



CES/JF

*Centro de Ensino Superior
de Juiz de Fora*

Disciplina

Gerência e Consulta em Banco de Dados

Prof. Jacimar Tavares
jacimar.tavares@gmail.com



Módulo 02



Abordagem Relacional

- **Conceitos Importantes:**

- Modelo:

- É um protótipo ou exemplo que se pretende reproduzir ou imitar.

- Modelo de Dados:

- Modelo que permite demonstrar como serão construídas as estruturas de dados que darão suporte aos processos de negócio, como esses dados estarão organizados e quais os relacionamentos que pretendemos estabelecer entre eles

- Modelo Relacional:

- É um modelo que opera com os dados organizados como um conjunto de **RELAÇÕES**
 - Modelo de Dados Relacional representa o Banco de Dados como uma coleção de tabelas, cada uma das quais associada a um nome único



Abordagem Relacional

- **Conceitos Importantes:**
 - Modelos Vistos:
 - Modelo Conceitual,
 - Modelo Lógico
 - Modelo Físico
 - Se resumem a **Representação Tabular...**
 - Toda relação pode ser vista como uma **TABELA**, onde cada linha é uma **TUPLA** e em cada coluna estão valores de um mesmo domínio.
 - Uma tabela é o formato de apresentação de dados mais entendido universalmente.



Abordagem Relacional

- **Representação Tabular...**

- Relação = TABELA
- Tupla = LINHA
- Atributo = COLUNA

	TABELA		COLUNAS
LINHAS			

FORNECIMENTO

FORNECEDOR	PEÇA	PROJETO	QTDE
3	1	3	46
4	6	1	32
7	2	5	15



Abordagem Relacional

- **Regras de Codd**

- Dr. Edgar Frank Codd, o criador dos bancos de dados relacionais, publicou uma lista de 12 regras que definem os Sistemas de Banco de Dados Relacional (SBDR) na Computerworld, de 14 e 21 de outubro de 1985, com o título ***“Is your DBMS Really Relational?”*** e ***“Does Your DBMS Run by the Rules?”***.
- Motivo:
 - Evitar que os fornecedores comercializassem seus produtos como relacionais, ainda que não estivessem sendo atendidas as regras mínimas dos padrões.
 - Suas regras servem como modelo de referência sobre o que deve ser um Banco de Dados Relacional.



Abordagem Relacional

- Regras de Codd

- 1) Toda informação num banco de dados relacional é apresentada a nível lógico por valores em tabelas;
- 2) Todo dado em um banco de dados relacional tem a garantia de ser logicamente acessível, recorrendo-se a uma combinação do nome da tabela, um valor de chave e o nome da coluna;
- 3) Tratamento sistemático de valores nulos (ausência de dado);
- 4) O dicionário de dados (catálogo) relacional ativo é baseado no modelo relacional;



Abordagem Relacional

- Regras de Codd

5) O SGBD relacional deve ter uma linguagem para definição, detalhamento e manipulação dos dados;

6) Tratamento das atualizações de visões dos dados;

7) Tratamento de alto nível para inserção, atualização e eliminação de dados;

8) Independência dos dados físicos (mudança na memória e no método de acesso);

10) Independência de dados lógicos (mudança de qualquer tipo nas tabelas básicas, ex.: divisão de uma tabela por linha ou coluna);



Abordagem Relacional

- Regras de Codd

10) Independência das restrições de integridade;

11) Independência de distribuição; Os usuários finais e aplicativos não conhecem nem são afetados pela localização dos dados (BD Distribuídos VS. BD Locais)

12) Não subversão das regras de integridade ou restrições quando se utiliza uma linguagem de baixo nível. Se o sistema dá suporte a acesso de baixo nível aos dados, não deve haver um modo de negligenciar as regras de integridade do BD.



Abordagem Relacional

- Propriedades das Relações
 - Cada tabela tem um nome único através do qual é referenciada
 - Cada tabela contém um número fixo de colunas
 - Cada linha da tabela representa uma tupla ou registro da relação
 - Todas as linhas são distintas uma das outras (não existem linhas iguais)
 - A ordem das linhas da tabela é irrelevante
 - Cada coluna representa um atributo mono valorado
 - Cada coluna tem um nome único
 - A ordem das colunas da tabela é irrelevante



