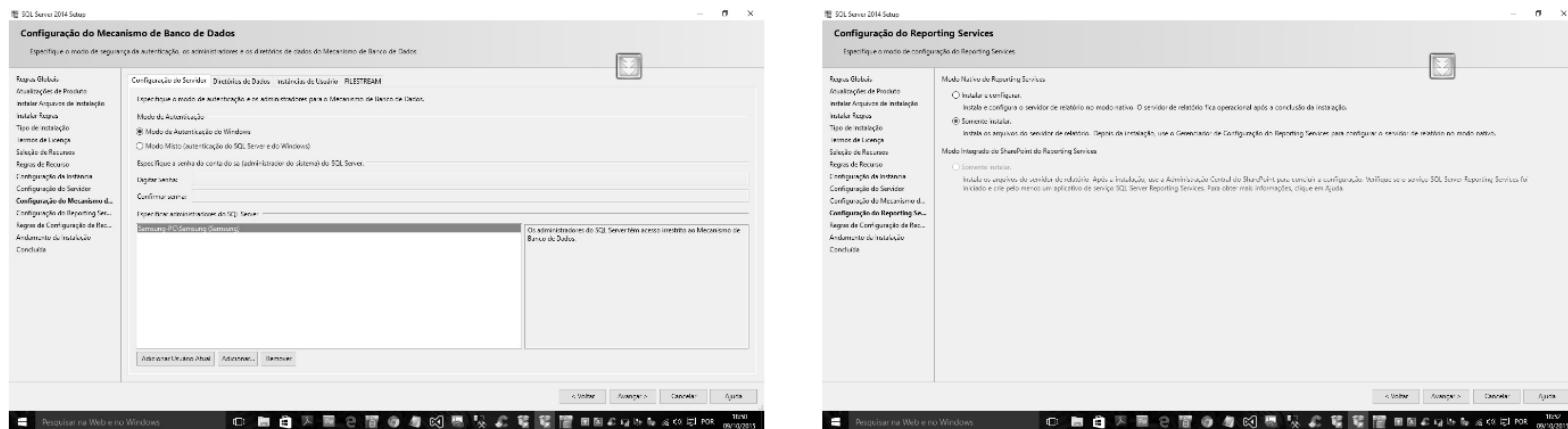


Imagen 2 – Intância Padrão: SQLExpress2



Modo de Autenticação do Windows 10 – Imagem 8

· Selecione Modo Padrão Autenticação Windows, assim ele autentica o acesso a instância SQLEXPRESS2 com senha e login de acesso ao Windows 10;

Configuração de Reporting Service - Imagem 9

- Em Configuração de Reporting Services imagem 9. Clique “somente instalar” e <avançar>;
- E aguarde a instalação. Aguarde por alguns minutos...

NOTA:

Caso Ocorra Problemas com SQL Management Studio:

Após instalação e se tudo ocorrer bem na barra de tarefas do Windows

10 digite “sql management studio” e abra-o. Se aparecer algum erro solicitado digitar no Prompt do Windows c:\ssms.exe\resetsettings , antes veja o ANEXO II. Confesso que esse erro me deixou impaciente, porém, não é tão complexo resolver veja no Anexo II usando o RegEdit.exe apaguei uma chave e pronto ele funcionou. Porém, até descobrir essa solução, aviso: paciência, paciência, persistência, persistência...

Pronto você instalou o Microsoft SQL Server 2014. Agora precisamos Administrar o Banco de Dados .

2. Introdução Administração de Banco de Dados

NOTA:

Necessário você abrir na pesquisa do Windows digite “SQL Server Configuration Manager” a fim de “iniciar” a instância do SQL Server que configuramos para criar nosso banco de dados. Conforme imagem 4.

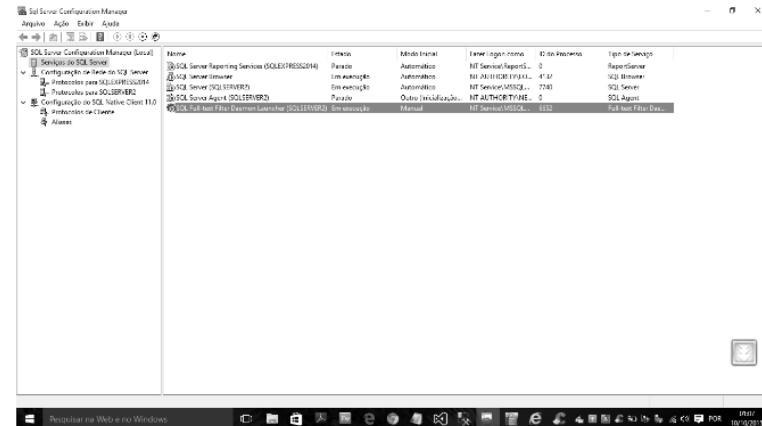


Imagen 4 – SQL Server Configuration Manager

. Observe que aparecem várias outras Instâncias, porém de todas a nossa é SQLSERVER2, usando o SQL Management Studio ou dbForge Studio par conectar e administrar e criando o banco de dados e tabelas;

. Usando o botão direito do mouse, clique sobre a Instância SQLSERVER2 e aparecerá uma cortina e depois clique em <iniciar>, assim acionamos o mecanismo de banco de dados SQL Server 2014

ativando essa Instância. Haverá necessidade dessa ação quando uso do dbForge Studio.

Gatilho Mental: Não antecipe o seu pensamento imaginando o quê é dbForge Studio, no momento será explicado e orientado o download dessa ferramenta de design e administração de banco de dados. É importante esse tipo de observação para melhor gerenciar o processamento da informação em nosso cérebro. Antecipar problemas ou criar não é correto, sabemos que há tempestade de informações chegando e nos responsabilizando. No entanto, na área de Tecnologia de Informação precisamos saber gerenciar, primeiro, essa tempestade de conteúdos. Não desista! Quando jovem após concluir meu Pós-Técnico em Processamento de Dados em 1992 na Escola de Processamento de Dados da Bahia em Salvador-BA. Lembro-me muito bem do meu Prof. Túlio Marcos, Professor Titular da UFBA – Universidade Federal da Bahia em Física, ele sempre nos orientava a sermos práticos não se apegando as informações “firulas”, nada de “plumas e paitês”, sejamos modestos e simples.

Após conclusão da Especialização no Ensino de Filosofia na UFBA, nos enriqueceu por demais como promover uma Metodologia de Ensino

que ajude o Estudante no seu aprendizado. E nessa Metodologia a inclusão de “Gatilhos Mentais”, servem como um momento de “STOP”, um parar para recomeçar, um pensar para agir.

NOTA:

INSTALAÇÃO CLEAN:

Há possibilidade de uma segunda opção de instalação com outra versão do SQL Server Express, com arquivos menores para download.

Estaremos utilizando o software “**dbForge Studio for SQL Server**” [“https://www.devart.com/dbforge/sql/studio/”](https://www.devart.com/dbforge/sql/studio/) - mais rápido de instalação e não preciso de tanto recursos, o que o dbForge faz é equivalente ao SQL Management Studio ou Visual Studio, porém o dbForge é mais “clean”, limpo e leve instalação.

Primeiro a instalação do SQL Server (selecione 64 bits ou 32 bits) no link <https://www.microsoft.com/pt-br/download/details.aspx?id=42299> – selecione
- selecione
Express 32BIT\SQLEXPRESS_X86_PTB.exe , arquivo menor de 183MB;
Segundo
Segundo instale a versão dbForge Studio for SQL Server, v4.5

ExpressdbForge Studio for SQL Server, v4.5 Express livre no

dbForge Studio for SQL Server, v4.5 Express

link <https://www.devart.com/dbforge/sql/studio/download.html#header>

Com o dbForge Studio dbForge Studio instalado, abra-o. Faça a conexão com o Servidor SQL

dbForge Studio

Server conforme imagem 6. Essa configuração é acionar a Instância criada no SQL Server quando na fase de instalação foi configurado.

Veja 1.2.1 Configurando SQL Server Express .

Ferramentas de Administração de Banco de Dados

Estarei utilizando dbForge Studio e SQL Management Studio – interagindo em ações diferentes a fim de fortalecer a sua compreensão quanto as ferramentas diferentes em administração de banco de dados. É importante essa flexibilidade nos dá reforços para nos adaptarmos as diversidades de tecnologias que tem a Linguagem SQL como seu código fonte.

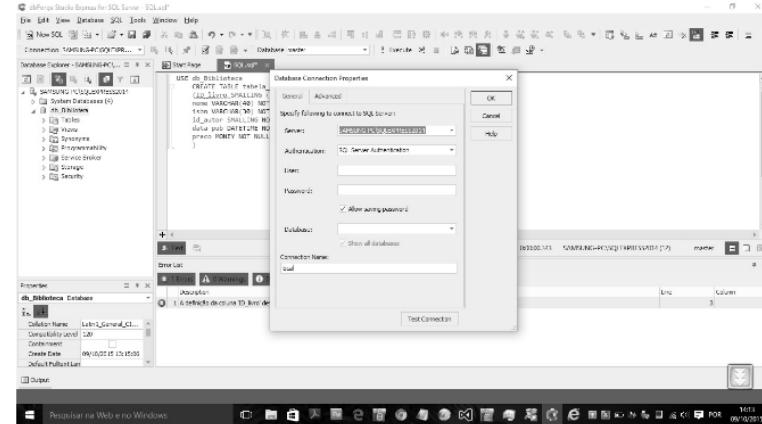


Imagen 6 – Configurando dbForge Studio SQL Server

- . Aqui você irá se conectar com o Servidor na Instância: “SQLEXPRESS”;
- . Selecione “Modo de Autenticação do Windows”;
- . Clique no canto inferior “Test Connection”;

2.1 - Criando Banco de Dados

- Criando Banco de Dados

2.1.1 – SQL Management Studio

- . Começamos interagindo abrindo, agora, digite na barra de tarefas do

Windows “SQL Management Studio”;

- No menu <Exibir> selecione “Servidores registrados” conforme imagem

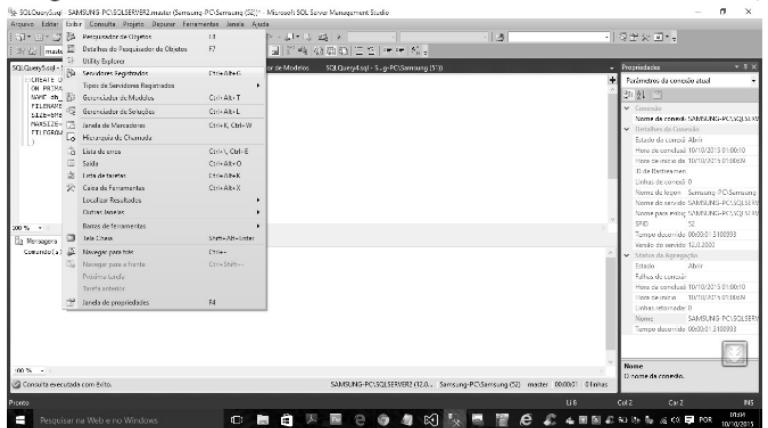
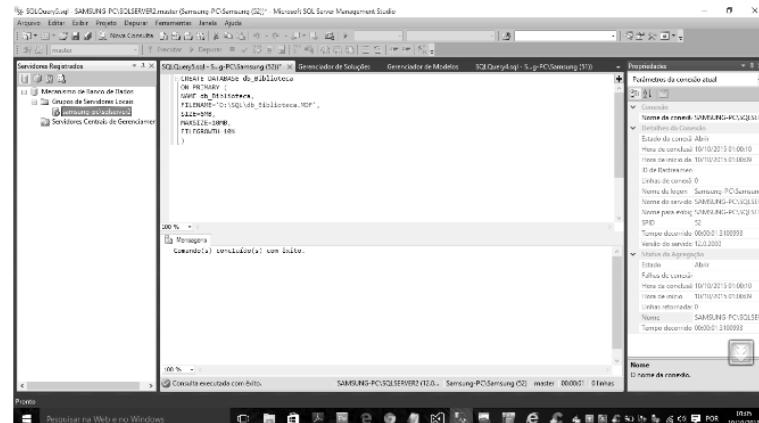


Imagen 3.2



Servidores Registrados - Imagem 3.3.

Lembrando, agora, selecione a Instância chamada “SQLserver2”. Para isso é necessário abrir “SQL Server Configuration Management” imagem 3.4

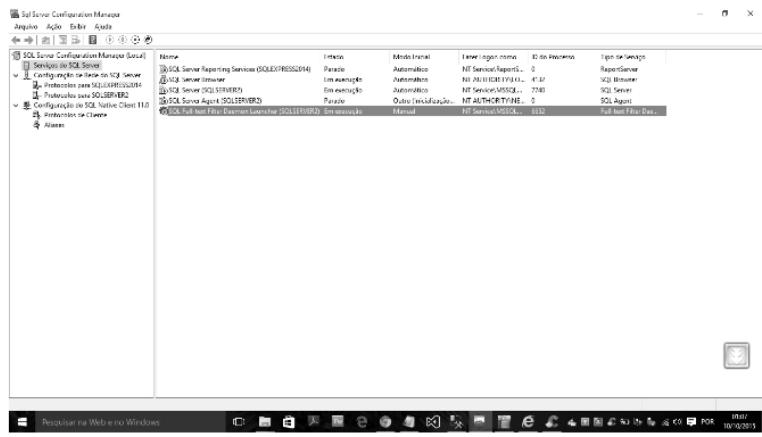


Imagen 3.4

. Na imagem 3.4 observe “Serviços de SQL Server” e clique para abrir as opções que aparecerão logo à direita: SQL Server(SQLEXPRESS2). Selecione e use o botão direito do mouse, abre-se uma cortina, selecione <iniciar> conforme imagem 3.4. Assim ativa a Instância e podemos conectar. Retorno ao SQL Management Studio vamos nos conectar.

2.1.2 - SQL Server Configuration Manager

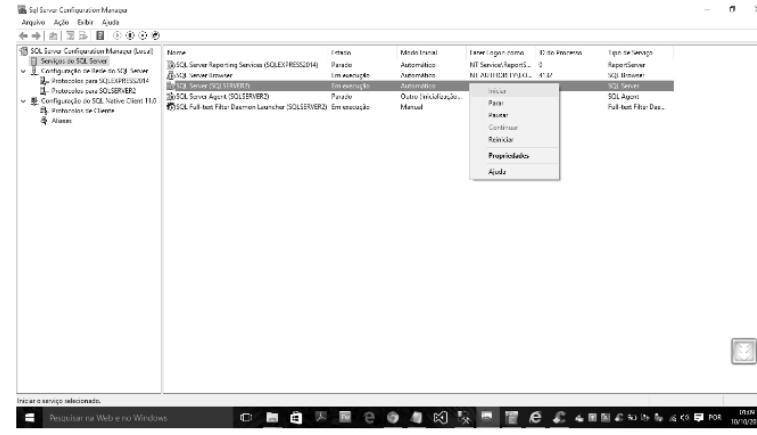


Imagen 3.5

. Agora com SQL Management Studio aberto clique em “nova consulta”;

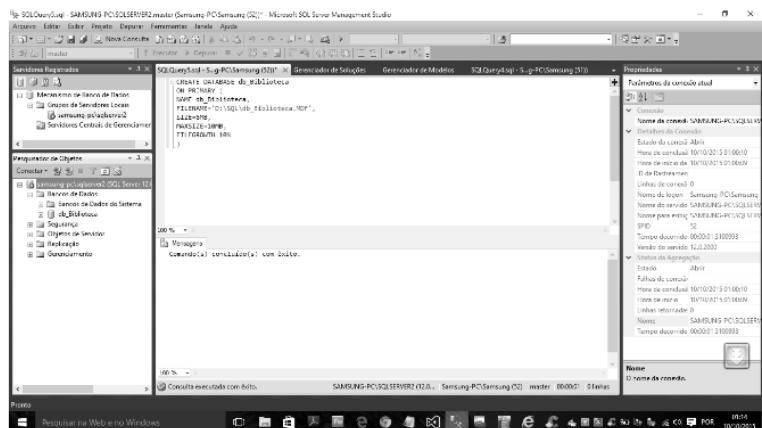
Escreva o comando SQL conforme imagem 3.6:

```
CREATE DATABASE db_Biblioteca
ON PRIMARY (
NAME=db_Biblioteca,
```

```

FILENAME='C:\SQL\db_Biblioteca.MDF',
SIZE=6MB,
MAXSIZE=10MB,
FILEGROWTH=10%
)

```



SQL Management Studio – Cria Banco “db_Biblioteca” - imagem 3.6

. Veja em “Servidores Registrados” aparece “Samsung-pc\sqlserver2”

nossa nova Instância configurada usando o Aplicativo de Instalação do SQL Server Express;

. E após execução do comando SQL conforme está na imagem 3.6 – foi criado o Banco de Dados “db_Biblioteca” que aparece no “Pesquisador de Objetos” na imagem 3.6;

Nome do banco de dados: db_Biblioteca

PRIMARY Chave primária: nome

FILENAME Nome do arquivo e gravar no caminho: D:\SQL
\\db_Biblioteca.MDF SIZE Tamanho: 6MB

Os comando SQL são mais rápidos para execução e desempenho no Servidor seja na WEB até mesmo em conexão de Internet lentas, ou em seu computador DESKTOP.

2.1.3 - Usando dbForge Studio for SQL

Vamos agora, usar, também o gerenciador de administração de banco de dados, interagindo com os dois. Um dia será muito útil, acredite em mim. Abra o “dbForge SQL”

- Clique em “Nova consulta”
- Digite o código fonte:

```

CREATE DATABASE db_Biblioteca ON PRIMARY (
    NAME=db_Biblioteca,
    FILENAME='D:\Ângelo\livros      angelo\Fundamentos      Database
\db_Biblioteca.MDF',
    SIZE=6MB,
    MAXSIZE=10MB,
    FILEGROWTH=10%
)
- Clique em "Executar"

- Veja que foi criado o Banco de dados: " db_Biblioteca "

```

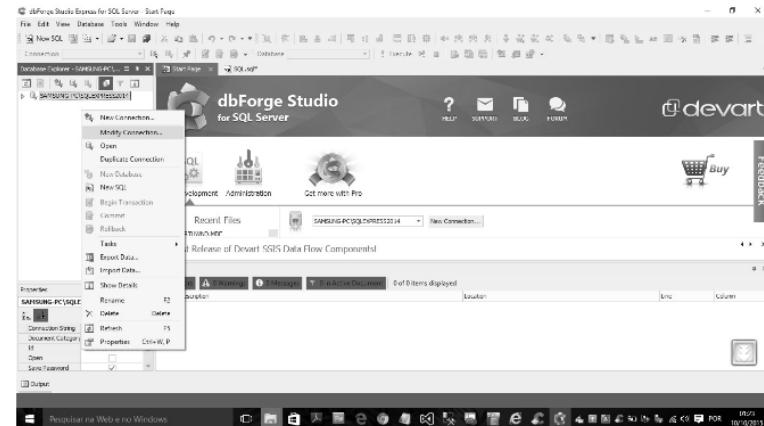


Imagen 3.9 – Nova Conexão ou Modificar Conexão com Servidores

Nesse caso vamos modificar a conexão, no dbForge SQL, é permitido trabalhar com um de cada vez, um dos pontos positivos a fim de evitar poluição visual e, pior, confusão;

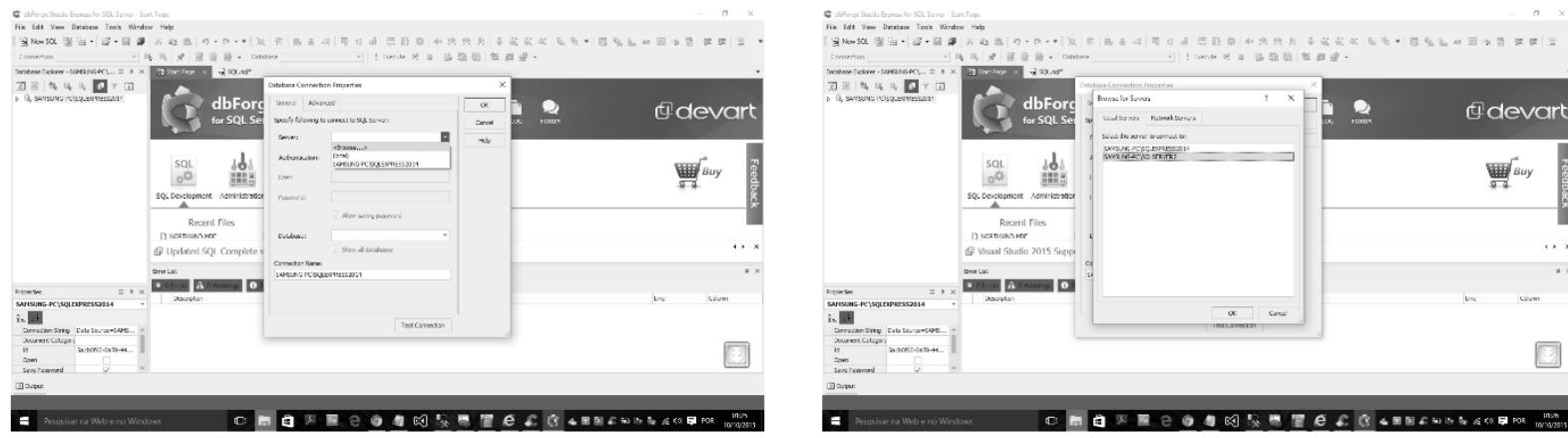


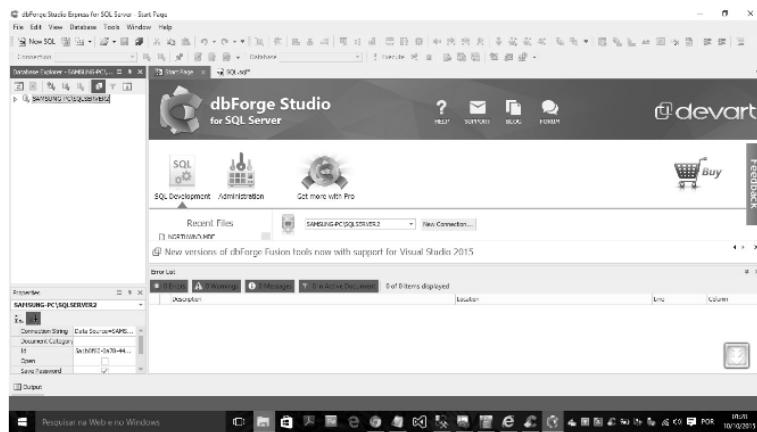
Imagen 3.9 – Modificando conexão de servidor

- Cliquei em “modify connection” e apareceu a imgem 3.8. Aqui nesse caso, a modificação é porque necessitou “atualizar” o Database Explorer onde aparecê a Instância SQLServer2. Selecione “browser” para aparecer as Instâncias ocultas, veja Tela 4.0 – Selecionar novo Servidor.

- Aqui selecionei a opção <browser> apareceu a tela 4.0;

Tela 4.0 – Selecionar novo Servidor

- Clique depois em <OK> para confirmar a seleção do novo Servidor;
- Novamente clique em <OK> para confirmar a nova conexão, conforme Tela 4.1



Tela 4.1

No canto superior à esquerda aparece “Database Explorer” e as opções do novo banco de dados “db_Biblioteca” afinal já tínhamos criado no “SQL Management Studio”. Que legal essas duas plataformas de gerenciamento de banco e dados;

2.2 - Uso comando "CONSTRAINTS" - restrições

- São regras aplicadas nas colunas das tabelas;

- Limitar o tipo de dados em cada coluna;
- Após criar a tabela (CREATE) podemos alterar a tabela com (ALTER)

2.2.1 - Not Null

- Força o Banco de Dados não aceitar campo nulo.
- Obrigatório digitar o campo.

2.2.2 - Unique

Informação que não se repete em um conjunto de colunas.
Uma chave primária PRIMARY KEY automaticamente possui uma restrição UNIQUE

2.2.3 - Primary Key

Uma chave primária composta por várias colunas. Contém valores únicos;
E não pode ter valores NULL.

2.2.4 - FOREIGN KEY

Um campo na Tabela que aponta para outra Tabela: Ex.
CONSTRAINT codigo_autor FOREIGN KEY(cod_aut) REFERENCES
tabela_autores(cod_aut)

NOTA: Depois do FOREIGN KEY é o nome da coluna e tem cod_aut como chave estrangeira.

2.2.5 - CHECK

Limita uma faixa de valores. O que é permitido ou não na coluna.

2.2.6 - DEFAULT

Assume um valor padrão em uma coluna. Que pode ser alterado.

2.3 - Criando Tabelas

CENÁRIO

“Raj atualmente controla seu estoque usando lápis e papel. Pedidos de peças e acessórios ou contagem de estoques levam vários dias. Raj está fazendo um curso de gerenciamento de banco de dados na escola e ele percebe que o negócio de seu pai poderia ser altamente beneficiado pela implementação de um sistema de gerenciamento de banco de dados.

Raj precisa revisar alguns conceitos básicos antes de começar, e fez uma lista de tópicos que ele sabe que serão importantes nos primeiros passos do projeto.

Identificar quais Tabelas Raj pode criar para o banco de dados.

a.) uma tabela de peças, uma tabela de automóveis, e uma tabela de acessórios;

b.) uma tabela para todas as peças, automóveis e acessórios;

c.) uma tabela para cada peça, cada bicicleta, e cada tipo de acessórios (uma tabela para capacetes, uma tabela para luvas para ciclistas, e assim por diante).

Identifique os campos que Raj deveria usar para os cabeçalhos das colunas da tabela de pecas

a.) número de peça, nome de peça, número da bicicleta, e nome da bicicleta

b.) número da peça e quantidade vendida

c. número da peça, nome da peça, quantidade, cor, e identificador da bicicleta

Qual é o nome dos dados localizados na intersecção de uma linha e coluna? a. campo

b. registro

c. variável

Respostas:

1.a

2.c
3.a²⁵

2.3.1 - Tipos de Dados no SQL Server

25 Certiport, Documento Oficial Exame 98-364 - Apostila Preparação para Certificação do MTA,http://certiport.com/Marketing/Mta/Localization/PTB/MTA_SSG_DbAdmin_individual_PTB.pdf, 03/10/2015

2.3.1.1 - Tipos Alfanuméricos Tipo dados char(n)

varchar(n)
varchar(max)
text
nchar
nvarchar
nvarchar(max)
ntext
bit binary(n)
varbinary
varbinary(max)

Descrição

Cadeia de caracteres de largura fixa. Máximo de 8.000 caracteres
Variável seqüência de caracteres de largura. Máximo de 8.000 caracteres
Variável seqüência de caracteres de largura. Máximo 1,073,741,824 personagens
Variável seqüência de caracteres de largura. 2GB máximo de dados de texto
Seqüência de caracteres Unicode de largura fixa. Máximo de 4.000 caracteres
Largura variável de cadeia
Unicode. Máximo de 4.000 caracteres Largura variável de cadeia
Unicode. Máximo 536,870,912 personagens
Largura variável de cadeia
Unicode. 2GB máximo de dados de texto
Permite a 0, 1, ou NULL
String binária largura fixa. Máximo de 8.000 bytes
Largura variável string binária. Máximo de 8.000 bytes
Largura variável string binária. 2GB

Armazenamento Largura definida

2 bytes + número de caracteres

2 bytes + número de caracteres

4 bytes + número de caracteres

2 x largura definida máximo

Largura variável string binária. 2GB_{image} máximo

2.3.1.2-Tipos de Números: Tipo dados

tinyint

smallint

int

bigint

Descrição Armazenamento

Permite números inteiros de 0 a 255 1 byte Permite que os números inteiros entre -32768 e 32767 2 bytes Permite que os números inteiros entre -2147483648 e 2147483647 4 bytes

Permite que os números inteiros entre -9.223.372.036.854.775.808 e 8 bytes 9.223.372.036.854.775.807

Precisão fixo e números da escala.

Permite números de -10^{38} 1 a 10^{38} -1.

decimal(p,s)

numeric(p,s) O parâmetro p indica o número total máximo de dígitos que pode ser armazenado 5-17 bytes(tanto para a esquerda e para a direita do ponto decimal). p deve ser um valor de 1 a 38. O padrão é 18.

O s parâmetro 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 fixo e números da escala.

Permite números de -10^{38} 1 a 10^{38} -1. 5-17 bytes

O parâmetro p indica o número total máximo de dígitos que pode ser armazenado

(tanto para a esquerda e para a direita do ponto decimal). p deve ser um valor de 1 a 38. O padrão é 18.

O s parâmetro 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

smallmoney Os dados monetários de -214,748.3648 para 214,748.3647 4 bytes **money** Os dados monetários de -922,337,203,685,477.5808 para 8 bytes 922,337,203,685,477.5807

Dados de número flutuante de precisão de -1.79E + 308 para 1.79E + 308.

float(n) O parâmetro n indica se o campo deve conter 4 ou 8 bytes. float (24) possui um 4 ou 8 bytes campo de 4 bytes e float (53) possui um campo de 8 bytes. O valor padrão de n é 53.

real Dados de número flutuante de precisão de -3.40E + 38 para 3.40e + 38 4 bytes

2.3.1.3 - Tipos de datas: Tipo dados

datetime
datetime2
smalldatetime

date
time
datetimeoffset timestamp

Descrição Armazenamento

A partir de primeiro de janeiro de 1753 a 31 de Dezembro de 8 bytes 9999

com uma precisão de 3,33 milissegundos
A partir de 1 de janeiro de 0001-31 de dezembro de 9999 com 6-8

bytes suma precisão de 100 nanosegundos
De 1 de Janeiro de 1900 a 06 de junho de 2079 com uma 4 bytes precisão de 1 minuto

Armazenar apenas uma data. A partir de 1 de janeiro de 3 bytes 0001-31 de dezembro de 9999

Armazenar uma vez apenas com uma precisão de 100 3-5 bytes nanosegundos
O mesmo que datetime2 com a adição de um deslocamento 8-10 bytes de fuso horário

Armazena um número exclusivo que é atualizado sempre que uma linha é criado ou modificado. O valor timestamp baseia-se em um relógio interno e não corresponde ao tempo real. Cada tabela pode ter apenas uma variável timestamp

2.3.1.4 - Outros tipos de dados:

Tipo dado **Descrição**
sql_variant

Armazena até 8.000 bytes de dados de vários tipos de dados, exceto text, ntext, e timestamp uniqueidentifier Armazena um identificador exclusivo global (GUID) xml Lojas de dados XML formatado. 2GB máximo

cursor

Armazena uma referência a um cursor utilizado para operações de bases de dados

table

Armazena um resultado de fábrica para processamento posterior

- Livros

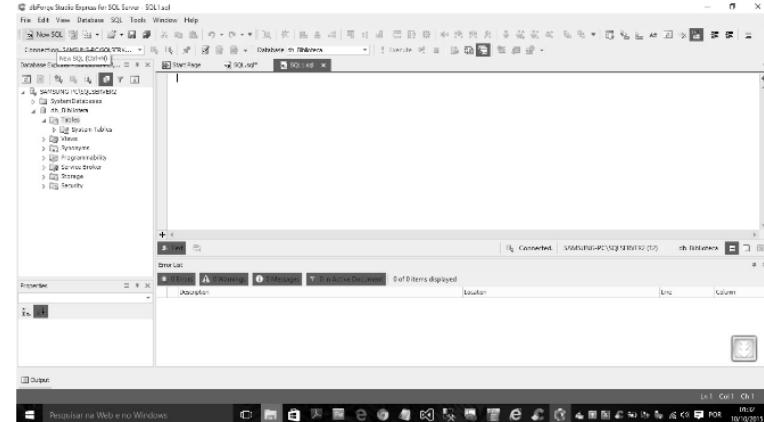
ID_livro, nome, isbn, preco, id_autor, data_pub

- Autores

ID_autor, nome, isbn

Com o dbForge Studio aberto vamos criar essa tabela:

. Clique em “New SQL” e nesse editor vamos copiar o código fonte para criar a Tabela Livros. Tela 4.3;



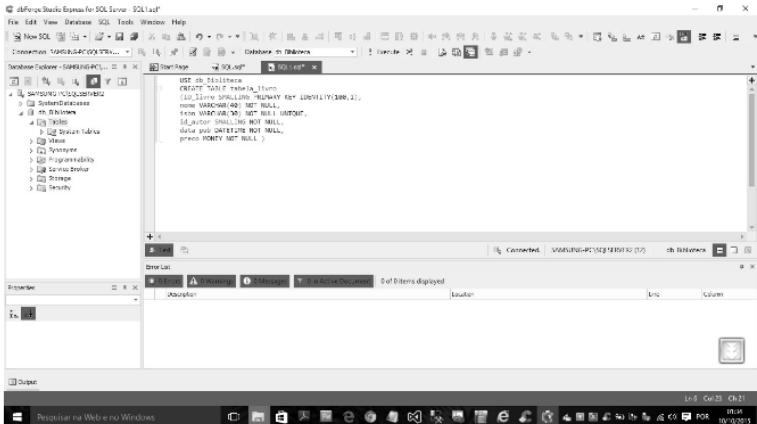
Tela 4.3 – New SQL

Comando SQL:

- CRIAR TABELA LIVRO:

```
USE db_Biblioteca
CREATE TABLE tabela_livro
ID_livro SMALLINT PRIMARY KEY IDENTITY(100,1),
nome_livro VARCHAR(40) NOT NULL,
```

isbn VARCHAR(30) NOT NULL UNIQUE, id_autor SMALLINT NOT
NULL,
data_pub DATETIME NOT NULL, preco MONEY NOT NULL)



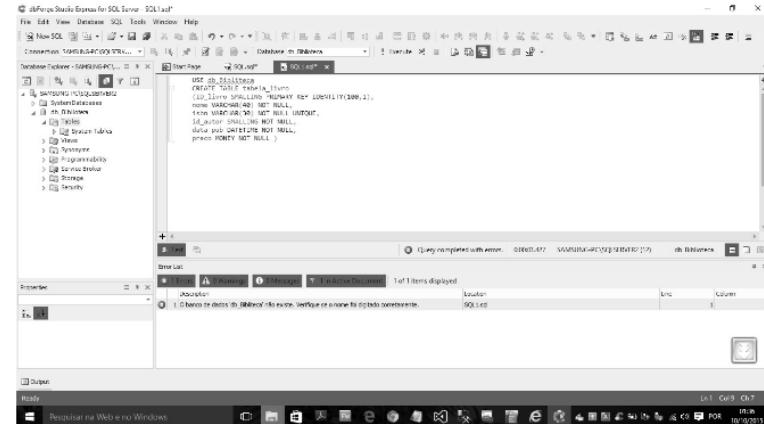
Tela 3.5 – Editor New SQL

- . Na Tela 3.5 - preste atenção que o código informado na área “New SQL” pode estar incorreto. Veja na linha “`nome_livro VARCHAR(40) NOT NULL,`” se estão iguais;
 - . Aparece o código fonte que colamos para criar a Tabela Livro. Porém,

agora clique em no botão “Execute”;²⁶

Aparecerá um erro informativo Tela 3.6, “banco de dados não existe”, porque está.

escrito c“db_Biblioteca”. Presta atenção cabra-macho;



Tela 3.6 – Erro de digitação

- . Na Tela 3.6 - preste atenção que o código informado na área “New SQL” pode está

26 http://www.ist
st=3860142, 11/10/2015

incorreto. Veja na linha “nome_livro VARCHAR(40) NOT NULL,” se estão iguais;

- . Corrija o erro e clique novamente em “Execute”;
- . Outro erro “SMAILLING”. De novo “cabra-macho” se escreve “SMALLINT”, um tipo de

dados “smallint” como tem “number”, “text”, “varchar” etc;

- . Ô moço veja que têm duas palavras “SMAILLIN” erradas. O Certo é “SMALLINT”.

NOTA:

- Criar tabela “**nome da tabela**”, depois declara cada coluna/campos. **Id_livro** tipo de dados **SMALLINT**, chave primaria, com auto incremento **IDENTITY (100,1)**

- Começa de 100 e sobe de 1 em 1.

- Copiar o código fonte no “nova consulta” e execute.

- CRIAR TABELA AUTORES:

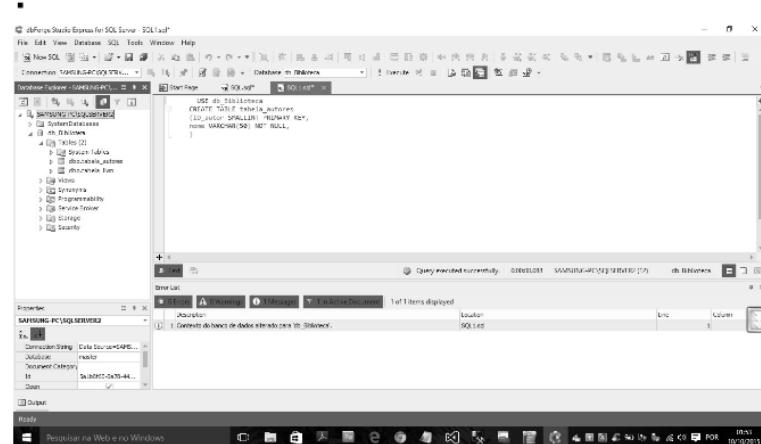
USE db_Biblioteca

```
CREATE TABLE tabela_autores (ID_autor SMALLINT PRIMARY KEY,  
nome VARCHAR(50) NOT NULL, )
```

. Será que digitou “SMALLINT” correto? Claro que sim! Ufa, ainda bem... . Após digitar o código no New SQL no dbForge SQL clique no botão <execute>; . Então cabra-macho, espero que tenha notado, novamente, “db_Biblioteca”

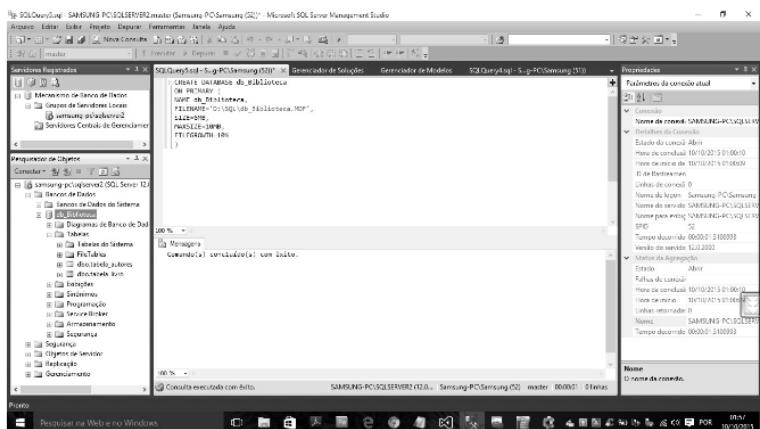
escrito errado. Conchte-o pois

estamos acionando um banco de dados existente nomeado “db_Biblioteca”;



Tela 4.6 – Criado Tabelas

- Observe “Database Explorer” no canto esquerdo. Clique duas vezes no ícone “db_Biblioteca” e abre a cortina de “Tabelas” aqui clique no segundo botão do mouse e selecione “Refresh”, assim é atualizado e aparecem as tabelas: dbo.tabela_autores, dbo.tabela_livro.



Tela 4.7 – Tabelas no SQL Management Studio

- Veja que aparecem as duas tabelas na Tela 4.7 . Porém, visualize selecionando “db_Biblioteca” com segundo botão do mouse e clique

“atualizar”, e clique duas vezes em “Tabelas” que “dbo.tabela_autores” e “dbo.tabela_livros” sorriem pra você.

2.4 - Alterar e Excluir Tabelas e Colunas

Alterar a estrutura depois de ser criada, acrescenta ou exclui atributos dos campos.

