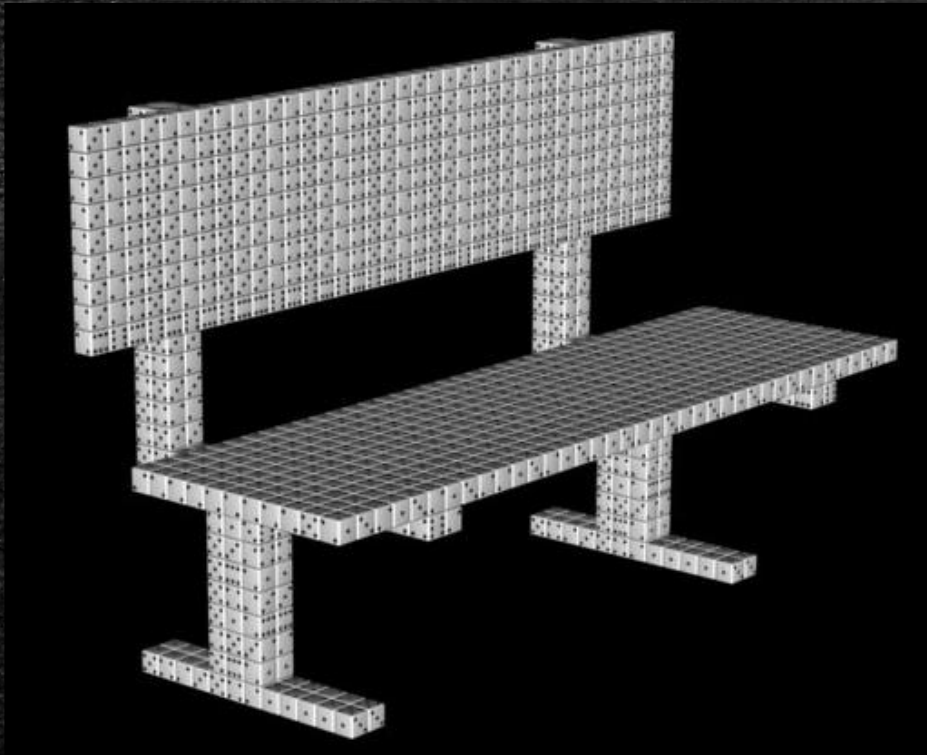


revisão!

Banco de Dados – BD

Aula de revisão

Banco de Dados



- Um banco de dados “é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”, ou seja, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, posso dizer que tenho um banco de dados.
- Exemplos de Banco de Dados:
- Um livro;
- Um catalogo de peças;
- Um sistema de vendas;

Dados e Informação



- Dado: São fatos brutos, em sua forma primária. E muitas vezes os dados podem não fazer sentido sozinhos.
- Informação: Consiste no agrupamentos de dados forma organizada para fazer sentido e gerar conhecimento.
- Exemplo: O número 2001 isoladamente faz algum sentido?
- E se eu dissesse: "Ano do atentado terrorista às torres gêmeas foi 2001"
- Agora faz sentido! Isso é uma informação.

Sistema Gerenciador de Banco de Dados - SGBD



- Um Sistema de Gerenciamento de Banco de Dados (SGBD) é o conjunto de programas de computador (softwares) responsáveis pelo gerenciamento de banco de dados(BD).
- Exemplos de SGBD são: Oracle, SQL Server, PostgreSQL, MySQL, Sybase, MariaDB, MongoDB.