Abordagem Relacional

- Falando sobre **Chaves**
 - Conjunto de atributos de uma relação R com a propriedade de que nenhum par de tuplas na instância r de R tem a mesma combinação de valores para aqueles atributos
 - É comum uma relação ter mais de uma **chave**; neste caso, cada uma das chaves é chamada **chave candidata**
 - A **chave candidata** cujos valores são usados para identificar tuplas em uma relação é chamada **chave primária**
 - Se um atributo (ou atributos) chave não-primário em uma relação é **chave primária** em outra relação, então este atributo na primeira relação é chamado de **chave estrangeira**



Abordagem Relacional




- Falando sobre **Chaves**

- **CHAVE CANDIDATA (CANDIDATE KEY-CK)**

- É todo atributo(ou conjunto de atributos) para o qual não pode haver repetição de valores na tabela.
 - Identifica,de maneira inequívoca, uma linha da tabela
 - Toda relação tem pelo menos uma **chave candidata** → a concatenação de todos os atributos (Não existem linhas iguais)

CHAVES CANDIDATAS



NUM_FUNC	NOME_FUNC	DEPTO_FUNC	CPF_FUNC



Abordagem Relacional


- Falando sobre **Chaves**

- **CHAVE PRIMÁRIA (PRIMARY KEY-PK)**

- É a chave candidata **escolhida** para identificar unívocamente uma linha de uma relação (TABELA).
 - A **chave primária** de uma tabela T1, corresponde a uma ou mais colunas de T1 de tal forma que identifique unívocamente as ocorrências (linhas) de T1, ou seja, não há duas ou mais linhas de T1 que tenham o mesmo valor de **chave primária**

CHAVE PRIMÁRIA

CHAVE CANDIDATA NÃO ESCOLHIDA também chamada de CHAVE SECUNDÁRIA ou ALTERNATIVA



NUM_FUNC	NOME_FUNC	DEPTO_FUNC	CPF_FUNC




Abordagem Relacional



- Falando sobre **Chaves**


- **CHAVE ESTRANGEIRA (FOREIGN KEY-FK)** É chave primária de outra tabela; colocada como atributo para mostrar o relacionamento entre tabelas
- A **chave estrangeira** de uma tabela T2, corresponde a uma ou mais colunas de T2 de tal forma que, para **cada valor não nulo** da **chave estrangeira** de T2, **há um valor igual** a uma chave primária da tabela T1 ou da própria T2
- Ex: FUNCIONÁRIO (T2) com DEPARTAMENTO (T1)

CHAVE PRIMÁRIA



COD_DEPTO	NOME_DEPTO	LOCAL_DEPTO
1		
3		

CHAVE ESTRANGEIRA



NUM_FUNC	NOME_FUNC	DEPTO_FUNC	CPF_FUNC
		1	
		3	



Atividade

- **Apresente o Modelo Lógico completo para o minimundo abaixo:**
 - Deseja-se construir um banco de dados de um sistema de informação para controlar a execução de conferências. Após a entrevista inicial foram obtidas as seguintes informações:
 - Um congresso, que tem como informações nome, data e local é composto de diversas sessões;
 - Cada sessão é identificada por um número. Registra-se ainda o local da sessão, horário de início e término da sessão. Dentro de cada sessão são apresentados diversos artigos;
 - Cada artigo tem como informações o título e o número de páginas. Um artigo pode ser escrito por um ou mais autores; Para os autores é necessário armazenar o nome, data de nascimento, e-mail e instituição a qual o mesmo está vinculada. Um autor pode escrever vários artigos;
 - Ainda registram-se os participantes do congresso. Para os participantes deve-se registrar o número de identificação, nome, data de nascimento, telefone e e-mail. Os participantes podem ser de 3 tipos: estudantes, profissionais associados e profissionais não associados. Para cada tipo temos um valor de inscrição diferenciado;
 - Para o participante estudante é emitido um diploma especial de participação, que tem uma numeração específica e data de impressão.





Atividade

- Crie o DTR considerando as informações:

- Cliente

- cod_cliente varchar(20)
- nome varchar(90)
- endereco varchar(120)
- cidade varchar(50)

- Filmes

- cod_filme integer
- cod_categoria integer
- descricao varchar(90)
- valor numeric

