

Banco De Dados Prova

O que é um Sistema?

- Qualquer conjunto de partes unidas si pode ser considerado um sistema, que a relação entre as partes e o comportamento do todo seja foco de atenção
- É algo que tem por finalidade tratar, armazenar e fornecer dadas as informações que possam apoiar os processos e funções realizados em um negócio.

O que é um Informação?

- Dados organizados com significado. Ex: Folha de Pagamento

O que é um Dados?

- Elementos Brutos sem significado. Ex: nomes, horas, salário

O que é um Conhecimento?

- Informação interpretada. Ex: Estudo das faixas de salários

O que é um Sistema de Informação ?

- Um **Sistema de Informação** é um conjunto de **processos inter-relacionados** , sejam eles pessoas, softwares, hardwares, redes de comunicação e dados coletados, que ajudam na **tomada de decisões, administração e metas** de uma empresa. De modo geral, **SI's** juntam processos tecnológicos, pessoas e informações para traçar e alcançar metas, sendo nesse processo extremamente importante a interação do lado social, que inclui, além das **pessoas, documentos e dados**, também, o lado automatizado, responsável por máquinas, **redes e computadores**.
- **Partes** que o compõem:
 - **Hardware/Software**
 - **Pessoas/Procedimentos**
- **Objetivo:**
 - Processar/organizar dados para gerar informações/conhecimento
- **Exemplo: Sistema de controle acadêmico**

O que é um Banco De Dados?

- Conjunto de dados que tem uma estrutura regular e que está organizado de tal forma que um computador pode facilmente encontrar as informações desejadas
- São conjuntos de arquivos relacionados entre si
- **Descrição:**
 - Representa algum aspecto do mundo real
 - Coleção logicamente coerente de dados com algum significado inerente
 - Construído para uma finalidade específica
 - Fatos conhecidos que podem ser registrados e possuem significado implícito

Sistema de Arquivos e Banco de Dados ?

- A relação entre Banco de Dados e Sistema de Arquivos é que o banco de dados guarda todas informações estabelecidas que podem ser estruturadas como um sistema de arquivos

O que é um Sistemas de Arquivos?

- Sistema de arquivo é um conjunto de estruturas lógicas desenvolvidas diretamente via software, que permite ao sistema operacional ter acesso e controlar os dados gravados no disco.

- **Pontos negativos:**
 - Dificuldade de acesso aos dados
 - Redundância e inconsistência
 - Isolamento dos dados
 - Problemas de integridade
 - Problemas de atomicidade
 - Problemas de segurança

O que é um Atomicidade?

- A atomicidade é uma propriedade que garante que cada transação seja tratada como uma entidade única, a qual deve ser executada por completo ou falhar completamente. Desta forma, todas as operações da transação devem ser executadas com sucesso para que a transação tenha sucesso. Se uma única operação que seja do bloco da transação falhar, toda a transação deverá ser cancelada – as transações são aplicadas de uma forma “**tudo ou nada**”.

O que é um Sistema Gerenciador de Banco de Dado(SGBD)?

- É um conjunto de programas e ferramentas utilizadas para configurar, atualizar e manter um banco de dados.
- **Exemplo:**

- Recursos para administrar usuários/permisões
- Recursos para criar/alterar tabelas e banco de dados
- Recursos para backup e restauração de dados
- Recursos para otimizar a performance do banco

- O principal **OBJETIVO** é fornecer uma maneira de recuperar informações de banco de dados que seja tanto conveniente quanto eficiente.
- **Exemplo de SGBD:** IMS, DB2, Ingres, Informix Dynamic Server, Oracle Server, Sybase SQL server, Microsoft SQL Server, Access, MySql, etc.

Vantagens da Abordagem SGBD?

- Representando relacionamentos complexos entre dados
 - Pode incluir muitas variedades de dados que estão inter-relacionados de diversas maneiras
- Impondo restrições de integridade
 - Restrição de **integridade referencial**
 - Cada registro de turma deve estar relacionado a um registro de disciplina
- Restrição de **chave ou singularidade**
 - Cada registro de tabela deve ter um código único
- Oferecendo backup e recuperação
 - Subsistema de backup e recuperação de SGBD é responsável pela recuperação

Classificação do SGBD?