ALTERAR e EXCLUIR:

Exemplo Um:

```
ALTER TABLE tabela_livro DROP COLUMN id_autor
```

NOTA:

No exemplo 1 altera tabela TABELA_LIVROS e excluir coluna id_autor

Exemplo Dois:

Código fonte SQL:

```
USE db_Biblioteca
ALTER TABLE tabela_livro
ADD CONSTRAINT id_autor
FOREIGN KEY (id_autor)
REFERENCES tabela_autores(id_autor)
```

- Observe que a imagem 6.9, traz o resultado do código fonte SQL, acima mencionado. Ele tem por finalidade alterar o atributo da coluna “id_autor” na “tabela_livros” e coluna “id_autor” para ser uma chave estrangeira (Foreing Key) e essa chave é referência na “tabela_autores” na coluna “id_autor”.
- Na imagem 6.9 observe que a coluna “id_autor” recebe uma chave prata indicando chave estrangeira. E a chave ouro da coluna “id_livro” como primária da “tabela_livros”.

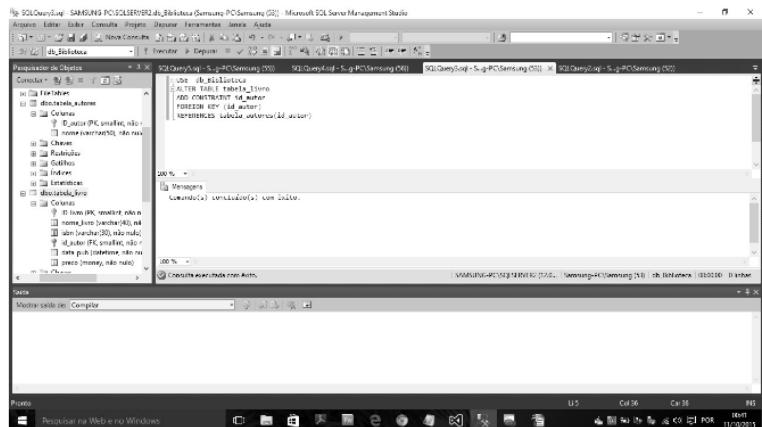


Imagen 6.9 – Criar uma Chave Estrangeira na Tabela_livros

. NOTA:

Consulte o ANEXO V a fim de reforça a compreensão do uso de ALTER TABLE

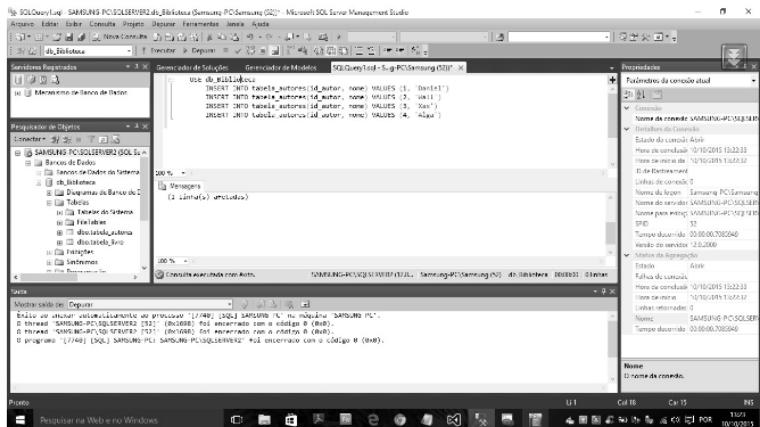
2.5 - Inserindo dados

Sintaxe:

INSERT INTO tabela (coluna, coluna2,...) VALUES (valor, valor2)

2.5.1 - Comandos SQL inserindo conteúdo na tabela_autores:

Usando o SQL Management Studio, porém o dbForge Studio pode receber os mesmos comandos INSERT:



. Execute o código em “Nova Consulta”:

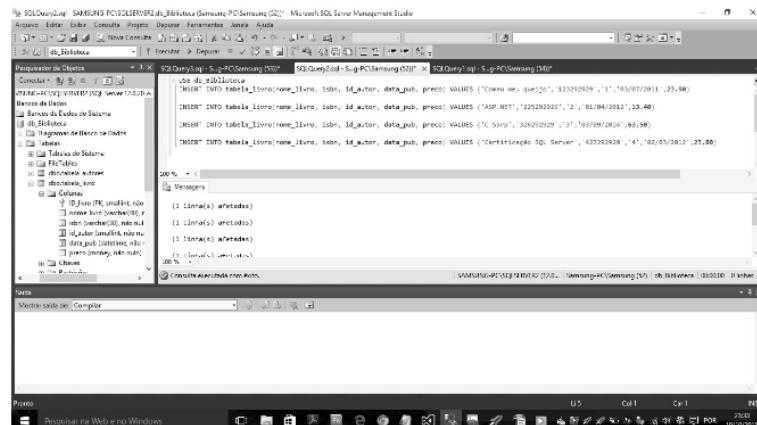
```

USE db_Biblioteca
INSERT INTO tabela_autores(id_autor, nome) VALUES (1, 'Daniel')
INSERT INTO tabela_autores(id_autor, nome) VALUES (2, 'Wall')
INSERT INTO tabela_autores(id_autor, nome) VALUES (3, 'Xax')
INSERT INTO tabela_autores(id_autor, nome) VALUES (4, 'Alga')
  
```

2.5.2 - Comandos SQL inserindo conteúdo na tabela_livro:

NOTA:

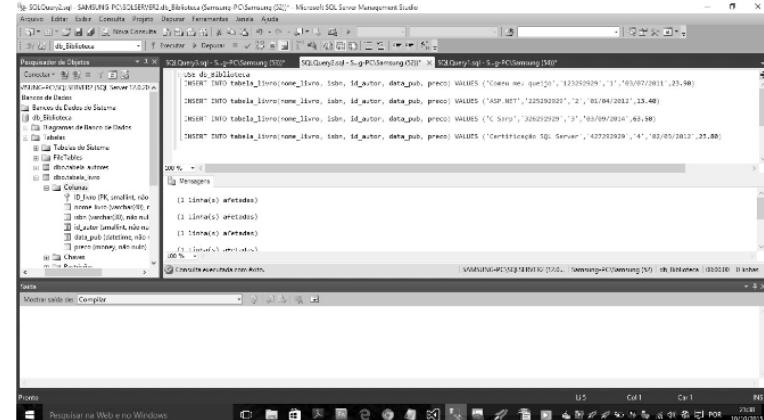
- Há dois campos nessa Tabela de Livros dependente de outra Tabela Autores;
- Cadastrar um id_autor nessa tabela tem que existir na Tabela Autores;
- Lembre-se o campo "id_livro" está incrementado, com o comando SQL "(ID_livro SMALLINT PRIMARY KEY IDENTITY(100,1),"



Código fonte comandos SQL:

```
USE db_Biblioteca
INSERT INTO tabela_livro(nome_livro, isbn, id_autor,
data_pub, preco) VALUES ('Comeu meu
queijo','123292929','1','03/07/2011',23.90)

INSERT INTO tabela_livro(nome_livro, isbn, id_autor, data_pub,
preco) VALUES ('ASP.NET','225292929','2','01/04/2012',13.40)
INSERT INTO tabela_livro(nome_livro, isbn, id_autor, data_pub,
preco) VALUES ('C Sharp','326292929','3','03/09/2014',63.50)
INSERT INTO tabela_livro(nome_livro, isbn, id_autor,
data_pub, preco) VALUES ('Certificação SQL
Server','427292929','4','02/03/2012',23.80)
```



Tela 6.7 – INSERT adicionando informações a Tabela_Livro

NOTA:

Não é declarado o campo "id_livros" está incrementado e será gravado automaticamente com o incremento.

2.5.2.1 - Verificar os dados gravados

Vamos ver os registros inseridos. Digite o comando SQL: USE db_Biblioteca

```
SELECT * FROM tabela_autores
```

The screenshot shows the SQL Server Management Studio interface. In the center, there is a results grid titled 'tabela_autores' with the following data:

ID	nome	id_livro	id_autor	date_pub	preco
1	J. R. F. Corrêa	1	1	2014-09-01 00:00:00	33,90
2	W. E. Sharpen	2	2	2015-03-01 00:00:00	18,90
3	C. P. Net	2	3	2014-01-01 00:00:00	14,90
4	C. G. M.	3	4	2014-04-01 00:00:00	33,90
5	D. C. D. Oliveira	4	5	2014-04-01 00:00:00	73,90

Tela 6.4 – Verificando existência de dados gravados com INSERT na Tabela_Livros

2.6 - Consultar os dados

```
USE db_Biblioteca
```

```
SELECT nome FROM tabela_autores
```

Aqui aparecerá, somente, o nome do livro da tabela_autores

```
SELECT * FROM tabela_autores
```

Aqui todos os campos do livro da tabela_autores

The screenshot shows the SQL Server Management Studio interface. In the center, there is a results grid titled 'tabela_autores' with the following data:

ID	nome	id_livro	id_autor	date_pub	preco
1	Daniel	1	1	2014-09-01 00:00:00	33,90
2	Paul	2	2	2015-03-01 00:00:00	18,90
3	João	3	3	2014-01-01 00:00:00	14,90
4	Alice	4	4	2014-04-01 00:00:00	33,90

Imagen 8.1 – Uso de SELECT aparece os dados gravados id_autor, nome:

2.6.1 - Especificar a ordem da consulta

```
USE db_Biblioteca
```

```
SELECT nome,id_autor FROM tabela_autores ORDER BY id_autor ASC
```

Aqui ele só imprime as colunas nome e id_autor na ordem da coluna id_autor na ordem ascendente. Pode ser ASC-ordem ascendente e DESC-ordem descendente.

2.6.2 - Filtros de registros em uma consulta - WHERE

USE db_Biblioteca

```
SELECT nome,id_autor FROM tabela_autores WHERE id_autor='1'  
Aqui é impresso o nome e id_autor da tabela_autores mas, somente, a linha que tem o id_autor igual '1'
```

NOTA: esse filtro cai no Exame, atento.

2.6.3 - Filtros de registros com cadeia de caracteres corresponde - LIKE

USE db_Biblioteca

```
SELECT nome_livro,id_autor FROM tabela_livro WHERE nome_livro  
LIKE 'a%' Busque todos os livros que começam com a letra 'A' e não importa os caracteres depois, selecionando os campos "nome_livro" e "id_autor" da tabela_livros. O resultado veja na imagem 8.9.
```

- . '%' - uso de porcentagem qualquer cadeia de 0 ou mais
- . '[']' - letras de "a" até "h" ([a-h])

USE db_Biblioteca

db_Biblioteca

```
SELECT nome_livro,id_autor FROM tabela_livro WHERE nome_livro  
LIKE '[AC] %'
```

Busque todos os livros que começam com a letra 'A' ou 'C' e não importa os caracteres depois. Aqui será selecionado somente os livros com letras "A" e "C". O resultado é visto na imagem 7.9

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a results grid titled 'Tabela' (Table) with two columns: 'nome_livro' and 'id_autor'. The data returned is:

nome_livro	id_autor
Elton P. Costa	1
ASP.NET	2
C-Sas	3
Cartografia SQL Server	4

Imagen 7.9 – Uso de filtro no LINK '[AC]%'

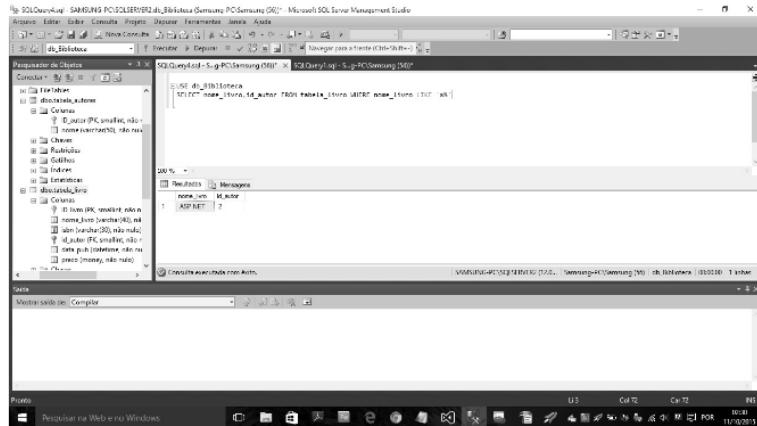


Imagen 8.9 – Uso de filtro LINK 'a%'

Exemplos de WHERE

-- Exibir todas as informações de todos os alunos

```
SELECT * FROM ALUNO;
```

-- Exibir somente o título de cada curso **SELECT CURSO FROM CURSO;**

-- Exibir somente o título e valor de cada curso cujo preço seja maior que 200 **SELECT CURSO, VALOR FROM CURSO WHERE VALOR > 200;**

-- Exibir somente o título e valor de cada curso cujo preço seja maior que 200 e menor que 300

```
SELECT CURSO, VALOR FROM CURSO WHERE VALOR > 200 AND VALOR < 300;
```

-- Outra solução para o exercício seria esta

```
SELECT CURSO, VALOR FROM CURSO WHERE VALOR BETWEEN 200 AND 300;
```

-- Exibir as informações da tabela PEDIDOS para os pedidos realizados entre 15/04/2010 e 18/04/2010

15/04/2010 e 18/04/2010

04-19';

-- Outra solução para o exercício seria esta

```
19';
```

-- Exibir as informações da tabela PEDIDOS para os pedidos realizados na data de 15/04/2010

```
SELECT * FROM PEDIDO WHERE DATE(DATAHORA) = '2010-04-15';
```

CAPÍTULO 2

1 - Introdução ao Banco de Dados

Precisamos de um espaço no disco rígido do computador onde os arquivos sejam específicos com tabelas e as tabelas sejam preenchidas com seus campos, assim, conseguimos conectar ao banco de dados e recortar ou separar as tabelas a fim de inserir, alterar, excluir e selecionar os campos. Assim começa o Banco de Dados, quando estruturamos a informação, como ela se propaga, o quê ela deve propagar, para quem se propagar.

1.1 - Definição de SGBD:

“Um Sistema de Gerenciamento de Banco de Dados (SGBD) – do inglês Data Base Management System (DBMS) – é o conjunto de programas de computador (softwares) responsáveis pelo gerenciamento de uma base de dados. Seu principal objetivo é retirar da aplicação cliente a responsabilidade de gerenciar o acesso, a manipulação e a organização dos dados. O SGBD disponibiliza uma interface para que seus clientes possam incluir, alterar ou consultar dados previamente armazenados. Em bancos de dados relacionais a interface é constituída pelas APIs (Application Programming Interface) ou drivers do SGBD,

que executam comandos na linguagem SQL (Structured Query Language).” Fonte²⁷

É um conjunto de softwares que acessam com usuários e delimitam e construem através da manipulação uma base de dados para fins da aplicação.

²⁷ Gustavo Furtado, O que é um SGBD, Wikipedia, <http://www.dicasdeprogramacao.com.br/o-que-e-um-sgbd>, 01/10/2007

Portanto, um SGBD comanda e gerencia vários bancos de dados em seu complexo sistemas de aplicações para devidas finalidades: incluir, excluir, consultar, etc.

O procedimento de execução do SGBD segue: APLICAÇÕES → PROCESSAR PROGRAMAS → ACESSAR DADOS ← → IDENTIFICAR BASE DA DOS ← → BASE DE DADOS.

E para a execução dessas atividades temos os Profissionais de Administrador de Banco de Dados que monitoram o acesso da base de dados e Implementadores de Banco de Dados que definem estruturas adequadas para armazenarem os dados.

1.1.1 – Consistência e Validade dos Dados

O SGBD verifica os dados de entrada com regra de integridade, permitindo o que são corretos. E valida se os dados são verdadeiros ou falsos referente ao domínio de valores.

PROPRIEDADE DO SGBD - VALIDADE NOME ENDEREÇO TELEFONE
Jose R. Xuy 09-394893 Manoel R. Kay 90-383838 Aparecido R. Lov
Bairro

NOTA:

Na coluna “telefone” o dado “Bairro” é inválido, pois foi digitado caractéres quando se pede número, então, entraria em ação uma máscara ou formatação do campo de entrada de dados validando esse tipo de dados.

1.2 - Sistema de Banco de Dados

Precisamos saber que o corpo humano funciona com vários sistemas:

reprodutor, respiratório, digestivo, nervoso, excretor, urinário, ósseo. Sendo que um sistema não funciona sem o outro, senão o corpo humano entra em falha ou enfermidade. E quem detecta esse mau funcionamento? Cada sistema tem sua resposta de informação, é emitido um sinal e essa informação precisa ser codificada ou interpretada. Assim, também, é um SGBD – um sistema que entram e saem informações que precisam ser interpretadas, quem as interpretam e gerenciam essas informações? Os Técnicos Certificados em Banco de Dados. E, por este motivo, é preciso uma especialização ou capacitação específica para Gerenciar o Banco de Dados, pois trabalhará com várias “Entidades”, e cada entidade tem suas “Tabelas” e cada Tabela seus “Campos”.

Aqui nesse livro compartilharemos informações que são a Porta de Entrada da Certificação Microsoft Technology Associate – MTA”. Claro que existem muitos materiais nesse segmento, o que trazemos é um diferencial onde exploraremos os “Mapas Mentais”.

1.3 - Conceitos Gerais de Banco de Dados

“Banco de Dados - Representa o arquivo físico de dados, armazenado

em dispositivos periféricos, onde estão armazenados os dados de diversos sistemas, para consulta e atualização pelo usuário.

- . Dados: É o valor do campo quando é armazenado no Banco de Dados;
- . Tabelas Lógicas: Representam as estruturas de armazenamento de dados (arquivos) dos sistemas;
- . S.G.D.B. (Sistema Gerenciador de Banco de Dados) : É o software responsável pelo gerenciamento (armazenamento e recuperação) dos dados no Banco de Dados;
- . Conteúdo do campo:- É o valor do campo armazenado no Banco de Dados;
- . Modelo de Banco de Dados: Modelo Relacional, Modelo Hierárquico e Modelo em Rede;
- . Representa a estrutura física no qual o armazenamento dos dados foram projetados. O modelo identifica a estrutura interna de recuperação e armazenamento dos dados no qual o SGBD foi projetado.” Fonte ²⁸

1.3.1 – Modelagem de Banco de Dados

Se faz muito necessário antecipar o que se fazer. Esse planejamento na construção de um Banco de Dados é indispensável, pois por esse caminho enxergamos a viabilidade de entrada e saída de informação. Pois, surgem uma tempestade de ideias através das Equipes envolvidas na Administração da Empresa, no Gerenciamento, no Controle, ou melhor em todos os departamentos. E por isso, um “Mapa Mental” usando de simbologia e organização de → entrada → processamento → saída.

- . Quantos departamentos na Empresa?
- . O que faz cada departamento?
- . O que entra dos departamentos?
- . O que executam, processam nesses departamentos?
- . O que sai desses departamentos?

NOTA:

No Capítulo 3, seguinte, mais detalhes sobre Modelagem de Dados. O leitor absorve o conteúdo interagindo com outros conteúdos que,

também, está contido em Modelagem de Dados. Mas, não pule esse Capítulo, siga a sequência de estudos.

28 Introdução à Banco de Dados, <http://people.ufpr.br/~ademirlp/BancoDados.pdf>, 01/10/2015

1.3.1.1 – Estrutura da Modelagem

- . Com quem se relacionar cada departamento/entidade;
- . Atributos dos dados de cada departamento;
- . Quais as entidades ;
- . Quais os problemas;
- . O que definir como solução dos problemas;
- . Quais os valores únicos das entidades: chaves que visam integridade dos dados;

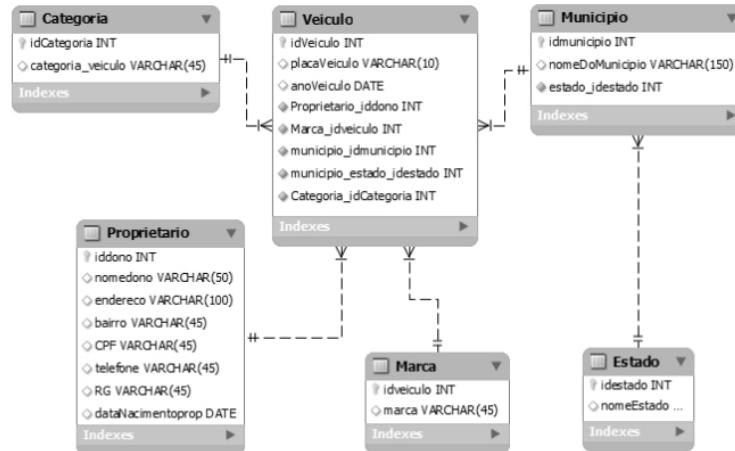
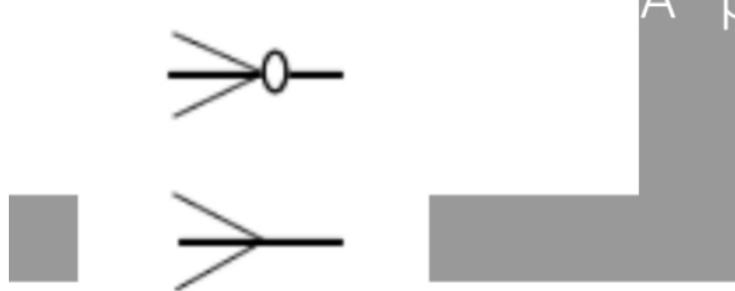


Diagrama ou “Mapa Mental” Banco Dados Relacional - Imagem 1

1.3.1.1.1 – Modelo Relacional

NOTA:



Aparece a presença de um destes símbolos de uma coluna como condição das entidades (tabelas).

ímbolos é indicação da chave estrangeira no relacionamento

presença
onamento

No “Mapa Mental” acima observe as “Entidades”, categoria, veículos, município, proprietário, marca e estado. E cada “Entidade” tem seus campos específicos.

- . Abra o “dbForge Studio Express for SQL Server”, vamos administrar o banco de dados com ele, agora;
- . Conecte com o banco de dados “db_Biblioteca” clicando em “New Connection” ou poderá já está aberto a conexão na área “Database Explorer”, caso sim, clique em “Samsung-pc\SQLSERVER2” no seu

computador poderá ser diferente a palavra “Samsung-pc” mas deverá se igual a Instância SQLSERVER2;

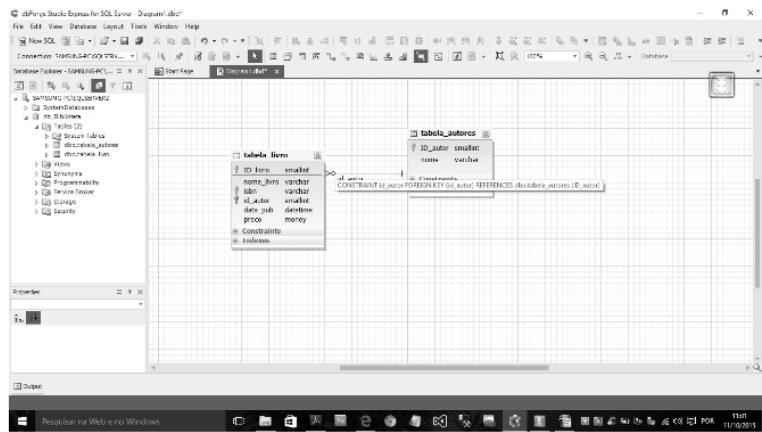


Imagen 5.5 – Database Diagrama

- Clique o menu “Database” selecione “Database Diagram” aparece a tela da imagem 5.5;
- Ao lado esquerdo “Database Explorer” aparece o banco de dados já em conexão e as Tabelas “tabela_autores” e “tabela_livro”. Agora clique

e segure no primeiro botão do mouse sobre ícone “tabela_autores” e arraste para área do “Database Diagram”, repita esse passo, também, com a “tabela_livro” e veja que se forma o layout da tela imagem 5.5;

. Observe que na “tabela_livros” o sinal de chave estrangeira do campo “id_autor” presente na “tabela_livros”. Automaticamente identificado por o “Database Explorer”.

1.3.1.1.2 – Modelo Hierárquico

O modelo que segue um diagrama com simbolos de retângulos, e linhas de links. Esse procedimento, também, conhecido com fluxograma, a fim de melhor criar o “Mapa Mental”, facilitando a percepção de entrada e saída de informações.

“Um diagrama com estrutura de árvore é um esquema para um banco de dados hierárquico. Tal diagrama consiste em dois componentes básicos: retângulos, que correspondem a tipos de registro, e linhas, que correspondem a ligações. O diagrama com estrutura de árvore serve para os mesmos propósitos que um diagrama

entidade de relacionamento; a saber, ele especifica a estrutura lógica geral do banco de dados. "Fonte"²⁹

Esses diagramas são elaborados juntamente com Equipe de Administração Geral da Empresa, afinal de contas, são eles que convivem e vivem com uma realidade interna e o Profissional de Tecnologia de Informação é um filtro das informações mais importantes que devem chegar até aos Gestores da Empresa a fim de muní-los de conhecimento e informação para a tomada de decisões, elaboração de estratégias, implementação de atividades etc.

29 Felipe Giovanangelo, Um pouco sobre banco de dados hierárquico, 2001,
[http://www.profissionaisti.com.br/2011/12/um-pouco-sobre-banco-de-dados-hierarquico/, 02/10/2015](http://www.profissionaisti.com.br/2011/12/um-pouco-sobre-banco-de-dados-hierarquico/)

Veja na imagem 2, tem a frase "Entidade Relacional", isso é porque o Diagrama tem essa finalidade é esclarecer quem se relaciona com quem e, essas ações e reações são demonstradas no "Mapa Mental em Diagramação Hierárquica".

Modelagem Conceitual do Banco de Dados (Projeto Febem)

Entidade Relacional (ER)

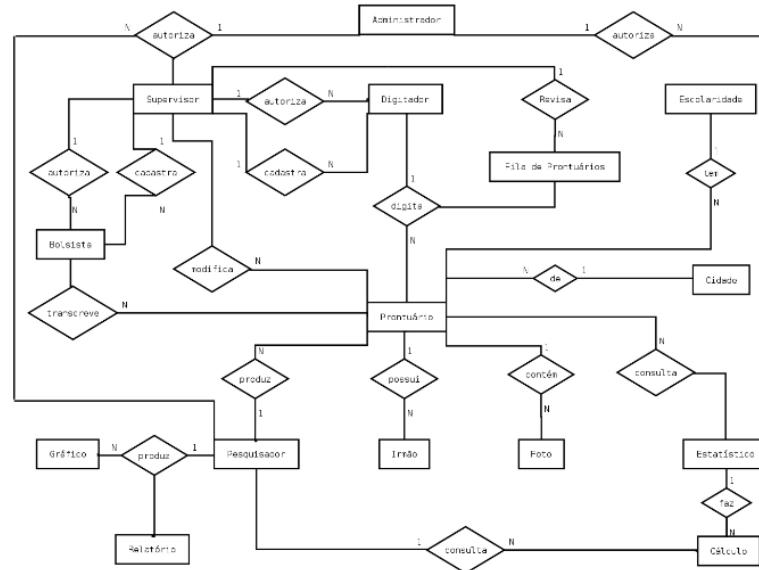
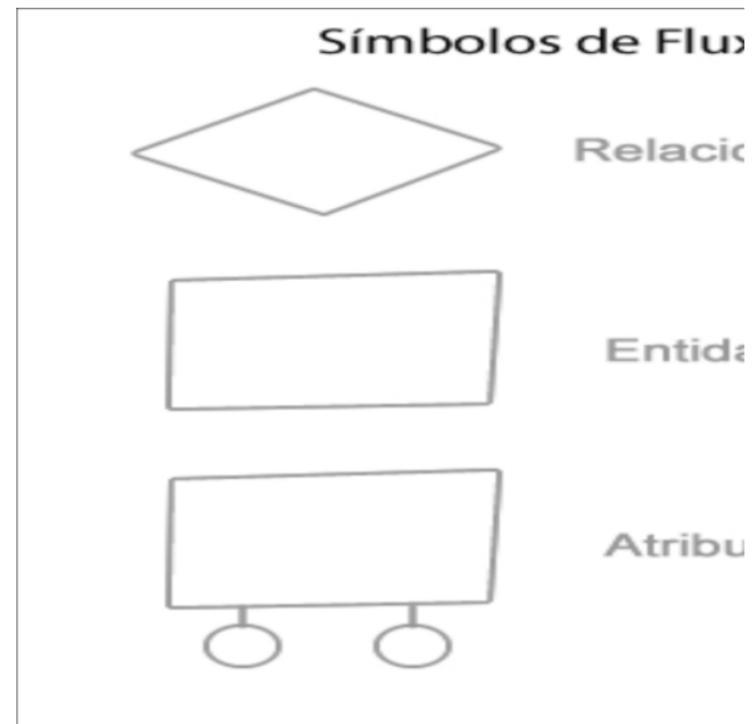


Diagrama ou fluxograma hierárquico - Imagem 2

A grande utilidade desse "Mapa Mental" ou Diagrama é o conhecimento de cima para baixo, onde o topo do Diagrama informa

a principal das entradas de dados, na imagem 2, no topo temos “Administrador” ele é quem primeiro recebe as informações para tão somente deliberar atividades e executar planejamentos.

Observe que o “Administrador” → “Autoriza” → “Supervisor”, a este não será permitido uma decisão sem o conhecimento prévio do Plano de Atividade da Empresa, pois assim, a tomada de decisões será diminuído em seus riscos de erros, pois, várias cabeças pensam melhor que uma, e mesmo que uma cabeça tenha uma ideia excelente deverá ser submetido a Equipe, pois estamos falando de um “Sistema”, onde um depende do outro, e como é o funcionamento do corpo humano, assim é uma Empresa.



kograma

onamento

ades

itos

1.3.1.1.3 - Modelo em Rede

Estamos na modelagem de banco de dados mais avançada de todas, aqui não é analisado que toda entrada de dados seja por uma raiz, ou seja, por um “caminho”, mas por “vários caminhos” essas informações podem entrar, e por isso é definido como “modelo em rede”. O que exige dos Administradores de Banco de Dados um maior cuidado com o item Segurança da Informação, pois um Funcionário pode digitar informações e, por isso, exige um algoritmo de “Auditoria de Informação”.

“A o contrário do Modelo Hierárquico, em que qualquer acesso aos dados passa pela raiz, o modelo em rede possibilita acesso a qualquer nó da rede sem passar pela raiz.” Fonte³⁰

Lembrando que quando criamos uma Tabela “cliente” no Banco de Dados os campos “codigo do cliente”, “nome do cliente”, além de definir “o tipo do dados” que pode ser “texto”, “alfanumérico”, “numérico”, “data”, “inteiro”, agora, também, é definido a “ocorrência do registro”.

“ A organização do Modelo em Redes é semelhante a do Modelo Hierárquico, mas com a diferença de que cada registro filho pode ser ligado a mais de um registro pai, criando conexões bastante complexas

e são bastante utilizados em sistemas para computadores de grande porte³¹

Em nosso Sistema de Gestão Educacional, desenvolvido em Linguagem ASP, SQL e banco de Dados Access, com migração para banco de dados MySQL, utilizamos esse modelo em rede (Veja o nosso livro “Programando na WEB com ASP Clássico, Ed.

30 Eduardo Javarotti ,Modelo de Redes em Banco de Dados , 2008,
<http://www.administradores.com.br/artigos/negocios/modelo-de-redes-em-banco-de-dados/26541/>, 02/10/2015

31 Ibidem

Viena, 2015”). E por isso, temos um script que “Audita quem acessou, o que digitou, onde digitou, hora e data que entrou com informações”, dessa maneira evitamos culpar a quem não é de direito por algum erro humano. Por que cada Professor lançará notas dos seus alunos em uma conexão pessoal, na sua casa, ou conexão corporativa na sua Empresa. Porém, a responsabilidade é do Professor, que não impede de transferir essa responsabilidade para outrem, e, portanto, em caso de um erro ou equívoco é guardado o “registro da ocorrência: dia, data, ip

da máquina, horário, e o que fez”.

NOTA:

Em nosso livro Programando na WEB com ASP Clássico, citamos o exemplo de conexão com o Banco de Dados MySQL para demonstração de poder e interatividade da Linguagem Nativa da Microsoft ASP Clássico que funciona sobre a Plataforma Sistema Operacional Windows Server. Sendo, assim, nosso maior objetivo foi mostrar o poder do ODBC que faz essa ponte de ligação com Drivers de Conexão do ASP (Microsoft) e Driver do Banco de Dados MySQL (Oracle).

Lembrando que MySQL é o banco de dados da Oracle de livre uso, sendo usado o Servidor Apache que funciona tanto em 32bits ou 64bits sobre os Sistemas Operacionais Windows ou Linux.

1.3.1.1.4 - Revisando Criação de Banco de Dados

“1. A sintaxe ANSI SQL apropriada refere-se ao conjunto de regras que:
a. decide se um campo pode conter um valor em branco ou não
b. determina o tamanho de todos os campos de dados c. governa a estrutura e conteúdo das instruções

- 2. Qual comando SQL é usado para inserir uma nova tabela?**
- a. CREATE TABLE table_name (column_name data type null/not null, column_name data type null/not null, e assim por diante)
- b. ADD TABLE table_name (column_name data type null/not null, column_name data type null/not null, e assim por diante)
- c. INSERT TABLE table_name (column_name data type null/not null, column_name data type null/not null, e assim por diante)"

Respostas

1. A sintaxe ANSI SQL apropriada refere-se ao conjunto de regras que:
- c. governa a estrutura e conteúdo das instruções**
2. O comando SQL usado para inserir uma nova tabela é: a. CREATE TABLE table _ name (column _ name data type null/not null, **column_name data type null/not null, etc.**)
3. O campo de dados que Yan pode usar para uma chave exclusiva de

acesso aos dados na tabela é:
b. nome da gravadora do CD³²

1.3.1.1.4.1 – Auditoria de Informação

Tem como finalidade acompanhar, monitorar as informações inseridas, excluídas e alteradas, sendo registrado em memorando o tipo da ocorrência, o dia, a data, ip da máquina e horário a fim de, até mesmo:

- . Medir desempenhos de funcionários;
- . Analisar poder de resposta da infraestrutura de Servidores;
- . Obter informações das entidades mais acessados;
- . O que mais são acessados no Banco de Dados;
- . Quem acessou e porquê acesso;
- . Produzir relatórios de planejamentos que antecipam erros e equívocos;
- . Segurança de Informação;
- . Monitorar o período de backup ou cópia de segurança do Banco de Dados;

1.3.1.1.4.1.1 – O que é uma auditoria em BD?

O mecanismo que funciona o SQL Server requer um controle com registro de “log” dos eventos que ocorreram no mecanismo do Banco de Dados.

32 Ibidem

- . “Todas as edições do SQL Server oferecem suporte a auditorias no nível do servidor ;
- . Gravar grupos de ação de auditoria de servidor por instância ;
- . O evento de auditoria ocorrerá sempre que a ação auditável for encontrada;

A Microsoft define “Auditoria é a combinação de vários elementos em um único pacote de um grupo específico de ações de servidor ou de banco de dados.Os componentes de auditoria do SQL Server são combinados para produzir uma saída conhecida como auditoria, da mesma forma como uma definição de relatório combinada com elementos gráficos e de dados produz um relatório.”³³

1.3.1.1.4.1.1.a – Como auditar um DB?

Como estamos focados em elaborar um conteúdo que atenda as terminologias, bem como, os termos técnicos utilizados na elaboração do Exame 98-364 da Certificação Microsoft SQL Server, então vamos ver o quê diz os Técnicos dessa Empresa. Atentemos para o estilo de elaboração do conteúdo Oficial da Microsoft, cito:

“Ao definir uma auditoria, especifique o local de saída dos resultados. Esse é o destino da auditoria. A auditoria é criada em um estado ‘desabilitado’ e não audita automaticamente nenhuma ação. Após a habilitação da auditoria, o destino da auditoria recebe dados da auditoria.”³⁴

33 Equipe MSDN, Auditoria do SQL Server – Mecanismo de Banco de Dados, [https://msdn.microsoft.com/ptbr/library/cc280386\(v=sql.120\).aspxm](https://msdn.microsoft.com/ptbr/library/cc280386(v=sql.120).aspxm), 02/10/105

34 Ibidem

1.3.1.1.4.1.b – O quê auditar em DB?

1.3.1.1.4.1.b.1 – Recursos e tarefas no Banco de Dados

1.3.1.1.4.1.b.1.a - Nas Tabelas

Segundo a Equipe da Microsoft em “Tabelas”, “ Por exemplo, você poderá criar restrições em uma coluna para não permitir valores nulos ou fornecer um valor padrão se um valor não for especificado, ou você poderá atribuir uma restrição chave na tabela que impõe exclusividade ou define uma relação entre tabelas”.³⁵

Bancos de dados Cursor Fazer backup e restaurar bancos de dados do SQL Server

Tabelas Números de sequência Importação e exportação em massa de dados (SQL Server)

OLTP na memória (otimização na memória)

Gatilhos DDL Compactação de dados

Índices Gatilhos DML Objetos de automação OLE em TransactSQL

Tabelas e índices particionados Sinônimos (Mecanismo de Notificações de eventos 35 MSDN, Recursos Tarefas no Banco de Dados, [https://msdn.microsoft.com/ptbr/library/ms189084\(v=sql.120\).aspx](https://msdn.microsoft.com/ptbr/library/ms189084(v=sql.120).aspx), 02/10/2015 Banco de Dados)

Exibições Dados XML (SQL Server) Monitorar e ajustar o desempenho
Procedimento armazenados (Mecanismo de Banco de Dados)

Dados espaciais (SQL Server)

Pesquisar (SQL Server)

Dados de objeto binário grande (Blob) (SQL Server)

Funções definidas pelo usuário Aplicativos da camada de dados

Estatísticas

O log de transações (SQL Server)

Guias de plano

Pontos de verificação de banco de dados (SQL Server)

1.3.1.1.4.1.b.1.b - Fazer backup e restaurar bancos de dados

“O componente de backup e restauração do SQL Server oferece

uma proteção essencial para dados críticos armazenados em bancos de dados do SQL Server. Para minimizar o risco de perda de dados catastrófica, você precisa fazer backup dos bancos de dados para preservar as modificações feitas nos dados regularmente. Uma estratégia de backup e restauração bem-planejada ajuda a proteger bancos de dados contra perda de dados causada por várias falhas. Teste sua estratégia restaurando um conjunto de backups e recuperando depois seu banco de dados para se preparar para responder com eficiência a um desastre.”³⁶

1.3.1.1.4.1.b.1.c - Monitorar e ajustar o desempenho

“A meta do monitoramento de bancos de dados é avaliar o desempenho do servidor. Um monitoramento eficaz envolve a criação de instantâneos periódicos do desempenho atual para isolar processos que estão ocasionando problemas, e a coleta contínua de dados para o controle das tendências de desempenho.

A avaliação contínua do desempenho de banco de dados ajuda a

minimizar tempos de resposta e a maximizar a taxa de transferência, permitindo alcançar desempenho ótimo. Tráfego de rede, E/S de disco e uso de CPU eficientes são fundamentais para um desempenho ótimo. É preciso analisar minuciosamente os requisitos de aplicativos, compreender a estrutura lógica e física dos dados, avaliar o uso de banco de dados e negociar compensações entre usos conflitantes...”³⁷

“Monitorando os tempos de resposta a consultas utilizadas com frequência, é possível determinar se são necessárias alterações na consulta ou nos índices das tabelas em que as consultas são executadas.

Monitorando consultas Transact-SQL à medida que são executadas, é possível determinar se elas estão escritas corretamente e produzindo os resultados esperados;

36 Ibidem

37 MSDN, Monitorar e ajustar o desempenho, [https://msdn.microsoft.com/ptbr/library/ms189081\(v=sql.120\).aspx](https://msdn.microsoft.com/ptbr/library/ms189081(v=sql.120).aspx), 02/10/2015

. Monitorando os usuários que tentam se conectar a uma instância do SQL Server, é possível determinar se a segurança está configurada adequadamente e testar aplicativos ou sistemas de desenvolvimento.”³⁸

Não são poucos os casos que nos deparamos com lentidão de conexão com o servidor, lentidão de navegação provocado por falta de memória no computador cliente. São piores quando as informações congelam e pensamos que são vírus, quando não são, o que está acontecendo é um sistema ou site mau gerenciável, quando não se mede o poder de respostas por desconhecer: o tempo de resposta e a taxa de transferência do processamento desses dados que estão trafegando na Internet.