- Locacao

- cod_cliente varchar(20)
- cod_filme integer
- data_locacao date

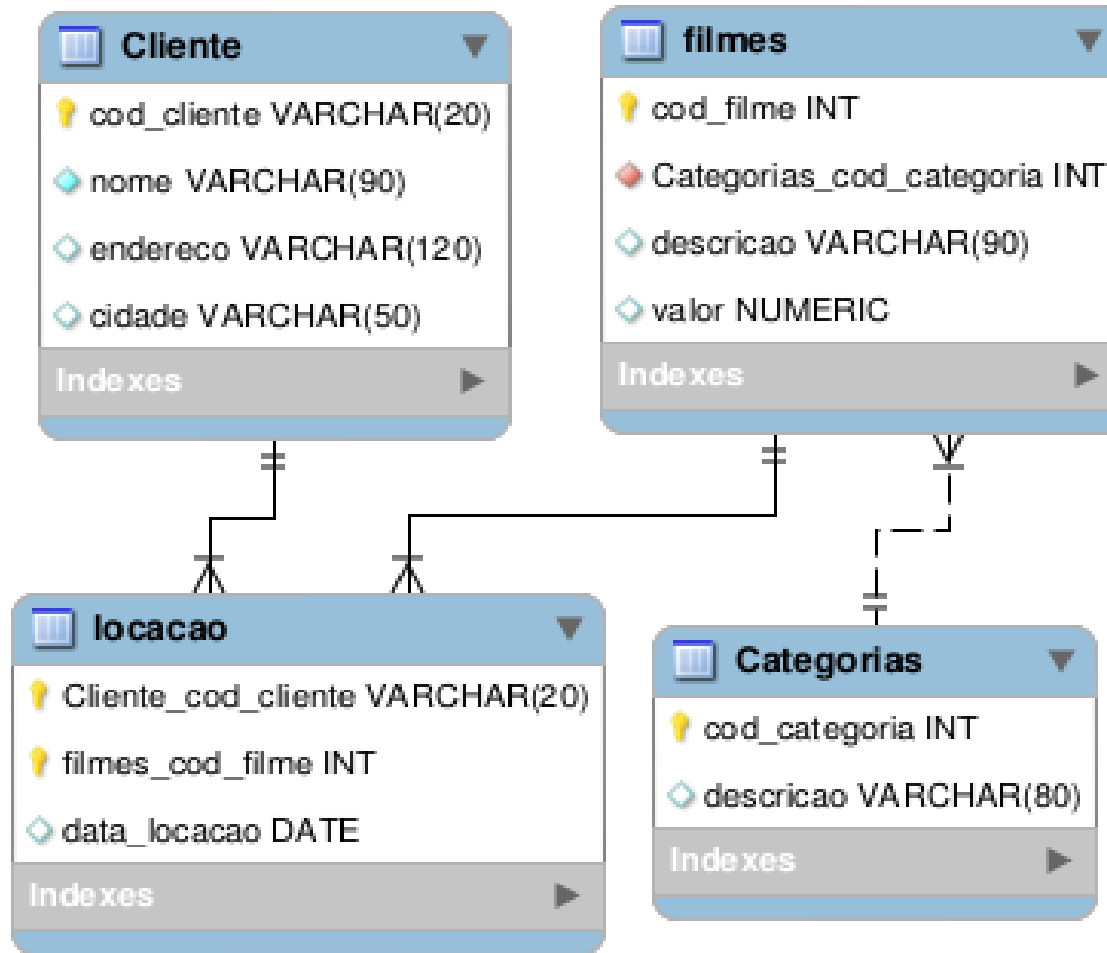
- Categorias

- cod_categoria integer
- descricao varchar(80)





Atividade (Resultado)





Linguagens Relacionais

- **Linguagens de Consulta**

- Permitem ao usuário interagir diretamente com o software de banco de dados, a fim de executar as tarefas de processamento de informações.
- É normalmente uma linguagem de computador fácil de usar, que se baseia em palavras básicas (comandos).
- Usando linguagem de consulta e um teclado de computador, o usuário digita comandos para recuperar dados de uma base de dados ou atualizar de um banco de dados.



Linguagens Relacionais

- **Linguagens de Consulta**
 - Formais:
 - **Álgebra Relacional**
 - **Cálculo Relacional**
 - Comerciais:
 - **SQL : Structured Query Language**
 - QUEL : linguagem de Consulta – INGRES – 1976
 - QBE : Query By Example – IBM - 1975



Linguagens Relacionais

- **Álgebra Relacional**

- Matematicamente falando, uma tabela (relação) é um conjunto de linhas.
- No modelo relacional temos o BD representado como uma coleção de tabelas, quando queremos manipular (recuperar) dados em geral o resultado nos é apresentado como uma tabela, derivada de alguma forma de outras tabelas.
- A álgebra relacional é um conjunto de operações e relações.



