**Banco de Dados:**

**O que é um dado?**

Seria uma informação única, um item da tabela (nome, idade, etc);

Fica dentro de uma entidade (~~coluna de~~ uma tabela);

**Informação:** Quando se adquire um ou mais dados (nome, data de nascimento, email, etc), a construção de um ou mais dados, se chama informação.

**MER**(Modelo de Entidade e Relacionamento)

**Modelo, Entidade e relacionamento**

Entidade: Os “participantes”;

-É o tipo do dado que estará na tabela, ~~colunas~~;

Relacionamento: As relações entre as entidades;

-A relação que cada tipo de dado da tabela (entidade) estabelecerá.

**Ex)** tabela\_clientes (usuário,nome e senha, perfilId) 🡨--🡪 Perfis (Usuario, Admin, Vip, Premium, etc);

**Modelo Conceitual:** Expôe-se o nome das tabelas junto de seus relacionamentos e cardinalidades;

**Modelo Lógico:** Além do nome das tabelas e seus relacionamentos, dentro de cada tabela há seus respectivos atributos (chaves primárias, estrangeiras e atributos);

**Modelo Físico:** Reside totalmente no Banco de Dados (Sistema de Gerenciamento de Banco de Dados – SGBD), seria os Scripts, a construção da tabela fisicamente no servidor do banco de dados.

Em banco de dados existem dados relacionados (entidades relacionadas)

Identificar sempre os dados relacionáveis como por exemplo, um sistema de consulta médica veterinária.

**Tabela PET:**

-ID;

-NomePET;

-DataNascimento;

-DonoId;

-RaçaId;

**Tabela Raça:**

-ID;

-Raça;

**Tabela TipoPet:**

-ID;

-Nome;

-TipoPetId;

**Tabela Medico:**

-ID

-Nome do médico;

**Tabela Dono:**

-Id;

-Nome;

**Consulta:**

-ID;

-PetID;

-MedicoId;

-DataConsulta;

OBS: tabela consulta cria a RELAÇÂO das tabelas PET e MÉDICO.

**Chave primária e secundaria (estrangeira)**

Chave primária são os ID’s de cada registro da tabela (Primary Key – PK)

Estrangeira ou secundária, seria os atributos que se relacionam com outras entidades. (Foreing Key – FK)

**Cardinalidade**

1:1, um para um; Um usuário tem um email

1:N, um para muitos; Uma professora tem X alunos ou Uma pessoa tem vários e-mails

N:N, muitos para muitos; Alunos assistem X aulas

(Passou de um, já é considerado como muitos(n))/

**Ex:**

**Tabela Alunos:**

-Id;

-Nome;

**Tabela Aulas:**

-Id;

-NomeAula;

**Tabela AlunosXaulas:**

-Id;

-NomeAlunoId;

-NomeAulaId;

1)

**Tabela Pessoa:**

-Id;

-NomePessoa;

-CnhId;

**Tabela Telefone:**

-Id;

-PessoaId;

-Telefone;

**Tabela Email:**

-Id;

-PessoaId;

-Email;

**Tabela CNH:**

-Id;

-Cnh;

-Expedição;

-Vencimento;

**Tipos de dados**

Diferença entre CHAR(n) e VARCHAR(n)

Em char, quando define-se um tamanho para a variável, ela irá utilizar todo este tamanho em sua totalidade.

Em Varchar, independente do tamanho (se for menor ou igual ao máximo), ele irá “eliminar” o tamanho excedente para e economizar memória.

**Procedure**

Store Procedures (Procedimento Armazenado), funcionam como fossem métodos para armazenar estruturas de códigos repetitivos ao decorrer da aplicação, como buscas (select) de alguma entidade específica, etc.

Armazena tarefas repetitivas e aceita parâmetros de entrada caso precise.

**Comandos:**

-OR: em comandos de consulta (DQL), retorna se encontrado, uma informação OU outra.

-LIKE %: (funciona como o IndexOf do C#), procura uma informação específica na tabela, frase, etc. Comparação de texto. Funciona como filtro.

WHERE, ON (Onde, em) funcionam quase da mesma forma.

DESC (ordem DECRESCENTE)

ASC (ordem CRESCENTE)

UNIQUE: Referencias UNIQUE’s nas tabelas, dizem que não pode-se repetir os mesmos valores dentro da tabela.

**Tipo de dado BIT:** BIT default (1) (declara o tipo de variável e diz que o padrão é 1, verdadeiro)