Características de um SGBD

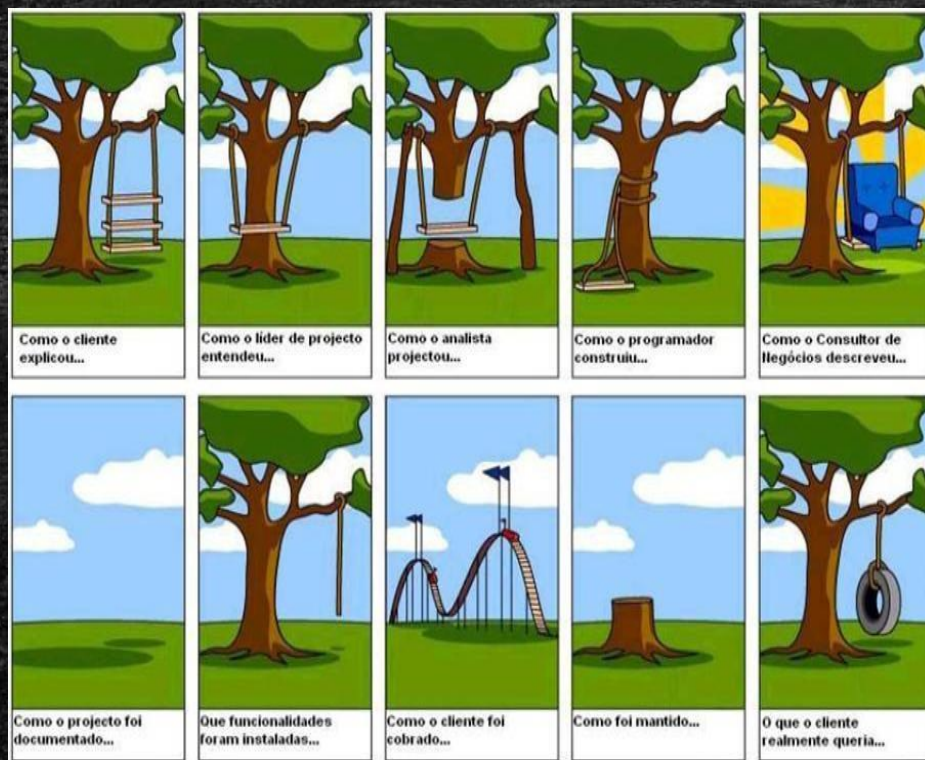
- Controle de redundância: Informações devem possuir um mínimo de redundância visando manter a estabilidade do modelo.
- Compartilhamento de dados: As informações devem estar disponíveis para qualquer número de usuários de forma rápida e segura.
- Controle de acesso: Necessidade de saber quem pode realizar qual função dentro do Banco de Dados.
- Esquematização: Os relacionamentos devem estar armazenados no Banco de Dados para garantir a facilidade de entendimento e aplicação do modelo. A integridade das informações deve ser garantida pelo Banco de Dados.
- Backup ou cópia de segurança: Deve haver rotinas específicas para realizar a cópia de segurança dos dados armazenados.