“O tempo de resposta é o tempo necessário para que a primeira linha do conjunto de resultados seja retornada para o usuário, na forma de uma confirmação visual de que uma consulta está sendo processada. A taxa de transferência é o número total de consultas manipuladas pelo servidor durante um período de tempo especificado.”³⁹

A relação de motivos que fazem as pessoas desistirem de comprar em um site é grande:

38 Ibidem 39 ibidem

- . “Erro de configuração de site;
- . Instabilidade de Provedor de Hospedagem do site;
- . Falta de informação das instabilidades;
- . Falta de interação com os clientes através do site.”⁴⁰

Esses são alguns motivos técnicos de falta de monitoramento, e sem levar em consideração design inadequado, uso de cores sem padrão ou desarmonia de cores, imagens JPG ou GIF pesados etc.

1.3.1.1.4.1.b.1.d - Índices e chave primárias

“

Tipo de índice

Hash

Descrição

Com um índice de hash, os dados são acessados por meio de uma tabela de hash na memória. Os índices de hash consomem uma quantidade fixa de memória, que é uma função do número de buckets.

índices não clusterizados com otimização de memória
Clusterizado

Para índices não clusterizados com otimização de memória, o consumo de memória é uma função da contagem de linhas e do tamanho das colunas de chave de índice.

O índice clusterizado classifica e armazena as linhas de dados da tabela ou exibição em uma ordem com base na chave do índice clusterizado. O índice clusterizado é implementado como uma estrutura de índice da árvore B que oferece suporte à recuperação rápida de Não clusterizado

40 Edmar Ferreira, Os 10 principais motivos que fazem seus clientes cancelarem, <http://startupceo.com.br/os10-principais-motivos-que-fazem-seus-clientes-cancelarem/>, 02/10/2015

Exclusivo

Columnstore linhas com base em seus valores da chave de índice clusterizado.

Um índice não clusterizado pode ser definido em uma tabela ou exibição com um índice clusterizado ou em um heap⁴¹. Cada linha do

índice não clusterizado contém o valor da chave não clusterizada e um localizador de linha. Esse localizador aponta para a linha de dados no índice clusterizado ou heap que possui o valor da chave. As linhas do índice são armazenadas na ordem dos valores da chave de índice, mas não há garantias de que as linhas de dados estejam em uma determinada ordem, a menos que um índice clusterizado seja criado na tabela.

Um índice exclusivo garante que a chave de índice não contenha valores duplicados; portanto, cada linha em uma tabela ou exibição é, de alguma forma, exclusiva. A exclusividade pode ser uma propriedade de índices clusterizados e não clusterizados. Um índice columnstore na memória armazena e gerencia dados usando o armazenamento de dados baseado em coluna e o processamento de consulta baseado em coluna.

Os índices columnstore funcionam bem para as cargas de trabalho de “data warehouse” que

41 Wikipedia, <https://pt.wikipedia.org/wiki/Heap>, **Heap** é uma estrutura de dados organizada como árvore binária balanceada, seguindo algumas regras. Este pode ser implementado em um arranjo,

para que ele seja acessado como uma árvore binária, usando as seguintes operações:

```
PAI(i)  
return i/2  
ESQUERDA(i) return i*2  
DIREITA(i)  
return i*2 + 1
```

Índice com
colunas incluídas Índice em colunas computadas.

Filtrado
Espacial
XML

Texto completo executam principalmente carregamentos em massa e consultas somente leitura. Use o índice columnstore para obter um ganho de desempenho de consulta até 10 vezes maior sobre o armazenamento tradicional orientado por linha e decompressão de dados até 7 vezes maior sobre o tamanho dos dados não compactados.

Um índice não clusterizado que é estendido para incluir colunas que não são de chave, além das colunas de chave.

Um índice em uma coluna que é derivada do valor de uma ou mais colunas ou certas entradas deterministas.

Um índice não clusterizado aperfeiçoado, especialmente indicado para abranger consultas que selecionam de um subconjunto bem definido de dados. Ele usa um predicado de filtro para indexar uma parte das linhas da tabela. Um índice filtrado bem projetado pode melhorar o desempenho da consulta, e reduzir os custos de manutenção e de armazenamento do índice em comparação com os índices de tabela completa.

Um índice espacial permite a execução de determinadas operações de forma mais eficiente em objetos espaciais (dados espaciais) em uma coluna do tipo de dados geométrico. O índice espacial reduz o número de objetos nos quais operações espaciais relativamente dispendiosas precisam ser aplicadas.

Uma representação fragmentada e persistente de BLOBS (objetos binários grandes) XML na coluna de tipo de dados xmlxml.

Um tipo especial de índice funcional com base em token que é criado e mantido pelo Mecanismo de Texto Completo da Microsoft para o

SQL Server. Ele fornece suporte eficiente para pesquisas sofisticadas de palavras em dados de cadeias de caracteres.

Fonte de informação ⁴² [42 MSDN, Índices,](https://msdn.microsoft.com/pt-br/library/ms175049(v=sql.120).aspx) [https://msdn.microsoft.com/pt-br/library/ms175049\(v=sql.120\).aspx,](https://msdn.microsoft.com/pt-br/library/ms175049(v=sql.120).aspx) 02/10/2015

1.3.1.1.4.1.b.1.e – Notificação de Eventos

. “Registrar em ‘ arquivo log ’ e examinar alterações ou atividade ocorridas no banco de dados;

. Executar uma ação em resposta a um evento de maneira assíncrona (que não acontece simultaneamente), em vez de síncrona (que acontece simultaneamente);

As notificações de evento podem oferecer uma alternativa de programação a gatilhos (ou seja disparos programados dia, hora e o que fazer) com DDL e ao Rastreamento do SQL.

Notificações de evento são executadas de forma assíncrona (que

não acontece simultaneamente), fora do escopo de uma transação. Portanto, ao contrário dos gatilhos DDL - Linguagem de Definição de Dados, as notificações de evento podem ser usadas dentro de um aplicativo de banco de dados para responder a eventos sem usar nenhum recurso definido pela transação imediata.

Ao contrário do Rastreamento do SQL, as notificações de evento podem ser usadas para executar uma ação dentro de uma instância do SQL Server em resposta a um evento do Rastreamento do SQL.

Os dados de evento podem ser usados por aplicativos que se encontram em execução junto com o SQL Server para rastrear o andamento e tomar decisões. Por exemplo, a notificação de evento a seguir envia um aviso a determinado serviço todas as vezes que uma instrução ALTER TABLE é emitida no banco de dados de exemplo AdventureWorks2012AdventureWorks2012 ..

AdventureWorks2012 ..

Código fonte SQL:

```
USE AdventureWorks2012;
GO
```

```
CREATE EVENT NOTIFICATION NotifyALTER_T1 ON DATABASE
FOR ALTER_TABLE
TO SERVICE '//Adventure-Works.com/ArchiveService',
'8140a771-3c4b-4479-8ac0-81008ab17984';43
```

Por fim, “ Os resultados de uma auditoria são enviados a um destino que pode ser um arquivo, o log de evento de Segurança do Windows ou o log de eventos de Aplicativo do Windows. É necessário verificar e arquivar os logs periodicamente para certificarse de que o destino tenha espaço suficiente para gravar mais registros”⁴⁴.

Eslcarecemos que as principais “auditorias” foram mencionadas com seus respectivos detalhes, haja vista que no Exame 98-364 da Microsoft SQL Server não é tão requisitado esses assuntos, no entanto, não gostaríamos que o leitor não ampliasse seus conhecimentos, haja vista que, essas informações condizem com uma grande necessidade em Gestão de TI em Banco de Dados.

43 MSDN, Notificação de Eventos,[https://msdn.microsoft.com/pt-br/library/ms186376\(v=sql.120\).aspx](https://msdn.microsoft.com/pt-br/library/ms186376(v=sql.120).aspx), 02/10/2015

44 Technet, Auditoria do SQL Server, <https://technet.microsoft.com/>

pt-br/library/cc280386(v=sql.110).aspx, 02/10/2015

2 - Instruções BACKUP e RESTORE (Transact-SQL)

NOTA:

No Primeiro Capítulo no item 1.1.1.6 - Entender conceitos de segurança do banco de dados é um dos conteúdos programático da Certificação Database Fundamental Exame 98-364.

Entender os backups e restauração do banco de dados. Portanto, o conteúdo a seguir é Documento Oficial da Microsoft SQL Server, haja vista que é um conteúdo expansivo, rico em detalhes. Alertamos ao leitor que essas instruções do Transact-SQL são as requisitadas e relacionadas no Conteúdo Programático do Exame 98-364. Assim o conteúdo é focado, informações necessárias a promover segurança de roteiro correto de estudos. Além de que nesse caso não inventamos a roda, ela é redonda, se já descobriram como funcionam vamos aperfeiçoá-la, se for viável. Quero dizer – o conteúdo a seguir

funciona, não precisaríamos inventar caminhos longos, filtramos aqui o fundamental para o bom estudo.

2.1 - Instrução BACKUP⁴⁵

45 [https://msdn.microsoft.com/pt-br/library/ms186865\(v=sql.120\).aspx](https://msdn.microsoft.com/pt-br/library/ms186865(v=sql.120).aspx)

Faz backup de um banco de dados completo do SQL Server para criar um backup de banco de dados ou um ou mais arquivos ou grupos de arquivos do banco de dados para criar um backup de arquivo (BACKUP DATABASE).

The screenshot shows the SSMS interface with two panes. The top pane displays a T-SQL script for a simple backup:

```
USE db_Biblioteca
-- BACKUP SIMPLES DE TODO O BANCO DE DADOS
BACKUP DATABASE db_Biblioteca
TO DISK = 'C:\DOCUMENTS\db_biblioteca.bak';
GO
```

The bottom pane shows the execution results in a table:

Linha	Coluna
1	1
2	1
3	1
4	1

Details of the execution steps are listed below the table:

- 1. Começou backup de banco intitulado 'db_Biblioteca'.
- 2. Processou 380 objetos para o banco de dados 'db_Biblioteca', incluindo 'db_Biblioteca' no arquivo 1.
- 3. Processou 2 logins para o banco de dados 'db_Biblioteca', incluindo 'sa' no arquivo 1.
- 4. FIMEXECUTARBACKUP processou cerca de 250 páginas em 0,37 segundos (0,411 MB/s).

Imagen 8.7 – Backup Simples do Banco de Dados

O código fonte SQL:

1. USE db_Biblioteca
2. GO
3. -- BACKUP SIMPLES DE TODO O BANCO DE DADOS
4. BACKUP DATABASE db_Biblioteca
5. TO DISK = 'C:\DOCUMENTS\db_biblioteca.bak';

6. GO

7.

- . Na linha 1 você abre o banco de dados;
 - . Na linha dois envia a instrução “vá” ou “avançe”;
 - . Na linha 3 um comentário usando no início “--”;
 - . Na linha 4 é informado a realizar o backup do banco de dados db_Biblioteca;
 - . Na linha 5 o destino onde será gravado esse arquivo tipo BAK, 'C:\DOCUMENTS\db_biblioteca.bak' – poderia ser também uma porta USB ou outra unidade de disco rígido – HD;
 - . Na linha 6, novamente, o comando GO – ou seja vá ou avance;
 - . Nossa caso usamos aspas simples ' - não use aspas “ ”;
 - . Na imagem 8.8 verifique a pasta se está gravado o arquivo "db_biblioteca.bak":
- .

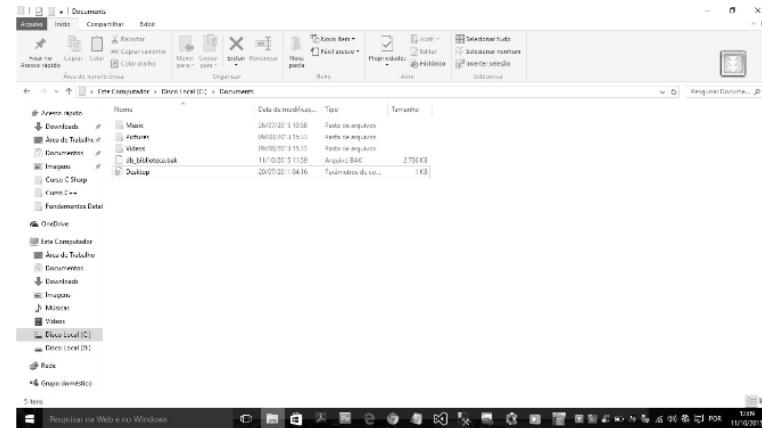


Imagen 8.8

A seguir detalhes fornecidos por Documentação Oficial Microsoft, a fim de nos munir de informações que poderão ser questões do Exame.

2.1.A - Criando um backup completo de arquivos dos grupos de arquivos secundários

O exemplo a seguir cria um backup de arquivo completo de todos os arquivos dos dois grupos de arquivos secundários.

--Backup do arquivo SalesGroup1:

BACKUP DATABASE Sales

```
FILEGROUP = 'SalesGroup1',
FILEGROUP = 'SalesGroup2'
TO DISK = 'Z:\SQLServerBackups\SalesFiles.bck';

GO
```

2.1.B - Criando um backup diferencial de arquivos dos grupos de arquivos secundários O exemplo a seguir cria um backup diferencial de cada arquivo nos dois grupos de arquivos secundários.

--Backup do arquivo SalesGroup1:

```
BACKUP DATABASE Sales
FILEGROUP = 'SalesGroup1',
FILEGROUP = 'SalesGroup2'
TO DISK = 'Z:\SQLServerBackups\SalesFiles.bck' WITH
DIFFERENTIAL;
```

GO

2.1.C - Criando e fazendo backup em um conjunto de mídias espelhado de uma única família

O exemplo a seguir cria um conjunto de mídias espelhado que contém uma única família de mídia e quatro espelhos nos quais faz backup do banco de dados AdventureWorks2012 .

```
BACKUP DATABASE AdventureWorks2012 TO TAPE = '\\.\tape0'
MIRROR TO TAPE = '\\.\tape1'
MIRROR TO TAPE = '\\.\tape2'
MIRROR TO TAPE = '\\.\tape3'
WITH
FORMAT,
```

MEDIANAME = 'AdventureWorksSet0';

2.1.D - Criando e fazendo backup em um conjunto de mídias espelhado de várias famílias

O exemplo a seguir cria um conjunto de mídias espelhado no qual cada espelho consiste em duas famílias de mídia. Em seguida, o exemplo faz backup do banco de dados AdventureWorks2012 nos dois espelhos.

```
BACKUP DATABASE AdventureWorks2012 TO TAPE = '\\.\tape0', TAPE
= '\\.\tape1' MIRROR TO TAPE = '\\.\tape2', TAPE = '\\.\tape3' WITH
```

FORMAT,

MEDIANAME = 'AdventureWorksSet1';

2.1.E - Fazendo backup em um conjunto de mídias espelhado existente

O exemplo a seguir anexa um conjunto de backup ao conjunto de mídias criado no exemplo anterior.

```
BACKUP LOG AdventureWorks2012  
TO TAPE = '\\.\tape0', TAPE = '\\.\tape1' MIRROR TO TAPE = '\\.\tape2',  
TAPE = '\\.\tape3' WITH
```

NOINIT,

MEDIANAME = 'AdventureWorksSet1';

2.2 - Instrução RESTORE⁴⁶

Restaura backups feitos por meio do comando BACKUP. Esse comando permite executar os seguintes cenários de restauração:

Restaurar um banco de dados inteiro de um backup de banco de dados completo.

(uma restauração completa).

- . Restaurar parte de um banco de dados (uma restauração parcial).
- . Restaurar arquivos ou grupos de arquivos específicos para um banco de dados (uma restauração de arquivo)
- . Restaurar páginas específicas para um banco de dados (uma restauração de página).

46 [https://msdn.microsoft.com/pt-br/library/ms186858\(v=sql.120\).aspx#restoring_full_db](https://msdn.microsoft.com/pt-br/library/ms186858(v=sql.120).aspx#restoring_full_db)

- . Restaurar um log de transações em um banco de dados (uma restauração de log de transações).
- . Reverter um banco de dados ao momento determinado capturado por um instantâneo do banco de dados.

Abra o Microsoft SQL Management Studio:

- _ Faça conexão com o banco de dados “db_Biblioteca”;
- _ Vamos restaurar o banco de dados que nós criamos “db_biblioteca.bak”. Veja imagem 8.9;

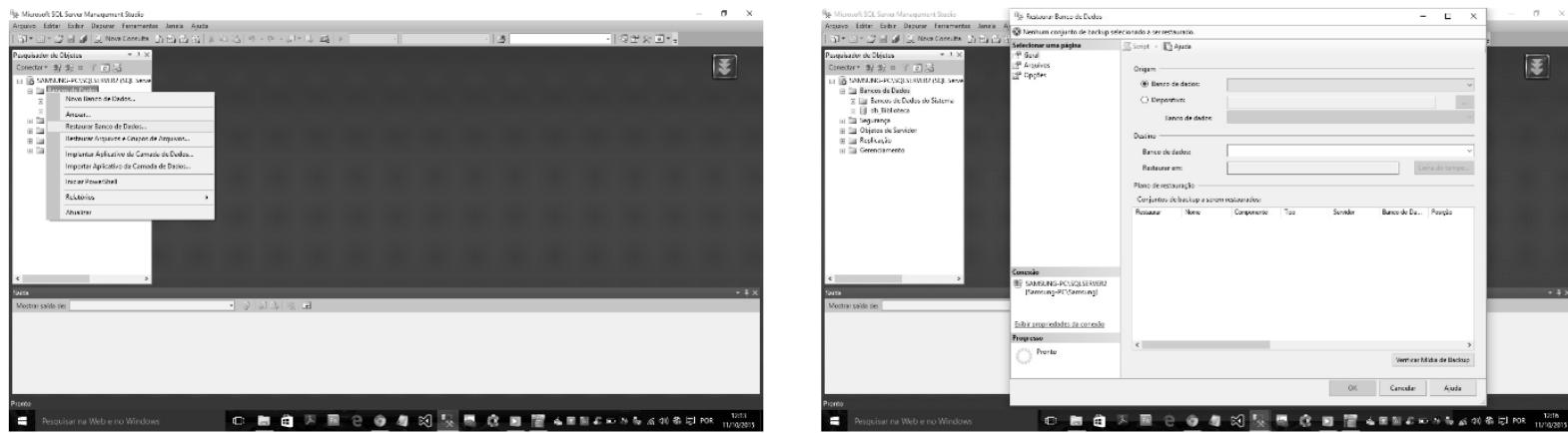


Imagen 8.9

_ Clique no segundo botão do mouse na pasta “banco de dados”, conforme imagem 8.9. Após abrir a cortina selecione “Restaurar Banco de Dados” e abre a tela da imagem 9.1;

Imagen 9.1 – Tela de Restaurar Banco de Dados

. Em “Origem” selecione “banco de dados” conforme imagem 9.2;

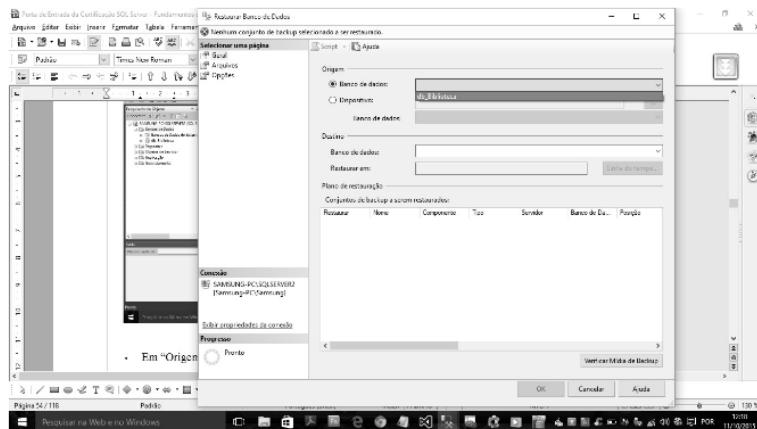


Imagen 9.2

. Agora clique no botão <OK> e aparecerá a imagem 9.3;

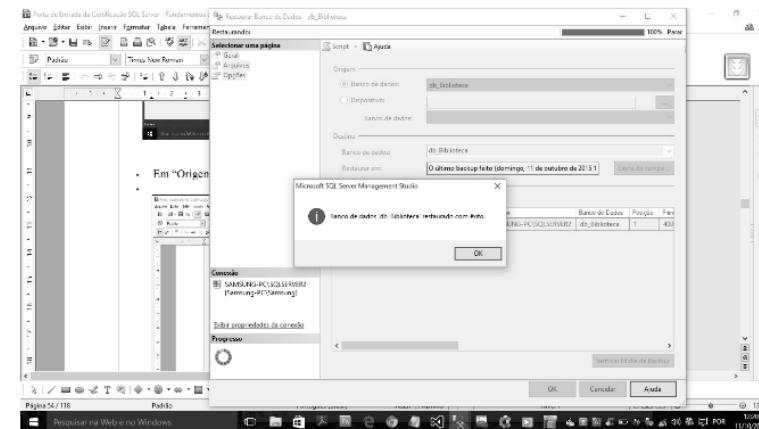


Imagen 9.3

Restaurar um banco de dados que não está em conexão, não está sendo utilizado. Vejamos como:

- . Execute o procedimento da imagem 9.2, agora;
- . Selecione “dispositivo” conforme imagem 9.3;

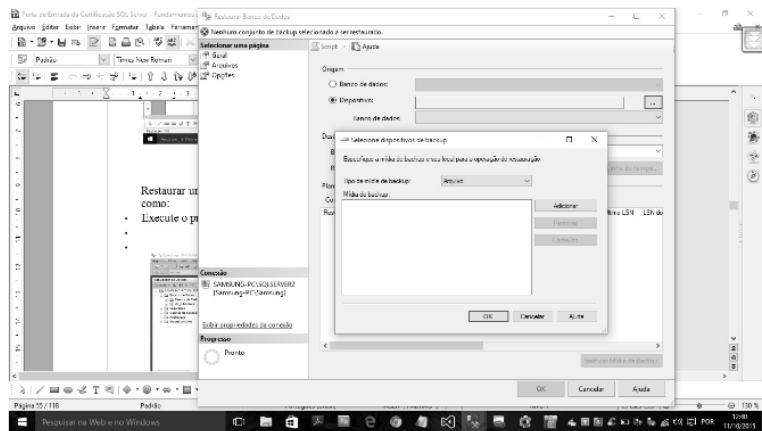


Imagen 9.3

. Clique, agora, em “adicionar” e procurei o caminho “c:\documents\” e selecionei o arquivo “test.bak” é um outro banco de dados inativo que tenho para exemplo;

. Aparece a tela da imagem 9.4;

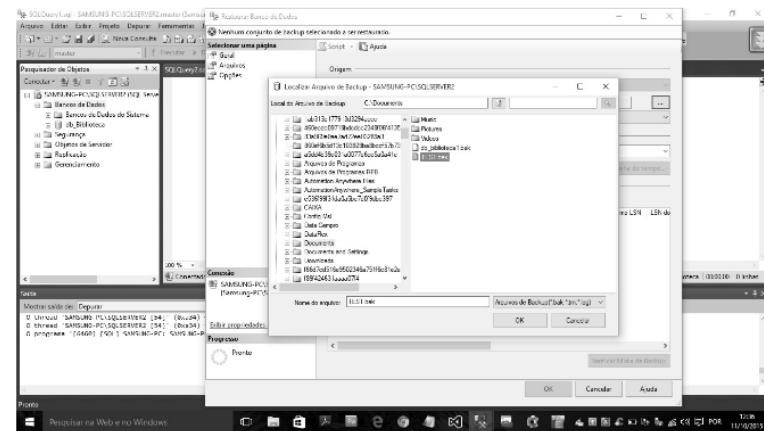


Imagen 9.4

. Seleccionei o arquivo “teste.bak” e depois cliquei em <ok>;

. Cliquei em <ok> novamente e aparece a tela imagem 9.5;

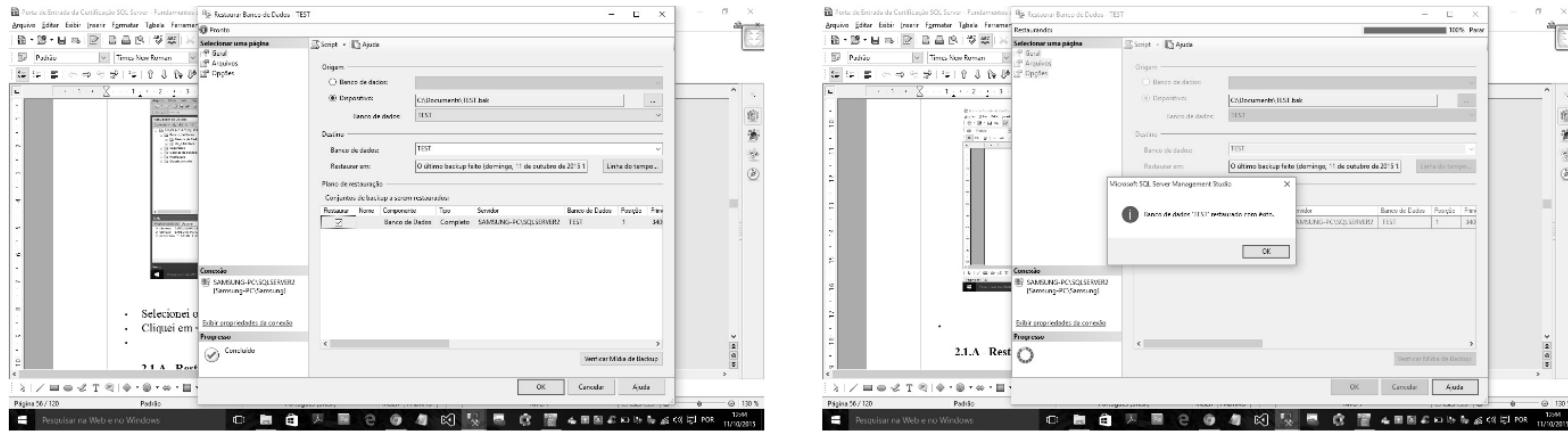


Imagen 9.5

. Novamente cliquei em <ok> e apareceu a imagem 9.6;

Imagen 9.6

Agora RESTORE com código fonte comando da SQL:

USE TEST

GO

```
-- RESTORE SIMPLES DE TODO O BANCO DE DADOS RESTORE
DATABASE TEST
FROM DISK = 'C:\DOCUMENTS\TEST.bak'; GO
```

. Restaure o banco de dados TEST que se encontra no caminho 'C:\DOCUMENTS\TEST.bak';

- . Após execução do código SQL, agora atualize a sua área de trabalho do SQL Management Studio, conforme a imagem 10.1. Selecione “Banco de Dados” use o segundo botão do mouse e selecione “Atualizar”;
- . Observe a imagem 10.2 que aparece em “Pesquisador de objetos” o banco de dados TEST;

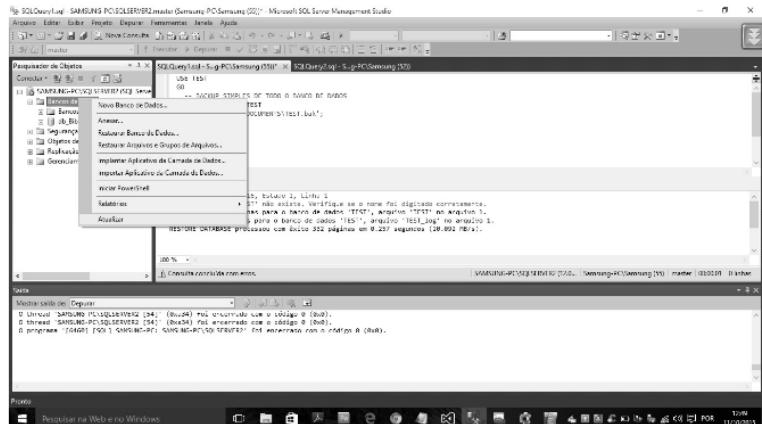


Imagen 10.1 – Atualizando a área do SQL Management Studio

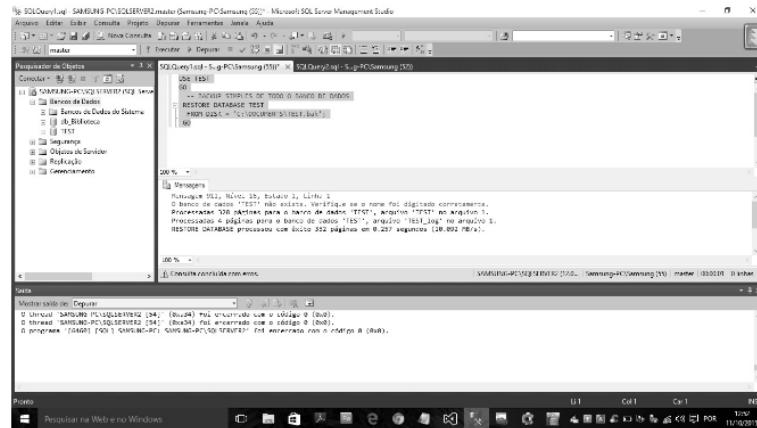


Imagen 10.2 – Visualização atualizada do Banco de Dados TEST

2.2.A - Restaurando um banco de dados completo

O exemplo a seguir restaura um backup de banco de dados completo:
USE TEST

GO

```
-- RESTORE SIMPLES DE TODO O BANCO DE DADOS RESTORE
DATABASE TEST
FROM DISK = 'C:\DOCUMENTS\TEST.bak'; GO
```

2.2.B - Restaurando backups de banco de dados diferenciais e completos

O exemplo a seguir restaura um backup de banco de dados completo seguido por um backup diferencial do dispositivo de backup Z:\SQLServerBackups\AdventureWorks2012.bak, que contém os dois backups. O backup de banco de dados completo a ser restaurado é o sexto conjunto de backup no dispositivo (FILE = 6), e o backup de banco de dados diferencial é o nono no dispositivo (FILE = 9). Assim que o backup diferencial é recuperado, o banco de dados é recuperado.

```
RESTORE DATABASE AdventureWorks2012
FROM DISK = 'Z:\SQLServerBackups\AdventureWorks2012.bak'
WITH FILE = 6
NORECOVERY;
RESTORE DATABASE AdventureWorks2012
FROM DISK = 'Z:\SQLServerBackups\AdventureWorks2012.bak'
WITH FILE = 9
RECOVERY;
```

2.2.C - Restaurando um banco de dados usando a sintaxe RESTART

O exemplo a seguir usa a opção RESTART para reiniciar uma operação RESTORE interrompida por uma falha de energia elétrica no servidor.

-- Esta base de dados RESTORE interrompido prematuramente devido à falha de energia.

```
RESTORE DATABASE AdventureWorks2012
```

```
FROM AdventureWorksBackups;
```

-- Aqui é a operação de restauração RESTART. RESTORE DATABASE AdventureWorks2012 FROM AdventureWorksBackups WITH RESTART;

2.2.D - Restaurando um banco de dados e movendo arquivos

O exemplo a seguir restaura um banco de dados completo e log de transações e move o banco de dados restaurado para o diretório C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Data.

```
RESTORE DATABASE AdventureWorks2012
FROM AdventureWorksBackups
WITH NORECOVERY,
```

```
MOVE 'AdventureWorks2012_Data' TO  
'C:\Program Files\Microsoft SQL  
Server\MSSQL12.MSSQLSERVER\MSSQL\Data\NewAdvWorks.mdf',
```

```
MOVE 'AdventureWorks2012_Log'  
TO 'C:\Program Files\Microsoft SQL  
Server\MSSQL12.MSSQLSERVER\MSSQL\Data\NewAdvWorks.ldf';  
RESTORE LOG AdventureWorks2012
```

```
FROM AdventureWorksBackups  
WITH RECOVERY;
```

2.2.E - Copiando um banco de dados usando BACKUP e RESTORE

O exemplo a seguir usa as instruções BACKUP e RESTORE para fazer uma cópia do banco de dados AdventureWorks2012 . A instrução MOVE causa a restauração dos dados e do arquivo de log nos locais especificados. A instrução RESTORE FILELISTONLY é usada para determinar o número e os nomes dos arquivos no banco de dados que está sendo restaurado. A nova cópia do banco de dados é nomeada TestDB.

```
BACKUP      DATABASE      AdventureWorks2012      TO
```

```
AdventureWorksBackups ;  
RESTORE FILELISTONLY  
FROM AdventureWorksBackups ;
```

```
RESTORE DATABASE TestDB  
FROM AdventureWorksBackups  
WITH MOVE 'AdventureWorks2012_Data' TO 'C:\MySQLServer  
\testdb.mdf', MOVE 'AdventureWorks2012_Log' TO 'C:\MySQLServer  
\testdb.ldf';
```

```
GO
```

2.2.F - Restauração pontual usando STOPAT

O exemplo a seguir restaura um banco de dados para seu estado de 12:00 AM em April 15, 2020 e mostra uma operação de restauração que envolve vários backups de log. No dispositivo de backup, AdventureWorksBackups, o backup de banco de dados completo a ser restaurado é o terceiro conjunto de backup no dispositivo (FILE = 3), o primeiro backup de log é o quarto conjunto de backup (FILE = 4), e o segundo backup de log é o quinto conjunto de backup (FILE = 5).

```

RESTORE      DATABASE      AdventureWorks2012      FROM      UPDATE Production.Product
AdventureWorksBackups
WITH FILE=3, NORECOVERY;

SET ListPrice = ListPrice * 1.10 WHERE ProductNumber LIKE 'BK-%';

GO
COMMIT TRANSACTION ListPriceUpdate; GO

-- O tempo passa. base de dados normal
- E backups do log são tomadas.
- Ocorre um erro no banco de dados. USE master;

GO

RESTORE      DATABASE      AdventureWorks2012      FROM
AdventureWorksBackups
WITH FILE = 3, NORECOVERY;
GO
RESTORE LOG AdventureWorks2012

FROM AdventureWorksBackups
WITH FILE=5, NORECOVERY, STOPAT = 'Apr 15, 2020 12:00 AM';

RESTORE DATABASE AdventureWorks2012 WITH RECOVERY;
2.2.G - Restaurando o log de transações até uma marca
O exemplo a seguir restaura o log de transações até a marca na
transação marcada denominada ListPriceUpdate.

USE AdventureWorks2012
GO
BEGIN TRANSACTION ListPriceUpdate

WITH MARK 'UPDATE Product list prices'; GO

```

```

FROM AdventureWorksBackups
WITH FILE = 4,
RECOVERY,
STOPATMARK = 'UPDATE Product list prices';

```

2.2.H - Restaurando com o uso da sintaxe de TAPE

O exemplo a seguir restaura um backup de banco de dados completo de um dispositivo de backup TAPE.

```
RESTORE DATABASE AdventureWorks2012 FROM TAPE = '\\.\tape0';
```

2.2.I - Restaurando usando a sintaxe FILE e FILEGROUP

O exemplo a seguir restaura um banco de dados nomeado MyDatabase que tem dois arquivos, um grupo de arquivos secundário e um log de transações. O banco de dados usa o modelo de recuperação completa.

No RESTORE DATABASE, observe que há dois tipos de opções de FILE. As opções FILE que precedem o nome do dispositivo de backup especificam os nomes de arquivos lógicos dos arquivos de banco de dados restaurados do conjunto de backup; por exemplo, FILE = 'MyDatabase_data_1'. Esse conjunto de backups não é o primeiro backup de banco de dados no conjunto de mídias; portanto, sua posição no conjunto de mídias é indicada com a opção

```
FILE na cláusula WITH, FILE=9. RESTORE DATABASE MyDatabase  
FILE = 'MyDatabase_data_1', FILE = 'MyDatabase_data_2',
```

```
FILEGROUP = 'new_customers' FROM MyDatabaseBackups WITH
```

```
FILE = 9,  
NORECOVERY;  
GO  
- Restaurar os backups do log.
```

```
RESTORE LOG MyDatabase FROM MyDatabaseBackups WITH FILE =  
10,
```

```
NORECOVERY;  
GO  
RESTORE LOG MyDatabase
```

```
FROM MyDatabaseBackups WITH FILE = 11,
```

```
NORECOVERY;  
GO  
RESTORE LOG MyDatabase
```

```
FROM MyDatabaseBackups WITH FILE = 12,
```

```
NORECOVERY;  
GO
```

- Recuperar o banco de dados:

```
RESTORE DATABASE MyDatabase WITH RECOVERY; GO
```

NOTA:

No Capítulo 1o. no item 1.1.1.6 - Entender conceitos de segurança do banco de dados

No Capítulo 1o. no item 1.1.1.6 - Entender conceitos de segurança do banco de dados 364.

. Entender os backups e restauração do banco de dados. Portanto, o conteúdo a seguir é Documento Oficial da Microsoft SQL Server, haja vista que é um conteúdo expansivo, rico em detalhes. Alertamos ao leitor que essas instruções do Transact-SQL são as requisitadas e relacionadas no Conteúdo Programático do Exame 98-364. Assim o conteúdo é focado, informações necessárias a promover segurança de roteiro correto de estudos. Além de que nesse caso não inventamos a roda, ela é redonda, se já descobriram como funcionam vamos aperfeiçoá-la, se for viável. Quero dizer – o conteúdo a seguir funciona, não precisaríamos inventar caminhos longos, filtramos aqui o fundamental para o bom estudo.

CAPÍTULO 3

Segundo gatilho mental: nunca diga que já sabe, encontre um espaço na sua memória para mais uma informação. Aprendendo a filtrar o quê você sabe, o quê você não sabe, o quê você está aprendendo, o quê tira atenção na compreensão e assimilação dos assuntos. Dá pra esquecer a balada do final de semana? Concentre-se.

1. Estrutura de Dados e Cardinalidade de Relacionamentos

O que é um relacionamento em banco de dados? Devido as regras do negócio as Entidades tem seus atributos que se interligam. E acontece entre duas Entidades ou mais.

- . Um aluno está em uma turma;
- . Uma turma tem vários alunos;
- . Uma turma tem várias disciplinas;
- . Um aluno está com várias disciplinas.

TURMA ALUNO DISCIPLINA 1 N

NOTA:

Cardinalidade é integridade do relacionamento entre as entidades pai e filhos. A entidade pai é aquela que não contém chave estrangeira, e a entidade filho contém a chave estrangeira de uma entidade pai.

1. Um aluno está em uma turma. Aqui é uma relacionamento 1:1, um para um;
2. Um aluno está cursando várias disciplinas. Aqui é um relacionamento 1:N, um para muitos;
3. Uma turma tem várias disciplinas. Porém, uma disciplina está em várias turmas. Aqui é um relacionamento N:N, vários para vários.

1.1 - Entidade: Tudo que está em contato, em movimento, ou inerte, que provoca ação ou

reação, que acolhe ou qualquer coisa que está no mundo real, Ex.: clientes (ação), peças (movimento), pedido (acolhe);

1.2 - RELACIONAMENTO: como as Entidade se relacionam entre si, observando “o que tem na Entidade” que se relaciona possa ser que tenha “com quem se relaciona”;

1.2.1 - TIPOS DE RELACIONAMENTOS (1:1) um para um

- Para cada registro da primeira tabela/entidade existe no máximo um correspondente na segunda tabela, vice-versa:

Tabela de PESSOAS Tabela HABILITAÇÃO-MOTORISTA NOME CPF ESTADO CPF HABILITAÇÃO José 2222222 BA 2222222 15151515
Manoel 3333333 RJ 3333333 16161616

Observando cada tabela visualizamos os campos definidos e específicos que representam cada entidade, organizadamente, facilitando a estrutura de dados e o trabalho dos Programadores e Analistas na Equipe de TI.