- Quanto ao **número de utilizadores**:
 - **Pessoal** - utilizada em computadores pessoais
 - **Multiutilizadores** - utilizado em estação de trabalho, minicomputadores e máquinas de grande porte
- Quanto ao **número de máquinas** onde a base de dados está armazenada.
 - Outra forma de classificar os SGBD é quanto à arquitetura que engloba o número de máquinas
 - **Centralizada**: Os dados ficam centralizados em um servidor
 - **Descentralizada**: O banco de dados fica em mais de um servidor, permitindo a descentralização dos dados.
 - **Distribuída**: Os dados ficarão distribuídos em vários servidores
 - **Replicado**: O banco de dados é replicado para vários servidores como num processo de espelhamento. Os bancos são iguais e na medida em que o primeiro banco é modificado, os demais também são modificados em cascata.

O que é um Abstração dos Dados?

- Representação conceitual dos dados. Através do Diagrama Entidade Relacionamento; Não inclui detalhes de como os dados são armazenados ou como as operações são implementadas.

Propriedade das transições de banco de dados?

- Tem que ser **ACID** – Atomicidade, Consistência, Isolamento e Durabilidade.
- **Atomicidade**: Todos os efeitos de uma transação são refletidos no banco de dados, ou nenhum deles ocorre
- **Consistência**: A execução da transação leva o banco de dados a um estado consistente
- **Isolamento**: As transações concorrentes são isoladas entre si
- **Durabilidade**: Atualizações de transações confirmadas não são perdidas, mesmo que ocorra falha do sistema.

O que é um Atores?

- Atores são usuários que possuem algum papel em relação ao sistema.

O que é um Modelagem de dados?

- Modelo de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados

O que é um Níveis do Modelo de Dados?

- **Alto nível** ou **modelo de dados conceitual** que fornece uma visão mais próxima do modo como os usuários visualizam os dados realmente

- Representação com **alto nível** de abstração que usa os conceitos de
 - **Entidade** que representa um objeto ou conceito do mundo real
 - Ex, funcionário
 - **Atributo** que representa alguma propriedade de interesse que descreve melhor uma entidade
 - Ex, nome ou o salário do funcionário
 - **Relacionamento** entre duas ou mais entidades representa uma associação entre elas
 - Ex, um relacionamento entre um funcionário e um projeto

entidade, atributo, relacionamento

- **Modelos de dados lógico** que são usados para determinar se o SGBD é relacional, orientado a objetos, hierárquico, etc

- Descrevem os **tipos de entidades lógicas**, os atributos de dados que descrevem essas entidades e os relacionamentos entre as entidades
 - Depende do Banco de Dados
 - Representam os dados em alguma estrutura lógica
 - Indica quais **tabelas** o banco contém e quais os nomes das **colunas**

CLIENTE

Numero do Cliente
CPF
Primeiro Nome
Sobrenome
Forma de Saudacao
Telefone

possui

ENDEREÇO

Rua
Cidade
Estado
Pais
CEP

TipoDeProduto (CodTipoProd, DescrTipoProd)

Produto (CodProd, DescrProd, PrecoProd, CodTipoProd)

CodTipoProd referencia TipoDeProduto

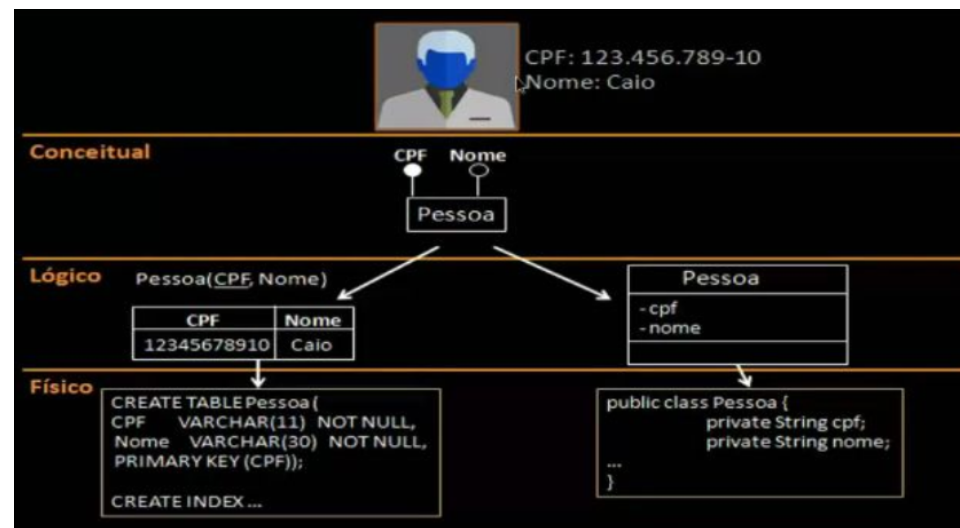
tabelas, colunas

- **Baixo nível** ou **modelo de dados físico**, que fornece uma visão mais detalhada do modo como os dados estão realmente armazenados no computador