Projeto de Banco de Dados



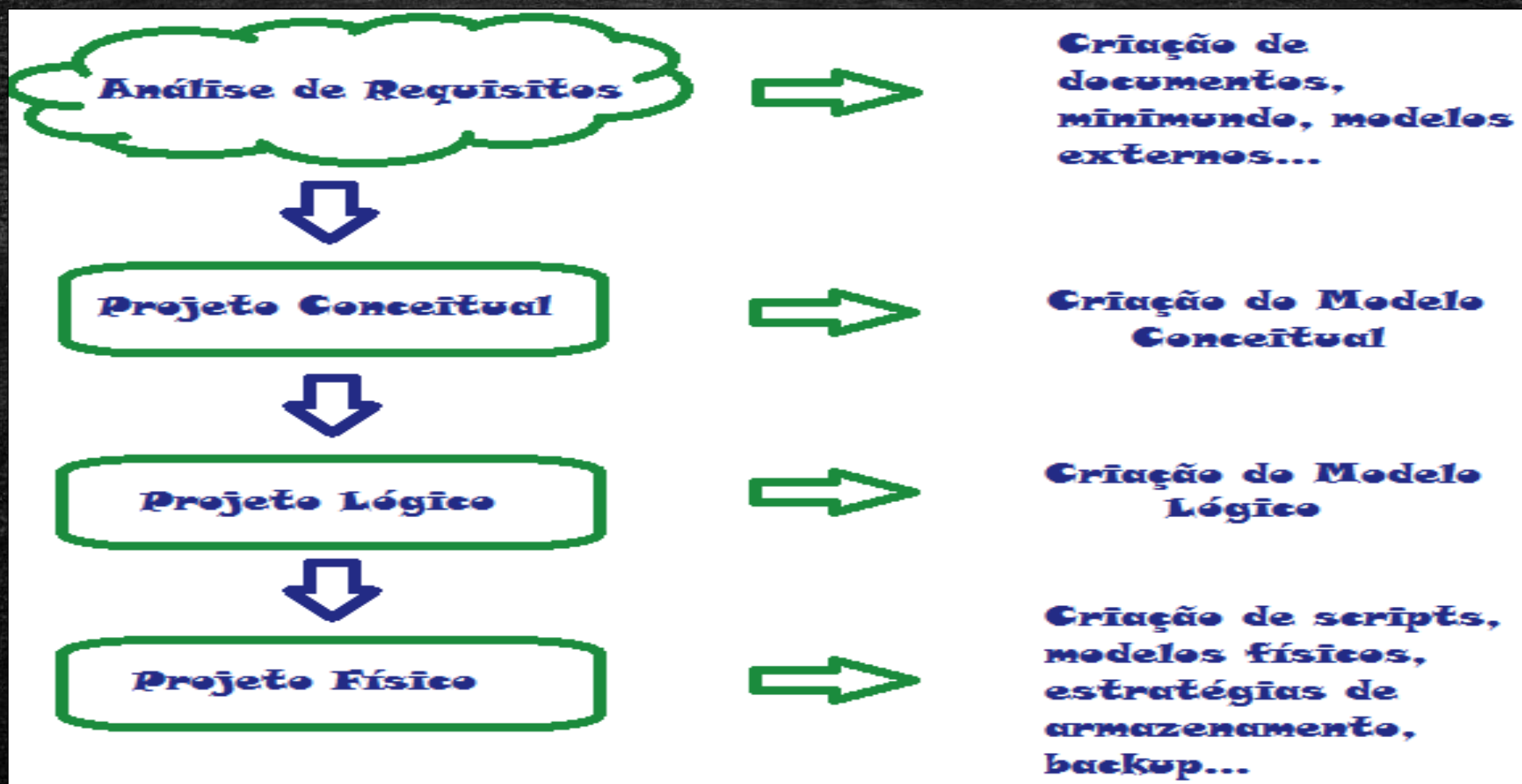
- Todo bom sistema de banco de dados deve apresentar um projeto, que visa à organização das informações e utilização de técnicas para que o futuro sistema obtenha boa performance e também facilite as manutenções que venham a acontecer.
- O projeto de Banco de Dados tem o objetivo de transformar as necessidades de informações do negócio em um banco de dados. Ou seja, foca na maneira em que o banco de dados será usado para armazenar e gerenciar os dados.