Linguagens Relacionais

- **Cálculo Relacional**

- Existem dois tipos:
 - Cálculo Relacional de Tuplas (CRT)
 - Cálculo Relacional de Domínio (CRD).
- No Cálculo Relacional existem variáveis, constantes, operadores lógicos, de comparação e quantificadores.
- As expressões de Cálculo são chamadas de fórmulas.



Linguagens Relacionais

- Cálculo Relacional

FORN	#FN	Fnome	Fcidade	PEÇA	#PN	Pnome	Pcor
	1313	Pavan	J. Fora		P1	pia	branca
	1320	ABC	T.Rios		P2	vaso	bege
	1330	DEF	J. Fora		P3	sifão	branca
	1350	Gil	J. Fora		P4	fio	preto
	1360	Visart	R.Janeiro				

FORNECIMENTO	#FN	#PN	QTDE
	1313	P1	25
	1313	P2	20
	1320	P2	25
	1320	P3	20
	1360	P3	25
	1313	P4	20



Linguagens Relacionais

- **Cálculo Relacional**
 - **Quais os nomes dos fornecedores que fornecem pelo menos 1 peça.**
 - PROJECT FORNECIMENTO OVER FN GIVING R1

a.1)R1	FN
	1313
	1320
	1360



Linguagens Relacionais

- **Cálculo Relacional**

- **Exemplos diversos:**

- 1) Encontre todos os empregados cujos salários estejam acima de R\$3.500,00.
 - $\{t \mid \text{EMPREGADO}(t) \text{ AND } t.\text{SALARIO} > 3500\}$
 - 2) Dê apenas os nomes e sobrenomes dos empregados cujos salários estejam acima de R\$3.500,00.
 - $\{t.\text{NOME}, t.\text{SOBRENOME} \mid \text{EMPREGADO}(t) \text{ AND } t.\text{SALARIO} > 3500\}$



Linguagem SQL

Histórico

- SEQUEL (Structured English Query Language) - Centro de Pesquisa San Jose da IBM (Projeto System R)
- SQL 1 (SQL-86): Padrão ISO/ANSI. Não havia cláusula para especificar chave; modificado em 1989.
- SQL 2 (SQL-92): Implementa conexão cliente/servidor.
- SQL 3 implementa o Modelo Orientado a Objeto.



Linguagem SQL

Structured Query Language

- São linguagens ou ferramentas para acesso aos bancos de dados de forma independente das aplicações. Os acessos obedecem aos direitos de acesso (leitura e gravação) estabelecidos para o usuário.
- Oferecem a oportunidade de apresentação das informações trazidas do banco de dados na tela, em relatório ou gravados em uma mídia de armazenamento.
- Normalmente estes softwares são disponibilizados para usuários finais, que adquirem certa independência em relação à área de Informática no tocante à criação de consultas e relatórios.



Linguagem SQL

Structured Query Language

- Mais que uma linguagem de consulta, a SQL oferece funções para DEFINIÇÃO, MANIPULAÇÃO e CONTROLE dos dados de um Banco de dados.
- A SQL permite a especificação não apenas de um conjunto de relações (tabelas), mas também das
- informações sobre cada tabela, incluindo:
 - O esquema de cada tabela;
 - O domínio de valores de cada atributo;
 - O conjunto de índices a ser mantido para cada tabela;
 - Informações de segurança e autorização de acesso para cada tabela;
 - Restrições de integridade;
 - Etc.