- São usados **para projetar o esquema interno de um banco de dados**, descrevendo as tabelas de dados, as colunas de dados das tabelas e o relacionamento entre as tabelas
 - Serve para definir o projeto do banco de dados
 - Define efetivamente de que maneira os dados estão armazenados
 - tipo
 - tamanho do campo
 - restrições
 - etc

```
CREATE TABLE FILME (  
  Codigo Filme INTEGER(6) NOT NULL,  
  Titulo VARCHAR(100),  
  Genero VARCHAR(50),  
  Ano INTEGER(4),  
  Duracao TIME,  
  Diretor VARCHAR(150),  
  FaixaEtaria INTEGER(2) UNSIGNED,  
  PRIMARY KEY(Codigo Filme)  
);
```

Abstração de Dados?



O que é Modelo Entidade-Relacionamento (ER)?

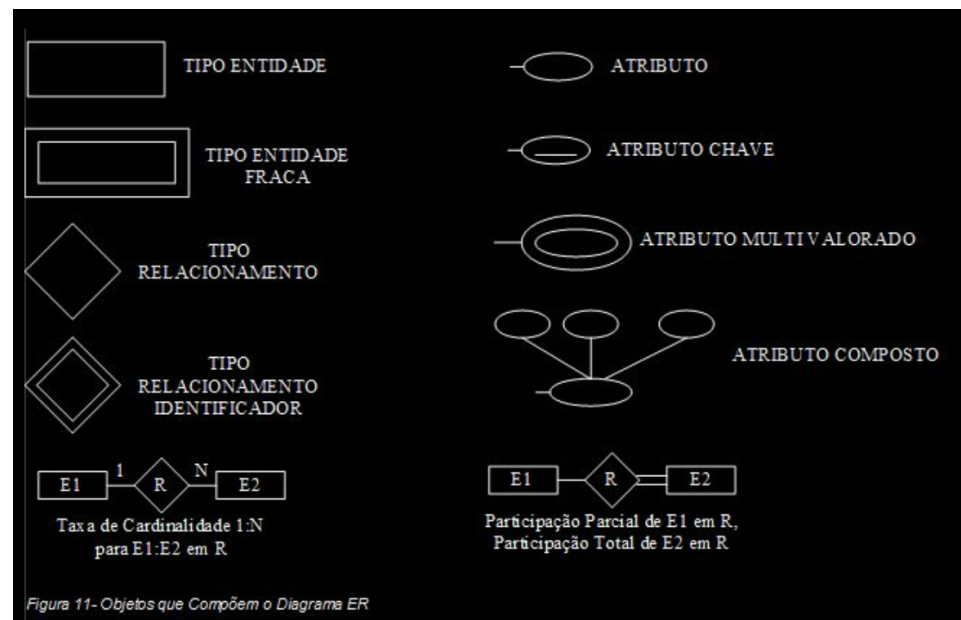
- O modelo Entidade-Relacionamento é o modelo de dados conceitual de **alto nível** ou modelo de dados conceitual
- Esse modelo descreve os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos)

O que é um Diagrama Entidade-Relacionamento?

- O modelo ER pode ficar abstrato demais para auxiliar no desenvolvimento do sistema, uma vez que ele é apresentado em forma de texto
- Por isso, o diagrama é tido muitas vezes como sinônimo de modelo
- O **diagrama** facilita a comunicação entre os integrantes da equipe, pois oferece uma linguagem comum utilizada tanto pelo analista, responsável por levantar os requisitos, e os desenvolvedores, responsáveis por implementar aquilo que foi modelado
- Notação do Diagrama ER:

- As **entidades** representadas por retângulos
- Os **atributos** aparecem listados na própria entidade
 - originalmente como elipses
- Os **relacionamentos** por losangos, ligados às entidades por linhas, contendo a cardinalidade (1..1, 1..n ou n..n)
- **Cardinalidade** é a quantidade máxima de ocorrência de entidades que podem estar associadas
- As **chaves primárias** são representadas por * ou sublinhado

- Notação:



Classificação de Entidades

São classificadas segundo o motivo de sua existência:

- **Entidades fortes** - são aquelas que por si só elas já possuem total sentido de existir.
 - Ex. funcionário
- **Entidades fracas** - são aquelas que dependem de outras entidades para existir
 - Não possui atributo chave
 - Ex.: dependente
- **Entidades associativas** - acontecem quando há um relacionamento do tipo muitos para muitos
 - É necessária a criação de uma entidade intermediária cuja identificação é formada pelas chaves primárias

forte; fraca; associativa

O que é um Atributos?