1.2.2 - TIPOS DE RELACIONAMENTOS (1:*) um para muitos

- Para cada registro da primeira tabela/entidade poderá existir um ou mais correspondente na segunda tabela e para cada registro

na segunda-tabela existe apenas um registro correspondente na primeira tabela

Tabela de PESSOAS Tabela VEÍCULOS NOME CPF ESTADO CPF
VEÍCULO PLACA José 2222222 BA 2222222 Carro 1001X Manoel
3333333 RJ 3333333 Moto 1002X

3333333 Carro 1003X

1.2.3 - TIPOS DE RELACIONAMENTOS (*:*) muitos para muitos

- Para cada registro da primeira tabela/entidade poderá existir um ou mais correspondente na segunda tabela e vice-versa.

Tabela de PESSOAS

NOME CPF ESTADO José 2222222 BA Manoel 3333333 RJ Tabela
PROFISSÕES
CODIGO PROFISSÃO 2 Programador 3 Webdesigner

Porquê essa nova tabela? Por que uma Aqui cria-se uma nova “Tabela pessoa pode ter várias profissões, com uma PESSOA_PROFESSÕES” (muitos para muitos) profissão está para várias pessoas. CPF PROFISSAO

2222222 2
2222222 3
3333333 2
3333333 3

Aqui nessa situação MUITOS PARA MUITOS, temos que criar uma terceira Tabela que se registre, somente, as chaves primárias, ou seja, os campos que não se repetem. Temos o CPF não se repete no cadastro de outra pessoa, cada um tem seu número único de CPF, bem como, o código de cada profissão poderá se repetir no cadastro de uma pessoa. Como o CPF da pessoa, a chave primária, vai se repetir, também, por que uma pessoa pode ter várias profissões, com uma profissão está para várias pessoas.

1.3 - Atributo: tudo que se relaciona com propriedade de uma Entidade ;

“Um campo é um local em um registro no qual um determinado tipo de dados está armazenado.

Uma coluna é um atributo para uma linha ou registro. Uma

linhalinhalinha é um registro dentro de uma tabela.

Uma tabelatabela é um objeto de banco de dados que consiste de linhas e colunas.

Um registro registro é uma estrutura de dados composta por uma coleção de campos (elementos), cada qual com seu próprio nome e tipo, que aparece em uma tabela como um grupo de campos em uma linha.”⁴⁷

2 - Conceitos de bancos de dados relacionais

“CENÁRIO: CENÁRIO: O próximo passo de Raj ao projetar o banco de dados para facilitar o negócio de automóveis de seu pai, é determinar os relacionamentos mais úteis entre as tabelas. Ele sabe que precisa criar um banco de dados relacional porque será importante fazer pesquisas usando dados em uma tabela para encontrar dados adicionais em outra. Criar o projeto que melhor atenda as necessidades da loja de automóveis é um passo crítico. Ao revisar seus atuais requisitos de negócios, ele percebe que precisa inserir outra tabela para incluir os fornecedores das peças para as automóveis. Esta nova tabela requer que Raj atualize a

tabela de peças para incluir uma chave estrangeira para a tabela de fornecedores.

Qual é a relação entre a tabela de fornecedores e a tabela de peças?

- a. um-para-um
- b. um-para-muitos

47 MSDN, [http://msdn.microsoft.com/en-us/library/aa174501\(SQL.80\).aspx](http://msdn.microsoft.com/en-us/library/aa174501(SQL.80).aspx), 02/10/2015 c. muitos-para-muitos

Qual é uma boa opção para uma chave estrangeira na relação entre a tabelas de automóveis e a tabela de peças?

- a. número da peça
- b. número do modelo da bicicleta
- c. nome da peça

Que sugestão você daria como chave primária da tabela de acessórios?

- a. número do acessório
- b. nome do acessório
- c. modelo do acessório

Resposta: 1.b; 2.a; 3.a”⁴⁸

2.1 – Chave Estrangeira⁴⁹

Esse exemplo a seguir já conhecemos, vamos revisá-lo:

Código fonte SQL:

⁴⁸ Certiport, Documento Oficial Exame 98-364 - Apostila Preparação para Certificação do MTA,http://certiport.com/Marketing/Mta/Localization/PTB/MTA_SSG_DbAdmin_individual_PTB.pdf, 03/10/2015

⁴⁹ MSDN, [https://msdn.microsoft.com/pt-br/library/ms179610\(v=sql.120\).aspx](https://msdn.microsoft.com/pt-br/library/ms179610(v=sql.120).aspx), 04/09/2015

NOTA: Não execute novamente, não funcionaria, pois já foi executado o código fonte.

```
USE db_Biblioteca
ALTER TABLE tabela_livro
ADD CONSTRAINT id_autor
FOREIGN KEY (id_autor)
REFERENCES tabela_autores(id_autor)
```

. Observe que a imagem 6.9, traz o resultado do código fonte SQL, acima mencionado. Ele tem por finalidade alterar o atributo da coluna “id_autor” na “tabela_livros” o coluna “id_autor” para ser uma chave estrangeira (Foreign Key) e essa chave é referência na “tabela_autores” na coluna “id_autor”;

. Na imagem 6.9 observe que a coluna “id_autor” recebe uma chave prata indicando chave estrangeira. E a chave ouro da coluna “id_livro” como primária da “tabela_livros”;

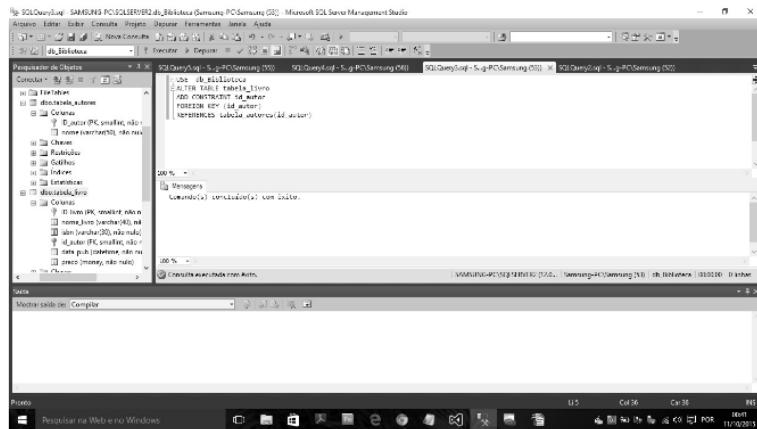


Imagen 7.4 – Criar uma Chave Estrangeira na Tabela_livros

2.1.1- Criar relações de chaves estrangeiras no SQL Server 2014⁵⁰

Usando o SQL Server Management Studio ou o Transact-SQL.Você cria uma relação entre duas tabelas quando deseja associar linhas de uma tabela com linhas de outra. 50 MSDN, [https://msdn.microsoft.com/pt-br/library/ms189049\(v=sql.120\).aspx](https://msdn.microsoft.com/pt-br/library/ms189049(v=sql.120).aspx),

03/09/2015

. Uma restrição de chave estrangeira não precisa estar vinculada apenas a uma restrição de chave primária em outra tabela; ela

também pode ser definida para referenciar as colunas de uma restrição UNIQUE em outra tabela;

. Quando um valor diferente de NULL é inserido na coluna de uma restrição FOREIGN KEY, o valor deve existir na coluna referenciada; caso contrário, será retornada uma mensagem de erro de violação de chave estrangeira.Para garantir que todos os valores de uma restrição FOREIGN KEY composta foram verificados, especifique NOT NULL em todas as colunas participantes;

. As restrições FOREIGN KEY só podem fazer referência a tabelas que estão no mesmo banco de dados e no mesmo servidor.A integridade referencial em todos os bancos de dados deve ser implementada por gatilhos;

. As restrições FOREIGN KEY podem fazer referência a outra coluna da mesma tabela.Isso se chama autorreferência;

. Uma restrição FOREIGN KEY especificada no nível da coluna pode listar apenas uma coluna de referência.Essa coluna deve ter o mesmo tipo de dados da coluna na qual a restrição foi definida;

. Uma restrição FOREIGN KEY especificada no nível da tabela deve ter o mesmo número de colunas de referência da lista de colunas de restrição. O tipo de dados de cada coluna de referência também deve ser igual ao da coluna correspondente na lista de colunas;

. O Mecanismo de Banco de Dados não tem um limite predefinido quanto ao número de restrições FOREIGN KEY que uma tabela pode conter para referenciar outras tabelas nem quanto ao número de restrições FOREIGN KEY que são propriedade de outras tabelas e fazem referência a uma tabela específica. Entretanto, o número real de restrições FOREIGN KEY que pode ser usado é limitado pela configuração do hardware e pelo design do banco de dados e do aplicativo. Recomendamos que uma tabela não contenha mais de 253 restrições FOREIGN KEY e que ela não seja referenciada por mais de 253 restrições FOREIGN KEY;

. As restrições FOREIGN KEY não são impostas a tabelas temporárias;
. Se a chave estrangeira for definida em uma coluna de tipo de dados CLR definido pelo usuário, a implementação do tipo deverá oferecer suporte a uma ordenação binária;

. A coluna do tipo varchar(max) poderá participar de uma restrição FOREIGN KEY somente se a chave primária que ela referencia também estiver definida como tipo varchar(max).

2.1.1.1 - Pemissões

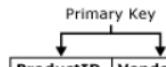
A criação de uma nova tabela com uma chave estrangeira requer a permissão CREATE TABLE no banco de dados e a permissão ALTER no esquema no qual a tabela está sendo criada.

Criar uma chave estrangeira em uma tabela existente requer a permissão ALTER na tabela.

2.2. - Chave Primária⁵¹

Geralmente, uma tabela tem uma coluna ou uma combinação de colunas que contém valores que identificam exclusivamente cada linha na tabela. Essa coluna, ou colunas, é chamada de chave primária (PK) da tabela e impõe a integridade da Entidade da mesma. Como as restrições PRIMARY KEY garantem dados exclusivos (não se repetem na tabela, CPF, RG, são chaves

primárias), elas são frequentemente definidas em uma coluna de identidade.



ProductID	VendorID	AverageLeadTime	StandardPrice	LastReceiptCost
1	1	17	47.8700	50.2635
2	104	19	39.9200	41.9160
7	4	17	54.3100	57.0255
609	7	17	25.7700	27.0585
609	100	19	28.1700	29.5785

ProductVendor table

Fonte

imagem 1.2⁵²

51 Ibidem

52 TecNET, <https://i-technet.sec.s-msft.com/dynimg/IC30184.gif>, 02/09/2015

. Na imagem 1.2 temos os campos “productID” e “vendorID” são chaves primárias compostas, pois são dois campos.

Uma tabela pode conter apenas uma restrição PRIMARY KEY;

Uma chave primária não pode exceder 16 colunas e o comprimento de chave total,

de 900 bytes;

. O índice gerado por uma restrição PRIMARY KEY não pode fazer com que o número de índices da tabela exceda 999

índices NONCLUSTERED (não clusterizados) e 1 índice CLUSTERED (clusterizado);

. Se CLUSTERED ou NONCLUSTERED (a seguir saberás o que são CLUSTED e NONCLUSTERED, 1.4) não estiver especificado para uma restrição PRIMARY KEY, CLUSTERED será usado se não houver índices clusterizados na tabela;

. Todas as colunas definidas em uma restrição PRIMARY KEY devem ser definidas como NOT NULL (Não aceitar campos nulos, vazios. Claro não podemos cadastrar um cliente sem validar o CPF e RG, nem mesmo não se permitir cadastrar vazio). Se a nulidade não for especificada, todas as colunas participantes de uma restrição PRIMARY KEY deverão ter sua nulidade definida como NOT NULL;

2.2.1 - Índice CLUSTED e NOCLUSTERED

“Um índice é uma estrutura em disco associado a uma tabela ou exibição que agiliza a recuperação de linhas da tabela ou exibição. Um índice contém chaves construídas a partir de uma ou mais colunas na tabela ou exibição. Estas chaves são armazenados em

uma estrutura (árvore-B) que permite SQL Server para encontrar a linha ou linhas associadas com os valores de chave rápida e eficiente.

Pessoalmente e por conhecimento prática em Programação é impossível gerenciar um banco de dados sem um índice, uma série de benefícios entre eles a velocidade de pesquisa para validar a existência de um dado. Ou seja no nosso SIGEO - Sistema de Gestão Educacional Online, é preciso verificar se o Professor X já não está na Disciplina X, em outro horário, lecionando. Assim, evita-se cadastrá-lo em lecionar duas Disciplinas no mesmo horário.

A tabela ou exibição pode conter os seguintes tipos de índices:

2.2.1.1 - CLUSTED - Índice Agrupados

. Índices agrupados classificar e armazenar as linhas de dados na tabela ou exibição com base em seus valores fundamentais. Estas são as colunas incluídos na definição do índice. Só pode haver um índice agrupado por tabela, porque as próprias linhas de dados podem ser classificados em apenas um fim;

. A única vez que as linhas de dados em uma tabela são armazenadas na ordem de classificação é quando a tabela contém um índice agrupado. Quando uma tabela tem um índice agrupado, a tabela é chamada de tabela em cluster. Se uma tabela não tem nenhum índice em cluster, suas linhas de dados são armazenados em uma estrutura desordenada.

2.2.1.2 - NONCLUSTERED – Índices Não Agrupados

. Índices não agrupados tem uma estrutura separada das linhas de dados. Um índice sem CLUSTED contém os valores de chave índice agrupado e cada entrada de chave tem um valor para um ponteiro na linha de dados que contém o valor da chave;

. O ponteiro (ele varre a tabela de cima para baixo) de uma linha de índice em um índice agrupado para uma linha de dados é chamado de localizador de linha. A estrutura do localizador de linha depende do facto das páginas de dados são armazenados em uma pilha ou uma tabela agrupado;

Ambos índices clusterizados e não clusterizados podem ser únicos. Isso significa que não há duas linhas, podem ter o mesmo valor para a chave de índice. Caso contrário, o índice não é única e várias linhas podem compartilhar o mesmo valor de chave.

Os índices são mantidas, automaticamente, para uma tabela ou exibição, sempre que os dados da tabela é modificado.

2.3 - Índices e restrições

Os índices são criados, automaticamente, quando PRIMARY KEY e UNIQUE são definidos em colunas da tabela. Por exemplo, quando você cria uma tabela e identificar uma determinada coluna para ser a chave primária, a Engenharia de Banco de Dados (Database Engine) cria, automaticamente, uma restrição de chave primária e índice nessa coluna.

2.3.1 - Como índices são usados pelo otimizador de consulta

otimizador de consulta

Índices bem concebidos podem reduzir operações de disco I/O (input e output) e consumir menos recursos do sistema, portanto, melhorar o desempenho da consulta. Os índices podem ser úteis para uma variedade de consultas que contêm SELECT, UPDATE, DELETE ou MERGE declarações.

Ao executar uma varredura (ponteiro) da tabela, o otimizador de consulta lê todas as linhas na tabela, e extraí as linhas que atendem aos critérios da consulta. A varredura da tabela requer muitos do disco I/O (Input e Output, entrada e saída) operações e pode ser recurso intensivo.

Quando o otimizador de consulta (SELECT) usa um índice, ele procura as colunas de chave de índice, encontra o local de armazenamento das linhas necessárias pela consulta e extraí as linhas correspondentes a partir desse local. Geralmente, pesquisando o índice é muito mais rápido do que procurar a tabela porque ao contrário de uma tabela, um índice frequentemente

contém muito poucas colunas por linha e as linhas são na ordem de classificação.

O otimizador de consulta normalmente seleciona o método mais eficiente ao executar consultas (SELECT). No entanto, se há índices estão disponíveis, o otimizador de consulta deve usar uma varredura da tabela. Sua tarefa é projetar e criar índices que são mais adequados para o seu ambiente de modo que o otimizador de consulta tem uma seleção de índices eficientes a partir da escolha.

CAPÍTULO 4

1. O que estudar para o Exame 98-64?

Todos os conteúdos dos capítulos anteriores se contextualizam em volta do que o SQL Server realiza, faz e executa. Sendo, portanto, importante conhecer a experiência de quem já fez o Exame, sem perder a capacidade de sempre adquirir informações através de uma leitura sistemática. Não podendo deixar de lado, principalmente, assuntos que já conhecemos nos outros capítulos deste livro.

No entanto o precisamos nos munir de conhecimento para vencer

essa primeir etapa, pois, o que ouvimos de muitos o conhecimento básico sobre o SQL Server, e diz “Fulano possui os requisitos básicos sobre determinado assunto e pronto”⁵³.

Renato Siqueira, Administrador de Banco de Dados (DBA), diz “Portanto, nessa última situação, recomendo instalar o SQL Server na máquina (ou o seu RDBMS amigo de fé, colega, amante, tanto faz), instalar um banco de exemplo(Adventure Works sempre é muito recomendado caso vá instalar no SQL Server, o que eu recomendo) e ir desbravando criando consultas, errando bastante e acertando muito mais...”⁵⁴

53 Renato Siqueira, Impressões sobre o exame 98-364: MTA: Database Administration Fundamentals

<http://renatomsiqueira.com/impressoes-sobre-a-98-364-mta-database-administration-fundamentals/>, 03/10/2015

54 ibidem

O Ângelo Silva, diz -"O exame falou apenas sobre conceitos de banco de dados, tipos de formas normais, etc, porém no momento da prova me deparei com questões sobre sintaxe de queries T-SQL e objetos do SQL Server (stored procedures, views, databases)."⁵⁵

O Luiz Mercante, Microsoft Associate Certification, diz - “Para se ter uma idéia, quando a avaliação quer testar seus conhecimentos nos comandos SQL, as comparações são feitas entre ALTER TABLE, MODIFY TABLE, CHANGE TABLE, coisas do tipo. Porém, pra quem nunca usa queries em suas atividades diárias, existem questões seletivas questionando como se cria um índice composto, por exemplo. Outras questões avaliam a ordem dos parâmetros na criação de índices, selects e afins”.⁵⁶

1. 1 - Algumas questões que caem no Exame 98-64

- Qual a diferença entre um Clustered Index e um Non-Clustered Index?
- Qual a diferença entre uma Stored Procedure e uma Function?
- Diversas questões sobre comandos DML (UPDATE, INSERT, DELETE) geralmente com pinguinhas de sintaxe. Muita atenção nisso caso você não tenha o hábito de construir queries;

[55 Ângelo Silva, Considerações sobre o exame 98-364 MTA Database](#)

Fundamentals, <https://angmaximo.wordpress.com/2013/01/27/mta-database-fundamentals/>, 03/10/2015

[56 Luiz Mercante, MTA – Database Fundamentals, http://sqldicas.com.br/dicas/mta-database-fundamentals/](http://sqldicas.com.br/dicas/mta-database-fundamentals/), 03/10/2015

- Duas questões sobre normalização (Uma pergunta sobre a vantagem da Desnormalização e outra sobre a característica da 2FN);
- Diferença entre DROP e DELETE (Que ainda pega muita gente, acredito);
- Uma questão sobre GROUP BY (Pra evitar aquele erro clássico de esquecer de agrupar quem tá por fora das funções de agregação);
- Algumas questões com comandos inseridos fora de ordem. Tipo um SELECT seguido de ORDER BY e depois WHERE. Isso vai contra a lógica de execução de consulta e eles embutiram essa pegadinha em várias questões;
- Comandos DML, DCL e DDL (Dado um comando, definir de qual categoria ele é. Por exemplo, o CREATE é um DDL).
- Uso do operador LIKE (especificamente a famosa sintaxe da

pegadinha do %malandro%. É, os percents, que chamamos no SQL de Wildcard);”⁵⁷ 57 Ibidem

NOTA

LIKE:

A consulta retornam resultados contendo um determinado padrão. Uma consulta que localiza todos os clientes em que todos nome inicie com “Jo”. Retornará os registros de “João Silva”, “João Santos”, “José Matos” e assim por diante.

A instrução SQL é criada assim:

```
SELECT codigo, nome, cidade FROM clientes WHERE nome LIKE 'Jo%';
```

Mas digamos que queremos uma instrução LIKE que termine com a letra “o”

Exemplo: SELECT * FROM clientes WHERE nome LIKE '%oo'; Dica:

O símbolo “%” é usado para definir (letras desaparecidas) antes

e depois do padrão. Você vai aprender mais sobre wildcards no próximo capítulo.

1.2 - Perguntas e Respostas do Exame 98-364

1.2.1 Parte I⁵⁸

1 - João trabalha como um administrador de base para Guia DBA.

A gestão instruiu-o a planejar um banco de dados onde a identificação dos objetos-chave ou entidades para a gestão de banco de dados é muito importante.

Qual das seguintes etapas de planejamento do banco de dados irá ajudá-lo a realizar essa tarefa?

- A. Identificação de objeto.
- B. Juntando informações.
- C. Identificar as relações entre os objetos. D. Modelagem de objetos.

Resposta: A

No processo de identificação de objetos no planejamento de banco

de dados, é necessário identificar os objetos-chave ou entidades para a gestão de banco de dados.

Em geral, há poucos objectos primários , e uma vez que estes são identificados , os itens relacionados tornam-se visíveis .

58 Alexandre Curvelo do Site Guia DBA, <http://guiadba.com.br/preparando-se-para-o-exame-de-certificacao98-364-mta-parte-1/>,
07/10/2015

2- Examine a seguinte tabela:

P_ID	First_Name	Last_Name	Score
001	Sharon	Hayes	25
002	James	Welch	23
003	Rick	Stuart	24
004	Tim	May	25
005	Mark	Smith	24

Qual é o grau dessa tabela?

- A. 20
- B. 25
- C. 5

D. 4

Resposta: D

O grau de uma tabela refere-se ao número de colunas que ela tem. o número de colunas contidas numa tabela é referido pelo termo grau da tabela.

3- Você trabalha como um administrador de banco de dados para Guia DBA .

A empresa tem um banco de dados e eles querem que você crie um projeto para um banco de dados que irá apoiar as operações e os objetivos da empresa. O que você vai usar para realizar a tarefa?

- A. Design do aplicativo.
- B. Design de banco de dados.
- C. Planejamento de banco de dados. D. Captura de requerimentos e análise.

Resposta: B

Design de banco de dados é o processo de criação de um projeto para um banco de dados que vai apoiar as operações e os objetivos da empresa para o sistema de banco de dados necessário .

Ela representa dados e relações entre os dados exigidos por todas as principais áreas de aplicação e grupos de usuários.

4- Qual dos seguintes são tipos de estratégias de prototipagem? Cada resposta correta representa uma solução completa. Escolha dois.

- A. Requerimentos de prototipagem. B. Prototipagem física.**
- C. Prototipagem revolucionária. D. Prototipagem evolucionária.**

Resposta: A e D

NOTA:

Existem dois tipos de estratégias de prototipagem:

Requerimentos prototipagem - Ela é usada para determinar as exigências de um sistema de base de dados e , em seguida, proposto

o protótipo é descartado. Prototipagem evolutiva - É utilizado para o mesmo objectivo , mas sem o descarte do protótipo e com o desenvolvimento se torna o sistema de base de dados de trabalho.

5- Qual das seguintes é uma atividade de gestão que permite as etapas do aplicativo de banco de dados a ser realizado da forma mais eficiente e eficaz possível?

- A. Planejamento de banco de dados. B. Informações de identificação de objetos. C. Identificação de objeto.**
- D. Requerimentos recolha e análise.**

Resposta: A

Planejamento de banco de dados é uma atividade de gestão que permite as etapas do aplicativo de banco de dados a ser realizado da forma mais eficiente e eficaz possível.

6- Uma empresa chamada Guia DBA tem muitos revendedores autorizados em todo o país que vendem os seus produtos. O gerente de vendas da empresa quer ver os detalhes dos revendedores autorizados, incluindo o nome, a região, e as vendas totais em ordem crescente de vendas.

Qual das seguintes consultas devem ser emitidos para obter o resultado desejado?

- A. Select max(TotalVendas) From Revendedor
- B. Select From Revendedor
- C. Select * From Revendedor Where max(TotalVendas)
- D. Select Nome, Regiao, TotalVendas From Revendedor Order by TotalVendas

Resposta: D

As opções B e C estão com sintaxe incorreta. A opção A só retorna o maior valor total de vendas.

7- Qual das seguintes são os comandos DML? Cada resposta correta representa uma solução completa.

Escolha todas que se aplicam.

- A. Select Into
- B. Delete
- C. Update
- D. Alter
- E. Select
- F. Insert

Resposta: A, B, C e F

Alter é um comando DDL.

O SELECT puramente é classificado como declaração SQL-DATA e, portanto, é considerado fora do DML.

`SELECT INTO` é considerado DML porque manipula ou modifica dados.

8- Qual é o tempo ideal para fazer backup do arquivo de log dinâmico?

- A. Quando o servidor está em execução.
- B. Você pode fazer backup de arquivos de log dinâmicos qualquer momento.
- C. Quando o servidor está parado.
- D. Quando o servidor é desligado.

Resposta: C

Arquivos dinâmicos, como os de logs são alterados quando o servidor está em execução, logo, o ideal é fazer backup quando o servidor está parado.

9- Qual dos seguintes comandos podem ser usados para alterar o tipo de

9- Qual dos seguintes comandos podem ser usados para alterar

o tipo de autorização no DB2 9? Cada resposta correta representa uma solução completa. Escolha todas que se aplicam.

- A. `UPDATE DATABASE CONFIGURATION`
- B. `SET AUTHORISATION`
- C. `RESET DATABASE CONFIGURATION`
- D. `UPDATE DBM CFG`

Resposta: A e D

Os comandos `UPDATE DBM CFG` e `UPDATE DATABASE CONFIGURATION` podem ser usados para alterar o tipo de autorização no DB2.

Eles são comandos equivalentes e usados para colocar as mudanças desejadas no arquivo de configuração de banco de dados DB2 9.

10- Qual das seguintes alternativas pode ser usada para preencher uma tabela? Cada

resposta correta representa uma solução completa. Escolha todas que se aplicam.

- A. Data Pump
- B. SQL*Loader
- C. Insert statement
- D. MERGE statement

Resposta: A, B, C e D MERGE é um comando DML usado para atualizar ou inserir linhas condicionalmente em uma tabela.

Insert é um comando DML usado para inserir dados em uma tabela. SQL * Loader (sqldr) é usado para cargas de dados de alto desempenho.

Data Pump(Oracle) é usado para mover dados e metadados entre bancos de dados e/ou a partir de arquivos do sistema operacional de forma muito eficiente.”

1.2.2 - Parte II⁵⁹ 1- Você está projetando um banco de dados para o Guia DBA. Um

jogador pode ser inscrito para vários jogos. Você está trabalhando em uma relação mostrada na imagem abaixo:

Game_ID	P_ID	P_Age	P_DOB
001	3011	25	2-2-1991
001	3012	23	3-4-1993
003	5078	24	3-3-1992
004	3011	25	2-2-1991
005	7023	24	12-8-1992

Qual das seguintes chaves candidatas serviria melhor como a chave primária?

- A. P_ID e P_DOB

59 Alexandre Curvelo do Site Guia DBA, <http://guiadba.com.br/preparando-se-para-o-exame-de-certificacao98-364-mta-parte-1/>, 07/10/2015

- B. Game_ID e P_ID
- C. Game_ID e P_Age
- D. P_ID e P_Age

Resposta: B

A chave primária deve ser exclusiva para cada linha. A única

combinação das chaves candidatas que o torna único para cada linha da tabela é Game_ID e P_ID.

2- Você trabalha como um desenvolvedor de banco de dados para o Guia DBA. A empresa tem um banco de dados chamado DB_GUIADBA que está hospedado em um Servidor SQL Server 2008. O banco de dados inclui uma tabela chamada Funcionario, que contém os detalhes dos funcionários. Você quer garantir que a qualquer momento quaisquer dados na tabela Funcionario que forem modificados com uma instrução UPDATE, sejam registrados em uma tabela chamada AlteraFuncionario . A tabela

AlteraFuncionario inclui colunas para gravar o que foi modificado, quem fez a modificação e quando eles fizeram a mudança. O que você deve criar para atender a essa necessidade?

A. Um gatilho DDL que consulta a tabela atualizada. B. Um gatilho DML que consulta a tabela inserida. C. Um gatilho DDL que consulta a tabela inserida. D. Um gatilho DML que consulta a tabela atualizada. Resposta:B

Um gatilho DML pode ser usado para capturar instruções DML (UPDATE, INSERT, DELETE). Gatilhos DML pode usar as tabelas inseridas e excluídas.

3- Qual das seguintes é uma linguagem aprovada pela ANSI para a comunicação e gestão de dados em um

sistema de gerenciamento de banco de dados relacional (RDBMS)?

- A. SQL
- B. QL
- C. DDL
- D. XML

Resposta:A

SQL (Structured Query Language) é usado para se comunicar e gerenciar dados em um sistema de gerenciamento de banco de dados relacional (RDBMS).

4- Qual dos seguintes fornece unidades confiáveis de trabalho que permitem recuperação correta de falhas e mantém um banco de dados consistente, mesmo em caso de falha do sistema?

- A. Segurança de banco de dados.
- B. Controle de concorrência.
- C.

Duas fases.

D. Transação de banco de dados.

Resposta: D

Uma transação de banco de dados compreende uma unidade de trabalho realizado dentro de um sistema de gerenciamento de banco de dados contra um banco de dados, e tratada de uma forma coerente e confiável independente de outras transações.

5- Alexandre trabalha como um administrador de banco de dados para GUIA DBA. A empresa tem um banco de dados SQL Server. A Gestão instrui-o a garantir que nenhum dado inconsistente

é inserido no banco de dados. Qual das seguintes irão ajudá-lo a realizar a tarefa?

- A. Integridade referencial.
- B. Abstração.
- C. Criptografia.
- D. Autenticação.
- E. Auditoria nativa.

Resposta: A

A integridade referencial é uma característica ou propriedade que impede que os usuários ou aplicativos de entrar em dados inconsistentes que, quando satisfeitas, exige que cada valor de uma coluna de uma tabela de existir como um valor de outra coluna em uma diferente ou a mesma tabela.

6- Qual dos seguintes comandos não pode ser revertida? Cada resposta correta representa uma solução completa. Escolha dois.

- A. DELETE
- B. TRUNCATE
- C. UPDATE
- D. COMMIT

Resposta: B e D

A instrução TRUNCATE é uma instrução DDL que é utilizada para remover todos os registros de uma tabela. Também libera a operação permanente.

A declaração COMMIT é a declaração de controle de transação que termina a transação do banco de dados atual e faz com que todas as alterações feitas nesta transação sejam permanentes.

7- Qual das seguintes afirmações é verdadeira em relação a uma chave composta em uma tabela?

- A. É uma chave primária.
- B. É uma chave primária ou estrangeira, que consiste em duas ou

mais atributos da tabela . C. É uma combinação de chave e a chave estrangeira primária . D. É uma chave estrangeira.

Resposta:B

Uma chave é um composto de chave primária ou estrangeira, que consiste em dois ou mais atributos da tabela. É uma chave que consiste em dois ou mais atributos que a identidade de, unicamente, uma entidade de ocorrência.

8- Você trabalha como um administrador de banco de dados para GUIA DBA. Uma tabela denominada Estudante, informacao existe na base de dados. A tabela tem colunas nomeadas nome_estudante, marcas_estudante e cursos_estudante. Alguns estudantes que optaram por vários cursos e têm diferentes marcas neles.

Isto implica que a tabela tem dois relacionamentos muitos para muitos. Você tem muitos para muitos. Você tem que garantir que estes dois relacionamentos não são representados em uma

que garantir que estes dois relacionamentos não são representados em uma

única tabela e que a coluna nome_estudante existe em ambas as tabelas. Qualambas as tabelas. Qual das seguintes formas normais que você vai usar para representar essas relações nas tabelas?

A. 4 NF B. 3 NF C. 2 NF D. 1 NF Resposta: A

Quarta forma normal (4 NF) é uma forma normal usada na normalização do banco de dados.

9- Qual dos seguintes termos de banco de dados está descrito na declaração abaixo ?

"Ela impede a operação de banco de dados atual de ler ou escrever um dado, enquanto o item de dados é acessado por outra operação."

- A. Bloqueio. B. Criptografia. C. Limitação. D. impasse.

Resposta: A

Bloqueio é o processo para impedir que as operações de banco de dados atuais de leitura e escrita de dados, aconteçam enquanto que um item de dados está sendo acessado por outra operação. Um bloqueio é utilizado quando vários usuários precisam acessar um banco de dados simultaneamente.

10- Qual das seguintes etapas no

planejamento de banco de dados ajuda a determinar as exigências do banco de dados através de entrevistas?

- A. Juntando informações.
- B. Identificar a relação entre os objetos.
- C. Identificar os objetos.
- D. Identificar o tipo de informação para cada objetos. E. A modelagem do objeto.

Resposta: A

Coleta de informações é o primeiro passo do planejamento de banco de dados e exige uma boa compreensão de quais são as tarefas a serem realizadas no banco de dados.

1.2.3 - Parte III⁶⁰

Estive pesquisando outras fontes de informações a fim de melhor atualização ao leitora. Então, a fonte fidedigna foi, também,

Microsoft Virtual Academy, e lá no site existem diversos cursos on-line, porém, com os Instrutores falando em inglês, todo o material em idioma inglês. Porém, já andado e habituado com esse ambiente de estudo, garimpei essas fontes de perguntas e respostas, onde são oficiais e certeiras que caem perguntas no Exame iguais:

Inclusive, após esse curso on-line caso alcance notas igual ou maior a 4 pontos em cada 5 perguntas você já está se auto-capacitando para a realização da Prova do Exame 98-364.

Porém, essas perguntas exigiram um bom tempo de estudo para acertá-las, devido ser escritas em inglês, porém, no caso a seguir a Microsoft Virtual Academy disponibilizou a 60 https://www.microsoftvirtualacademy.com/pt-pt/training-courses/conceitos-b-sicos-de-bancos-de-dados8243?l=vfVio0yFB_5904984382, 14/10/2015

tradução, facilitando, um pouco as respostas. Porém, com respostas que podem nos pegar na equivalência.

Conceitos Básico de Banco de Dados SQL Server:

“1.

“1. What is the purpose of the USE statement in SQL Server Management Studio?

What is the purpose of the USE statement in SQL Server Management Studio?

Qual é a finalidade da instrução USE no SQL Server Management Studio?

A. You can specify which table to use for queries. Você pode especificar que tabela usar para consultas.

B. You can specify which user account to use to log in to SQL Server. Você pode especificar que conta de usuário será usada para fazer logon no SQL Server.

C. You can select which specific database to use for queries. Você pode selecionar que banco de dados específico usar para consultas.

D. You can choose which SQL Server instance to connect to. Você pode escolher que instância do SQL Server usar para conexão.

2. You create a SQL Server instance. Which of the following databases are created automatically? (Choose all that apply.)

Você cria uma instância do SQL. Quais dos seguintes bancos de dados são criados automaticamente?

(Selecione todas as alternativas aplicáveis.) A. master B. tempdb C. model D. msdb

3. Which of the following statements about composite keys is true?
Qual das alternativas a seguir sobre chaves compostas é verdadeira?

A. A composite key is a constraint that you create by using more than one field from a single table.

Chave composta é uma restrição que você cria usando mais de um campo de uma tabela única.

B. A composite key is a constraint that you create by using more than one field from multiple tables.

Chave composta é uma restrição que você cria usando mais de um campo de várias tabelas.

C. A composite key is the combination of a primary key and a foreign key. Chave composta é a combinação de uma chave primária e uma chave estrangeira.

D. A composite key is the combination of a primary key and a unique constraint. Chave composta é a combinação de uma chave primária e uma restrição exclusiva.

4. A composite primary key is created by using one column from one table and another column from a second table.

Uma chave primária composta é criada usando uma coluna de uma tabela e outra coluna de uma segunda tabela.

A. True B. False

5. Which of the following normalization forms does a database conform to when you eliminate repeating columns of data in tables?

Com qual dos seguintes formulários de normalização um banco de dados está em conformidade quando você elimina colunas repetidas de dados em tabelas?

A. Fourth normal form

Quarto formulário de normalização

B. Third normal form

Terceiro formulário de normalização

C. First normal form

Primeiro formulário de normalização

D. Second normal form

Segundo formulário de normalização

6. A foreign key cannot reference a column in the table where it is defined. Uma chave estrangeira não pode referenciar uma coluna na tabela em que foi definida. A. True

B. False

7. Which character acts as a wildcard when used in the LIKE predicate? Qual personagem atua como um curinga quando utilizado no predicado LIKE? A. *

B. %
C. \$
D. #

8. Which character in a LIKE statement matches exactly one alphanumeric character? Qual personagem em uma declaração como corresponde exatamente um carácter alfanumérico?

A. _ B. & C. * D. +

9. Which style of language is SQL? Qual o estilo da linguagem SQL é?

A. processual
B. declarativa
C. orientado a objeto

D. funcional

10. You write a Transact-SQL query that returns the Cellphone column from the Sales.Customer table. Cellphone is a varchar column that permits NULL values. For rows where the Cellphone value is NULL, your query should return the text 'None'. Select the correct function to complete the following query:

Você escreve uma consulta Transact-SQL que retorna a coluna Celular da tabela Sales.Customer. Celular é uma coluna varchar que permite valores nulos. Para as linhas onde o valor do telemóvel é NULL, a consulta deve retornar o texto 'Nenhum'. Selecione a função correta para completar a seguinte consulta:

```
SELECT FirstName, LastName, _____(Cellphone, 'None') AS  
Cellphone FROM Sales.Customer;
```

A. COALESCE B. NULLIF
C. ISNULL

11. The term Principal is used to describe the individuals and groups that are allowed to access SQL Server.

O termo Entidade de Segurança é usado para descrever as pessoas e os grupos que estão autorizados a acessar o SQL Server.

- A. True
- B. False

12. What data is backed up when you create a differential backup?
Quais dados são salvos quando você cria um backup diferencial?

- A. Only the data in the transaction log that has changed since the last backup Somente os dados que foram alterados no log de transações desde o último backup
- B. Only the data in the database that has changed since the last backup Somente os dados que foram alterados no banco de dados desde o último backup
- C. All of the data in the transaction log Todos os dados no log de transações
- D. All of the data in the database Todos os dados no banco de dados

13. You use SQL Server Management Studio to create a backup.
What file extension will the backup file use?

Você usa o SQL Server Management Studio para criar um backup.
Que extensão de arquivo o arquivo de backup usará?

- A. log
- B. mdf
- C. bak
- D. ldf

14. SQL Server requires that users have a Windows user account to access databases. O SQL Server exige que os usuários tenham uma conta de usuário do Windows para acessar bancos de dados.

- A. False
- B. True

15. The guest user account can be removed from individual databases. A conta de usuário convidado pode ser removida de bancos de dados individuais.

- A. True
- B. False

16. What data is backed up when you create an incremental backup?
Quais dados são salvos quando você cria um backup incremental?

- A. All of the data in the database Todos os dados no banco de dados
- B. Only the data in the transaction log that has changed since

the last backup Somente os dados que foram alterados no log de transações desde o último backup

C. All of the data in the transaction log Todos os dados no log de transações

D. Only the data in the database that has changed since the last backup Somente os dados que foram alterados no banco de dados desde o último backup

17. You need to back up all data in a specific database and enough of the transaction log to recover all of the data. Which type of backup should you create? Você pode fazer backup de todos os dados em um banco de dados específico e parte suficiente do log de transações para recuperar todos os dados. Que tipo de backup você deve criar?

A. Full

Completo

B. Incremental Incremental

C. Transaction Log Log de transações D. Differential Diferencial

18. What keyboard shortcut can you use to run queries in SQL

Server Management Studio?

Que atalho do teclado você pode usar para executar consultas no SQL Server Management Studio?