Linguagem SQL

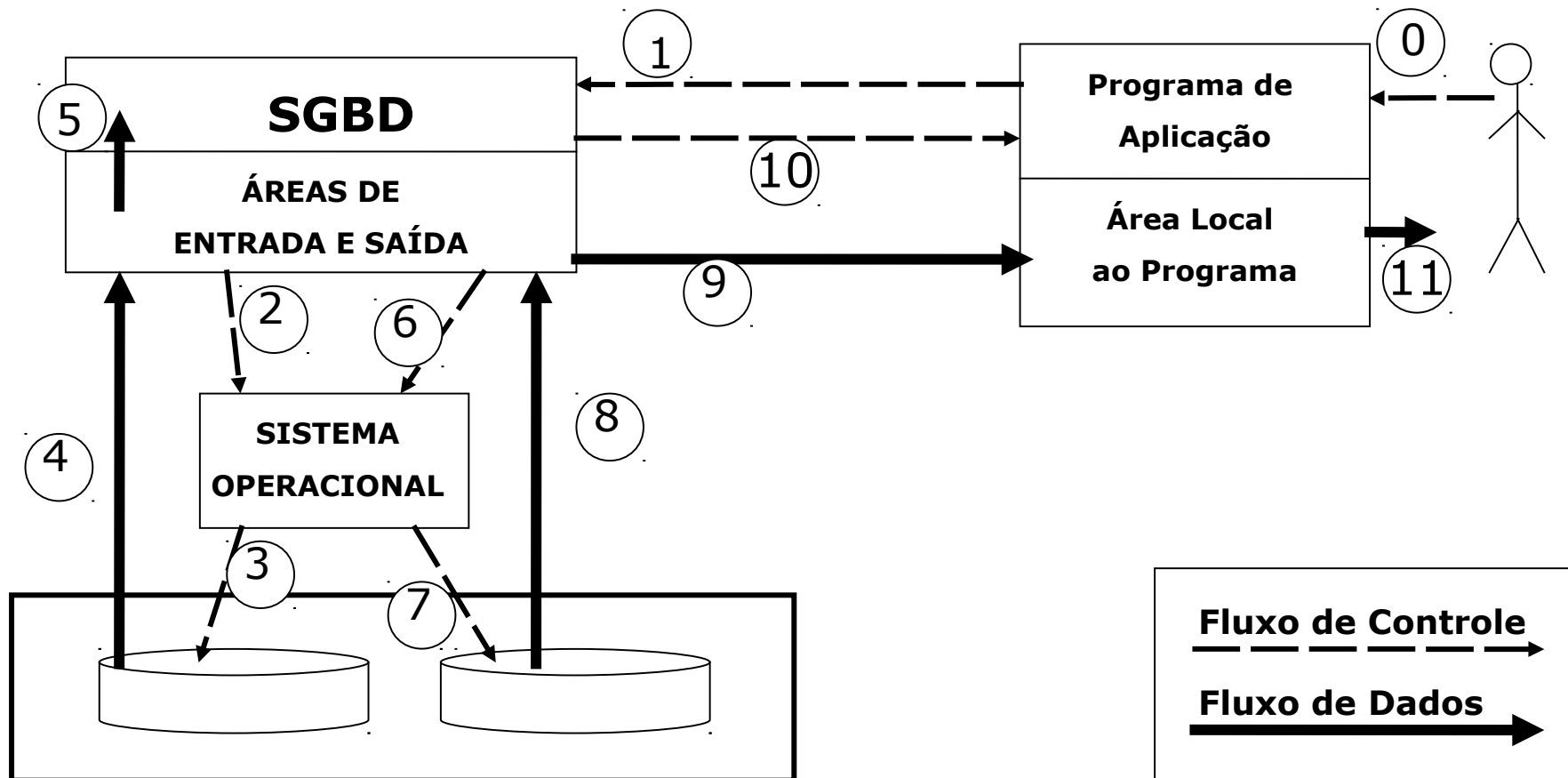
Structured Query Language

- Linguagem como o Inglês;
- Não inclui nenhuma referência a caminhos de acesso explícitos;
- É um modo de recuperar e manipular dados de um Banco de Dados;
- Pode ser usado por um terminal ON-LINE;
- Pode ser usado na forma de “SQL EMBUTIDA” em um programa de aplicação;
- Possui uma série completa de operações de atualização;
- Usado para administração.



Linguagem SQL

Structured Query Language – Processamento de Consulta





Linguagem SQL

Structured Query Language – Processamento de Consulta

- **0 - o usuário submete a consulta**
- **1 - o programa que implementa a transação ou o Processador de Consultas pede ao SGBD um Registro Lógico.**
- **2 - o SGBD pede ao Sistema Operacional a leitura dos esquemas.**
- **3 - o Sistema Operacional acessa os esquemas.**
- **4 - os Meta-dados são transferidos para a área do SGBD.**
- **5 - o SGBD consulta os meta-dados para saber como traduzir os comandos do usuário.**
- **6 - o SGBD emite os comandos para que o Sistema Operacional leia os registros físicos necessários.**



Linguagem SQL

Structured Query Language – Processamento de Consulta

- **7 - o Sistema Operacional acessa a Base de Dados.**
- **8 - os Registros físicos são transferidos para a área de E/S do SGBD.**
- **9 - o SGBD seleciona os dados necessários para formar o(s) registro(s) lógicos pedidos pelo usuário. Se necessário faz transformações, conversões e coloca na área do Usuário.**
- **10 - o SGBD envia ao Programa de Aplicação um código sinalizando final da operação.**
- **11 - o Programa de Aplicação exibe o resultado ao Usuário, ou gera relatório.**