Análise de Requisitos

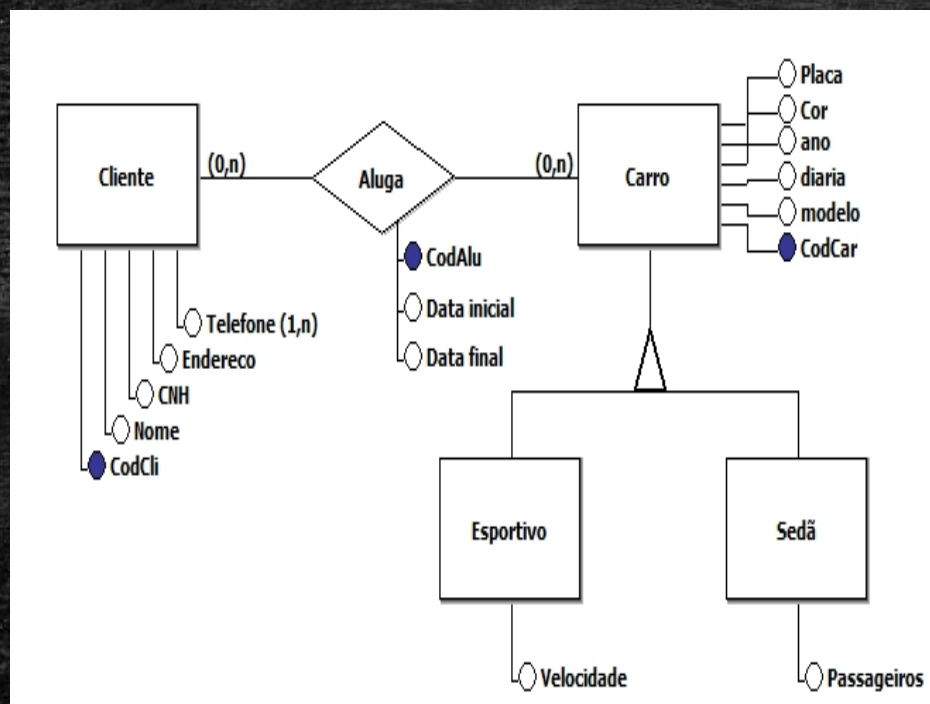


- A primeira etapa do projeto de Banco de Dados é a identificação dos requisitos que o Banco de Dados deve atender. Nessa fase, realiza-se entrevistas com as pessoas envolvidas no processo, cria-se uma descrição textual macro do processo (minimundos).

Modelagem de Dados

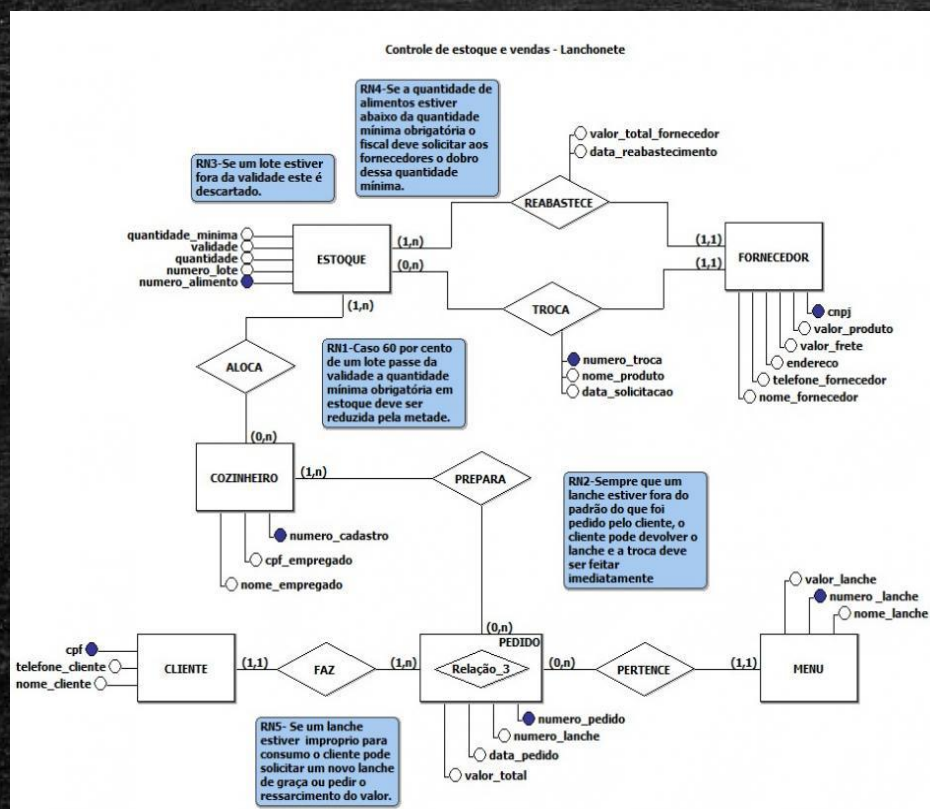


Modelo Conceitual



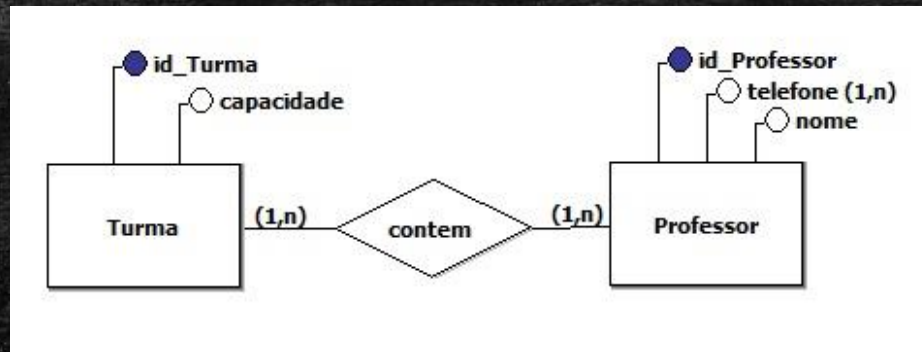
- É a primeira etapa da modelagem do banco de dados, a modelagem conceitual é a visão mais próxima do modo que o usuário visualiza as informações que serão armazenadas no BD.
- Tem por objetivo representar em “alto nível” os dados independente da implementação física do BD no computador.