- Atributos são as características que descrevem cada entidade dentro do domínio
 - Exemplo: nome da pessoa, endereço, função, etc...
- Os atributos podem ser divididos quanto a estrutura:

Atributo composto - é um atributo pode ser dividido em diversas subpartes com significado independente entre si

- Exemplos: nome, endereço, etc...

Atributo simples ou atômico - é um atributo que não pode ser subdividido

- Exemplos: sexo, cpf

Composto; simples ou atômico

- Os atributos podem ser classificados quanto à sua função:

Descritivos - representam característica intrínsecas de uma entidade, tais como nome ou cor

Nominativos - além de serem também descritivos, estes têm a função de definir e identificar um objeto, tais como nome, código ou número

Referenciais - representam a ligação de uma entidade com outra em um relacionamento

- Exemplo: uma venda possui o CPF do cliente, que a relaciona com a entidade cliente

Descritivos; Nominativos; Referenciais.

- Os atributos ainda são divididos em:

Atributo simplesmente valorado

- São atributos que podem assumir apenas um determinado valor em uma determinada instância
 - Ex.: nome

Multi valorado

- São atributo que podem assumir diversos valores em uma mesma instância
 - Ex. Telefone

Atributo derivado

- É um atributo que é gerado a partir de outro atributo e não precisa ser armazenado
 - Ex.: Idade (se no banco tem a data de nascimento)

Simplesmente valorado; Multi valorado; Derivado

- **Atributos referenciais** são chamados de Chave Estrangeira e geralmente estão ligados à chave primária da outra entidade

O que é um Modelo Lógico Relacional?

- Depois de criado o modelo Conceitual, o próximo passo é a criação do modelo lógico
- Para mudamos do modelo conceitual (Diagrama entidade relacionamento) para o modelo lógico relacional, usamos as seguintes regras:

1. Toda entidade vira uma relação (**Tabela**)
2. Atributo identificador se torna **chave primária** na relação
3. Atributos simples se tornam **colunas** (campos)
4. Atributos compostos **tornam-se atributos simples**, mapeados em colunas, uma coluna para cada atributo
5. Atributos derivados **não** são mapeados
6. Atributos multivalorados podem ser mapeados de duas formas:
 - Como **n colunas**, onde n é o número máximo de valores do atributo
 - Criando-se **uma nova relação**

Tabela; Chave primária; Colunas;

- O modelo lógico representa os dados de um banco de dados como uma coleção de relações