Linguagem SQL

Conceitos usados

- **Tabela (table)**
 - Equivale à relação, no Modelo Relacional.
 - Constituída de Linhas e Colunas
- **Linha (row) – registros**
 - Equivale à tupla, no Modelo Relacional.
 - Análogo a um registro de um arquivo;
 - Todas linhas de uma tabela tem o mesmo conjunto de colunas.
- **Coluna (column)**
 - Equivale aos atributos da relação, no Modelo relacional.
 - Semelhante à campo de um registro;
 - Cada coluna em cada linha tem apenas um conteúdo;
 - Cada coluna é só de um tipo de dados.



Linguagem SQL

Conceitos usados

- **Domínios (domain)**
 - Determina os valores possíveis para colunas, ou seja, é o conjunto de valores válidos para uma determinada coluna.
 - Domínios normalmente são definidos antes da criação das tabelas, para que possam ser usadas por estas.
- **Restrições (constraint)**
 - Condições necessária para manter a integridade dos dados.
- **Visões (view)**
 - É uma “tabela virtual” cujo conteúdo é derivado de tabelas do Banco de Dados.



Linguagem SQL

Conceitos usados

- **Tipos de dados (data type) mais comuns:**
- **Caracter**
 - CHAR(n) ou CHARACTER(n)
 - VARCHAR(n) ou CHARACTER VARYING(n) ou CHAR VARYING(n)
- **Numérico**
 - SMALLINT, INTEGER
 - FLOAT, DOUBLE PRECISION
 - NUMERIC(precision, scale), DECIMAL(precision,scale)
- **Datas**
 - DATE
 - TIME
 - TIMESTAMP
- **BLOB (Binary Large Objects)**



Linguagem SQL

- **SQL apresenta três divisões:**
 - **DDL (Data Definition Language)**
 - CREATE: criação de novas estruturas
 - ALTER: alteração de estruturas existentes
 - DROP: remoção de estruturas
 - **DML (Data Manipulation Language)**
 - INSERT: Inserção de registros
 - DELETE: remoção de registros
 - UPDATE: atualização de registros
 - SELECT: Seleção (consulta) de registros
 - **DCL (Data Control Language)**
 - GRANT: concessão de direitos de acesso a tabelas e visões
 - REVOKE: revogação de direitos de acesso a tabelas e visões



Restrições de Integridade

- **Definição:**

Conjunto de regras que existem para o modelo de dados, assim como um conjunto de regras de negócio, que regem a manipulação do Banco de Dados, de forma a não ferir nenhuma destas regras estabelecidas.





Restrições de Integridade

- **INTEGRIDADE DE ENTIDADE (ou DE CHAVE)**
- Para a **chave primária** ou qualquer parte dela não é permitido valores nulos, ou seja, nenhum atributo participante da **chave primária** de uma relação pode aceitar valor nulo
 - Deve garantir:
 - UNICIDADE - não é permitida a repetição da chave primária
 - MINIMALIDADE - a chave primária deve ser composta do menor número possível de colunas que mantenham a unicidade





Restrições de Integridade

- **INTEGRIDADE DE ATUALIZAÇÃO**

- Refere-se ao processo de garantir a consistência das ocorrências de entidades e relacionamentos, respeitando-se as cardinalidades mínima e máxima.





Restrições de Integridade

- **INTEGRIDADE INERENTE ao NEGÓCIO**

- São regras ditadas pelo negócio e não são mapeadas pelo M.E.R por se tratar de condições especiais. Entretanto, no Banco de Dados é preciso que estas regras sejam obedecidas.

- **Exemplos:**

- Valor mínimo de depósito para abertura de uma conta: R\$10.000,00
- Conta corrente sem movimento a 180 dias será encerrada.





Restrições de Integridade

- **INTEGRIDADE REFERENCIAL**

- “ O conteúdo de um atributo definido como uma **chave estrangeira** de uma tabela deve ser igual a um valor da **chave primária** associada de outra tabela ou ser nulo”.
- A **chave estrangeira** deve ter correspondência com a **chave primária** em outra tabela ou ser nula.
- Em outras palavras, se alguma tupla **t2** referencia alguma tupla **t1**, então a tupla **t1** deve existir.
- A **integridade referencial** deve ser garantida para as operações de inserção, deleção e atualização.