A. F12

B. F5

C. F10

D. F1

19. What is the purpose of the USE statement in SQL Server Management Studio? Qual é a finalidade da instrução USE no SQL Server Management Studio?

A. You can specify which user account to use to log in to SQL Server. Você pode especificar que conta de usuário será usada para fazer logon no SQL Server.

B. You can choose which SQL Server instance to connect to. Você pode escolher que instância do SQL Server usar para conexão.

C. You can select which specific database to use for queries. Você pode selecionar que banco de dados específico usar para consultas.

D. You can specify which table to use for queries. Você pode especificar que tabela usar para consultas.

20. Each database may contain only one table.

Cada banco de dados pode conter somente uma tabela. A. True

B. False

21. Which of the following statements describes a database schema? Qual das alternativas a seguir descreve um esquema de banco de dados?

A - A. The schema defines the structure of tables in a database. O esquema define a estrutura de tabelas em um banco de dados.

B. The schema is a list of all of the users that have permissions to access a database. O esquema é uma lista de todos os usuários que têm permissões para acessar um banco de dados.

C. The schema defines where the data resides and who has access to the data. O esquema define onde os dados residem e quem tem acesso aos dados.

D. The schema is a list of all of the tables in a database. O esquema é uma lista de todas as tabelas em um banco de dados.

Resposta:

1. c

2.a,b,c,d

3.c

4.b

5.3

6.b

7.b

8.a

9.c

10.c

11.b

12.b

13.c

14.a

15.b

16.b

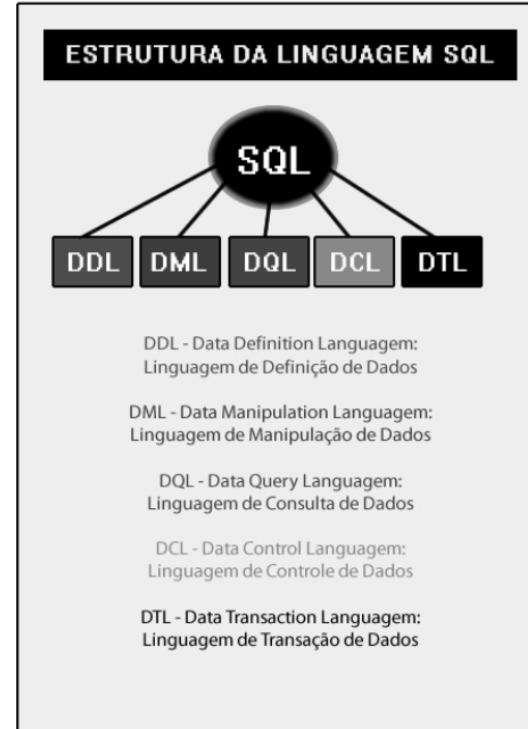
17.a

18.b

19.c

20.b
21.a

CAPÍTULO 5



Fonte da

imagem⁶¹

" O SQLSQL foi desenvolvido originalmente no início dos anos 70 nos laboratórios da IBM em San Jose, dentro do projeto System R, que tinha por objetivo demonstrar a viabilidade da implementação do modelo relacional proposto por E. F. Codd. O nome original da linguagem era SEQUEL, acrônimo para "Structured English Query Language" (Linguagem de Consulta Estruturada, em Inglês)[1] , vindo daí o facto de, até hoje, a sigla, em inglês, ser comumente pronunciada "síquel" ao invés de "és-kiú-él", letra a letra. No entanto, em português, a pronúncia mais corrente é a letra a letra: "éss-e-quê-éle".

A linguagem é um grande padrão de banco de dados. Isto decorre da sua simplicidade e facilidade de uso. Ela se diferencia de outras linguagens de consulta a banco de dados no sentido em que uma consulta SQL especifica a forma do resultado e não o caminho para chegar a ele. Ela é uma linguagem declarativa em oposição a outras linguagens procedurais. Isto reduz o ciclo de aprendizado daqueles que se iniciam na linguagem.

Embora o SQL tenha sido originalmente criado pela IBM,

rapidamente surgiram vários "dialectos" necessidade de realizada pela American National Standards Institute (ANSI) em 1986 e ISO em 1987."⁶²

desenvolvidos por outros produtores. Essa expansão levou à ser criado e adaptado um padrão para a linguagem. Esta tarefa foi **61** www.focosite.com.br, **11\10\2015**
62 Wikipedia , <https://pt.wikipedia.org/wiki/SQL>, **10\11\2015**,

1 - DML SQL⁶³



Linguagem de Manipulação de Dados: INSERT, UPDATE, DELETE, SELECT.

"

NOTA:

. A SQL (Structured query languageStructured query language)) é uma linguagem de banco de dados relacional usada para consultar, atualizar, e gerenciar bancos de dados relacionais e é o padrão de fato para produtos de bancos de dados;

.

A DML (Linguagem de manipulação de dadosLinguagem de manipulação de dados) é usada para inserir, atualizar, e Linguagem de manipulação de dados

63

excluir dados, além de consultar um banco de dados;

. A DDL (Linguagem de definição de dadosLinguagem de definição de dados) é usada para criar, alterar, ou remover bancos de dados relacionais, entidades, atributos, e outros objetos (por exemplo, exibições);

. A TCLTCL — Transaction Control LanguageTransaction Control Language – (Controle de Transações) são usados para gerenciar

as mudanças feitas por instruções DML . Ele permite que as declarações a serem agrupadas em transações lógicas;

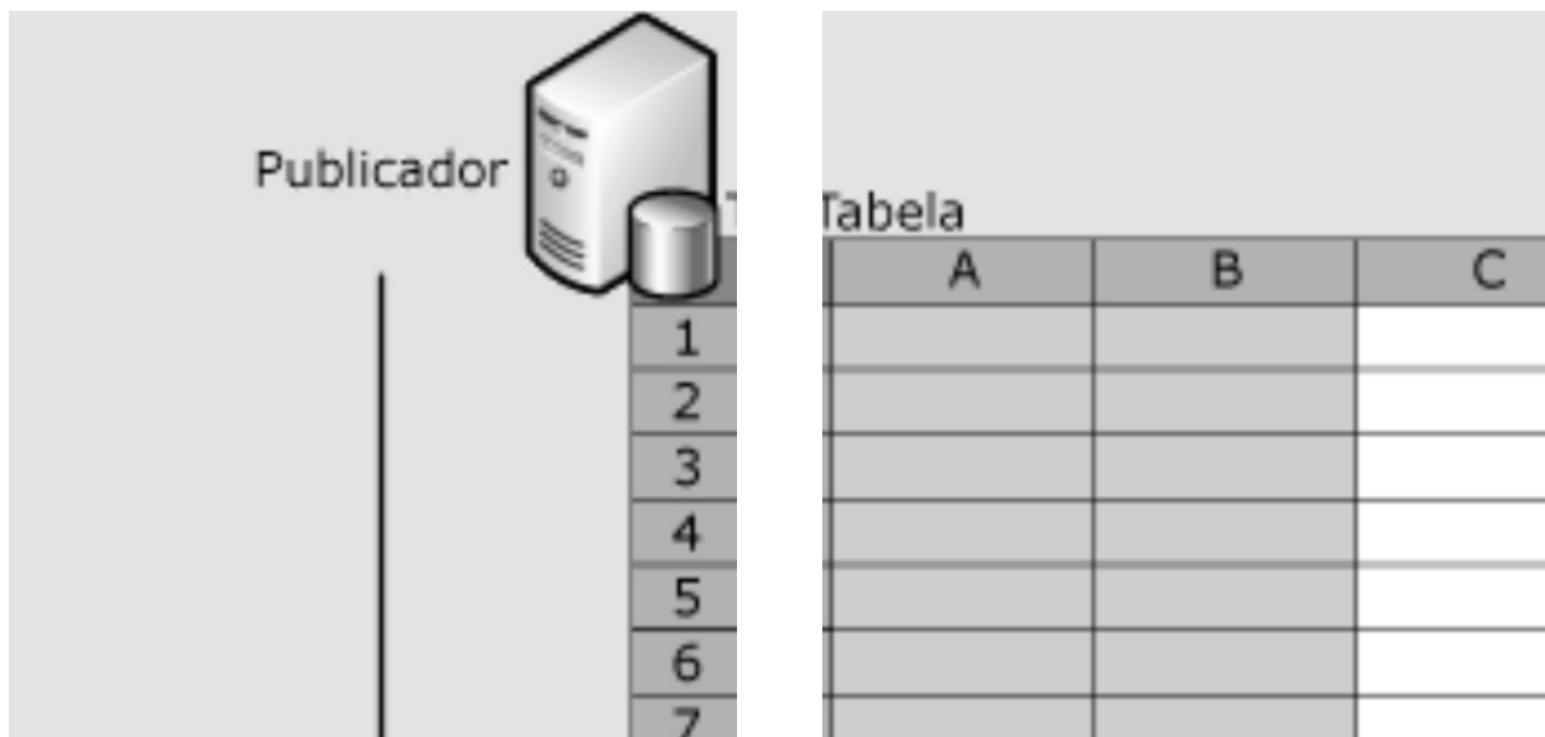
. A DCLDCL — Data Control Language

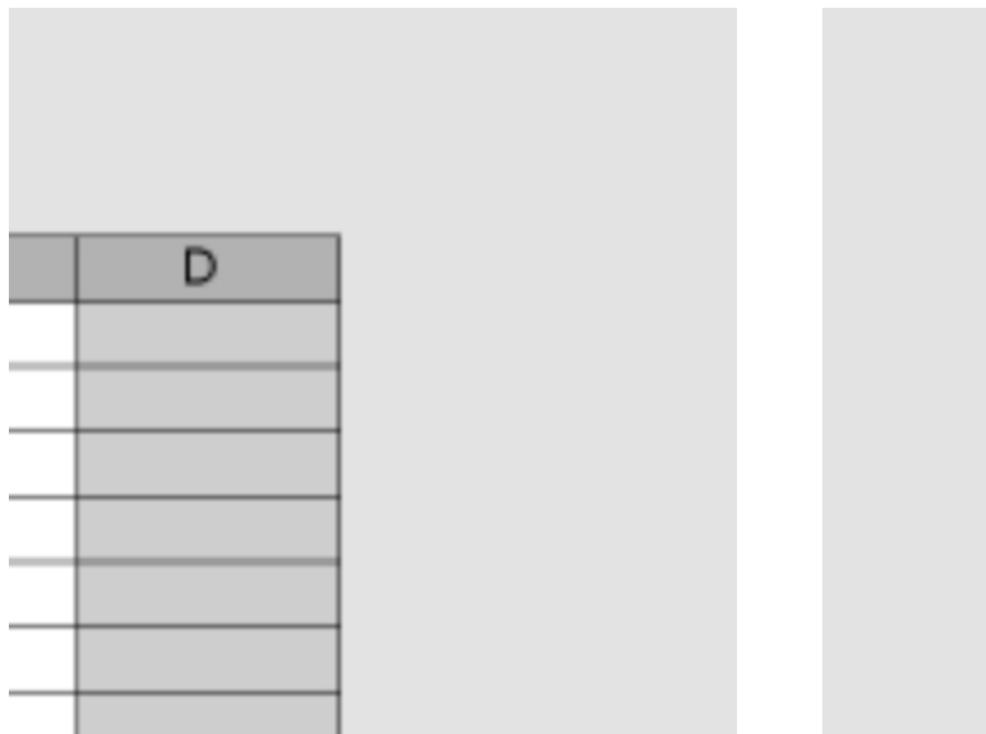
.

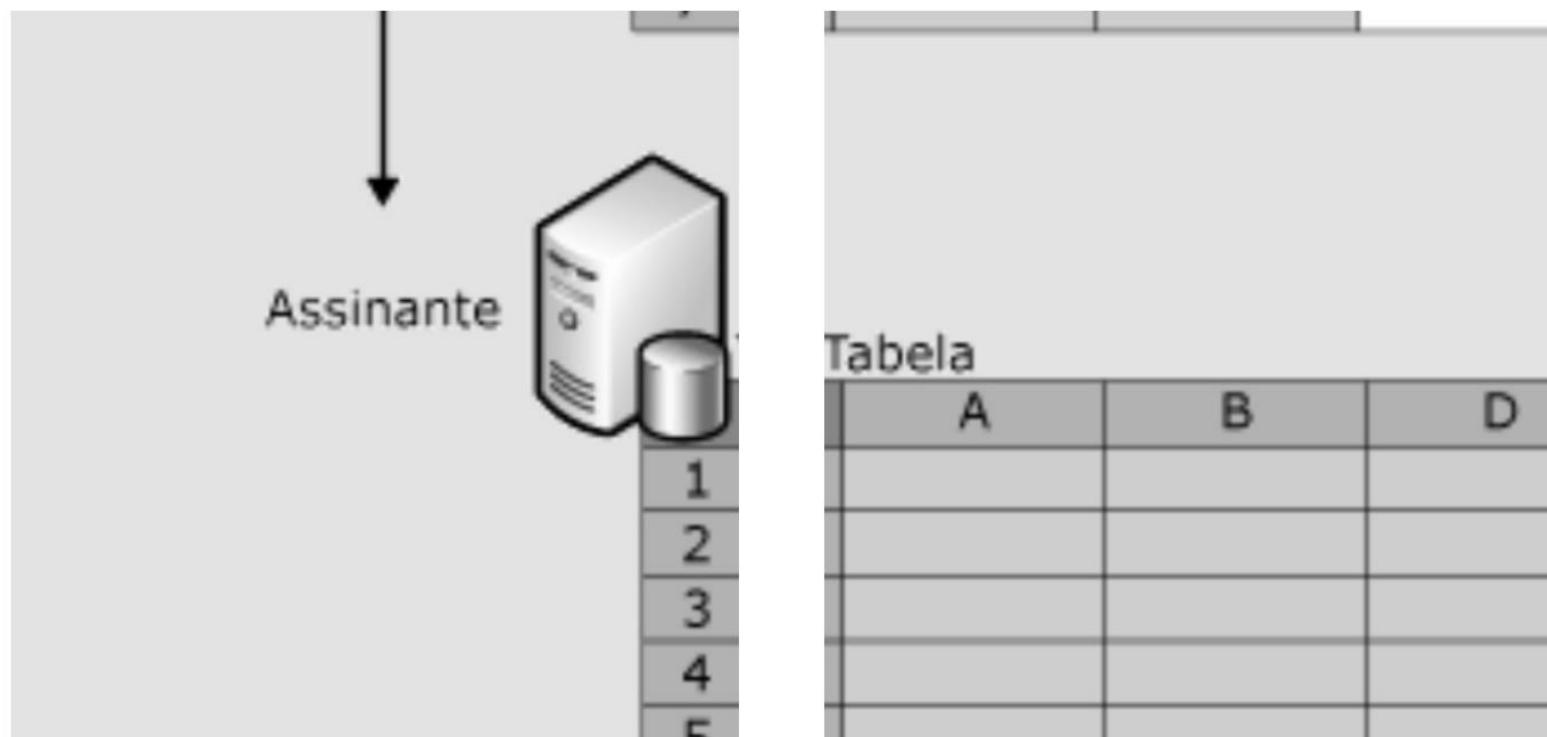
. 364: quando o uso da condição WHERE, ex.: SELECT automovel _ model WHERE automovel _ color = 'vermelho'

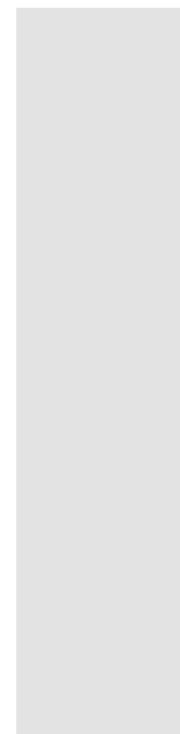
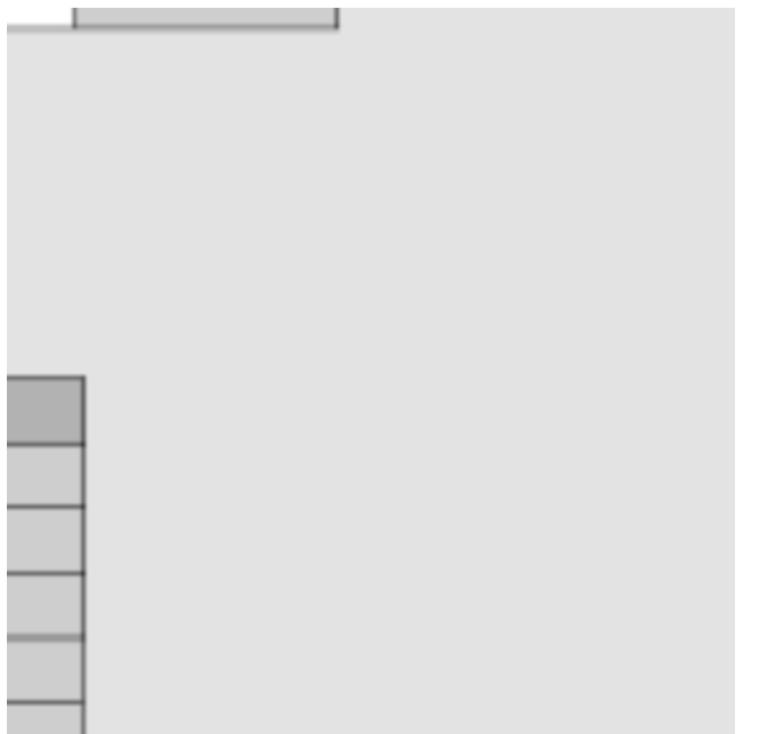
Filtro de Coluna:

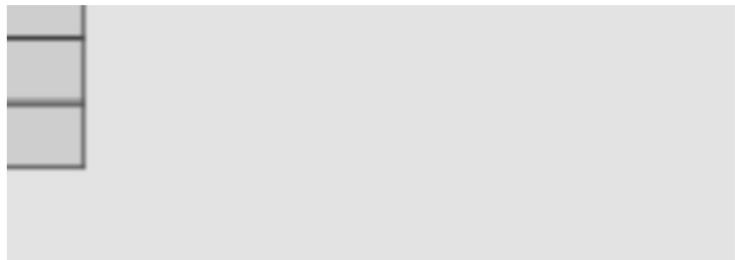
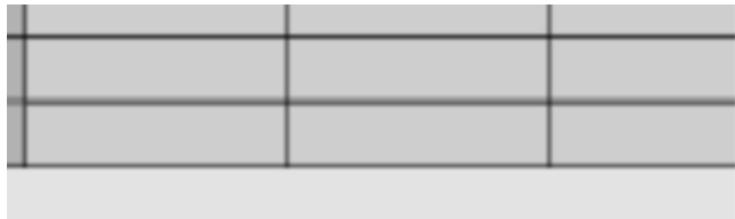
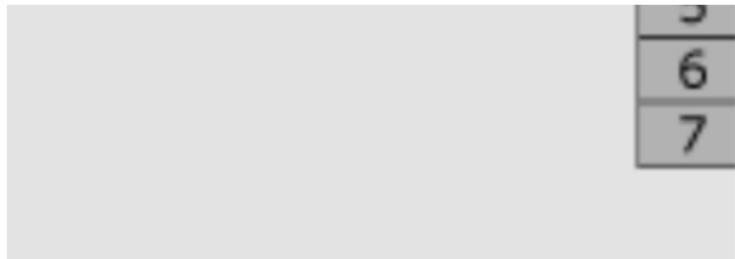
A imagem 1.3⁶⁴ - Uma publicação que filtra a coluna "C": **64 MSDN, Filtrando dados, [https://msdn.microsoft.com/pt-br/library/ms151775\(v=sql.100\).aspx](https://msdn.microsoft.com/pt-br/library/ms151775(v=sql.100).aspx),** 03/10/2015



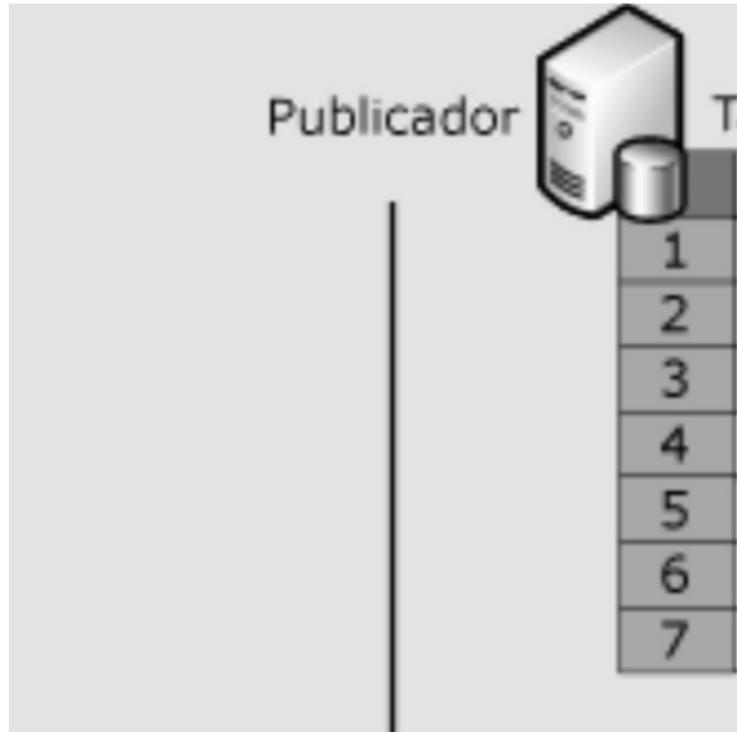








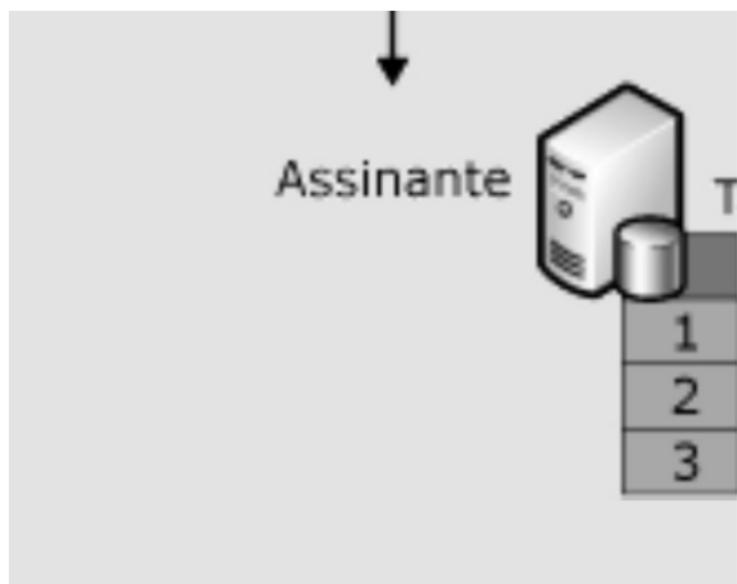
⁶⁵ MSDN, **Filtrando Dados**, <https://msdn.microsoft.com/pt-br/>



Tabela

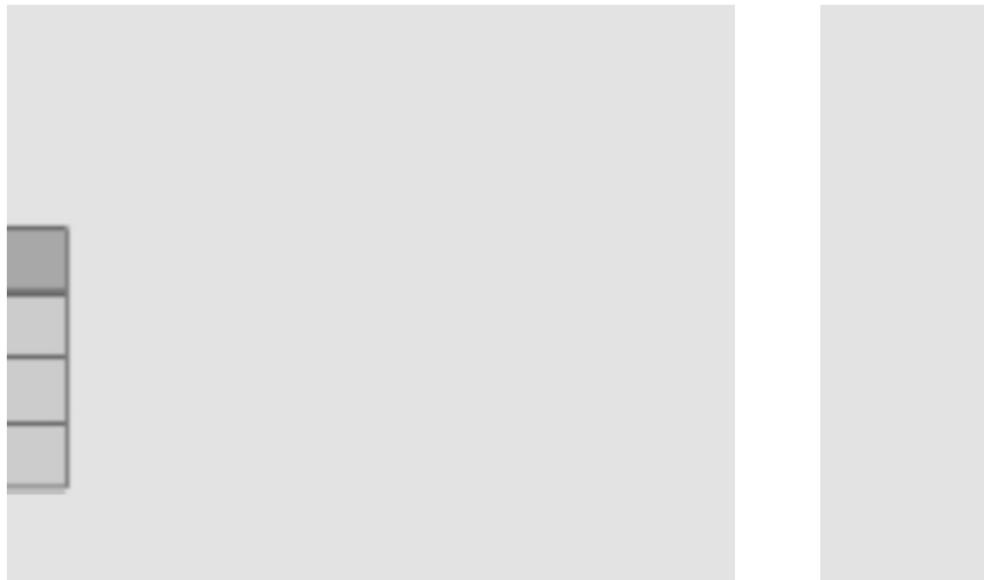
A	B	C

D	



abera

A	B	C



CENÁRIO: O proprietário conhece a configuração do banco de dados dos automóveis na loja de auto-peças. Agora é preciso saber quais operações ele fará com o banco de dados.

NOTA

T-SQL (Transact-SQLTransact-SQL) é uma linguagem de consulta sofisticada com recursos adicionais além do que é definido no ANSI SQL.

- Gerar vários relatórios de estoque
- Produzir relatórios de vendas (por modelo de automóvel, preço, e assim por diante)
- automóvel
 - Inserir novo estoque ao sistema quando este chegar
 - Alterar o custo das automóvel e peças conforme necessário
 - Remover automóvel do banco de dados quando eles forem vendidos
- automóvel
 - 1. Que comando DML (Linguagem de manipulação de dados) deve ser usado para indicar qual peça que foi vendida e será removida da tabela de automóvel? a. DELETE FROM automóvelautomóvel WHERE peca_id = p1234
b. REMOVE FROM automóvelautomóvel WHERE peca_id = p1234

c. ERASE FROM automóvelautomóvel WHERE automóvelautomóvel_id = T1234

- 2. Que comando informa a quantidade disponível no estoque de peças verdes? a. SELECT pecas _ model WHERE automóvelautomóvel_color = 'verde'
b. SELECT * FROM pecas WHERE automóvelautomóvel _ color = 'verde'
c. FIND * FROM pecas WHERE automóvelautomóvelautomóvel _ color = 'verde'
- 3. Uma nova peça é inserida à tabela de automóvel no banco de dados. Qual comando? a. INSERT INTO automóvelautomóvelautomóvel VALUES (p342, 'Tal', 'verde', 99977.00)
b. ADD INTO automóvelautomóvel VALUES (p3425, 'Rec', 'verde', 99977.00)
c. INSERT INTO automóvelautomóvel VALUES (p3421, 'Deer', 'verde', 89999.00)

Respostas

1. O comando DML (Linguagem de manipulação de dados) que deve ser usado para indicar que uma bicicleta foi vendida e deve ser removida da tabela de automóveis é:

a.

a. **DELETE FROM** automóvel WHERE automóvel_number = T1234

DELETE FROM automóvel_number = T1234

2. O comando usado para informar sobre a quantidade disponível atual de automóveis verdes é:

b. **SELECT * FROM** automóvel WHERE automóvel_color = 'verde'

3. Uma novo automóvel é inserido à tabela de automóvel no banco de dados com: c. **INSERT INTO** automóvel VALUES (p3425, 'Ro', 'verde', 9999.00)
VALUES (p3425, 'Ro', 'verde', 9999.00)"

"NOTA

Qual é a principal diferença entre o comando DML **DELETE** e o comando DDL **DROP**?

e o comando DDL **DROP**?

- a. Eles realizam a mesma tarefa; portanto, não há diferença
- b. **DELETE** somente remove todos os (ou um subconjunto de) registros da tabela; ele não remove a tabela em si
- c. **DROP** somente remove todos os registros da tabela; ele não remove a tabela

A principal diferença entre o comando DML **DELETE** e o comando DDL **DROP** é: b. **DELETE** somente remove todos os (ou um subconjunto de) registros da tabela; ele não remove a tabela em si.⁶⁶

66 Apostila Certiport, Fundamentos de administração de banco de dados MTA exame 98-364, ftp://ftp.certiport.com/Marketing/Mta/Localizations/PTB/MTA_SSG_DbAdmin_individual_PTB.pdf, 05/10/2015

2 - DDL SQL⁶⁷



DELETE, REMOVE, ERASE, DROP, ALTER

Linguagem de

Definição de dados: DELETE, REMOVE, ERASE, DROP, ALTER. “

NOTA:

NOTA:

. A linguagem SQL (Structured query language) é uma linguagem de banco de dados relacional usada para consultar, atualizar, e gerenciar bancos de dados relacionais e é o padrão de fato para produtos de bancos de dados;

. A linguagem DML (Linguagem de manipulação de dados) é usada para inserir, atualizar, e excluir dados, além de consultar um banco de dados;

. A linguagem DDL (Linguagem de definição de dados) é usada para

definição de dados) é usada para criar, alterar, ou remover bancos de dados relacionais, entidades, atributos, e outros objetos (por exemplo, exibições).

67 Ibidem

1. Que comando DDL deve ser usado para indicar que um automóvel foi vendido e deve ser removido da tabela de automóvel?

a.

a. DELETE FROM automóvelautomóvelautomóvel WHERE automóvelautomóvel_id = p1234
automóvel

b. REMOVE FROM automóvelautomóvel WHERE automóvelautomóvel_id = p1234

c. ERASE FROM automóvelautomóvel WHERE automóvelautomóvel_id = p1234

2. Que comando é usado para informar sobre a quantidade disponível atual de automóveis verde?

a. SELECT automóvelautomóvel_model WHERE automóvelautomóvel_color = 'verde'

- b. `SELECT * FROM automóvelautomóvel WHERE automóvelautomóvel_color = 'verde'`
c. `FIND * FROM automóvelautomóvel WHERE automóvelautomóvel_color = 'verde'`

3. Como um novo automóvel é inserida à tabela de automóvel no banco de dados? a. `INSERT INTO automovelautomovelautomovel (p3425, 'Rockrider', 'verde', 9999.00)`
b. `ADD INTO automovelautomovel VALUES (p3425, 'Rockrider', 'verde', 9999.00)` c. `INSERT INTO automovelautomovel VALUES (p3425, 'Rockrider', 'verde', 9999.00)"`

Respostas

1. O comando DDL que deve ser usado para indicar que um automovel automovelautomovel foi vendido e deve ser removido da tabela de automóvelautomóvel é:
a. `DELETE FROMDELETE FROM automovelautomovel WHEREWHERE automovelautomovel_number = p1234`

2. O comando usado para informar sobre a quantidade disponível

atual de automóveis verdes é:

- b. `SELECT * FROMSELECT * FROM automovelautomovel WHEREWHERE automovelautomovel_color = 'verde'`

3. Um novo automovelautomovelautomovel é adicionado à tabela de automovelautomovelautomovel no banco de dados com: c. `INSERT INTOINSERT INTO automovelautomovel VALUES (p3425, 'Audi', 'verde', 9999.00)"`

NOTA

Qual segmento de código cria um modo de exibição contendo o título, artista, e ano para todos os CDs com mais de 10 faixas?

- a. `CREATE VIEW CD _ More _ than _ 10 AS SELECT CD _ Title, CD _ Art- CD _ Year FROM CD _ Collection WHERE Tracks > 10`
b. `CREATE TABLE CD _ More _ than _ 10 AS SELECT CD _ Title, CD _ ArtistCD _ Year FROM CD _ Collection WHERE Tracks > 10`
c. `CREATE VIEW AS CD _ More _ than _ 10 FROM CD _ Collection WHERE Tracks > 10`

O código abaixo cria um modo de exibição contendo o título, artista, e ano para todos os CDs com mais de 10 faixas:

a. CREATE VIEW CD_More_than_10 AS SELECT CD_Title, CD_Artist, CD_Year FROM CD_Collection WHERE Tracks > 10

1. Qual comando DDL Raj pode usar para inserir um novo campo à tabela de automóveis, para armazenar o nome do arquivo da foto?

a. ALTER TABLE automovel ADD photo_file_name CHAR(30) NULL

b. ADD photo_file_name TO TABLE automovel

c. ALTER automovel TABLE USING photo_file_name CHAR(30) NULL

2. No exemplo da pergunta anterior, que efeito a palavra NULL tem ao se inserir um

No exemplo da pergunta anterior, que efeito a palavra NULL tem ao se inserir um

novo automóvel à tabela?

a. requer que o usuário digite o nome do arquivo da foto

b. não requer que o usuário digite o nome do arquivo da foto

c. automaticamente insere um nome de arquivo para a foto

3. Qual é a principal diferença entre o comando DML DELETE DELETE e o comando DDL e o comando DDL DROP? a. Eles realizam a mesma tarefa; portanto, não há diferença

b. DELETE somente remove todos os (ou um subconjunto de) registros da tabela; ele não remove a tabela em si

c. DROP somente remove todos os registros da tabela; ele não remove a tabela

Respostas

1. Raj pode inserir um novo campo à tabela de automóveis para armazenar o nome do arquivo da foto com:

a. ALTER TABLE automovel ADD photo_file_name CHAR(30) NULL

ALTER TABLE automovel ADD photo_file_name CHAR(30) NULL

2. Ao inserir um novo automóvel à tabela, a palavra NULL: b. não requer que o usuário digite o nome do arquivo da foto

3. A principal diferença entre o comando DML DELETE e o comando DDL DROP é: b. DELETEDELETE somente remove todos os (ou um

subconjunto de) registros da tabela; ele não remove a tabela em si”⁶⁸

NOTA

- CREATECREATE adiciona bancos de dados ou objetos.
- ALTERALTER modifica bancos de dados ou objetos.
- DROPDROP remove bancos de dados ou objetos.

DROP

“

NOTA

Qual comando SQL é usado para inserir uma nova tabela?

- a. CREATE TABLE table_name (column_name data type null/not null, column_name data type null/not null, e assim por diante)
- b. ADD TABLE table_name (column_name data type null/not null, column_name data type null/not null, e assim por diante)
- c. INSERT TABLE table_name (column_name data type null/not null, column_name data type null/not null, e assim por diante)”

O comando SQL usado para inserir uma nova tabela é:

- a. CREATE TABLE table _ name (column _ name data type null/not

null, column _ name data type null/not null, etc.)”⁶⁹

68 Ibidem

69 Apostila Certiport, Fundamentos de administração de banco de dados MTA exame 98-364, ftp://ftp.certiport.com/Marketing/Mta/Localizations/PTB/MTA_SSG_DbAdmin_individual_PTB.pdf,

Resumo de Alguns Comandos SQL

GERENCIANDO DADOS

- INSERT INTO Tabela VALUES
- SELECT Campos FROM Tabela
- UPDATE Tabela SET Campo = Valor
- DELETE FROM Tabela
- INSERT INTO PESSOAS VALUES ('Andre', '111')

DML

Linguagem de Manipulação de Dados

- INSERT: Insere dados
- UPDATE: Altera dados
- DELETE: Exclui dados

DQL

Linguagem de Consulta de Dados

- **SELECT:** Retorna dados
- Ordenação de dados
- Agrupamento de dados
- Filtros de seleção
- Funções aritméticas

FILTROS DE SELEÇÃO

Registros selecionados (WHERE) Operadores relacionais

- Igual (=), Diferente (!=)
- Maior (>), Maior ou igual (>=)
- Menor (<), Menor ou igual (<=)
- Nulo (IS NULL), ou não-nulo (IS NOT NULL)
- Entre intervalo (BETWEEN)
- Valor parcial (LIKE)

Operadores lógicos

- AND
- OR
- NOT

05/10/2015

CONSULTANDO DADOS COM FILTROS

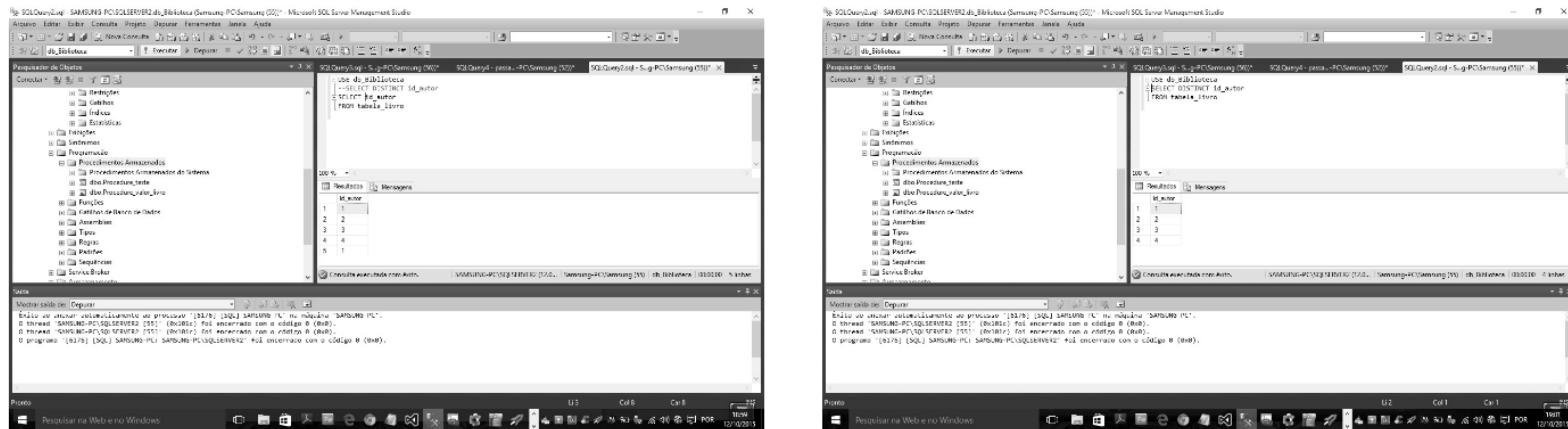
```
• SELECT Campos FROM Tabela WHERE Condição  
• UPDATE Tabela SET Campo = Valor WHERE Condição  
• DELETE FROM Tabela WHERE Condição  
SELECT * FROM PESSOAS WHERE CPF = '111.111.111-11' SELECT  
NOME FROM PESSOAS WHERE IDADE > 20  
UPDATE PESSOAS SET NOME = 'Fabio' WHERE CPF =  
'222.222.222-22' UPDATE PESSOAS SET IDADE = 25 WHERE NOME  
= 'Mario' DELETE FROM PESSOAS WHERE CPF = '222.222.222-22'  
DELETE FROM PESSOAS WHERE NOME = 'Mario'
```

1.1 - Select 1.1.1 - Uso de DISTINCT

Ex.:SELECT DISTINCT colunas FROM tabela

Código fonte SQL:

```
USE db_Biblioteca  
SELECT id_autor  
FROM tabela_livro
```



Tela 5.3.4 – Sem uso de DISTINCT

O que acontece sem não usamos o DISTINCT está o resultado na Tela 5.3.4, a coluna aparece repetindo as informações.

Agora vamos usar DISTINCT, vejamos o código fonte SQL:

USE db_Biblioteca

SELECT DISTINCT id_autor

FROM tabela_livro

Tela 5.3.4 – Com uso de DISTINCT

. O resultado está na Tela 5.3.4 – não se repete o “id_autor”.

“1. Qual comando asseguraria que visitantes em retorno sejam contados apenas uma vez ao executar a consulta contra o sistema de reservas?

- SELECT ONLY**
- SELECT DISTINCT**
- SELECT UNIQUE**

2. Para alguns relatórios será útil ver os resultados em ordem alfabetica. Qual comando produzirá uma lista em ordem alfabetica?

- a. ORDER BY column_name (ou ASC ou DESC)
- b. SORT BY column_name (ou ASC ou DESC)
- c. ARRANGE BY column_name (ou ASC ou DESC)

3. Uma tabela verdade é útil para visualizar os resultados de operadores lógicos. Ao comparar dois campos, qual condição sempre resulta em TRUE?

sempre resulta em TRUE?