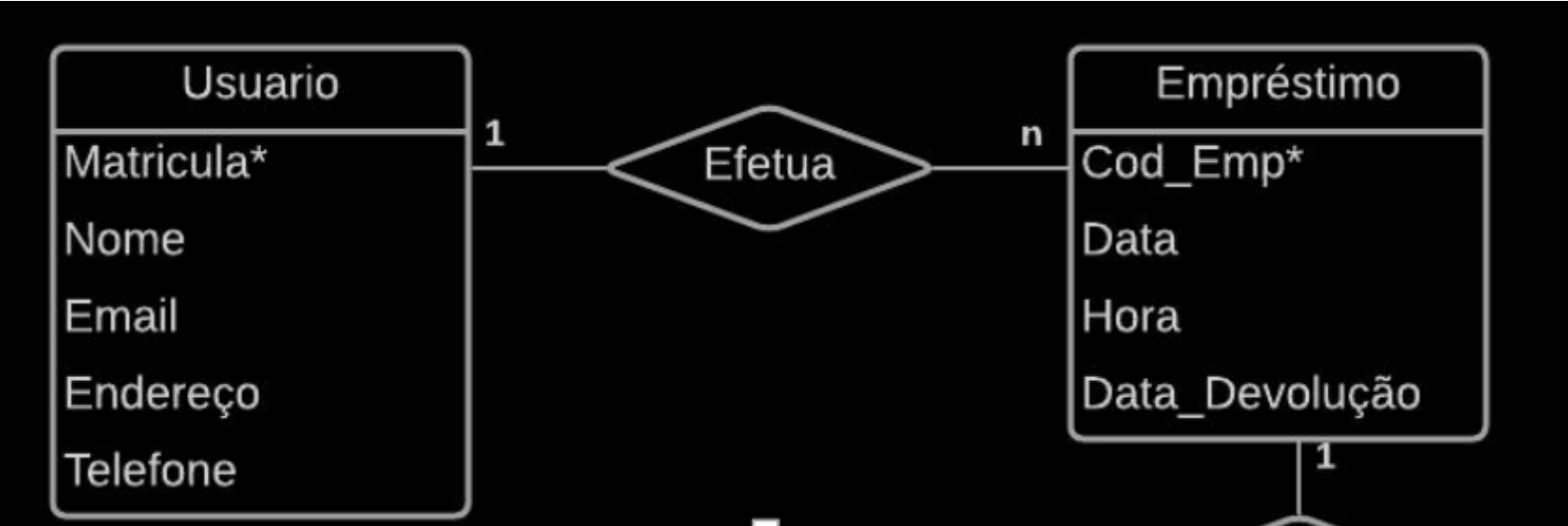
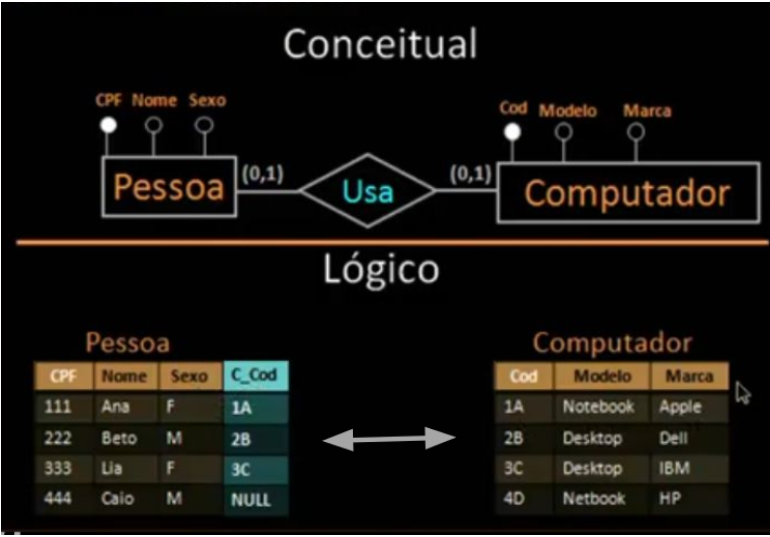
- Cada linha é chamada de **tupla**
- Um coluna é chamada de **atributo**
- E cada tabela é conhecida como **relação**

Exemplo Tupla e Relação

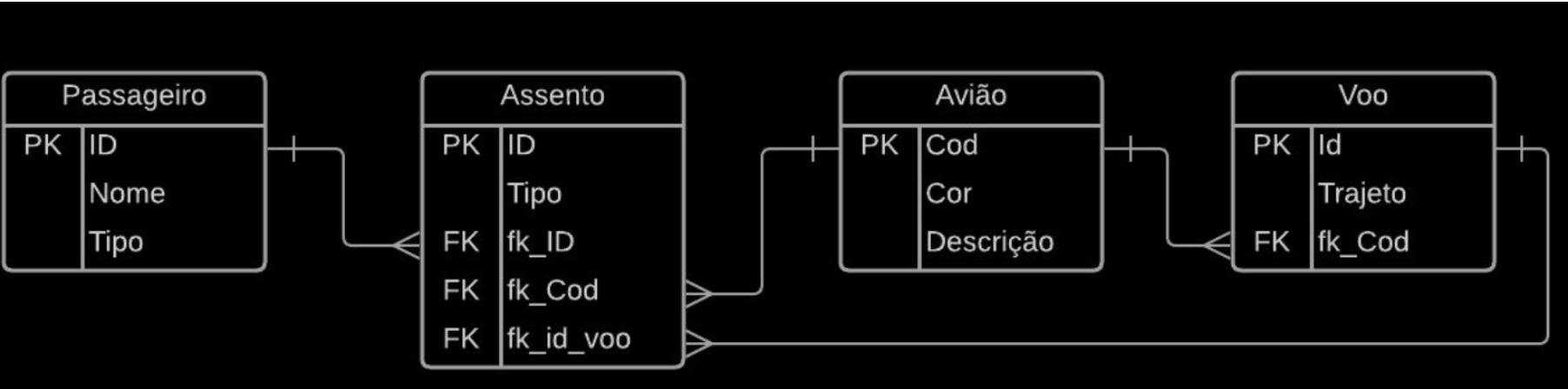
| Atributos | | |
|-----------|---------|-------|
| nome | cor | idade |
| Rex | Preto | 8 |
| Scoob | Marrom | 6 |
| Mel | Amarelo | 9 |