Representação do Modelo Conceitual - MER



- Modelo Entidade Relacionamento → MER
- Para a representação do Modelo Conceitual usaremos o Modelo Entidade Relacionamento [MER], que consistem em um modelo baseado na percepção do mundo real, que representa um conjunto de objetos básicos chamado de entidade, seus atributos e nos relacionamentos entre essas entidades.

Etapas a Construção do MER



- Identificar as entidades.
- Identificar os atributos das entidades.
- Classificar os atributos de acordo com o seu tipo.
- Identificar os relacionamentos entre as entidades.
- Estabelecer as cardinalidades.
- Gerar o diagrama com o MER.

Entidade

- Uma entidade é uma representação de um conjunto de informações sobre determinado conceito do sistema.
- Toda entidade possui ATRIBUTOS, que são as informações que referenciam a entidade.

FUNCIONARIO

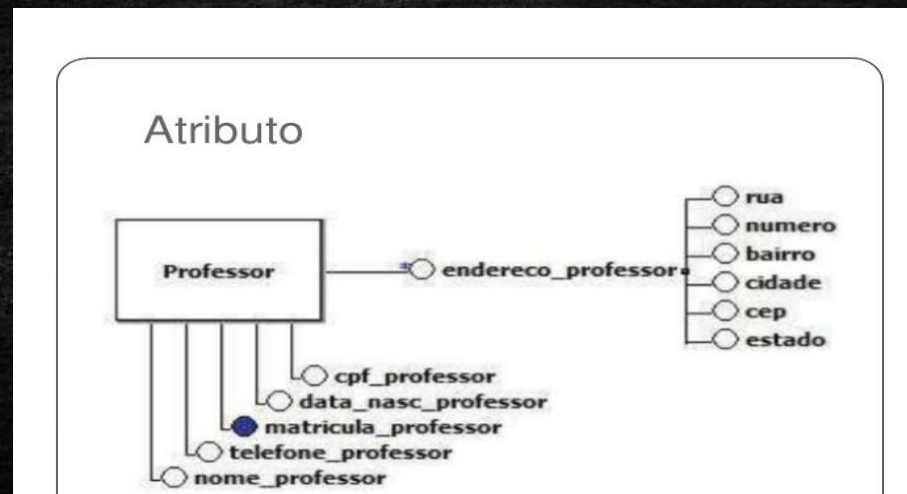
DEPARTAMENTO

CLIENTE

FORNECEDOR

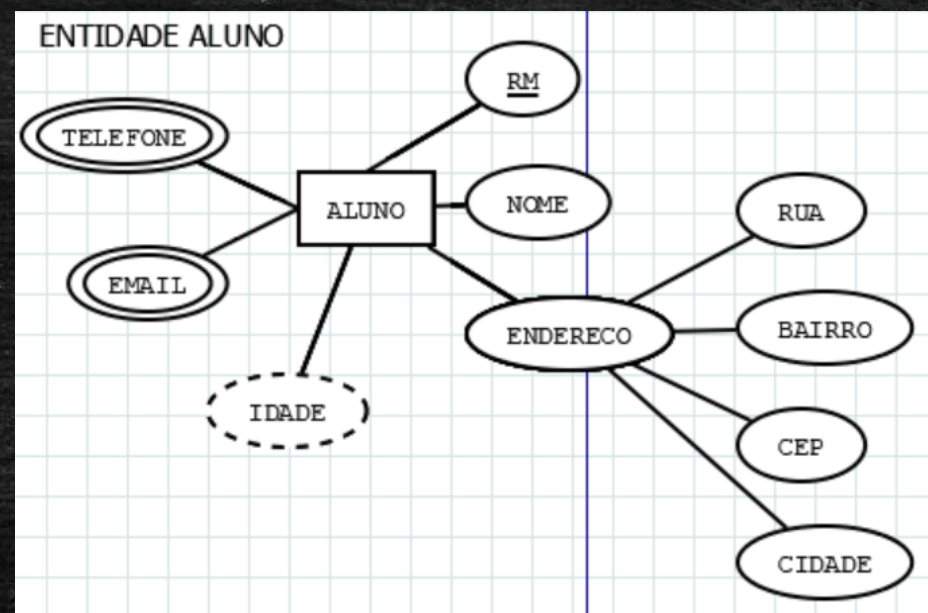
Atributos

- São informações que desejamos guardar sobre a instância da entidade, é o que caracteriza a entidade.
- Exemplo:
Da entidade Aluno temos, nome, endereço, telefone, e-mail, matricula entre outros.