- a. o operador AND e somente um campo = TRUE
- b. o operador OR e pelo menos um campo = TRUE
- c. o operador AND e pelo menos um campo = FALSE

Respostas

1. O comando que asseguraria que visitantes em retorno fossem contados apenas uma vez é:

- b. SELECT DISTINCT

2. O comando que retorna uma lista em ordem alfabetica é: a. ORDER BY column_name (ou ASC ou DESC)

3. A condição que sempre resulta em TRUE é:

- b. the operador OR e pelo menos um campo = TRUE

Exemplos de DISTINCT:

-
-- Selecione os nomes de todos os alunos que já fizeram alguma matrícula na FocoSite, sem repetição.

-
`SELECT DISTINCT(aluno.aluno) FROM pedido INNER JOIN aluno
ON pedido.aluno = aluno.codigo;`

-
-- Exiba o nome do aluno mais antigo da FocoSite

-
`SELECT DISTINCT(aluno.aluno) FROM pedido INNER JOIN aluno
ON pedido.aluno = aluno.codigo ORDER BY datahora ASC LIMIT 1;`

-- Exiba o nome do aluno mais recente da FocoSite

-

```
SELECT DISTINCT(aluno.aluno) FROM pedido INNER JOIN aluno  
ON pedido.aluno = aluno.codigo ORDER BY datahora DESC LIMIT 1;
```

-- Exiba o nome do terceiro aluno mais antigo da FocoSite

-

```
SELECT DISTINCT(aluno.aluno) FROM pedido INNER JOIN aluno  
ON pedido.aluno = aluno.codigo ORDER BY datahora ASC LIMIT 1  
OFFSET 2;
```

NOTA:

- SELECT SELECT é o comando SQL mais usado para selecionar os dados do banco de dados. “O conjunto de resultados da instrução SELECT forma a tabela virtual retornada pela exibição.”⁷⁰
- DISTINCTDISTINCT retorna somente valores distintos.
- WHEREWHERE quais são as linhas a recuperar.
- ORDER BYORDER BY ordem de retorno das linhas.

ORDER BY

- Operadores lógicos booleanos
- ANDAND exibe um registro caso tanto a primeira e a segunda condição sejam verdadeiras.
- OROR exibe um registro se a primeira ou a segunda condição for verdadeira.

⁷⁰ MSDN, Exibiçõe, [https://msdn.microsoft.com/pt-br/library/ms190706\(v=sql.100\).aspx](https://msdn.microsoft.com/pt-br/library/ms190706(v=sql.100).aspx), 03/10/2015

“Suponha uma tabela LIVROS e a tabela USUARIOS, e suponha o relacionamento entre elas onde o campo EMPRESTOU_ID da tabela LIVROS representa o valor do campo ID da tabela USUARIOS, armazenando o ID do usuário que emprestou o livro da biblioteca, ou então o valor NULL no caso do livro estar disponível na biblioteca. Qual a melhor forma de construir um SELECT de união entre as tabelas, visando trazer os dados de todos os livros, independente de estarem locados ou não?

- a. SELECT * FROM livros INNER JOIN usuarios ON livros.emprestou_id=usuario.id
- b. SELECT * FROM livros

```

RIGHT JOIN usuarios ON livros.emprestou_id=usuario.id c.
SELECT * FROM livros FULL JOIN usuarios ON
livros.emprestou_id=usuario.id d. SELECT * FROM livros LEFT JOIN
usuarios ON livros.emprestou_id=usuario.id

```

Resposta: D ⁷¹

INNER JOIN:

"INNER JOIN retorna somente as linhas correspondentes em ambas as tabelas (ou seja, retorna apenas as linhas para as quais a condição de junção satisfaz).

Demo 1: De acordo com os dados em nossas mesas de demonstração, os clientes com CustomerId 1 e 3 na tabela Clientes têm as encomendas na tabela Encomendas. Onde como o cliente com CustomerId 2 não tem qualquer ordem na tabela Pedidos. Assim, a Inner Join na coluna CustomerId entre Customers e Orders tabela retornará os

71 FocoSite, Apostila Curso SQL Completo, 20/09,/2015

detalhes do Cliente e Ordem dos Clientes com CustomerId 1 e apenas 3.

SELECT *

```

FROM Customers C
INNER JOIN Orders O
ON O.CustomerId = C.CustomerId72

```

Resultado

INNER JOIN

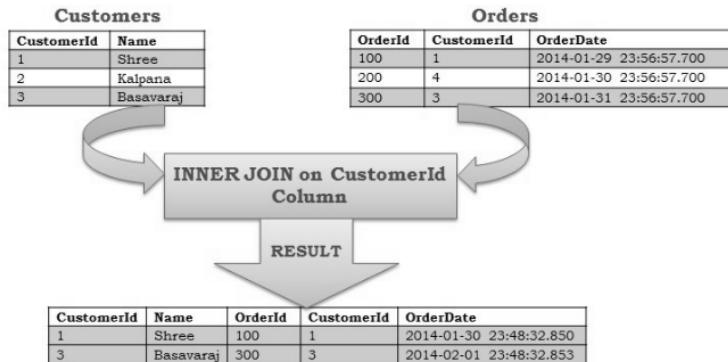


Gráfico INNER JOIN - Fonte imagem⁷³

72 Basavaraj Biradar, <http://sqlhints.com/tag/inner-join/>, 15/10/2015

73 <http://sqlhints.com/wp-content/uploads/2014/01/INNER-JOIN.jpg>, 15/10/2015

LEFT JOIN

Fonte Imagem⁷⁴

“Demo 1: De acordo com os dados em nossas mesas de demonstração, os clientes com CustomerId 1 e 3 na tabela Clientes têm as encomendas na tabela Encomendas. Onde como o cliente com CustomerId 2 não tem qualquer ordem na tabela Pedidos. Assim, o LEFT JOIN na coluna CustomerId entre Customers e Orders tabela retornará os detalhes do Cliente e Ordem dos Clientes com CustomerId 1 e 3 e para CustomerId 2 colunas ordem de tabela terá valor nulo no resultado.

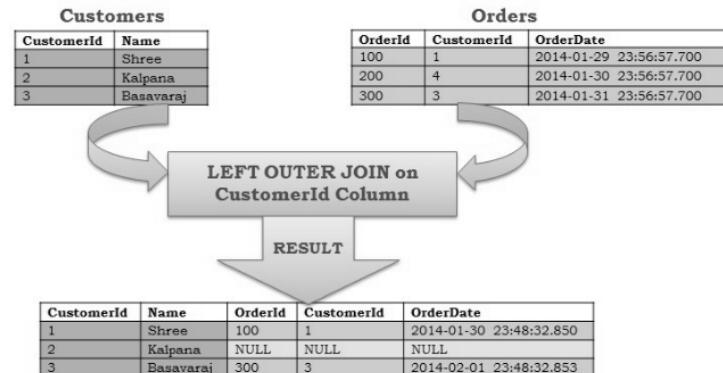
```
SELECT *
FROM Customers C
LEFT OUTER JOIN Orders O
ON O.CustomerId = C.CustomerId”75
```

74 ibidem

75 Basavaraj Biradar, <http://sqlhints.com/tag/inner-join/>,
15/10/2015

Resultado

LEFT OUTER JOIN



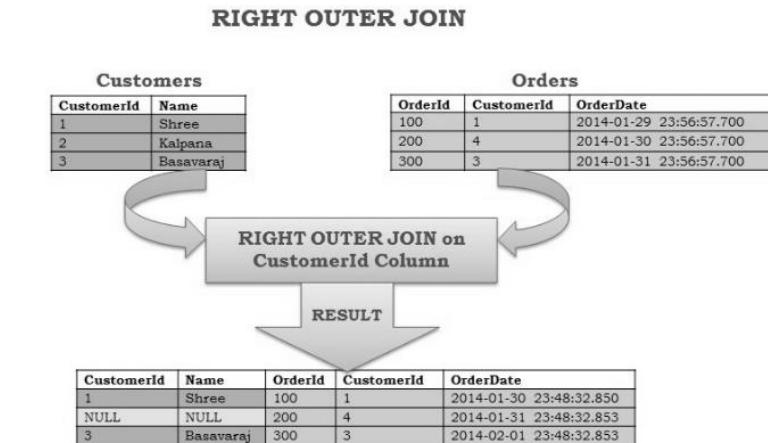
RIGH JOIN

“Demo 1: De acordo com os dados em nossas mesas de demonstração, apenas para o fim com OrderId 200 não temos é correspondente informações de clientes com CustomerId 4 na tabela Clientes. E para as outras duas ordens, a informação de cliente correspondente está presente na tabela Clientes. Assim, para os pedidos com CustomerId 1 e 3 terão detalhes do cliente e para a

ordem com CustomerId 4, as colunas da tabela Clientes terá valor nulo no resultado.

```
SELECT *  
FROM Customers C  
RIGHT OUTER JOIN Orders O  
ON O.CustomerId = C.CustomerId"76  
  
76 Basavaraj Biradar, http://sqlhints.com/tag/inner-join/,  
15/10/2015
```

Resultado



FULL JOIN

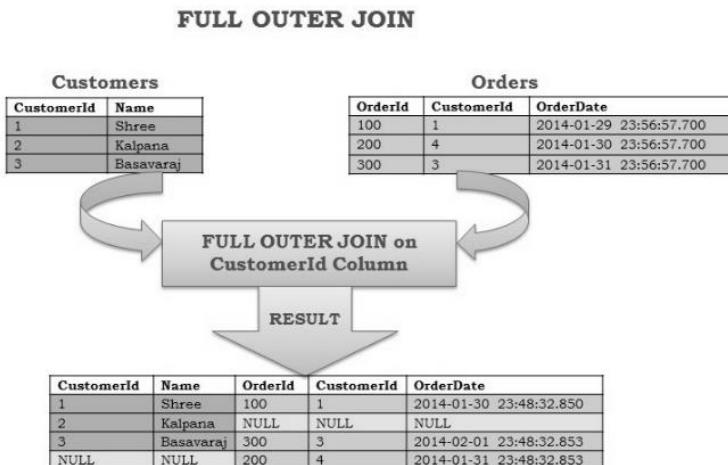
"Ele retorna todas as linhas de ambas as tabelas, se não houver nenhuma linha correspondente em qualquer um dos lados, em seguida, ele exibe valores nulos no resultado que colunas da tabela em tais linhas.

Junção externa completa = Left Outer Join + associação externa à direita

Demo 1: De acordo com os dados em nossas mesas de demonstração ao Cliente CustomerId 2 não tem a ordem na tabela Pedidos. Assim, no resultado do pleno Outer join entre Clientes e Encomendas tabela na coluna CustomerId terá valores nulos para as colunas da tabela Pedidos para o cliente com CustomerId 2.

E para a Ordem com OrderId 200 ter se CustomerId 4 não tem um registro correspondente na tabela de clientes com CustomerId 4. Assim, no resultado do pleno Outer join entre Clientes e Encomendas tabela na coluna CustomerId terão valores NULL para a tabela Clientes colunas para a Ordem com OrderId 200.

```
SELECT *
FROM Customers C
FULL OUTER JOIN Orders O
ON O.CustomerId = C.CustomerId"77
77
Resultado
```



CROSS JOIN

“CROSS JOIN também é referido como Produto cartesiano. Para cada linha na tabela à esquerda do CROSS JOIN todas as linhas da tabela direita são devolvidos e Vice-Versa (o resultado terá o produto cartesiano das linhas de tabelas associativas).

No.of registros no resultado da junção cruzada = (Número de linhas no lado esquerdo da tabela) * (Número de linhas em direito na

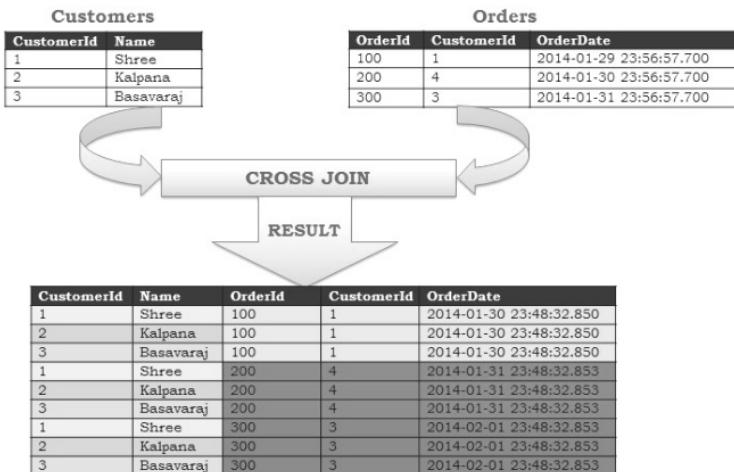
tabela)

SELECT *

FROM Customers C

CROSS JOIN Orders O⁷⁸

78 <http://sqlhints.com/tag/inner-join/>, 15/10/2015
CROSS JOIN



AUTO JOIN

"Se uma tabela é unida a si mesmo usando um dos tipos de juntar

explicado acima, em seguida, um tal tipo de junção é chamado de auto-junção.

Para esta demonstração juntar vamos criar uma tabela com os dados do empregado, conforme ilustrado na imagem abaixo com o seguinte roteiro:

EmployeeId	Name	ManagerId
1	Shree	1
2	Kalpana	1
3	Basavaraj	2
4	Monty	2

CREATE TABLE Employee

(EmployeeId INT, Name NVARCHAR(50), ManagerId INT)

GO

INSERT INTO Employee VALUES(1,'Shree',1)

INSERT INTO Employee VALUES(2,'Kalpana',1)

INSERT INTO Employee VALUES(3,'Basavaraj',2)

INSERT INTO Employee VALUES(4,'Monty',2)

GO

Demo 1: Agora, se nós precisamos de obter o nome do empregado e seu nome

Manager para cada empregado na tabela de funcionários. Então nós temos que Participe do empregado de mesa a si mesmo como empregado e seus dados Gestor está presente nesta tabela apenas como mostrado na consulta a seguir:

```
SELECT E.EmployeeId,  
E.Name 'Employee Name', M.Name 'Manager Name'  
FROM dbo.Employee E  
INNER JOIN Employee M  
ON M.EmployeeId = E.ManagerId
```

Resultado

EmployeeId	Employee Name	Manager Name
1	Shree	Shree
2	Kalpana	Shree
3	Basavaraj	Kalpana
4	Monty	Kalpana

Outros Exemplos:

-- Exibe uma lista com os títulos dos cursos da FocoSite e o tipo de curso ao lado

```
SELECT CURSO.CURSO, TIPO.TIPO FROM CURSO INNER JOIN TIPO  
ON CURSO.TIPO = TIPO.CODIGO;
```

-- Exiba uma lista com os títulos dos cursos da FocoSite, tipo do curso, nome do instrutor responsável pelo mesmo e telefone

```
SELECT CURSO.CURSO, TIPO.TIPO, INSTRUTOR.INSTRUTOR,  
INSTRUTOR.TELEFONE FROM CURSO INNER JOIN TIPO ON  
CURSO.TIPO = TIPO.CODIGO INNER JOIN INSTRUTOR ON  
CURSO.INSTRUTOR = INSTRUTOR.CODIGO;
```

-- Exiba uma lista com o código e data e hora dos pedidos e os códigos dos cursos de cada pedido

```
SELECT CODIGO, DATAHORA, CURSO FROM PEDIDO  
INNER JOIN PEDIDO_DETALHE ON PEDIDO.CODIGO =  
PEDIDO_DETALHE.PEDIDO;
```

-- Exiba uma lista com o código e data e hora dos pedidos e os títulos dos cursos de cada pedido

```
SELECT PEDIDO.CODIGO, DATAHORA, CURSO.CURSO FROM  
PEDIDO INNER JOIN PEDIDO_DETALHE ON PEDIDO.CODIGO  
= PEDIDO_DETALHE.PEDIDO INNER JOIN CURSO ON  
PEDIDO_DETALHE.CURSO = CURSO.CODIGO;
```

-- Exiba uma lista com o código e data e hora dos pedidos, nome do aluno e os títulos dos cursos de cada pedido

```
SELECT PEDIDO.CODIGO, DATAHORA, ALUNO.ALUNO,  
CURSO.CURSO FROM PEDIDO INNER JOIN PEDIDO_DETALHE ON  
PEDIDO.CODIGO = PEDIDO_DETALHE.PEDIDO INNER JOIN CURSO  
ON PEDIDO_DETALHE.CURSO = CURSO.CODIGO INNER JOIN  
ALUNO ON PEDIDO.ALUNO = ALUNO.CODIGO;
```

1.1.2 - Uso de UNION

Agora abra o SQL Management Studio 2014, vamos ao exemplo de UNION, essa condição com certeza cairá no Exame. UNION é uma condição que combina os resultados entre duas tabelas.

NOTA: Os colunas devem ter o mesmo tipo de dados, ou seja, o exemplo a seguir do código fonte SQL é campo "nome" da "tabela_autores" e "nome_livro" da "tabela_livro". Se informar campo "id_autor" da "tabela_autores" e "nome_livro" da "tabela_livro" - dará erro de sintaxe;

. Clique em "Nova Consulta" e digite o código fonte SQL:

```
USE db_Biblioteca
```

```
SELECT nome FROM tabela_autores WHERE ID_autor=2 UNION  
SELECT nome_livro FROM tabela_livro WHERE id_autor=2
```

. Esse código fonte irá unir a "tabela_autores" com a "tabela_livro" sendo filtrado por WHERE o id_autor=2, onde o campo "nome_livro" da tabela "tabela_livro" irá aparecer, somente, o filtro

do(s) livro(s) do “id_autor=2” . Veja a imagem 7.5 o resultado é Livro: ASP.NET do Autor Wall.

NOTA: Caso não consiga conexão com banco de dados veja o Anexo III – Erro de Instância Parada no Server Configuration Manager.

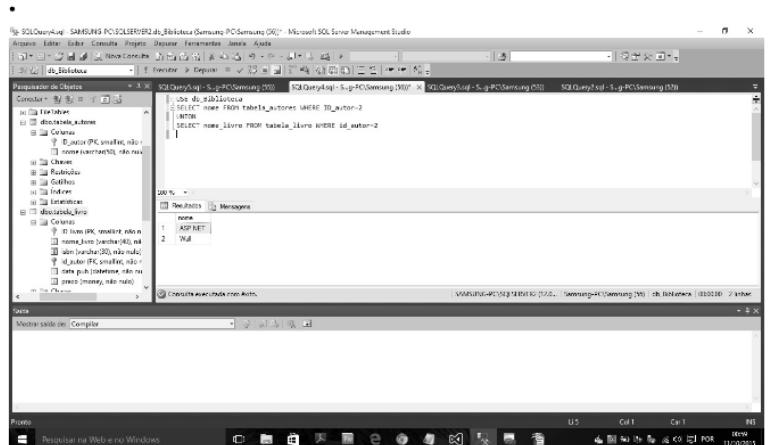


Imagen 7.5 – União das Tabelas com filtro de Livros por Autor

“NOTA

Qual é a diferença entre UNION e JOIN?

- UNION combina os resultados de duas consultas SQL quando há o mesmo número de colunas e tipos de dados ; JOIN retorna linhas quando há pelo menos uma correspondência de coluna.
- UNION combina os resultados de duas consultas SQL quando há pelo menos uma correspondência de coluna; JOIN retorna linhas quando há o mesmo número de colunas e tipos de dados.
- UNION somente retorna linhas que aparecem em ambas as tabelas; JOIN retorna linhas quando há pelo menos uma correspondência de coluna.

A diferença entre UNIONUNION ee JOINJOIN é:

A diferença entre UNION JOIN é:

- UNION combina os resultados de duas consultas SQL quando há o mesmo número de colunas e tipos de dados; JOIN retorna linhas quando há pelo menos uma correspondência de coluna”⁷⁹

1.1.3 - Uso de BETWEEN

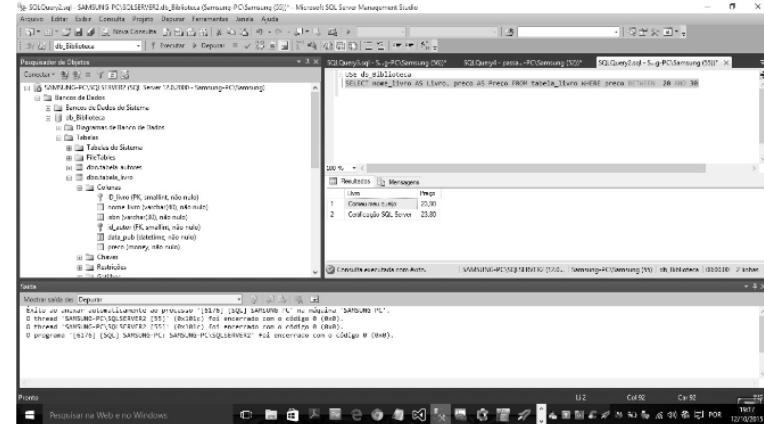
Seleção de intervalo, onde retornará valores entre o valor inicial e

final.

```
SELECT nome_livro AS Livro, preco AS Preço FROM tabela_livro  
WHERE preco BETWEEN 20 AND 30
```

O resultado são livros que os preços estão entre 20 e 30, veja a Tela 7.7.1

79 Apostila Certiport, Fundamentos de administração de banco de dados MTA exame 98-364, ftp://ftp.certiport.com/Marketing/Mta/Localizations/PTB/MTA_SSG_DbAdmin_individual_PTB.pdf, 05/10/2015



Tela 7.7.1

1.2 - Inserção

"1. Ao projetar o INSERT SQL, o que acontece se estiverem faltando dados de uma

Ao projetar o INSERT SQL, o que acontece se estiverem faltando dados de uma

coluna em particular?

- a. A instrução INSERT retorna um erro de sintaxe.
- b. A instrução INSERT usa o valor padrão para a coluna.
- c. A instrução INSERT insere um valor NULL na coluna.

2. Como Katarina pode configurar o INSERT para permitir ao programador copiar linhas de outras tabelas?

- a. Ela pode usar INSERT INTO com uma cláusula SELECT ... FROM.
- b. Isto não pode ser feito com um comando INSERT; ela precisa usar um JOIN.
- c. Ela pode usar construtores de valor de linha.

3. Qual é a sintaxe correta para inserir múltiplas linhas de uma só vez? a. INSERT INTO table (column1, val1a), (column2, val2a); b. INSERT (column1, column2) VALUES (val1a, val1b),
(val2a, val2b) INTO table;

c. INSERT INTO table (column1, column2) VALUES (val1a, val1b),
(val2a, val2b);

Respostas

1. Se há dados faltando para uma coluna:

b. A instrução INSERT usa o valor padrão para a coluna

2. Para configurar o INSERT a fim de permitir ao programador copiar linhas de outras tabelas:

- a. use o INSERT INTOINSERT INTO com uma cláusula com uma cláusula SELECTSELECT FROMFROM.

3.3. A sintaxe correta para inserir múltiplas linhas de uma só vez é: c. INSERT INTOINSERT INTO table (column1, column2) VALUES (val1a, val1b), (val2a, val2b);⁸⁰

INSERT INTOtable (column1, column2) VALUES (val1a, val1b), (val2a, val2b);"

NOTA:

- A instrução INSERT INTOINSERT INTOINSERT INTO é usada para inserir uma nova linha em uma tabela.

1.3 - Atualização

"1. O desenvolvedor quer atualizar todos os registros em um banco de dados para refletir um aumento no imposto sobre valor adicionado da província, de 8 por cento para 10 por cento. Qual opção abaixo tem a sintaxe correta?

- a. UPDATE RENTALS SET value_added_tax = .10;
b. SET sales_tax_rate = .10 IN RENTALS;
c. UPDATE sales_tax_rate = .10 IN RENTALS;
2. Katarina quer atualizar os dados se uma condição for verdadeira e realizar uma atualização alternativa se a condição for falsa. Qual das opções abaixo é a melhor escolha?
- a. instrução CASE
b. instrução LIKE
c. instrução IF/THEN/ELSE
3. Um dos desenvolvedores mencionou que às vezes ele precisa atualizar dados em uma tabela a partir de outra tabela. Qual opção abaixo tem a sintaxe correta para este tipo de atualização?
- a. UPDATE SET kayak = RENTALS.kayak + EQUIPMENT.kayak FROM RENTALS, EQUIPMENT;
b. UPDATE RENTALS SET RENTALS.kayak + EQUIPMENT.kayak;
c. UPDATE RENTALS SET kayak = RENTALS.kayak + EQUIPMENT.kayak FROM RENTALS, EQUIPMENT;

80 Ibidem

tipo de atualização?

- a. UPDATE SET kayak = RENTALS.kayak + EQUIPMENT.kayak FROM RENTALS, EQUIPMENT;
b. UPDATE RENTALS SET RENTALS.kayak + EQUIPMENT.kayak;
c. UPDATE RENTALS SET kayak = RENTALS.kayak + EQUIPMENT.kayak FROM RENTALS, EQUIPMENT;

Respostas

1. Para atualizar todos os registros em um banco de dados para refletir um aumento no imposto sobre valor adicionado da província de 8 por cento para 10 por cento o desenvolvedor deveria usar:

a. UPDATE RENTALS SET value_added_tax = .10

2. Katarina pode atualizar os dados se uma condição for verdadeira e realizar uma atualização alternativa se a condição for falsa com a:

a. instrução CASE

3. Para atualizar dados de uma tabela a partir de outra tabela, o desenvolvedor deveria usar:

c. UPDATE RENTALS SET kayak = RENTALS.kayak + EQUIPMENT.kayak FROM RENTALS, EQUIPMENT⁸¹

⁸¹ Apostila Certiport, Fundamentos de administração de banco de dados MTA exame 98-364, ftp://ftp.certiport.com/Marketing/Mta/Localizations/PTB/MTA_SSG_DbAdmin_individual_PTB.pdf, 05/10/2015

NOTA

- UPDATE é usado para atualizar registros existentes em uma tabela.
- A cláusula WHERE em uma instrução UPDATE especifica as linhas a atualizar.

1.4 - Exclusão

- “1. O que acontece se uma instrução WHERE não estiver incluída na instrução DELETE?
- a. A tabela inteira é excluída
 - b. Os dados da tabela são excluídos
 - c. O usuário recebe a mensagem de erro de sintaxe
2. Como as transações são úteis ao atualizar/excluir/inserir registros em um banco de dados?
- a.a. Elas permitem que instruções múltiplas sejam agrupadas juntas para evitar erros de integridade de dados
 - b. Elas contam o número de alterações feitas em registros na tabela
 - c. Elas restringem atualizações para permitir a execução de somente uma instrução de cada vez

3. Como um programador impõe regras de integridade de dados ao fazer alterações nos registros de um banco de dados?

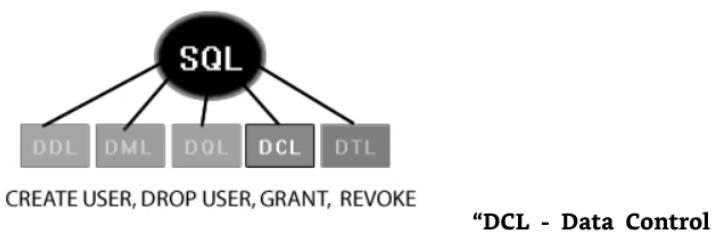
- a. Quando todas as instruções em uma transação forem completadas com êxito, emite uma instrução de confirmação
- b. Quando todas as instruções em uma transação forem completadas com êxito, emite uma instrução de reversão
- c. Quando um erro ocorre em uma instrução dentro de uma transação, emite uma instrução com- e vai para a próxima instrução

Respostas

1. Se você não incluir uma instrução WHERE na instrução DELETE:
- b. os dados da tabela são excluídos
2. Transações são úteis ao atualizar/excluir/inserir registros a um banco de dados porque:
- a. elas permitem que instruções múltiplas sejam agrupadas juntas para evitar erros de integridade de dados
3. Um programador pode impor regras de integridade de dados para fazer alterações nos registros de um banco de dados ao definir que:

- a. quando todas as instruções em uma transação forem completadas com êxito, seja emitida uma instrução de confirmação

3 - DCL SQL⁸²



Language

DCL - Data Control Language

- CREATE USER Nome: Cria um usuário
 - DROP USER Nome: Exclui um usuário
 - GRANT: Habilita acessos
 - REVOKE: Revoga acessos”⁸³
- . GRANT – atribui privilégios de acesso do usuário a objetos do banco de dados;

82 Apostila Certiport, Fundamentos de administração de banco de dados MTA exame 98-364, ftp://ftp.certiport.com/Marketing/Mta/Localizations/PTB/MTA_SSG_DbAdmin_individual_PTB.pdf, 05/10/2015

83

- . REVOKE – remove os privilégios de acesso aos objetos obtidos com o comando GRANT.

4. TCL SQL

- . COMMIT – salvar o trabalho feito;
- . SAVEPOINT – identificar um ponto em uma transação para que mais tarde você pode efetuar um ROLLBACK;
- . ROLLBACK – restaurar banco de dados ao original desde o último COMMIT.

CAPÍTULO 6 1 – Sintax Transact-SQL

Terceiro gatilho mental: tenha uma

atitude pró-ativa e pouco ré-ativa, pra frente é que se anda. Por que se não entende o que lês seja humilde, leia pra guardar a informação, leia pra familiarizar-se com o código ou informação.

Fonte de imagem⁸⁴

O Transact-SQL é um assunto que vai cair no Exame não tem para onde correr, por isso, não se detenha em preocupar-se em guardar tanta informação, preocupe-se em conhecer a informação a respeito de:

1.1 - SQL Server Syntax⁸⁵

As palavras em inglês não poderão ser traduzidas, é proposital essa inclusão de informação na Documentação Oficial da Microsoft e grande parte desse conteúdo é dessa fonte, pois a Certificação é em

Tecnologia Microsoft e eles devem orientar o quê e como estudar.

**84 [http://www.istockphoto.com/vector/pass-20934226?
st=f9bf3bd, 10\10\2015](http://www.istockphoto.com/vector/pass-20934226?st=f9bf3bd)**

**85 MSDN, [https://msdn.microsoft.com/pt-br/library/
ms189799\(v=sql.120\).aspx, 04/10/2015](https://msdn.microsoft.com/pt-br/library/ms189799(v=sql.120).aspx)**

O quê se segue é instruções em SQL, e absoluta certeza esse é o foco pois são elas que movem o motor do SQL Server.

1.1. 1- Trigger (desencadear em um INSERT, UPDATE, ou DELETE para uma tabela ou exibição (DML Trigger))

Trigger

3) Crie uma TRIGGER que altere de forma automática o valor da locação aos finais de semana. Aos sábados e Domingos, as locações devem ser acrescidas em 10% .

```
CREATE or replace TRIGGER trg_valor_locacao
BEFORE INSERT ON locacao
FOR EACH ROW
BEGIN
If to_char(:new.data_locacao,'d') in (1,7) then
:new.valor:=:new.valor*1.1;
End if;
END;
```

Imagen 3.41 - Fonte da Imagem – Exemplo Trigger⁸⁶

86 http://images.slideplayer.com.br/7/1851203/slides/slide_2.jpg, 10\10\2015

Significa: desencadear em um INSERT, UPDATE, ou DELETE para uma tabela ou exibição (DML trigger). Calma amigo, estamos aqui para facilitar, mas é necessário você guarda em sua memória a palavra “trigger”, “on”, “statement”, “table”, “view”.

A sintaxe do código fonte SQL:

```
CREATE TRIGGER [ schema_name . ]trigger_name
ON { table | view }
[ WITH <dml_trigger_option> [,..n] ]
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ][ , ][ UPDATE ][ , ][ DELETE ] }
[ WITH APPEND ]
[ NOT FOR REPLICATION ]
AS { sql_statement [ ; ][ ..n ] | EXTERNAL NAME <method specifier
[ ; ]> }
```

```

<dml_trigger_option> ::= [ ENCRYPTION ]
[ EXECUTE AS Clause ]

<methodSpecifier> ::=
assembly_name.class_name.method_name

Trigger on a CREATE, ALTER, DROP, GRANT, DENY, REVOKE, or
UPDATE STATISTICS statement (DDL Trigger)
CREATE TRIGGER trigger_name
ON { ALL SERVER | DATABASE }
[ WITH <ddl_trigger_option> [,..n] ]
{ FOR | AFTER } { event_type | event_group } [,..n]
AS { sql_statement [ ; ] [,..n] } | EXTERNAL NAME < method specifier
> [ ; ]

```

<ddl_trigger_option> ::= [ENCRYPTION]
[EXECUTE AS Clause]

2.1 - CREATE TRIGGER (Transact-SQL)

Microsoft SQL Server

◆ Triggers (Gatilhos)

```

CREATE TRIGGER member_delete
ON member FOR DELETE
AS
IF (SELECT COUNT(*)
    FROM loan,deleted
    WHERE loan.member_no = deleted.member_no)> 0
BEGIN
    PRINT 'A transação não pode ser executada.'
    PRINT 'Este membro possui livro não devolvido'
    ROLLBACK TRANSACTION
END

```

Imagen 3.42 - Fonte da Imagem – Exemplo Triggers⁸⁷

. Na imagem 3.42 é criado um Gatilho que dispara criando um

trigger nome member_delete a fim de apagar um número de registros maior que "0" para isso uma WHERE cria um filtro onde o total lido COUNT conta e safisfeito a condição maior que "0" então BEGIN ou seja, dispare na tela do computador no SQL Server 2014 a

87 http://images.slideplayer.com.br/3/389605/slides/slide_45.jpg, 10\10\2015

informação "A transação não pode ser executada", isso por que o membro possui livro não devolvido, ainda em posse, sendo então uma quantidade maior que "0", então ele tem 1 ou mais livros em sua possose.

"SQL - Structured Query LanguageSQL - Structured Query Language, ou seja Linguagem

Estruturada de Consulta. E SQL contém a formatacao de Conjunto de Linguagens como:

- DDL (Data Definition Language): Linguagem de Definicao de Dados;
- DML (Data Manipulation Language): Linguagem de Manipulacao de Dados;
- DQL (Data Query Language): Linguagem de Consulta de Dados;

Exemplos de DQL:

-
-- Utilizando subquery, exiba uma lista com os nomes dos cursos disponibilizados pela FocoSite informando para cada curso qual o seu menor valor de venda já praticado.

-

```
SELECT CURSO, (SELECT MIN(VALOR) FROM PEDIDO_DETALHE  
WHERE PEDIDO_DETALHE.CURSO = CURSO.CODIGO) AS  
MENOR_VALOR FROM CURSO;
```

-
-- Utilizando subquery e o parâmetro IN, exiba os nomes dos cursos disponibilizados pela FocoSite cujo tipo de curso seja 'Programação'.

-

```
SELECT CURSO FROM CURSO WHERE TIPO IN (SELECT CODIGO  
FROM TIPO WHERE TIPO = 'Programação');
```

-
-- Utilizando subquery e o parâmetro EXISTS, exiba novamente os

nomes dos cursos disponibilizados pela FocoSite cujo tipo de curso seja 'Programação'.

-
SELECT CURSO FROM CURSO WHERE EXISTS (SELECT CODIGO FROM TIPO WHERE TIPO.CODIGO = CURSO.TIPO AND TIPO.TIPO = 'Programação');

-- Exiba uma lista com os nomes dos instrutores da FocoSite e ao lado o total acumulado das vendas referente aos cursos pelo qual o instrutor é responsável.

-
SELECT INSTRUTOR, (SELECT SUM(PEDIDO_DETALHE.VALOR) FROM PEDIDO_DETALHE INNER JOIN CURSO ON PEDIDO_DETALHE.CURSO = CURSO.CODIGO WHERE CURSO.INSTRUTOR = INSTRUTOR.CODIGO) AS TOTAL_DE_VENDAS FROM INSTRUTOR;

-- Crie uma visão que exiba os nomes dos alunos e quanto cada um já

comprou em cursos

-

CREATE VIEW ALUNOS_E_COMPRA AS SELECT ALUNO, (SELECT SUM(PEDIDO_DETALHE.VALOR) FROM PEDIDO_DETALHE INNER JOIN PEDIDO ON PEDIDO_DETALHE.PEDIDO = PEDIDO.CODIGO WHERE PEDIDO.ALUNO = ALUNO.CODIGO) AS TOTAL_DE_COMPRA FROM ALUNO;

- DCL (Data Control Language): Linguagem de Controle de Dados;
- DTL (Data Transaction Language): Linguagem de Transação de Dados;⁸⁸

Cria um gatilho DML, DDL ou de logon.:

Um gatilho é um tipo especial de procedimento armazenado que é executado automaticamente quando um evento ocorre no servidor de banco de dados.

Os gatilhos DML são executados quando um usuário tenta modificar dados através de um evento DML (linguagem de manipulação de dados). Os eventos DML são instruções INSERT, UPDATE ou DELETE em uma tabela ou exibição. Esses gatilhos são disparados

quando qualquer evento válido é acionado, independentemente de quaisquer linhas da tabela serem afetadas ou não.

NOTA:

DML as instruções são: INSERT, UPDATE ou DELETE

Os gatilhos DDL são executados em resposta a diversos eventos DDL (linguagem de definição de dados). Esses eventos correspondem, basicamente, a instruções Transact-SQL CREATE, ALTER e DROP e determinados procedimentos armazenados do sistema que executam operações do tipo DDL. Os gatilhos de logon são disparados em resposta ao evento LOGON que é gerado quando as sessões de um usuário estão sendo estabelecidas. Os gatilhos podem ser criados diretamente à partir de instruções Transact-SQL ou de métodos de Assembly (Linguagem de Máquina) criados no CLR (Common Language Runtime) do Microsoft .NET Framework e carregados em uma instância do SQL Server. O SQL Server permite criar vários gatilhos para qualquer instrução específica.

88 Ângelo M. Almeida Santos, Programando na Web com ASP Clássico, 2015, São Paulo, Ed.Viena, 04/10/2015

NOTA:

DDL as instruções são: CREATE, ALTER e DROP

2.1.1 - Argumentos

2.1.1.1 - schema_name

É o nome do esquema ao qual o gatilho DML pertence. Os gatilhos DML são definidos no escopo do esquema da tabela ou na exibição na qual são criados. schema_name não pode ser especificado para gatilhos DDL ou de logon.

2.1.1.2 - trigger_name

É o nome do gatilho. Um trigger_name deve estar de acordo com as regras para identificadores, a menos que trigger_name não possa ser iniciado com # ou ##.

table | view

É a tabela ou exibição na qual o gatilho DML é executado e às vezes referenciado como a tabela de gatilho ou exibição de gatilho. A especificação do nome totalmente qualificado da tabela ou da

exibição é opcional. Uma exibição só pode ser referenciada por um gatilho INSTEAD OF. Gatilhos DML não podem ser definidos em tabelas temporárias locais ou globais.

2.1.1.2.1 - DATABASE

Aplica o escopo de um gatilho DDL ao banco de dados atual. Se for especificado, o gatilho será acionado sempre que event_type ou event_group ocorrer no banco de dados atual.

2.1.1.2.2 - ALL SERVER

Aplica o escopo de um gatilho DDL ou de logon ao servidor atual. Se for especificado, o gatilho será acionado sempre que event_type ou event_group ocorrer em qualquer local no servidor atual.

2.1.1.2.3 - WITH ENCRYPTION

Ofusca o texto da instrução CREATE TRIGGER. O uso de WITH ENCRYPTION impede que o gatilho seja publicado como parte

da replicação do SQL Server. WITH ENCRYPTION não pode ser especificado para gatilhos CLR.

3.1.1.2.4 - EXECUTE AS

Especifica o contexto de segurança no qual o gatilho é executado. Permite controlar a conta de usuário que a instância do SQL Server usa para validar permissões em quaisquer objetos do banco de dados referidos pelo gatilho.

3.1.1.2.4.a - AFTER especifica que o gatilho DML é disparado apenas quando

todas as operações especificadas na instrução SQL de gatilho são executadas com êxito. Todas as verificações de restrição e ações referenciais em cascata também devem obter êxito para que este gatilho seja disparado.