Tuplas →

Relação



Exemplo:



O que é um Modelo Físico?

No modelo físico fazemos a modelagem física do modelo de banco de dados por meio do SQL.

```
CREATE TABLE Usuario (
    ID varchar (4) NOT NULL,
    matricula varchar (11) UNIQUE,
    nome varchar (22),
    email varchar (18),
    endereco varchar (20) ,
    telefone varchar (8),
    PRIMARY KEY (MATRICULA)
);
```

O que é SQL?

- **SQL (Structured Query Language)**, ou “Linguagem de Consulta Estruturada”) é uma linguagem de banco de dados abrangente com instruções para definição de dados, consultas e atualizações.

O que é um Instâncias do banco de dados?

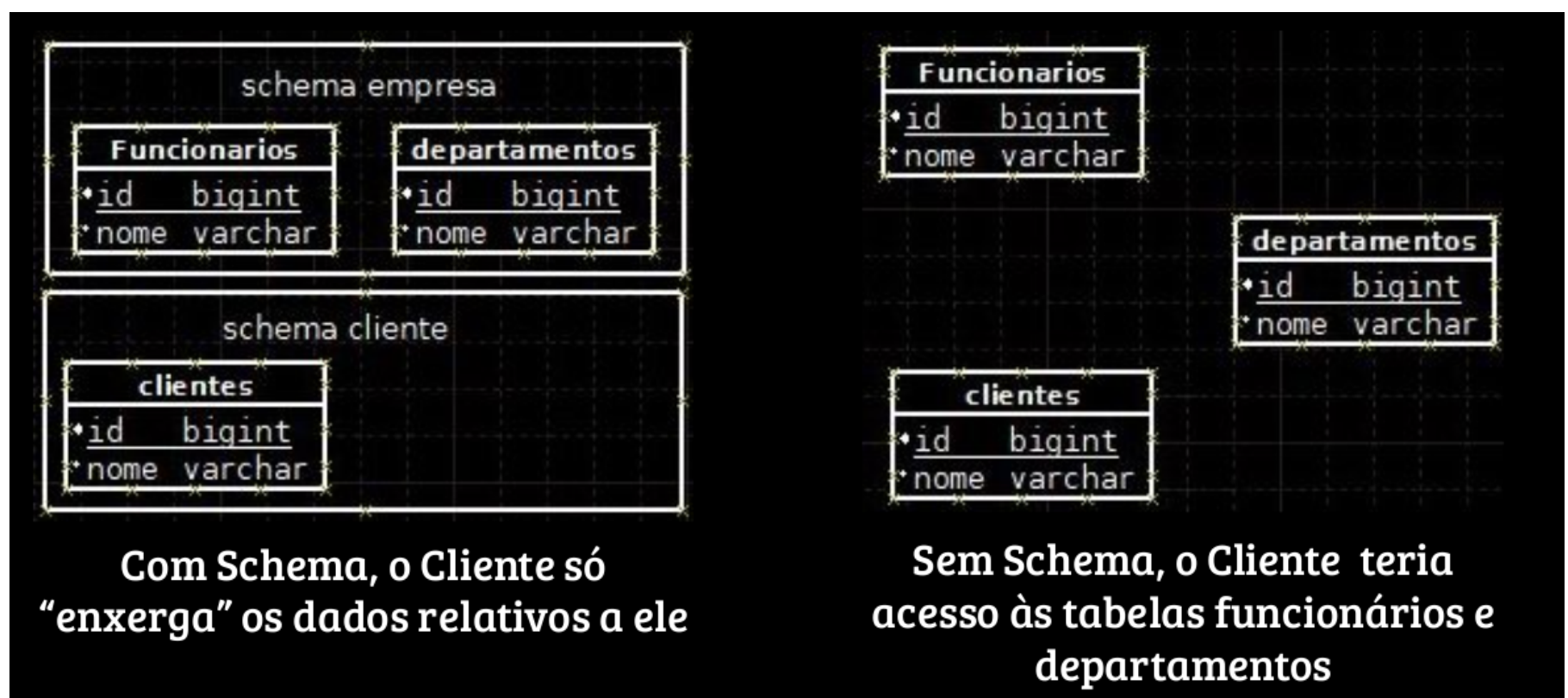
- É uma coleção de informações armazenada (mudam com frequência)

O que é um Esquema do banco de dados?