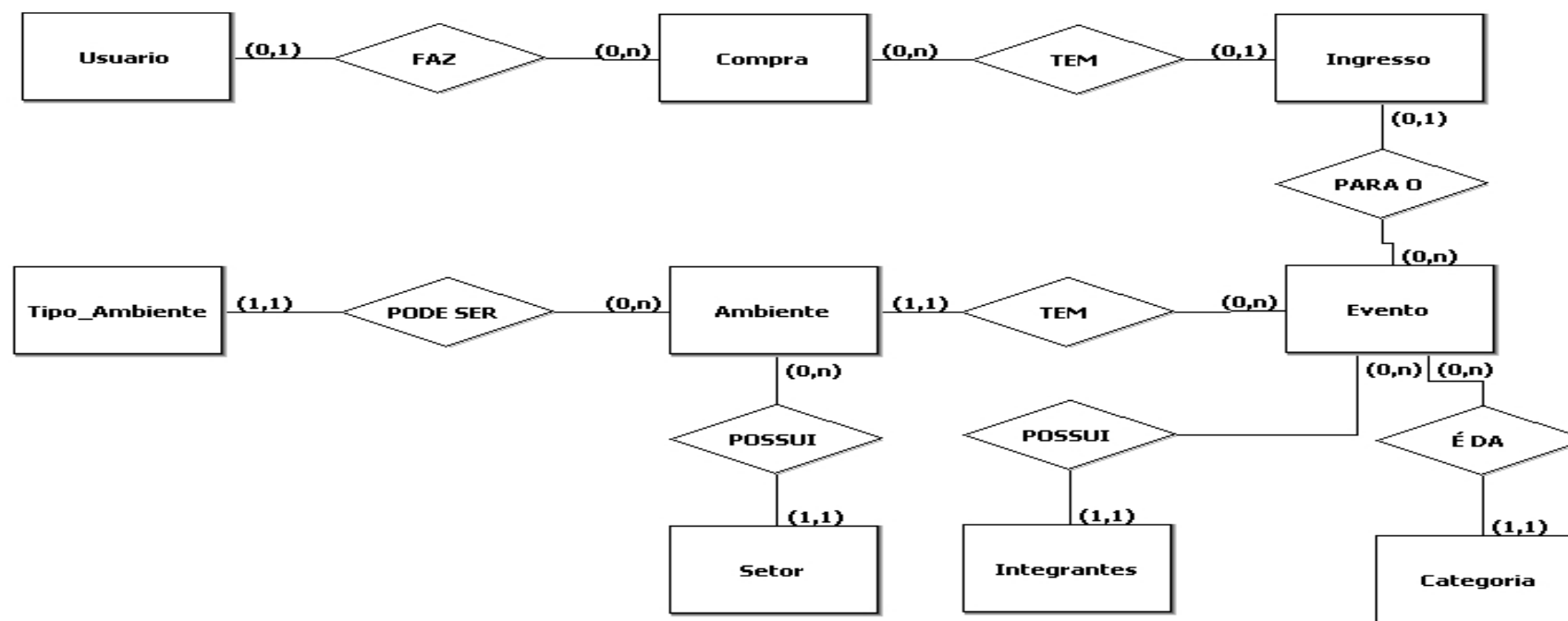
Tipos de Atributos

- *Identificador*
- *Simples*
- *Composto*
- *Multivalorado*
- *Derivado*



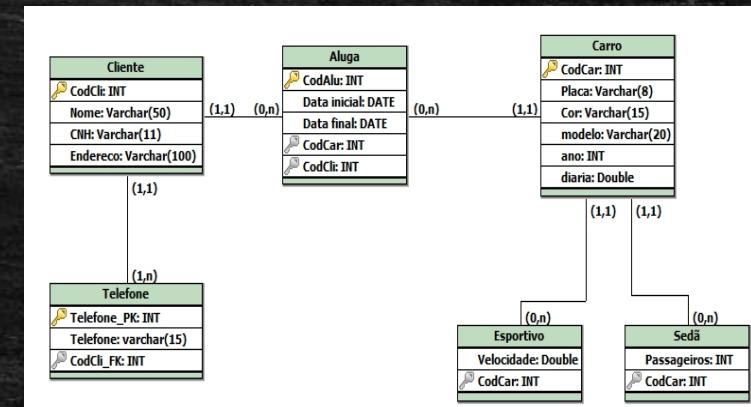
Relacionamentos

- É a associação, é a ligação entre as entidades, é representado por um losango.



Modelo Lógico

- Compreende uma descrição das estruturas que serão armazenadas no banco de Dados e que resulta numa representação gráfica dos dados de uma maneira Lógica.
- Leva em conta os limites e restrições impostas de acordo com o Sistema Gerenciador de Banco de Dados). Suas características são :
 - Deriva do Modelo Conceitual e vira a representação do negócio.
 - Possui entidades associativas em lugar de relacionamentos N:N.
 - Define as chaves primárias das entidades.
 - Define tabelas para os atributos composto e multivalorados.
 - Normalização até a 3a. forma normal.
 - Adequação ao padrão de nomenclatura.
 - Entidades e atributos documentados.



Mapeamento Modelo Lógico

- Mapeamento de entidades :
 - O primeiro conceito que devemos mapear do Modelo Entidade Relacionamento (MER) para o Modelo Lógico é o conceito de entidade. No Modelo Lógico, entidades são mapeadas para tabelas. Ou seja, cada entidade do Modelo Conceitual será mapeado para uma tabela no Modelo Lógico.

Mapeamento de Atributos

- Mapeamento de atributo Identificador:
 - Para cada atributo identificador da entidade do Modelo Conceitual criar um atributo PK (Primary key) na tabela do Modelo Lógico.