AFTER é o padrão quando **FOR** é a única palavra-chave especificada.
Gatilhos **AFTER** não podem ser definidos em exibições.

3.1.1.1 - INSTEAD OF

Especifica que o gatilho DML será executado E NÃO a instrução SQL de gatilho, substituindo assim as ações das instruções de gatilho. **INSTEAD OF** não pode ser especificado para gatilhos DDL ou de logon.

NOTA:

DDL as instruções são: **CREATE, ALTER e DROP**

DML as instruções são: **INSERT, UPDATE ou DELETE**

No máximo, um gatilho **INSTEAD OF** por instrução **INSERT, UPDATE ou DELETE** pode ser definido em uma tabela ou exibição. Entretanto, você pode definir exibições sobre exibições, onde cada uma tem seu próprio gatilho **INSTEAD OF**.

Os gatilhos **INSTEAD OF** não são permitidos em exibições atualizáveis que usam **WITH CHECK OPTION**. O SQL Server gera um erro quando um gatilho **INSTEAD OF** é adicionado a uma **WITH CHECK OPTION** de exibição atualizável especificado. O usuário deve

remover essa opção usando **ALTER VIEW** antes de definir o gatilho **INSTEAD OF**.

Especifica as instruções de modificação de dados que, quando tentadas nessa tabela ou exibição, ativam o gatilho DML. É necessário especificar pelo menos uma opção. É permitida qualquer combinação dessas opções em qualquer ordem na definição do gatilho.

Para gatilhos **INSTEAD OF**, a opção **DELETE** não é permitida em tabelas que tenham um relacionamento referencial que especifique uma ação **ON DELETE** em cascata. Da mesma maneira, a opção **UPDATE** não é permitida em tabelas que tenham um relacionamento referencial que especifique uma ação **ON UPDATE** em cascata.

3.1.1.2 - WITH APPEND

Especifica que um gatilho adicional de um tipo existente deve ser adicionado. **WITH APPEND** não pode ser usado com gatilhos **INSTEAD OF** ou se o gatilho **AFTER FOR** explicitamente declarado. **WITH APPEND** só pode ser usado quando **FOR** é especificado,

sem INSTEAD OF ou AFTER, por razões de compatibilidade com versões anteriores. WITH APPEND não poderá ser especificado se EXTERNAL NAME for especificado (quer dizer, se o gatilho for um gatilho CLR).

3.1.1.2.1 - event_type

É o nome de um evento da linguagem Transact-SQL que, após a execução, faz com que um gatilho DDL seja acionado. Os eventos válidos para gatilhos DDL estão listados em Eventos DDL.

3.1.1.2.2 - event_group

É o nome de um agrupamento predefinido de eventos da linguagem TransactSQL. O gatilho DDL será acionado após a execução de qualquer evento da linguagem Transact-SQL que pertença ao event_group.

Depois da conclusão de CREATE TRIGGER, event_group também atuará como uma macro por meio da adição dos tipos de evento que ele abrange à exibição do catálogo sys.trigger_events.

3.1.1.2 - NOT FOR REPLICATION

Indica que o gatilho não deve ser executado quando um agente de replicação modificar a tabela envolvida no gatilho.

3.1.1.2.1 - sql_statement

São as condições e as ações do gatilho. As condições de gatilho especificam critérios adicionais que determinam se os eventos DML, DDL ou de logon fazem com que as ações de gatilho sejam executadas.

NOTA:

DDL as instruções são: CREATE, ALTER e DROP

NOTA:

DML as instruções são: INSERT, UPDATE ou DELETE

As ações de gatilho especificadas nas instruções Transact-SQL entram em vigor quando a operação é tentada.

Os gatilhos podem incluir qualquer número e tipo de instruções Transact-SQL, com exceções.

Comentários. Um gatilho é criado para verificar ou alterar dados com base em uma instrução de definição ou modificação de dados. Ele não deve retornar dados ao usuário.

2 - CASE (Transact-SQL)

Aqui “CASE” é uma das instruções da Linguagem de Controle de Fluxo. Temos, também: IF...ELSE , BEGIN...END , etc. Entretanto, não nos será exigido com rigor nesse Primeiro Exame Microsoft MTA, sendo esse conteúdo para o Segundo Exame, no entanto, é importante nos familiarizar com ele, nos ajudará em raciocínio lógico de “Controle de Fluxo”. E poderá, também, acontecer cair alguma expressão do tipo “Controle de Fluxo”, portanto, já se tem uma familiaridade com o termo.

Avalia uma lista de condições e retorna uma das várias expressões de resultado possíveis.

A expressão CASE tem dois formatos:

- . A expressão CASE simples compara uma expressão com um conjunto de expressões simples para determinar o resultado;
- . A expressão CASE pesquisada avalia um conjunto de expressões booleanas para determinar o resultado.

Os dois formatos dão suporte a um argumento ELSE opcional.

CASE pode ser usada em qualquer instrução ou cláusula que permita uma expressão válida. Por exemplo, você pode usar CASE em instruções, como SELECT, UPDATE, DELETE e SET, e em cláusulas, como select_list, IN, WHERE, ORDER BY e HAVING.

2.1 - Sintaxe do CASE

Simple CASE expression:

CASE input_expression

WHEN when_expression THEN result_expression [...n] [ELSE else_result_expression]

END

Searched CASE expression:

CASE

WHEN Boolean_expression THEN result_expression [...n] [ELSE else_result_expression]

END

2.2 - Argumentos

Argumentos

2.2.1 - input_expression

É a expressão avaliada quando o formato CASE simples é usado. input_expression é qualquer expressão válida.

2.2.2 - WHEN when_expression

when_expression

É uma expressão simples à qual input_expression é comparada quando o formato CASE simples é usado. when_expression é qualquer expressão válida. Os tipos de dados de input_expression e cada when_expression devem ser iguais ou ser uma conversão implícita.

2.2.3 - THEN result_expression

É a expressão retornada quando input_expression igual a when_expression é avaliada como TRUE ou quando Boolean_expression é avaliada como TRUE. result_expression é qualquer expressão válida.

2.2.4 - ELSE else_result_expression

É a expressão retornada se nenhuma operação de comparação for avaliada como TRUE. Se esse argumento comparação for avaliada NULL, else_result_expression é qualquer expressão válida. Os tipos de dados de else_result_expression e qualquer result_expression devem ser iguais ou ser uma conversão implícita. for omitido e nenhuma operação de

como TRUE, CASE retornará

2.2.5 - WHEN Boolean_expression

É a expressão booleana avaliada quando o formato CASE simples é usado. Boolean_expression é qualquer expressão booleana válida.

3 - Gatilhos DML⁸⁹

Os gatilhos DML são um tipo especial de procedimento armazenado que entra em vigor automaticamente quando um evento DML (linguagem de manipulação de dados) ocorre e afeta a tabela ou exibição definida no gatilho. Os eventos DML são instruções INSERT, UPDATE ou DELETE (Provável cai no Exame).

Os gatilhos DML podem ser usados:

- . Para impor regras de negócios;
- . Integridade de dados;
- . Consultar outras tabelas;
- . Incluir instruções Transact-SQL complexas.

O gatilho e a instrução que o dispara são tratados como uma transação simples, que pode ser revertida dentro do gatilho. Se um erro grave for detectado (espaço em disco insuficiente, por exemplo), toda a transação será revertida automaticamente (volta para onde partiu, sem alterações).

1.1 - Benefícios do gatilho DML

⁸⁹ MSDN, [https://msdn.microsoft.com/pt-br/library/ms178110\(v=sql.120\).aspx](https://msdn.microsoft.com/pt-br/library/ms178110(v=sql.120).aspx), 02/09/2015

Os gatilhos DML são semelhantes a restrições, pois podem impor integridade de entidade ou integridade de domínio. Em geral, a integridade da entidade sempre deve ser imposta no menor nível por índices que fazem parte das restrições PRIMARY KEY e UNIQUE ou que são criados independentemente de restrições. A integridade de domínio deve ser imposta por restrições CHECK e a RIN (integridade referencial) deve ser imposta por restrições FOREIGN KEY. Os gatilhos DML são muito úteis quando os recursos suportados por restrições não atendem às necessidades funcionais do aplicativo.

A lista a seguir compara gatilhos DML com restrições e identifica quando os gatilhos DML mais benefícios que ela.

- . Os gatilhos DML podem colocar as alterações em cascata através das tabelas relacionadas no banco de dados; no entanto, essas

alterações podem ser executadas com mais eficiência utilizando restrições de integridade referencial em cascata. As restrições FOREIGN KEY só podem validar um valor de coluna com uma combinação exata de valor em outra coluna a menos que a cláusula REFERENCES defina uma ação referencial em cascata;

. Podem proteger contra operações mal-intencionadas ou incorretas do tipo INSERT, UPDATE, e DELETE, e fazer cumprir as outras restrições mais complexas do que aquelas definidas nas restrições CHECK;

Diferentemente das restrições CHECK, os gatilhos DML podem fazer referência a colunas em outras tabelas. Por exemplo, um gatilho pode usar um SELECT de outra tabela para comparar com os dados atualizados ou inseridos e para efetuar ações adicionais, como modificar os dados ou exibir uma mensagem de erro definida pelo usuário;

. Podem avaliar o estado de uma tabela antes e depois da modificação dos dados e efetuar ações com base nessa diferença;

. Vários gatilhos DML do mesmo tipo (INSERT, UPDATE, ou

DELETE), em uma tabela, permitem que múltiplas ações diferentes ocorram em resposta à mesma instrução de modificação;

. Restrições só podem comunicar erros através de mensagens de erro padronizadas do sistema. Se o aplicativo exigir ou beneficiar-se de mensagens personalizadas e tratamento de erros mais complexo, é necessário usar um gatilho;

. Gatilhos DML podem desabilitar ou reverter alterações que violam a integridade referencial, cancelando assim a tentativa de modificação de dados. Pode ser que tais gatilhos só tenham efeito ao alterar uma chave estrangeira e se o novo valor não combinar com sua chave primária. Porém, as restrições FOREIGN KEY normalmente são usadas com este propósito;

. Se houver restrições na tabela de gatilhos, elas serão verificadas após a execução do gatilho INSTEAD OF, mas antes da execução do gatilho AFTER. Se as restrições forem violadas, as ações do gatilho INSTEAD OF serão revertidas e o gatilho AFTER não será executado.

1.2 - Tipos de Triggers (gatilhos) DML

Comparação entre Triggers		
Característica	INSTEAD OF	AFTER
Declaração DML	Simulada, mas não executada	Executada, mas pode ser revertida no trigger (Roll back)
Timing	Antes das constraints PK e FK	Após a transação completa, mas antes do commit
Nº de eventos por tabela	Um	Múltiplos
Aplicável em Views?	Sim	Não
Permite Aninhamento?	Depende das opções do servidor	Depende das opções do servidor
É Recursivo?	Não	Depende das opções do Banco de Dados.

Imagen 9.3 -Tabela Triggers⁹⁰

1.2.1 - Gatilho AFTER

90 https://www.youtube.com/watch?v=HQgJY96Xb_A&index=49&list=PLucm8g_ezqNqI5cW3alteV50l cMCcHYRK, 11\10\2015

Os gatilhos AFTER são executados depois que a ação das instruções INSERT, UPDATE, MERGE ou DELETE é executada. Os gatilhos AFTER jamais são executados em caso de uma violação de restrição; por isso, estes gatilhos não podem ser usados em processamentos que possam evitar as violações de restrição. Para cada ação INSERT, UPDATE ou DELETE especificada em uma instrução MERGE, o gatilho correspondente é disparado para cada operação DML.

1.2.2 - Gatilho INSTEAD OF

Os gatilhos INSTEAD OF substituem as ações padrão da instrução de gatilho. Portanto, eles podem ser usados para executar uma verificação de erro ou valor em uma ou mais colunas, e executar ações adicionais antes de inserir, atualizar ou excluir uma ou mais linhas. Por exemplo, quando o valor que estiver sendo atualizado em uma coluna de salário calculado por hora, de uma tabela de folha de pagamento, exceder um valor especificado, um gatilho poderá

ser definido para produzir uma mensagem de erro e reverter a transação, ou inserir um novo registro em uma trilha de auditoria, antes de inserir o registro na tabela de folha de pagamento. A principal vantagem dos gatilhos INSTEAD OF é que eles habilitam exibições que não seriam atualizáveis para oferecer suporte a atualizações. Por exemplo, uma exibição baseada em várias tabelas base deve usar um gatilho INSTEAD OF para oferecer suporte a inserções, atualizações e exclusões que referenciam dados em mais de uma tabela. Outra vantagem dos gatilhos INSTEAD OF é que eles o habilitam a codificar lógica que pode rejeitar partes de um lote e, ao mesmo tempo, permitir que outras partes do lote tenham êxito.

CAPÍTULO 7

1. Stored Procedures

Controle de Decisões, as stored procedures, usa a janela de execução de comandos T-SQL e SQL Management Studio, quando instalamos o Data Tools.

“Melhora a velocidade em execução, compila os comandos stored

procedure, na primeira vez que este é executado, e mantém estes comandos na memória cache, para agilizar as próximas execuções. Isto faz com que as chamadas subsequentes executem bem mais rápido.”⁹¹

- . Com a Procedures se centraliza a lógica de acesso aos dados em um único local, facilitando a manutenção e otimização de código;
- . Define o usuário com permissão de acesso, quem pode ou não executá-las: inserções, atualizações ou exclusão de dados;

1.1 – Criando Stored Procedures

Abra o SQL Management Studio

Código fonte SQL:

91 Júlio Battisti, SQL Server 2005 Curso Completo 2005, Ed. Axcel, 05.10.2010

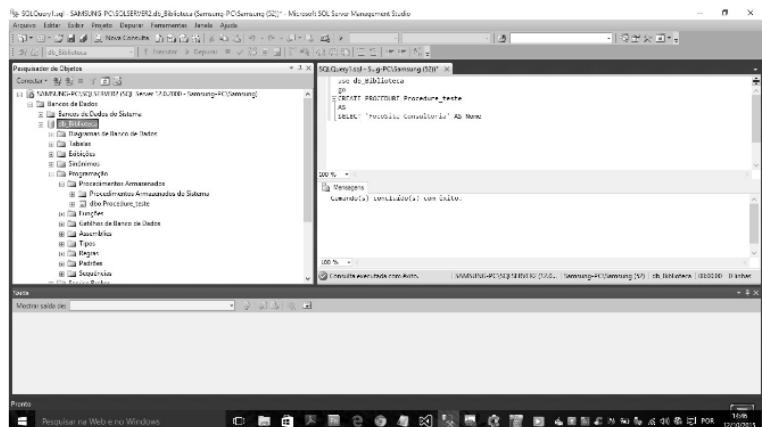
USE db_Biblioteca

GO

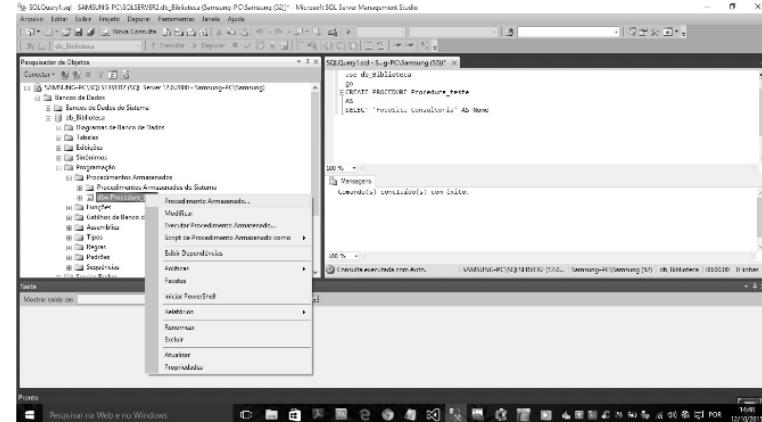
CREATE PROCEDURE teste

AS

SELECT 'FocoSite Consultoria' AS Nome



'Tela 4.2.1 - Criando Procedures'

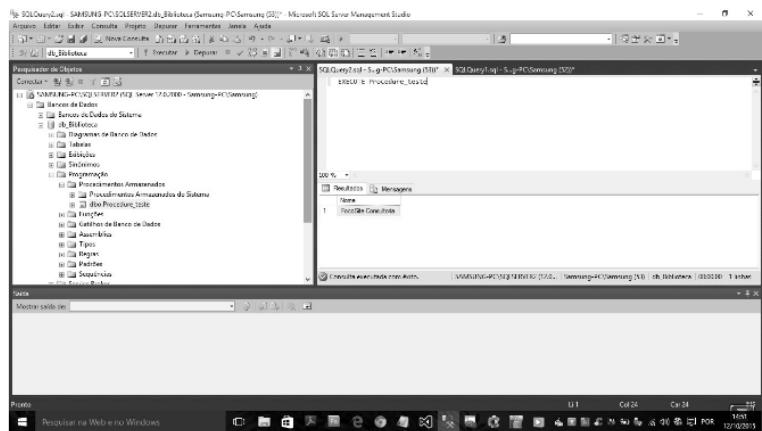


Tela 4.2.2

- . Com o botão direito do mouse sobre o nome da “Procedure_teste” abre-se uma cortina de opções para executar;
- . Agora clique em “Nova Consulta” e digite o código fonte SQL:

EXECUTE procedure_teste

. Observe a Tela 4.2.3 e veja que aparece “Focosite Consultoria”



Tela 4.2.3

Nova Procedure. Digite o código fonte SQL:

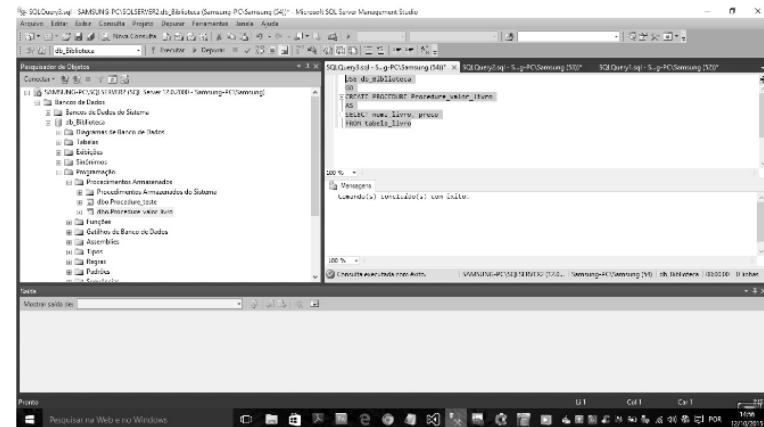
```

USE db_Biblioteca
GO
CREATE PROCEDURE Procedure_Valor_Livro AS
SELECT nome_livro, preco
FROM tabela_livro

```

. Após execução “atualize” os Procedimentos Armazenados do

Sistema, usando o botão direito do mouse;

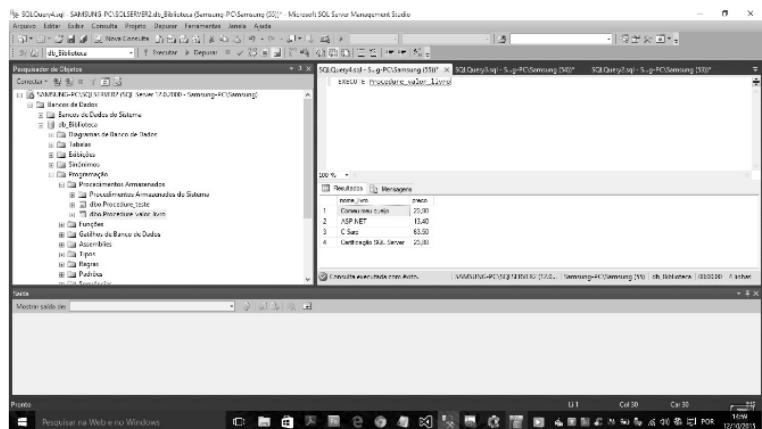


Tela 4.2.3

Execute a procedure com o código fonte:

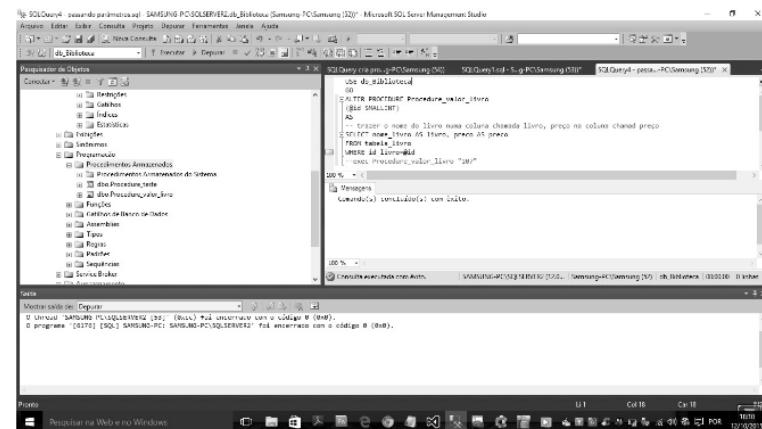
EXECUTE Procedure_Valor_Livro

. Clique “executar”. O resultado na Tela 4.2.4



Tela 4.2.4

1.2 – Alterando Procedures passando parâmetros



Tela 4.2.5

Código fonte SQL:

```
USE db_Biblioteca
GO
ALTER PROCEDURE Procedure_valor_livro
(@id SMALLINT)
AS
-- trazer o nome do livro numa coluna chamada livro, preço na
-- coluna chamad preço
SELECT nome_livro AS livro, preco AS preco
FROM livro
WHERE id_livro = @id
-- exec Procedure_valor_livro "10"
```

```

FROM tabela_livro
WHERE id_livro=@id
--exec Procedure_valor_livro "107"

```

. A procedure criada nomeada “Procedure_valor_livro” agora será alterada recebendo parâmetros de entrada ID que filtrará o nome do livro e preço na tabela livro. Veja na Tela 4.2.5 ;

. Agora, quando digitado o código fonte SQL:

```
EXEC Procedure_valor_livro "107"
```

. Tem como resultado a Tela 4.2.6. Onde aparece o nome do livro numa coluna chamada livro, preço na coluna chamad preço. Os demais códigos estão comentados com “--”.

```

use db_biblioteca
--ALTER PROCEDURE Procedure_valor_livro
--(@id INT)
--trazer o nome do livro numa coluna chamada livro, preco na coluna chamad preco
--@id INT
--@livro NVARCHAR(50)
--@preco DECIMAL(10,2)
--IMPRENTA AL LIVROS
exec Procedure_valor_livro "107"

```

Tela 4.2.6

NOTA

Na janela de execução T-SQL iremos utilizar o comando “CREATE PROCEDURES” e podemos utilizar CREATE DEFAULT, CREATE RULE, CREATE TRIGGER e CREATE VIEW.

Sintaxe:

```
CREATE PROCEDURES nome_da_procedure  
[ (@nome_parametro tipo_de_dados_do_parametro) [ =  
valor_padrao] [OUTPUT]]  
AS  
comando 1  
comando 2  
comando n....
```

Vamos utilizar o banco de dados “NwindAcces” ele vem instalado com Microsoft Acces com dados inclusos.

1. Abra o Management Studio depois o banco de dados ;
2. Clique com o botão direito do mouse sob o banco de dados NwindAcces, selecione New Query;
3. Com janela T-SQL aberta. Execute:

```
USE NwindAcces  
GO  
CREATE PROCEDURES usp_pedidosdobrasil AS  
SELECT * FROM pedidos  
WHERE paisdedestino = 'Brasil'
```

ATENÇÃO: use aspas simples ''

1.3 - Revisando Procedures e Funções

“1. Que funções agregadas Yan pode usar para calcular o número total de CDs no banco de dados da coleção de CDs?

- a. SUM(column name)
- b. COUNT(column name)
- c. AVG(column name)

2. Yan não tem muita certeza sobre como funções escalares funcionam. Qual das opções abaixo é uma função escalar?

- a. FIRST(column name) retorna o primeiro campo da coluna especificada
- b. SUM(column name) retorna o total de todos os valores da coluna
- c. UCASE(column name) retorna o valor do campo todo em letras maiúsculas

3. Como um procedimento armazenado é chamado? a. RUN (procedure name, input values)

- b. EXECUTE (procedure name, input values) c. PERFORM (procedure name, input values)”

Respostas

1. A função agregada que Yan pode usar para calcular o número total de CDs no banco de dados da coleção de CDs é:
b. COUNT(column name)

2. O exemplo abaixo é uma função escalar:
c. UCASE(column name) retorna o valor do campo todo em letras maiúsculas

3. Um procedimento armazenado é chamado com: b. EXECUTE (procedure name, input values)

NOTA:

- FunçõesFunções são comandos de uma palavra que retornam um valor único escrito no conjunto de comandos do SQL.
- Procedimentos armazenadosProcedimentos armazenados são grupos pré-compilados de instruções SQL salvas no banco de dados.”⁹²

Exemplos de funções MIN,MAX,EXITS,SUM:

-- Utilizando subquery, exiba uma lista com os nomes dos cursos disponibilizados pela FocoSite informando para cada curso qual o seu menor valor de venda já praticado.

SELECT CURSO, (SELECT MIN(VALOR) FROM PEDIDO_DETALHE WHERE PEDIDO_DETALHE.CURSO = CURSO.CODIGO) AS MENOR_VALOR FROM CURSO;

92 Ibidem

-- Utilizando subquery e o parâmetro IN, exiba os nomes dos cursos disponibilizados pela FocoSite cujo tipo de curso seja 'Programação'.

SELECT CURSO FROM CURSO WHERE TIPO IN (SELECT CODIGO FROM TIPO WHERE TIPO = 'Programação');

-- Utilizando subquery e o parâmetro EXISTS, exiba novamente os nomes dos cursos disponibilizados pela FocoSite cujo tipo de curso

seja 'Programação'.

```
-  
SELECT CURSO FROM CURSO WHERE EXISTS (SELECT CODIGO  
FROM TIPO WHERE TIPO.CODIGO = CURSO.TIPO AND TIPO.TIPO =  
'Programação');
```

```
-- Exiba uma lista com os nomes dos instrutores da FocoSite e ao  
lado o total acumulado das vendas referente aos cursos pelo qual o  
instrutor é responsável.
```

```
-  
SELECT INSTRUTOR, (SELECT SUM(PEDIDO_DETALHE.VALOR)  
FROM PEDIDO_DETALHE INNER JOIN CURSO  
ON PEDIDO_DETALHE.CURSO = CURSO.CODIGO  
WHERE CURSO.INSTRUTOR = INSTRUTOR.CODIGO) AS  
TOTAL_DE_VENDAS FROM INSTRUTOR;
```

```
-- Crie uma visão que exiba os nomes dos alunos e quanto cada um já  
comprou em cursos
```

```
-  
CREATE VIEW ALUNOS_E_COMPRA AS SELECT ALUNO, (SELECT  
SUM(PEDIDO_DETALHE.VALOR) FROM PEDIDO_DETALHE  
INNER JOIN PEDIDO ON PEDIDO_DETALHE.PEDIDO =  
PEDIDO.CODIGO WHERE PEDIDO.ALUNO = ALUNO.CODIGO) AS  
TOTAL_DE_COMPRA FROM ALUNO;
```

```
-- Selecione os nomes de todos os alunos que já fizeram alguma  
matrícula na FocoSite, sem repetição.
```

```
-  
SELECT DISTINCT(ALUNO.ALUNO) FROM PEDIDO INNER JOIN  
ALUNO ON PEDIDO.ALUNO = ALUNO.CODIGO;
```

```
-- Exiba o nome do aluno mais antigo da FocoSite
```

```
SELECT DISTINCT(ALUNO.ALUNO) FROM PEDIDO INNER JOIN  
ALUNO ON PEDIDO.ALUNO = ALUNO.CODIGO ORDER BY  
DATAHORA ASC LIMIT 1;
```

-- Exiba o nome do aluno mais recente da FocoSite

```
SELECT DISTINCT(ALUNO.ALUNO) FROM PEDIDO INNER JOIN  
ALUNO ON PEDIDO.ALUNO = ALUNO.CODIGO ORDER BY  
DATAHORA DESC LIMIT 1;
```

-- Exiba o nome do terceiro aluno mais antigo da FocoSite

```
SELECT DISTINCT(ALUNO.ALUNO) FROM PEDIDO INNER JOIN  
ALUNO ON PEDIDO.ALUNO = ALUNO.CODIGO ORDER BY  
DATAHORA ASC LIMIT 1 OFFSET 2;
```

-- Exiba a quantidade de cursos que já foram vendidos pela FocoSite

-

```
SELECT COUNT(*) FROM PEDIDO_DETALHE;
```

-- Exiba o valor total já arrecadado pelos cursos vendidos pela
FocoSite

```
SELECT SUM(VALOR) FROM PEDIDO_DETALHE;
```

-- Exiba o valor médio cobrado por curso para o pedido cujo CODIGO
é 2

```
SELECT AVG(VALOR) FROM PEDIDO_DETALHE WHERE PEDIDO =  
2;
```

-- Exiba o valor do curso mais caro da FocoSite

-

```
SELECT MAX(VALOR) FROM CURSO;
```

-
-- Exiba o valor do curso mais barato da FocoSite

-

```
SELECT MIN(VALOR) FROM CURSO;
```

-
-- Exiba o valor total de cada pedido realizado na FocoSite

-

```
SELECT PEDIDO, SUM(VALOR) FROM PEDIDO_DETALHE GROUP BY  
PEDIDO;
```

-
-- Exiba os nomes dos instrutores da FocoSite e a quantidade de
cursos que cada um tem sob sua responsabilidade

-

```
SELECT INSTRUTOR.INSTRUTOR, COUNT(*) FROM CURSO INNER  
JOIN INSTRUTOR ON CURSO.INSTRUTOR = INSTRUTOR.CODIGO
```

```
GROUP BY INSTRUTOR;
```

-
-- Exiba o número do pedido, nome do aluno e valor para todos os
pedidos realizados na FocoSite cujo valor total sejam maiores que
500

-

```
SELECT PEDIDO, ALUNO.ALUNO, SUM(VALOR) FROM  
PEDIDO_DETALHE INNER JOIN PEDIDO ON  
PEDIDO_DETALHE.PEDIDO = PEDIDO.CODIGO INNER JOIN ALUNO  
ON PEDIDO.ALUNO = ALUNO.CODIGO GROUP BY PEDIDO HAVING  
SUM(VALOR) > 500;
```

-
-- Exiba o número do pedido, nome do aluno e quantos cursos foram
comprados no pedido para todos os pedidos realizados na FocoSite
que compraram dois ou mais cursos.

-

```
SELECT PEDIDO, ALUNO.ALUNO, COUNT(*) FROM  
PEDIDO_DETALHE INNER JOIN PEDIDO ON
```

```
PEDIDO_DETALHE.PEDIDO = PEDIDO.CODIGO INNER JOIN ALUNO  
ON PEDIDO.ALUNO = ALUNO.CODIGO GROUP BY PEDIDO HAVING  
COUNT(*) > 1;
```

-
-- Exiba o nome e endereço de todos os alunos que morem em
Avenidas (Av.)

```
SELECT ALUNO, ENDERECO FROM ALUNO WHERE ENDERECO  
LIKE 'Av%';
```

-
-- Exiba os nomes dos cursos de Java da FocoSite

```
SELECT CURSO FROM CURSO WHERE CURSO LIKE '%java%';
```

1.4 - Regras e Prática Recomendada com Stored Procedures ⁹³

A equipe de Suporte da Microsoft em SQL Server, detectou algumas regras para uso correto de Procedures, pois o mau uso desse recurso pode sobrecarregar o processamento do Servidor, portanto vejamos o que eles avisam:

93 MSDN, [https://msdn.microsoft.com/pt-br/library/ms187926\(v=sql.120\).aspx](https://msdn.microsoft.com/pt-br/library/ms187926(v=sql.120).aspx), 11\10\2015

. “ Use a instrução SET NOCOUNT ON como a primeira instrução no corpo do procedimento. Ou seja, coloque-a logo após a palavra-chave AS. Isso desativa as mensagens que o SQL Server envia ao cliente após a execução de qualquer instrução SELECT, INSERT, UPDATE, MERGE e DELETE. O desempenho global do banco de dados e do aplicativo melhora ao eliminar essa sobrecarga de rede desnecessária. Para obter informações, consulte SET NOCOUNT (TransactSQL).

. Use nomes de esquemas ao criar ou referenciar objetos de banco de dados no procedimento. Isso exigirá menos tempo de processamento para o Mecanismo de Banco de Dados resolver nomes de objetos se ele não precisar pesquisar vários esquemas. Além disso, evitara problemas de acesso e permissão causados pelo

esquema padrão de um usuário sendo atribuído quando são criados objetos sem a especificação do esquema;

. Evite ajustar funções ao redor de colunas especificadas nas cláusulas WHERE e JOIN. Isso torna as colunas não determinísticas e impede o processador de consultas de usar índices;

. Evite usar funções escalares em instruções SELECT que retornam muitas linhas de dados. Como a função escalar deve ser aplicada a cada linha, o comportamento resultante é como o processamento baseado em linha e afeta o desempenho;

. Evite o uso de SELECT *. Em vez disso, especifique os nomes de colunas necessários. Isso pode evitar alguns erros do Mecanismo de Banco de Dados que param a execução do procedimento. Por exemplo, uma instrução SELECT * que retorna dados de uma tabela de 12 colunas e, em seguida, insere os dados em uma tabela temporária de 12 colunas terá êxito até o número ou a ordem das colunas mudar em uma das tabelas;

. Evite processar ou retornar dados em excesso. Delimite os resultados o quanto antes no código do procedimento, para que

quaisquer operações subsequentes executadas pelo procedimento sejam efetuadas com o menor conjunto de dados possível. Envie apenas os dados essenciais ao aplicativo cliente. Além disso, enviar somente os dados essenciais ao aplicativo cliente é mais eficiente do que enviar dados adicionais pela rede e forçar o aplicativo cliente a trabalhar com conjuntos de resultados desnecessariamente grandes.”

NOTA:

-

-- Baseado na CONTA_BANCARIA, construa uma transação que atualiza o valor aplicado em cada conta bancária baseado no índice de correção percentual de 3% (Primary Key);

-

-- Cria a tabela CONTA_BANCARIA

-

**CREATE TABLE CONTA_BANCARIA (
CÓDIGO INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, --
Código interno (PK) TITULAR VARCHAR(32) NOT NULL, -- Nome do**

titular da conta SALDO DOUBLE NOT NULL, -- Representa o saldo da conta PRIMARY KEY(CODIGO) -- Define o campo CODIGO como PK

```
INSERT INTO CONTA_BANCARIA VALUES (1, 'André', 213); INSERT  
INTO CONTA_BANCARIA VALUES (2, 'Diogo', 489); INSERT INTO  
CONTA_BANCARIA VALUES (3, 'Rafael', 568); INSERT INTO  
CONTA_BANCARIA VALUES (4, 'Carlos', 712); INSERT INTO  
CONTA_BANCARIA VALUES (5, 'Peter', -38);
```

```
START TRANSACTION;  
UPDATE CONTA_BANCARIA SET SALDO = SALDO + (SALDO / 100 *  
3) WHERE SALDO > 0; COMMIT;
```

CAPÍTULO 8

1.1 - Como obter todas as tabelas com ou sem restrição de chave primária no SQL Server?⁹⁴

Podemos escrever uma consulta como abaixo para obter todas as tabelas com nenhuma restrição de chave primária:

```
SELECT T.name 'Table without Primary Key'  
FROM SYS.Tables T  
WHERE OBJECTPROPERTY(object_id,'TableHasPrimaryKey') = 0
```

AND type = 'U'

Podemos escrever uma consulta como abaixo para obter todas as tabelas com restrição de chave primária:

```
SELECT T.name 'Table with Primary Key'  
FROM SYS.Tables T  
WHERE OBJECTPROPERTY(object_id,'TableHasPrimaryKey') = 1
```

AND type = 'U'

94 Basavaraj Biradar , <http://sqlhints.com/category/sql-server/tipstricks/>, 15/10/2105

Vamos entender isso com exemplo:

```
CREATE DATABASE SqlHintsDemoDB  
GO  
USE SqlHintsDemoDB  
GO
```

```

/*Let us create Customers table with Primary Key.*/ CREATE TABLE
dbo.Customers (

CustomerId int IDENTITY (1, 1)
PRIMARY KEY CLUSTERED NOT NULL ,
FirstName Varchar(50),
LastName Varchar(50))
GO

/*Let us create Orders Table without any primary key.*/
CREATE TABLE dbo.Orders (
OrderId int IDENTITY (1, 1) NOT NULL ,
CustomerId int NOT NULL ,
CreationDT DATETIME NOT NULL)

GO

```

1.2 - PRINT / mensagens instrução SELECT dentro de loop while ou lote

de declaração Executando é exibida imediatamente depois que é Sql Server

Você está enfrentando o problema em que as mensagens de impressão / instruções SELECT não estão sendo exibidas como a da explicado no cenário abaixo dois de? Vamos passar por estes cenário da e também ver como resolvê-los.

[LEIA TAMBÉM] while em Sql Server

PROBLEMA: PRINT / valores instrução SELECT dentro de um loop while não são exibidos quando eles são executados em vez disso, são exibidas no final do loop while.

SET NOCOUNT ON

GO

DECLARE @iteration AS INT

SET @iteration = 1

WHILE(@iteration<=10)

BEGIN

```
SELECT 'Start of Iteration ' + CAST(@iteration AS VARCHAR)
WAITFOR DELAY '00:00:01'
PRINT 'End Of Iteration ' + CAST(@iteration AS VARCHAR)
SET @iteration = @iteration + 1

END
```

No script acima estamos passando por 10 iterações e cada iteração tem atraso de 1 segundo. E durante cada iteração estamos exibindo mensagens usando PRINT e instruções SELECT. Aqui se observarmos a execução do script acima, todas as mensagens impressão / instrução SELECT são exibidas no final do ciclo ou seja, depois de 10 segundos em vez de ser exibido durante cada iteração (ou seja, quando ele é executado).

Como mostrado na imagem abaixo mesmo após 8 segundos de execução do script acima nem mesmo uma única mensagem de do PRINT / SELECT são exibidos.

PRINT_AND_SELECT_MESSAGES_ARE_NOT_DISPLAYED

SOLUÇÃO: A razão pela qual a impressão / Mensagens instrução SELECT dentro de um lote não são exibidos é porque o SQL Server armazena os dados de saída.

A seguir estão os dois maneira de resolver esta questão. Aqui eu estou usando o RAISERROR COM NOWAIT, e nível de gravidade é 10, que é suficiente para o SQL Server para considerá-lo como uma mensagem em vez de ser um erro. RAISERROR instrução WITH NOWAIT libera todas as mensagens em buffer.

Observação isso funciona bem para os primeiros 500 RAISERROR COM NOWAIT declaração única de. Depois que o SQL Server inicia rubor 50 mensagens por vez.

```
SET NOCOUNT ON
GO
DECLARE @iteration AS INT,@message as VARCHAR(100)
SET @iteration = 1
WHILE(@iteration<=10)
BEGIN

SET @message = 'Start of Iteration ' + CAST(@iteration AS
VARCHAR) RAISERROR(@message, 10,1) WITH NOWAIT
WAITFOR DELAY '00:00:01'
SET @message = 'End Of Iteration ' + CAST(@iteration AS
```

```

VARCHAR) RAISERROR( @message, 10,1) WITH NOWAIT
SET @iteration+=1

END
----- OR
SET NOCOUNT ON
GO
DECLARE @iteration AS INT
SET @iteration = 1
WHILE(@iteration<=10)
BEGIN

SELECT 'Start of Iteration ' + CAST(@iteration AS VARCHAR)
WAITFOR DELAY '00:00:01'
PRINT 'End Of Iteration ' + CAST(@iteration AS VARCHAR) SET
@iteration+=1

RAISERROR("",10,1) WITH NOWAIT END
RESULTADO: Abaixo imagem demonstra que as mensagens são
exibidos como e quando a declaração correspondente são
executadas:
PRINT_AND_SELECT_MESSAGES_ARE_NOT_DISPLAYED_SOLUTI

```

ON

PROBLEMA: PRINT / valores instrução SELECT dentro de um lote de declarações não são exibidos quando eles são executados em vez disso, são exibidas no final da execução do lote.

```

PRINT 'PRINT MESSAGE' SELECT 'SELECT MESSAGE' WAITFOR
DELAY '00:00:10' GO

```

Como mostrado na imagem abaixo mesmo após 8 segundos de execução do script acima do PRINT / selecione Mensagens não são exibidos.