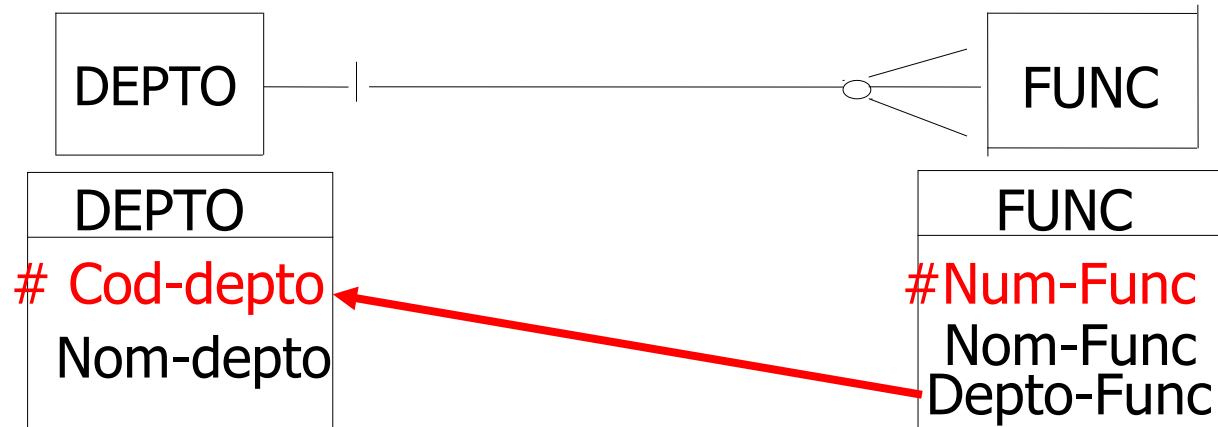




Restrições de Integridade

- **INTEGRIDADE REFERENCIAL**

- Qualquer chave estrangeira em uma tabela deve:
 - 1 - Ou **se referir** a uma chave primária de uma tabela relacionada (os **valores** dos atributos devem ser os mesmos)
 - 2 - Ou **ser nula**(não se relaciona com outra tabela)





Restrições de Integridade

- INTEGRIDADE REFERENCIAL

- Exemplo

DEPTO	<u>NUMDEPTO</u>	DESCRIÇÃO	
	100	Recursos Humanos	
	200	Cobrança
	300	Informática

FUNC	<u>NUMFUNC</u>	NOMEFUNC	<u>DEPTOFUNC</u>
	1100	Ney	100
	8888	Carla	300
	2500	Rodrigo	200

É possível inserir o funcionário 8899? Por que ?

8899	Augusto	250
------	---------	-----	-------



Restrições de Integridade

- **INTEGRIDADE REFERENCIAL DE INSERÇÃO**

- Antes de ser inserida uma nova linha de uma tabela que contém um valor de chave estrangeira, é necessário que já exista uma linha em uma tabela com este valor de chave primária. Caso contrário, a operação de inserção deve ser rejeitada ou uma linha com a chave primária deverá ser, previamente, inserida na respectiva tabela.

- No exemplo anterior:

8899	Augusto	250
------	---------	-----	-------

- É necessário que já exista o depto 250, caso contrário não será permitido a inclusão do novo empregado na tabela correspondente, ou então, deverá ser incluído o depto 250 e depois o empregado 8899 nas respectivas tabelas





Restrições de Integridade

- INTEGRIDADE REFERENCIAL DE DELEÇÃO

- Exemplo

DEPTO	<u>NUMDEPTO</u>	DESCRIÇÃO	
	100	Recursos Humanos	
	200	Cobrança
	300	Informática
FUNC	<u>NUMFUNC</u>	NOMEFUNC	<u>DEPTOFUNC</u>
	1100	Ney	100
	8888	Carla	300
	2500	Rodrigo	200

É possível deletar o departamento 200?