- O projeto geral do banco de dados é chamado **Esquema (Schemas)** do banco de dados (não mudam com frequência)
- São coleções de objetos dentro de um determinado banco de dados, que organizam vários aspectos e são importantes para segmentação da segurança, facilitando a administração dos objetos e dos dados
- Um esquema SQL visa agrupar tabelas e outras construções que pertencem à mesma aplicação de banco de dados
- Um esquema SQL é identificado por um **nome**, e inclui um **identificador de autorização** para indicar o usuário ou conta proprietário do esquema, bem como **descritores** para cada elemento
- Vamos criar um esquema chamado **EMPRESA** pertencente ao usuário com identificador de autorização **'Jsilva'**.

CREATE SCHEMA EMPRESA AUTHORIZATION 'Jsilva';

- Os esquemas fornecem a oportunidade para simplificar a administração de segurança, backup/restauração e gestão de banco de dados, permitindo que os objetos de banco de dados, ou entidades, a ser logicamente agrupadas



Integridade Referencial

- Garante que todos os relacionamentos entre as tabelas serão respeitadas resultando em um banco de dados íntegro

▼ ON DELETE SET NULL

- Na remoção da linha que contém o valor da chave estrangeira deve-se colocar NULL

```
CREATE TABLE Emprego(
    ...,
    FOREIGN KEY(CodDepartamento) REFERENCES Departamento(Dcod)
    ON DELETE SET NULL
);
```

▼ ON UPDATE CASCADE

- Na alteração do valor da chave estrangeira deve-se alterar em cascata as chaves

```
CREATE TABLE Emprego(
    ...,
    FOREIGN KEY(CodDepartamento) REFERENCES Departamento(Dcod)
    ON UPDATE CASCADE
);
```

Comandos

▼ CREATE TABLE

- Cria relações (tabelas).

```
CREATE TABLE <nome_da_tabela> (
    <nome_da_coluna1> <tipo_da_coluna1> <restrição>
    <nome_da_coluna2> <tipo_da_coluna2> <restrição>
    <nome_da_coluna3> <tipo_da_coluna3> <restrição>

    PRIMARY KEY <nome_da_coluna_que_vai_ser_PK>
    FOREIGN KEY <nome_da_coluna_que_vai_ser_PF>
    REFERENCES <nome_da_tabela_que_vai_ser_referenciada>
    (<nome_da_coluna_que_vai_ser_referenciada>)
);
```

```
CREATE TABLE <nome_da_tabela>
( <nome_da_col1> <tipo_da_col1> NOT NULL,
  <nome_da_col2> <tipo_da_col2> NOT NULL,
  ...
  PRIMARY KEY <lista_de_nomes_de_col>,
  FOREIGN KEY <nomes_de_col>
  REFERENCES <nome_tab_ref>(<nome_da_col_ref>)
);
```

```
CREATE TABLE Empregado ( ...,
  NOME DA RESTRIÇÃO → CONSTRAINT TrabalhaEm
  NOME DA CHAVE → FOREIGN KEY(Cod_Dept)
  ESTRANGEIRA
  NOME DA OUTRA → REFERENCES Departamento(Dcod), ... );
  TABELA E COLUNA
```

▼ DROP TABLE

- Elimina completamente a tabela (vazia ou não).

```
DROP TABLE <nome_da_tabela>;
```

▼ ALTER TABLE

- Modifica uma tabela

```
ALTER TABLE <nome_da_tabela> <alterações>
```

```
ALTER TABLE Empregado
ADD Telefone VARCHAR(30)
ADD sexo CHAR(1)
DEFAULT 'F';
```