PRINT_AND_SELECT_MESSAGES_ARE_NOT_DISPLAYED_1

SOLUÇÃO:

```

PRINT 'PRINT MESSAGE'
SELECT 'SELECT MESSAGE'
RAISERROR("",10,1) WITH NOWAIT
WAITFOR DELAY '00:00:10' GO
----- OR - ou você pode optar por esse outro código
RAISERROR('PRINT MESSAGE',10,1) WITH NOWAIT
RAISERROR('SELECT MESSAGE',10,1) WITH NOWAIT
WAITFOR DELAY '00:00:10' GO

```

1.3 Como encontrar todas as tabelas com nenhum índice em tudo no SQL Server?

Podemos escrever uma consulta como abaixo para obter todas as tabelas no banco de dados que não tem nenhum índices:

```
SELECT Name 'Tables without any Indexes'  
FROM SYS.tables  
WHERE OBJECTPROPERTY(OBJECT_ID,'TableHasIndex')=0
```

Vamos entender isso com um exemplo:

```
CREATE DATABASE SqlHintsDemoDB  
GO  
USE SqlHintsDemoDB  
GO  
/*Let us create Customers table with Clustered  
and Non-Clustered Indexes.*/  
CREATE TABLE dbo.Customers (  
CustomerId int IDENTITY (1, 1) PRIMARY KEY CLUSTERED NOT  
NULL,  
FirstName Varchar(50),  
LastName Varchar(50))
```

```
GO  
CREATE NONCLUSTERED INDEX IX_Customers  
ON dbo.Customers(FirstName,LastName)  
GO  
/*Let us create Orders Table without any indexes.*/  
CREATE TABLE dbo.Orders (  
OrderId int IDENTITY (1, 1) NOT NULL ,  
CustomerId int NOT NULL ,  
CreationDT DATETIME NOT NULL) GO
```

1.4 - A função MID ()⁹⁵

A função MID () é usado para extrair caracteres a partir de um campo de texto.

SQL MID () Sintaxe

```
SELECT MID(column_name,start[,length]) AS some_name FROM  
table_name;
```

SQL MID () Exemplo

A seguinte instrução SQL seleciona os quatro primeiros caracteres da coluna "Cidade" da tabela "Clientes":

Exemplo

```
SELECT MID(City,1,4) AS ShortCity  
FROM Customers; 95 http://www.w3schools.com/sql/default.asp,  
16/10/2015
```

Resultado:

Cidade	Berl
Méxi	
Méxi	
Lond	
Lule	
Mann Stra	
Madr	

1.5 - A função LEN ()⁹⁶

A função LEN () retorna o comprimento do valor em um campo de texto.

SQL LEN () Sintaxe

```
SELECT LEN(column_name) FROM table_name; 96 http://www.w3schools.com/sql/default.asp, 16/10/2015
```

SQL LEN () Exemplo

A instrução SQL a seguir seleciona o "CustomerName" e o comprimento dos valores na coluna "Endereço" da tabela "Clientes":

Exemplo

```
SELECT CustomerName,LEN(Address) as LengthOfAddress FROM  
Customers;
```

Resultado:

NOME	TAMANHO
Alfreds	Futterkiste
Ana Trujillo Emparedados y helados	29
Antonio Moreno Taquería	14
Around the Horn	15
Berglunds snabbköp	14
Blauer See Delikatessen	14

1.6 - A função Format ()⁹⁷

A função Format () é usado para formatar como um campo a ser exibido.

SQL FORMATO () Sintaxe

```
SELECT FORMAT(column_name,format) FROM table_name;
```

SQL FORMATO () Exemplo

A instrução SQL a seguir seleciona o nome do produto e preço para hoje (formatada como AAAA-MM-DD) do "produtos" da tabela:

Exemplo

```
SELECT ProductName, Price, FORMAT(Now(), 'YYYY-MM-DD') AS  
PerDate  
FROM Products; 97 http://www.w3schools.com/sql/default.asp,  
16/10/2015
```

Resultado:

Data	Produto	Preço
2015-10-15	Chais	18
2015-10-15	Aniseed Syrup	10
Chef Anton's Cajun		22

1.7 - A função GROUP BY ()⁹⁸

As funções de agregação muitas vezes precisam de um GROUP BY adicionado comunicado.

A instrução GROUP BY é utilizada em conjunto com as funções

de agregação para o grupo por uma ou mais colunas conjunto de resultados.

Sintaxe SQL GROUP BY

98 <http://www.w3schools.com/sql/default.asp>, 16/10/2015

```
SELECT column_name, aggregate_function(column_name) FROM  
table_name  
WHERE column_name operator value  
GROUP BY column_name;
```

Base de dados de demonstração

Neste tutorial, vamos utilizar o banco de dados exemplo bem conhecido Adamastor.

Abaixo está uma seleção da tabela "Orders":

SQL GROUP BY Exemplo

Agora queremos encontrar o número de pedidos enviados por cada transportadora.

A seguinte instrução SQL conta como ordens agrupadas por carregadores:

Exemplo:

```

SELECT      Shippers.ShipperName,COUNT(Orders.OrderID)      AS
NumberOfOrders   FROM Orders  LEFT JOIN Shippers  ON
Orders.ShipperID=Shippers.ShipperID
GROUP BY ShipperName;

```

Resultado:

Número de registros: 3

ShipperName	NumberOfOrders
Federal Shipping	68
Speedy Express	54
United Package	74

1.8 - Função AVG()

A função AVG () retorna o valor médio de uma coluna numérica.

SQL AVG () Sintaxe

SELECT AVG(column_name) FROM table_name

SQL AVG () Exemplo:

A instrução SQL a seguir obtém o valor médio da coluna "preço" da "produtos" da tabela:

Exemplo

```

SELECT      ProductName,      Price      FROM      Products      WHERE
Price>(SELECT AVG(Price) FROM Products);

```

Resultado:

A função AVG () retorna o valor médio de uma coluna numérica.

Código SQL:

SQL AVG () Sintaxe SELECT AVG(column_name) FROM table_name

Resultado:

ProductName	Price
Uncle Bob's Organic Dried Pears	30
Northwoods Cranberry Sauce	40
Mishi Kobe Niku	97
Ikura	31
Queso Manchego La Pastora	38
Alice Mutton	39
Carnarvon Tigers	62.5

Tabela Produtos:

Abaixo está uma seleção dos "produtos" da tabela:

ProductID	ProductName	SupplierID
CategoryID	Unit Price	

1 Chais 1 1 10 boxes 18 2 Chang 1 1 24
- 12ml 19 3 Aniseed Syrup 1 2 12 - 550ml 10 4 Chef Anton's Cajun 2
2 48 - 6 21.35 5 Chef Anton's Gumbo Mix 2 2 36 boxes 25

REFERÊNCIAS

- . SQLhints;
- . Equipe Microsoft MSDN;
- . Equipe Microsoft Thechnet;
- . Microsoft SQL Server 2014;
- . SQL Management Studio;
- . SQL Server Configuration;
- . dbForge Studio for SQL Server;
- . Pinale Dave, site SQLAuthority;
- . Charles Gagnonv – connect.bMicrosoft.com;
- . Felipe Giovananelo, Site Profissionais de TI;
- . Eduardo Javarotti, Site Administradores;
- . Edmar Ferreira, Site Startupco;
- . Renato Siqueira, Site Renatomsiqueira;
- . Ângela Silva, Site Angmaximo;

- . Luiz Mercante, Site SQLdicas;
- . Alexandre Curvel, Site Guia DBA;
- . SoftBlue – Curso Online SQL Completo;
- . Agência Digital FocoSite;
- . FocoSite, Curso SQL Completo;
- . Ângelo M. Almeida Santos, Programando na WEB com ASP Clássico, Ed. Viena;
- . Youtube, Curso Triggers;
- . Júlio Battisti, SQL Server 2005 Curso Completo, Ed. Axcel;
- . Suporte Microsoft, Site suporte.microsoft.com;
- . Bóson Treinamento;
- . Google Trends;
- . W3schools;
- . Certiport;
- . Wikipedia;
- .

GLOSSÁRIO

- . DDL - Data Definition Languagem: Linguagem de Definição de

Dados

- . DML - Data Manipulation Languagem: Linguagem de Manipulação de Dados
- . DQL - Data Query Languagem: Linguagem de Consulta de Dados
- . DCL - Data Control Languagem: Linguagem de Controle de Dados
- . DTL - Data Transaction Languagem: Linguagem de Transação de Dados
- . I/O (Input e Output, entrada e saída)

- . Heap - é uma estrutura de dados organizada como árvore binária balanceada, seguindo algumas regras. Este pode ser implementado em um arranjo, para que ele seja acessado como uma árvore binária, operações:

usando as seguintes

PAI(i) return i/2; ESQUERDA(i) return i*2; DIREITA(i) return i*2 + 1

- . Data warehouse é um depósito de dados digitais que serve para armazenar informações detalhadas relativamente a uma empresa, criando e organizando relatórios através de históricos que são depois usados pela empresa para ajudar a tomar decisões importantes com base nos fatos apresentados.

ANEXOS

Anexo I – Erro de conexão com banco de dados

SQL SERVER - Fix: erro: 40 - não foi possível abrir uma conexão com o servidor SQL.

“Ocorreu um erro ao estabelecer uma conexão com o servidor ao conectar ao servidor SQL 2005, essa falha pode ser causada pelo fato de que sob as configurações padrão SQL Server não permite conexão remota. (Provider: Named Pipes Provider, erro: 40 - não foi possível abrir uma conexão com o servidor SQL.)

Fix / Solução / Solução:

Passo 1) Certifique-se o SQL Server está-se a instância e você tentar se conectar está em execução.
Passo 2) Seu sistema de Firewall não deve bloquear a porta SQL Server. Passo 3) Vá para Gerenciamento do Computador >> Serviço e Aplicação >> SQL Server 2005 Configuração >> Configuração de Rede

Ativar protocolo TCP / IP. Certifique-se de que a porta SQL SERVER é, por padrão 1433. Passo 4) Agora siga este artigo KB do MSDN dependendo do seu servidor: <http://support.microsoft.com/default.aspx?scid=kb;EN-US;914277>

UPDATE: Se a solução acima não ajudar a remeter o acompanhamento pós SQL SERVER - Fix: Erro: 1326 não é possível conectar ao banco de dados do servidor de erro: 40 - não foi possível abrir uma ligação ao SQL Server onde problemas adicionais com o Firewall é explicado com imagens⁹⁹.

Anexo II – Erro no SQL Management Studio

Erro de inicialização do SQL Management Studio solicitando que execute na linha de comando do prompt c:\ssms.exe\resetsettings

Procedimentos Erro SQL Management Studio:

Este erro geralmente ocorre devido a um problema com os quadros de comando VS.

A desinstalação / reinstalação normal deve corrigi-lo, mas uma vez que não é você poderia fazer o seguinte:

1. Desinstalar SSMS (no mínimo, se você é capaz de desinstalar

todos os componentes do SQL na máquina por favor);

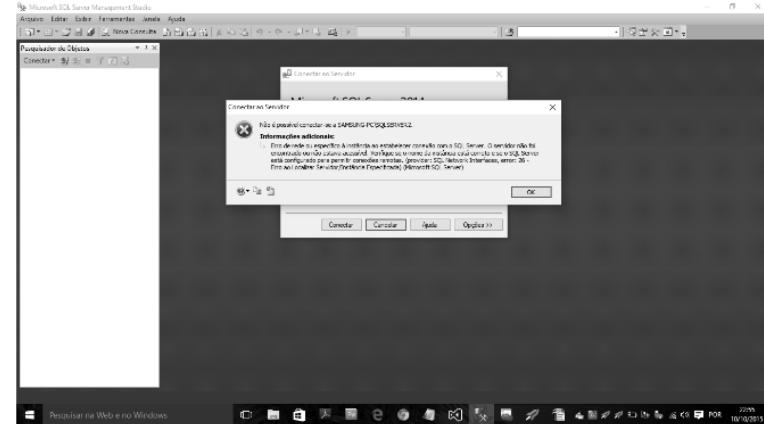
99 Pinal Dave (<http://blog.SQLAuthority.com>), 10\11\2015

- 2. Verifique se o "C: \ Arquivos de Programas (x86) \ Microsoft SQL Server \ 110 \ Tools" pasta está desaparecido. Se ele ainda está em torno de excluí-lo manualmente;**
- 3. Navegue até "C: \ Users \ <usuário> \ AppData \ Roaming \ Microsoft \ SQL Server Management Studio" e excluir a pasta 11,0, onde <usuário> é o nome de usuário da conta. Faça isso para ambos os usuários (sua conta normal e um privilegiado);**
- 4. Desinstalar o "Microsoft Visual Studio 2010 Shell (isolado)" programa 5;**
- 5. Execute o seguinte comando, substituindo <usuário> com uma das contas de usuário executar / user: <USUÁRIO> regedit.exe 6;**
- 6. Navegue até "HKEY_CURRENT_USER \ Software \ Microsoft" e eliminar a chave de "SQL Server Management Studio" se ele existir 7;**
- 7. Sair regedit 8. Repita os passos 5-7 para a outra conta 9;**

8. Reinstalar SSMS Obrigado, Charles Gagnon
(chgagnon@microsoft.com)¹⁰⁰

100 - Charles Gagnon, Procedimentos Erros Server Management Studio,
<https://connect.microsoft.com/SQLServer/feedback/details/795182/ssms-2012-will-not-openusing-run-as>, Microsoft Connect, 09/10/2015

Anexo III – Erro de Instância Parada no Server Configuration Manager



. Para corrigir esse erro precisamos abrir o SQL Server Configuration Manager:

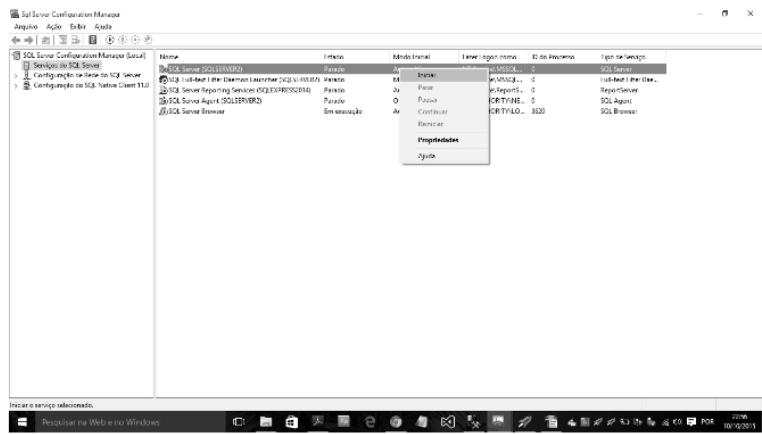


Imagen 7.7

- . Com o botão direito do mouse sobre a Instância SQLSERVER2 clique em “iniciar”. Aguarde finalizar a inicialização da Instância;

- . Retorno ao SQL Management Studio 2014 e agora clique no botão “conectar” o Servidor SQLSERVER2. Se tudo estiver ok abre-se o Banco de Dados “db_Biblioteca”;

ANEXO IV - Introdução ao API Sistema Windows

Reserve um momento para assistir esses vídeos a fim de familiarizar com alguns termos que iremos utilizar:

Instalação de Servidores

[Link do vídeo https://technet.microsoft.com/pt-br/hh925584](https://technet.microsoft.com/pt-br/hh925584)

Armazenamento

[Link do vídeo http://technet.microsoft.com/pt-br/hh925588](http://technet.microsoft.com/pt-br/hh925588)

Gerenciamento de Servidores

[Link do vídeo http://technet.microsoft.com/pt-br/hh925587](http://technet.microsoft.com/pt-br/hh925587)

Serviços

[Link do video http://technet.microsoft.com/pt-br/hh925586](http://technet.microsoft.com/pt-br/hh925586)

Gerenciador de Dispositivos

[Link do vídeo http://technet.microsoft.com/pt-br/hh925585](http://technet.microsoft.com/pt-br/hh925585)

Active Directory

[Link do vídeo http://www.microsoftvirtualacademy.com/trainingcourses/introducao-ao-active-directory](http://www.microsoftvirtualacademy.com/trainingcourses/introducao-ao-active-directory)

DNS

[Link do vídeo: http://technet.microsoft.com/pt-br/hh925592](http://technet.microsoft.com/pt-br/hh925592)

Troubleshooting

Link do vídeo <http://technet.microsoft.com/pt-br/hh925589>

Backup Link do vídeo <http://technet.microsoft.com/pt-br/hh925591>"

ANEXO V – ALTER TABLE e CREATE TRIGGERCREATE TRIGGER¹⁰¹

Categoria

Adicionando colunas e restrições

Descartando colunas e restrições

Alterando uma definição de coluna

Alterando uma definição de tabela

Desabilitando e

habilitando restrições e gatilhos

Elementos de sintaxe em destaque

ADD · PRIMARY KEY com opções de índice · colunas esparsas e conjuntos de colunas ·

DROP

alterar o tipo de dados · alterar o tamanho da coluna · agrupamento
DATA_COMPRESSION · SWITCH PARTITION · LOCK ESCALATION ·
controle de alterações
CHECK · NO CHECK · ENABLE TRIGGER · DISABLE TRIGGER

Adicionando colunas e restrições

Os exemplos desta seção demonstram a adição de colunas e restrições em uma tabela.

A. Adicionando uma nova coluna

O exemplo a seguir adiciona uma coluna que permite valores nulos e que não tem nenhum valor fornecido por uma definição DEFAULT. Na nova coluna, cada linha terá NULL.

101MSDN, <https://msdn.microsoft.com/pt-br/library/ms190273.aspx?f=255&MSPPError=-2147217396#add, 15/10/2015>

```
CREATE TABLE dbo.doc_exa (column_a INT);
GO
ALTER TABLE dbo.doc_exa ADD column_b VARCHAR(20) NULL; GO
```

B. Adicionando uma coluna com uma restrição

O exemplo a seguir adiciona uma nova coluna com uma restrição UNIQUE.

```
CREATE TABLE dbo.doc_exc (column_a INT);
GO
ALTER TABLE dbo.doc_exc ADD column_b VARCHAR(20) NULL
```

```
CONSTRAINT exb_unique UNIQUE;
```

```
GO
```

```
EXEC sp_help doc_exc;
```

```
GO
```

```
DROP TABLE dbo.doc_exc;
```

```
GO
```

C. Adicionando uma restrição CHECK não verificada a uma coluna existente O exemplo a seguir adiciona uma restrição a uma coluna existente na tabela. A coluna tem um valor que viola a restrição.

Portanto, WITH NOCHECK é usado para evitar que a restrição seja validada contra as linhas existentes e para permitir que a restrição seja adicionada.

```
CREATE TABLE dbo.doc_exd ( column_a INT ) ; GO
INSERT INTO dbo.doc_exd VALUES (-1);
GO
ALTER TABLE dbo.doc_exd WITH NOCHECK ADD CONSTRAINT
exd_check CHECK (column_a > 1) ; GO
EXEC sp_help doc_exd ;
GO
DROP TABLE dbo.doc_exd ;
GO
```

D. Adicionando uma restrição DEFAULT a uma coluna existente

O exemplo a seguir cria uma tabela de duas colunas e insere um valor na primeira coluna, sendo que a outra permanece NULL. Depois, uma restrição DEFAULT é adicionada à segunda coluna. Para verificar se o padrão está aplicado, outro valor é inserido na primeira coluna e a tabela é consultada.

```

CREATE TABLE dbo.doc_exz ( column_a INT, column_b INT ) ; GO
INSERT INTO dbo.doc_exz (column_a)VALUES ( 7 ) ; GO
ALTER TABLE dbo.doc_exz
ADD CONSTRAINT col_b_def
DEFAULT 50 FOR column_b ;
GO
INSERT INTO dbo.doc_exz (column_a) VALUES ( 10 ) ; GO
SELECT * FROM dbo.doc_exz ;
GO
DROP TABLE dbo.doc_exz ;
GO

```

E. Adicionando várias colunas com restrições

O exemplo a seguir adiciona várias colunas com restrições definidas com a nova coluna. A primeira coluna nova tem uma propriedade IDENTITY. Cada linha na tabela tem novos valores com incremento na coluna de identidade.

```

CREATE TABLE dbo.doc_exe ( column_a INT CONSTRAINT
column_a_un UNIQUE ) ; GO
ALTER TABLE dbo.doc_exe ADD

```

```

-- Adiciona a chave primária PRIMARY KEY identificando a coluna
column_b INT IDENTITY
CONSTRAINT column_b_pk PRIMARY KEY,

```

-- Add a column that references another column in the same table.

```

column_c INT NULL
CONSTRAINT column_c_fk
REFERENCES doc_exe(column_a),

```

-- Add a column with a constraint to enforce that
-- nonnull data is in a valid telephone number format.

```

column_d VARCHAR(16) NULL
CONSTRAINT column_d_chk
CHECK
(column_d LIKE '[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]' OR column_d
LIKE
'([0-9][0-9][0-9]) [0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]'),
-- Add a nonnull column with a default. column_e DECIMAL(3,3)
CONSTRAINT column_e_default DEFAULT .081 ;

```

```
GO  
EXEC sp_help doc_exe;  
GO  
DROP TABLE dbo.doc_exe; GO
```

F.Adicionando uma coluna que permite valor nulo com valores padrão

O exemplo a seguir adiciona uma coluna que permite valor nulo com uma definição DEFAULT e usa WITH VALUES para fornecer valores para cada linha existente na tabela. Se WITH VALUES não for usado, cada linha terá o valor NULL na nova coluna.

```
CREATE TABLE dbo.doc_exf( column_a INT); GO  
INSERT INTO dbo.doc_exf VALUES (1); GO  
ALTER TABLE dbo.doc_exf  
ADD AddDate smalldatetime NULL  
CONSTRAINT AddDateDflt  
DEFAULT GETDATE() WITH VALUES ;  
GO  
DROP TABLE dbo.doc_exf;  
GO
```

G.Criando uma restrição PRIMARY KEY com opções de índice
O exemplo a seguir cria a restrição PRIMARY

KEY PK_TransactionHistoryArchive_TransactionID e especifica as opções FILLFACTOR, ONLINE e PAD_INDEX. O índice clusterizado resultante terá o mesmo nome da restrição.
Aplica-se a: do SQL Server 2008 ao SQL Server 2014.

```
USE AdventureWorks2012;  
GO  
ALTER TABLE Production.TransactionHistoryArchive  
WITH NOCHECK ADD CONSTRAINT  
PK_TransactionHistoryArchive_TransactionID PRIMARY KEY  
CLUSTERED (TransactionID)  
WITH (FILLFACTOR = 75, ONLINE = ON, PAD_INDEX = ON);  
GO
```

H.Adicionando uma coluna esparsa
Os exemplos a seguir mostram a adição e modificação de colunas esparsas na tabela T1. O código para criar a tabela T1 é o seguinte:

```
CREATE TABLE T1  
(C1 int PRIMARY KEY,  
C2 varchar(50) SPARSE NULL, C3 int SPARSE NULL,  
C4 int );  
GO
```

Para adicionar uma outra coluna esparsa C5, execute a seguinte instrução:

```
ALTER TABLE T1  
ADD C5 char(100) SPARSE NULL ;  
GO
```

Para converter a coluna não esparsa C4 a uma coluna esparsa, execute a seguinte instrução:

```
ALTER TABLE T1  
ALTER COLUMN C4 ADD SPARSE ;  
GO
```

Para converter a coluna esparsa C4 a uma coluna não esparsa, execute a seguinte instrução:

```
ALTER TABLE T1  
ALTER COLUMN C4 DROP SPARSE; GO
```

I. Adicionando um conjunto de colunas

Os exemplos a seguir mostram a adição de uma coluna à tabela T2. Um conjunto de colunas não poderá ser adicionado a uma tabela se ela já contiver colunas esparsas. O código para criar a tabela T2 é o seguinte:

```
CREATE TABLE T2 (C1 int PRIMARY KEY, C2 varchar(50) NULL, C3  
int NULL,  
C4 int );  
GO
```

As três instruções a seguir adicionam um conjunto de colunas chamado CS e, depois, modificam colunas C2 e C3 para SPARSE.

```
ALTER TABLE T2  
ADD CS XML COLUMN_SET FOR ALL_SPARSE_COLUMNS ; GO
```

```
ALTER TABLE T2  
ALTER COLUMN C2 ADD SPARSE ; GO
```

```
ALTER TABLE T2  
ALTER COLUMN C3 ADD SPARSE ; GO
```

Descartando colunas e restrições

Os exemplos desta seção demonstram o descarte de colunas e restrições.

A. Descartando uma coluna ou colunas

O primeiro exemplo modifica uma tabela para remover uma coluna.

O segundo exemplo remove várias colunas.

```
CREATE TABLE dbo.doc_exb  
(column_a INT  
,column_b VARCHAR(20) NULL  
,column_c datetime  
,column_d int);  
  
GO  
-- Remove a single column.  
ALTER TABLE dbo.doc_exb DROP COLUMN column_b;  
GO  
-- Remove multiple columns.  
ALTER TABLE dbo.doc_exb DROP COLUMN column_c, column_d;
```

B. Descartando restrições e colunas

O primeiro exemplo remove uma restrição UNIQUE de uma tabela.

O segundo exemplo remove duas restrições e uma única coluna.

```
CREATE TABLE dbo.doc_exc ( column_a int NOT NULL  
CONSTRAINT my_constraint UNIQUE);  
GO  
  
-- Example 1. Remove a single constraint. ALTER TABLE  
dbo.doc_exc DROP my_constraint; GO  
  
DROP TABLE dbo.doc_exc; GO  
  
CREATE TABLE dbo.doc_exc ( column_a int  
NOT NULL CONSTRAINT my_constraint UNIQUE  
,column_b int  
NOT NULL CONSTRAINT my_pk_constraint PRIMARY KEY);  
GO  
  
-- Example 2. Remove two constraints and one column  
-- The keyword CONSTRAINT is optional. The keyword COLUMN is  
required. ALTER TABLE dbo.doc_exc
```

```
DROP      CONSTRAINT      CONSTRAINT      my_constraint,
my_pk_constraint, COLUMN column_b ;
GO
```

C.Descartando uma restrição PRIMARY KEY no modo ONLINE
O exemplo a seguir exclui uma restrição PRIMARY KEY com a opção
ONLINE definida como ON.

```
ALTER TABLE Production.TransactionHistoryArchive
DROP          CONSTRAINT
PK_TransactionHistoryArchive_TransactionDate WITH (ONLINE =
ON);
GO
```

D.Adicionando e descartando uma restrição FOREIGN KEY

O exemplo a seguir cria a tabela ContactBackup e, em seguida, altera
a tabela, adicionando uma restrição FOREIGN KEY que referencia
a tabela Person.Person e, depois, descartando a restrição FOREIGN
KEY.

```
CREATE TABLE Person.ContactBackup (ContactID int);
```

```
GO
ALTER TABLE Person.ContactBackup
ADD CONSTRAINT FK_ContactBacup_Contact FOREIGN KEY
(ContactID) REFERENCES Person.Person (BusinessEntityID);
GO
```

```
ALTER TABLE Person.ContactBackup
DROP CONSTRAINT FK_ContactBacup_Contact ; GO
```

DROP TABLE Person.ContactBackup ;

 [Início]

Alterando uma definição de coluna

A.Alteração do tipo de dados de uma coluna

O exemplo a seguir altera uma coluna de uma tabela de INT para DECIMAL.

```
CREATE TABLE dbo.doc_exy (column_a INT );
GO
INSERT INTO dbo.doc_exy (column_a) VALUES (10);
GO
ALTER TABLE dbo.doc_exy ALTER COLUMN column_a DECIMAL (5,
2) ; GO
DROP TABLE dbo.doc_exy ;
GO
```

B.Alterando o tamanho de uma coluna

O exemplo a seguir aumenta o tamanho de uma coluna varchar varchar e a precisão e escala de uma coluna decimaldecimal. Como essas colunas contêm dados, o tamanho da coluna só pode ser aumentado. Além disso, observe que col_a está definido como um índice exclusivo. O tamanho de col_a ainda pode ser aumentado, pois o tipo de dados é umvarcharvarchar e o índice não é o resultado

de uma restrição PRIMARY KEY.

```
IF OBJECT_ID ( 'dbo.doc_exy', 'U' ) IS NOT NULL
DROP TABLE dbo.doc_exy;
GO
-- Create a two-column table with a unique index on the varchar
column.
CREATE TABLE dbo.doc_exy ( col_a varchar(5) UNIQUE NOT NULL,
col_b decimal
(4,2));
GO
INSERT INTO dbo.doc_exy VALUES ('Test', 99.99);
GO
-- Verify the current column size.
SELECT    name,    TYPE_NAME(system_type_id),    max_length,
precision, scale
FROM sys.columns WHERE object_id = OBJECT_ID(N'dbo.doc_exy');
GO
-- Increase the size of the varchar column.
ALTER TABLE dbo.doc_exy ALTER COLUMN col_a varchar(25);
GO
```

```
-- Increase the scale and precision of the decimal column.
ALTER TABLE dbo.doc_exy ALTER COLUMN col_b decimal (10,4);
GO
-- Insert a new row.
INSERT INTO dbo.doc_exy VALUES ('MyNewColumnSize',
99999.9999') GO
-- Verify the current column size.
SELECT name, TYPE_NAME(system_type_id), max_length,
precision, scale FROM sys.columns WHERE object_id =
OBJECT_ID(N'dbo.doc_exy');
```

C.Alterando o agrupamento de colunas

Os exemplos a seguir mostram como alterar o agrupamento de uma coluna. Primeiro, uma tabela é criada com o agrupamento de usuário padrão.

```
CREATE TABLE T3 (C1 int PRIMARY KEY, C2 varchar(50) NULL, C3
int NULL,
C4 int );
GO
```

Em seguida, o agrupamento da coluna C2 é alterado para

Latin1_General_BIN. Observe que o tipo de dados é obrigatório, mesmo que não tenha sido alterado.

ALTER TABLE T3

```
ALTER COLUMN C2 varchar(50) COLLATE Latin1_General_BIN; GO
```

Alterando uma definição de tabela

Os exemplos desta seção demonstram como alterar a definição de uma tabela.

A.Modificando uma tabela para alterar a compactação

O exemplo a seguir altera a compactação de uma tabela não particionada. O heap ou índice clusterizado será recriado. Se a tabela for um heap, todos os índices não clusterizados serão recriados.

ALTER TABLE T1

```
REBUILD WITH (DATA_COMPRESSION = PAGE);
```

O exemplo a seguir altera a compactação de uma tabela particionada. A sintaxe REBUILD PARTITION = 1 faz com que

somente o número de partição 1 seja recriado.
Aplica-se a: do SQL Server 2008 ao SQL Server
2014.

```
ALTER TABLE PartitionTable1  
REBUILD PARTITION = 1 WITH (DATA_COMPRESSION = NONE);  
GO
```

A mesma operação usando a sintaxe alternada a seguir faz com que todas as partições na tabela sejam recriadas.

Aplica-se a: do SQL Server 2008 ao SQL Server 2014.

```
ALTER TABLE PartitionTable1  
REBUILD PARTITION = ALL  
WITH (DATA_COMPRESSION = PAGE ON PARTITIONS(1));
```

Para obter exemplos adicionais de compactação de dados, consulte Compactação de dados.

B.Modificando uma tabela columnstore para alterar a compactação de arquivamento O exemplo a seguir compacta ainda mais uma partição de tabela columnstore aplicando um algoritmo de compactação adicional. Isso reduz a tabela para um

tamanho menor, mas também aumenta o tempo necessário para armazenamento e recuperação. Isso pode ser útil para fins de arquivamento, ou em outras situações que exijam menos espaço e possam dispensar mais tempo para armazenamento e recuperação.

Aplica-se a: do SQL Server 2014 ao SQL Server
2014.

```
ALTER TABLE PartitionTable1  
REBUILD PARTITION = 1 WITH (DATA_COMPRESSION =  
COLUMNSTORE_ARCHIVE) GO
```

O exemplo a seguir descompacta uma partição de tabela columnstore compactada com a opção COLUMNSTORE_ARCHIVE. Quando os dados forem restaurados, eles continuarão sendo compactados através da compactação columnstore usada em todas as tabelas columnstore.

Aplica-se a: do SQL Server 2014 ao SQL Server 2014.

```
ALTER TABLE PartitionTable1  
REBUILD PARTITION = 1 WITH (DATA_COMPRESSION =  
COLUMNSTORE) GO
```

C.Alterando partições entre tabelas

O exemplo a seguir cria uma tabela particionada, pressupondo que o esquema de partição myRangePS1 já esteja criado no banco de dados. Em seguida, uma tabela não particionada é criada com a mesma estrutura de uma tabela particionada e no mesmo grupo de arquivos que PARTITION 2 da tabela PartitionTable. Depois, os dados daPARTITION 2 da tabela PartitionTable são inseridos na tabela NonPartitionTable.

```
CREATE TABLE PartitionTable (col1 int, col2 char(10)) ON  
myRangePS1 (col1);  
GO  
CREATE TABLE NonPartitionTable (col1 int, col2 char(10))  
ON test2fg ;  
GO  
ALTER TABLE PartitionTable SWITCH PARTITION 2 TO  
NonPartitionTable ; GO
```

D.Permitindo escalonamento de bloqueios em tabelas particionadas

O exemplo a seguir habilita o escalonamento de bloqueios no nível de partição em uma tabela particionada. Se a tabela não estiver particionada, o escalonamento de bloqueios será definido no nível TABLE.

Aplica-se a: do SQL Server 2008 ao SQL Server 2014.

```
ALTER TABLE dbo.T1 SET (LOCK_ESCALATION = AUTO);
```

GO

E.Configurando o controle de alterações em uma tabela

O exemplo a seguir habilita o controle de alterações na tabela Person.Person. Aplica-se a: do SQL Server 2008 ao SQL Server 2014.

```
USE AdventureWorks2012; ALTER TABLE Person.Person  
ENABLE CHANGE_TRACKING;
```

O exemplo a seguir habilita o controle de alterações e também o controle de colunas que são atualizadas durante uma alteração.

Aplica-se a: do SQL Server 2008 ao SQL Server 2014.

```
USE AdventureWorks2012;
GO
ALTER TABLE Person.Person
ENABLE CHANGE_TRACKING
WITH (TRACK_COLUMNS_UPDATED = ON)
```

O exemplo a seguir desabilita o controle de alterações na tabela Person.Person. Aplica-se a: do SQL Server 2008 ao SQL Server 2014.

```
USE AdventureWorks2012; Go
ALTER TABLE Person.Person DISABLE CHANGE_TRACKING;
```

Desabilitando e habilitando restrições e gatilhos

A. Desabilitando e reabilitando uma restrição

O exemplo a seguir desabilita uma restrição que limita os salários aceitos nos dados. NOCHECK CONSTRAINT é usada com ALTER TABLE para desabilitar a restrição e permitir uma inserção que normalmente violaria a restrição. CHECK

CONSTRAINT reabilita a restrição.

```
CREATE TABLE dbo.cnst_example
(id INT NOT NULL,
name VARCHAR(10) NOT NULL,
salary MONEY NOT NULL

CONSTRAINT salary_cap CHECK (salary < 100000);

-- Valid inserts
INSERT INTO dbo.cnst_example VALUES (1,'Joe Brown',65000);
INSERT INTO dbo.cnst_example VALUES (2,'Mary Smith',75000);

-- This insert violates the constraint.
INSERT INTO dbo.cnst_example VALUES (3,'Pat Jones',105000);

-- Disable the constraint and try again.
ALTER TABLE dbo.cnst_example NOCHECK CONSTRAINT
salary_cap; INSERT INTO dbo.cnst_example VALUES (3,'Pat
Jones',105000);
```

```
-- Re-enable the constraint and try another insert; this will fail.  
ALTER TABLE dbo.cnst_example CHECK CONSTRAINT salary_cap;  
INSERT INTO dbo.cnst_example VALUES (4,'Eric James',110000);
```

B.Desabilitando e reabilitando um gatilho

O exemplo a seguir usa a opção DISABLE TRIGGER de ALTER TABLE para desabilitar o gatilho e permitir uma inserção que normalmente violaria o gatilho. ENABLE TRIGGER é usado para reabilitar o gatilho.

```
CREATE TABLE dbo.trig_example  
(id INT,  
name VARCHAR(12),  
salary MONEY);  
GO  
-- Create the trigger.  
CREATE TRIGGER dbo.trig1 ON dbo.trig_example FOR INSERT AS  
IF (SELECT COUNT(*) FROM INSERTED  
WHERE salary > 100000) > 0  
BEGIN
```

```
print 'TRIG1 Error: you attempted to insert a salary > $100,000'  
  
ROLLBACK TRANSACTION  
END ;  
GO  
-- Try an insert that violates the trigger.  
INSERT INTO dbo.trig_example VALUES (1,'Pat Smith',100001) ; GO  
-- Disable the trigger.  
ALTER TABLE dbo.trig_example DISABLE TRIGGER trig1 ; GO  
-- Try an insert that would typically violate the trigger. INSERT  
INTO dbo.trig_example VALUES (2,'Chuck Jones',100001) ; GO  
-- Re-enable the trigger.  
ALTER TABLE dbo.trig_example ENABLE TRIGGER trig1 ; GO  
-- Try an insert that violates the trigger.  
INSERT INTO dbo.trig_example VALUES (3,'Mary Booth',100001) ;  
GO
```

Operações online

A.Recompilação de índice online usando opções de espera de baixa prioridade

O exemplo a seguir mostra como executar uma recompilação de índice online que especifica as opções de espera de baixa prioridade.
Aplica-se a: do SQL Server 2014 ao SQL Server
2014.

```
ALTER TABLE T1 REBUILD WITH (
    PAD_INDEX = ON,
    ONLINE = ON ( WAIT_AT_LOW_PRIORITY ( MAX_DURATION = 4
    MINUTES, ABORT_AFTER_WAIT = BLOCKERS ) )
)
;
AUTOR
```



Ângelo Mário Almeida dos Santos, genuinamente baiano de Capela do Alto Alegre-BA, Técnico em Processamento de Dados desde 1992 e Diretor de Tecnologia na Focosite desde 2005. Atua na área de Webdesigner, Programador WEB com ASP Clássico, Banco de Dados SQL Server e MySQKL, SQL, Jquery. Trabalhou, também, com dBase III Plus, Pascal, Clipper 5.0 hoje xHabour e Visual Basic 6.0 em Programação Desktop

desenvolvendo vários Sistemas Comerciais, Igreja, Ouvidoria, Laboratório Clínico etc. É, também, Consultor de TI e Identidade

Visual já alguns anos dos clientes: www.m2ddecor.com.br , www.sigeo.net.br , www.consultoriadeplanejados.com.br , www.pro3academia.com.br

- . Pós-Técnico em Processamento de Dados por EPDBA-BA, 1992;**
- . Bacharelado em Teologia por Faculdade Unida-ES, 2010;**
- . Especialista no Ensino de Filosofia por Universidade Federal da Bahia, 2015;**
- . Autor de diversos livros na área de Ciências Humanas publicado na Amazon e Clube de Autores.**