Restrições de Integridade

- **INTEGRIDADE REFERENCIAL DE DELEÇÃO**

- Para operações de **deleção**, também há ações alternativas que possibilitam preservar a consistência dos dados:
 - **1-** Quando uma linha em uma tabela é **deletada**, então, todas as ocorrências de **chave estrangeira** desta chave primária também devem ser deletadas (efeito cascata) ou os valores de **chave estrangeira** devem ser atualizados para **nulos**
 - **2-** A operação de **deleção** pode ser rejeitada se existir uma ocorrência de **chave estrangeira** da chave primária a ser **deletada**
- **Exemplo 1**
 - A deleção em cascata pode ocorrer se um projeto é deletado da tabela de projetos, então todas as linhas de subprojeto devem ser deletados
- **Exemplo 2**
 - Um determinado departamento não pode ser deletado da tabela de departamentos enquanto existir empregados alocados a este departamento



Restrições de Integridade

- **INTEGRIDADE REFERENCIAL DE DELEÇÃO**
 - **Ações possíveis no caso de exclusão de linha:**
 - a) Exclusão automática de todas as linhas que fazem referencia à chave excluída
 - b) Exclusão não permitida para linhas que sejam referenciadas por outras tabelas
 - c) Inserção de valor nulo na chave estrangeira das linhas que referenciam à chave excluída





Restrições de Integridade

- **INTEGRIDADE REFERENCIAL DE DELEÇÃO**

DEPTO	<u>NUMDEPTO</u>	DESCRIÇÃO	
	100	Recursos Humanos	
	200	Cobrança
	300	Informática
FUNC	<u>NUMFUNC</u>	<u>NOMEFUNC</u>	<u>DEPTOFUNC</u>
	1100	Ney	100
	8888	Carla	300
	2500	Rodrigo	200

- Um determinado **DEPTO** mudou o NUM-DEPTO. O que fazer com todos os funcionários alocados neste depto?
 - Exemplo: NUM-DEPTO mudou de 300 para 901 e 902





Restrições de Integridade

- **INTEGRIDADE REFERENCIAL DE DELEÇÃO**

- Um determinado DEPTO mudou o NUM-DEPTO. O que fazer com todos os funcionários alocados neste depto?
 - Exemplo: NUM-DEPTO mudou de 300 para 901 e 902

300	INFORMÁTICA
↓ ?		
901	INFORMÁTICA - TREINAMENTO
902	INFORMÁTICA - DESENVOLVIMENTO





Restrições de Integridade

- **INTEGRIDADE REFERENCIAL DE DELEÇÃO**

- Para a operação de atualização, também há ações alternativas que possibilitam preservar a integridade dos dados
- Se uma **chave primária** é atualizada pode ser necessário:
 - mudar para nulas todas as ocorrências existentes de chave estrangeira com o antigo valor;
 - mudar todas as ocorrências de chave estrangeira do antigo para o novo valor;
 - rejeitar a atualização

- Exemplo:

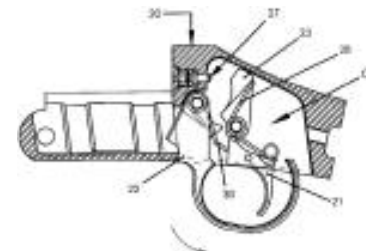
- Um determinado departamento mudou o código do depto. O que fazer com todos os funcionários alocados neste departamento?
 - deixar todos os funcionários deste departamento sem alocação em um departamento
 - alocar todos os funcionários deste departamento no novo departamento, ou
 - rejeitar a atualização, pois, não se pode mudar a identificação do depto enquanto houver funcionários alocados





Restrições de Integridade

- **INTEGRIDADE DE *TRIGGER* (Gatilho)**
 - Um gatilho é um comando executado automaticamente pelo sistema como um efeito de uma modificação no Banco de Dados.
 - Para projetar um *gatilho*, precisamos:
 - Especificar as condições sob as quais o *gatilho* deve ser executado.
 - Especificar as ações a serem tomadas quando o *gatilho* é executado.

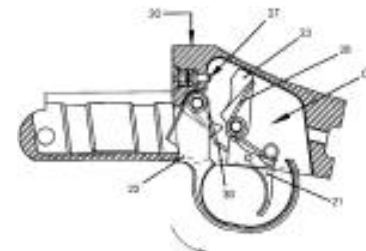




Restrições de Integridade

- **INTEGRIDADE DE TRIGGER (Gatilho)**

- Suponha que em vez de permitir saldos negativos, o banco trate saque a descoberto assim :
 - ajustando o saldo para zero
 - criando um empréstimo no valor da quantia saldo negativo
 - a este empréstimo é dado um número igual ao número da conta estourada.
- A condição para executar o *trigger* é uma atualização na relação depósito que resulte em um valor de saldo negativo





Atividade

- **Apresente situações reais para exemplificar cada um dos tipos de Restrições estudadas:**
 - Chave
 - Atualização
 - Inerente ao Negócio
 - Referencial
 - Inserção
 - Deleção
 - Trigger