▼ Restrições

- **NOT NULL**

```
CREATE TABLE Tabela(
  nome TEXT NOT NULL,
);
```

- **DEFAULT**

```
CREATE TABLE Tabela(
  cep NUMERIC DEFAULT 0,
);
```

- **CHECK**

```
CREATE TABLE Tabela(
  Dnumero NUMERIC NOT NULL CHECK (Dnumero > 0 AND Dnumero <10),
);
```

- **UNIQUE**

```
CREATE TABLE Tabela(
  CPF INT UNIQUE,
);
```

- **PRIMARY KEY**

```
CREATE TABLE Tabela(
  CPF INT UNIQUE,
  PRIMARY KEY (CPF)
);
```

- **FOREIGN KEY** : Atenção esse comando precisa vim acompanhado do comando **REFERENCES** para que os valores não fiquem repetidos.

```
CREATE TABLE Empregado(
  ...,
  cod_Dept INTEGER,
  FOREIGN KEY (cod_Dept) REFERENCES Departamento (Dcod)
);
```

▼ INSERT

- Em sua forma mais simples, **INSERT** é usado para acrescentar uma única linha em uma tabela. No comando abaixo é preciso passar todos os valores e precisam está na mesma ordem quando foi criado os atributos.

```
INSERT INTO <nome_da_tabela> VALUES (<valores_separados_por_virgula>);
```

- Outra maneira de usar o **INSERT** é informando só os valores que você deseja inserir na linha.

```
INSERT INTO <nome_da_tabela> (<nome_dos_atributos_desejados>)  
VALUES (<valores_separados_por_virgula>);
```

▼ DELETE

- Remove linhas da tabela.

```
DELETE FROM <nome_da_tabela> WHERE <condição>;
```

- **Obs:** Se a condição **WHERE** não for especificada, o **DELETE** vai remover todas as linhas da tabela.

▼ UPDATE

- O comando UPDATE é usado para modificar valores de atributo de uma ou mais linhas selecionadas

```
UPDATE <nome_da_tabela> SET <nome_coluna> = <novo_valor> WHERE <condição>
```

▼ SELECT

```
SELECT <nome_dos_atributos> FROM <nome_da_tabela> WHERE <condição>
```

- **Obs:** Se o **WHERE** não for especificado o **SELECT** retornará todos os valores da tabela.

```
SELECT * FROM <nome_da_tabela>
```

o comando acima retorna todos os atributos da tabela especificada.

- **Exemplo:**

```
SELECT F.nome, F.salario * 1.1 FROM Funcionario AS F  
WHERE F.salario < 1000;
```

- **BETWEEN:** É um operador lógico, que retorna os valores que estejam entre dois valores.

```
SELECT * FROM Funcionario WHERE (salario BETWEEN 2000 AND 6000)
```

- **Junção de tabelas**

```
SELECT C.codigo, C.Nome, P.data FROM Clientes AS C, Pedidos AS P  
WHERE C.codigo = P.codCliente
```

- **LIKE:** O operador de comparação **LIKE** permitirá condições de comparação apenas sobre partes de uma cadeia de caracteres:

```
SELECT Nome FROM Funcionario WHERE endereco LIKE '%SÃO';
```

Funções de Agregação


```
SELECT    SUM ( Salario ), MIN ( Salario ),  
          MAX ( Salario ), AVG ( Salario )  
FROM      PROFESSOR;
```

▼ COUNT

- A função de agregação **COUNT** é responsável por trazer o total de ocorrências da coluna informada
-

▼ MIN

- Retorna o menor valor da coluna especificada.

▼ MAX

- Retorna o maior valor da coluna especificada.

▼ SUM

- Retorna a soma de todos os valores da coluna especificada.

▼ AVG

- Retorna a média dos valores da coluna especificada.

- **FROM** – utilizada para especificar a tabela, que contém os campos listados na cláusula SELECT
- **WHERE** – utilizada para especificar as condições que devem reunir os registros que serão selecionados
- **GROUP BY** – utilizada para separar os registros selecionados em grupos específicos
- **HAVING** – utilizada para expressar a condição que deve satisfazer cada grupo

- **ORDER BY** – utilizada para ordenar os registros selecionados com uma ordem específica
- **DISTINCT** – utilizada para selecionar dados sem repetição
- **UNION** – combina os resultados de duas consultas SQL em uma única tabela para todas as linhas correspondentes

- **UNION** - união de conjunto ($A \cup B$)
- **EXCEPT** - diferença de conjunto ($A - B$)
- **INTERSECT** - interseção de conjunto ($A \cap B$)