- Mapeamento de atributos Simples :
 - Para cada atributo simples da entidade do Modelo Conceitual criar um atributo simples na entidade do Modelo Lógico.

Mapeamento de Atributos

- Mapeamento de atributos Compostos :
 - Para cada atributo composto criar uma tabela no Modelo Lógico com o atributo, fazendo referência a chave primária (PK) da tabela que deu origem a nova tabela.
- Mapeamento de atributos Multivalorados :
 - Para cada atributo multivalorado criar uma nova tabela no Modelo Lógico contendo o atributo chave estrangeira (Foreign Key) fazendo referência a chave primária PK da tabela que deu origem a nova tabela.

Mapeamento de Relacionamentos

- Mapeamento de relacionamentos 1:1:
 - Neste tipo de relacionamento escolhe-se uma das entidades e exporta seu atributo chave primária (Primary Key) para a outra entidade, sendo a chave estrangeira (Foreign Key) desta entidade.
- Mapeamento de relacionamentos 1:N:
 - Para cada relacionamento um para muitos exportar o atributo chave primária PK da entidade do lado 1 para o lado N. Este atributo exportado tem o nome de Foreign Key (FK).
- Mapeamento de relacionamentos N:N ou N:M:
 - Para cada relacionamento muitos para muitos criar uma nova entidade tendo como chave os atributos das entidades participantes do relacionamento que serão as chaves primárias desta nova entidade além de serem foreign keys.

Dicionário de Dados

- O dicionário de dados descreve a terminologia utilizada para o desenvolvimento do modelo de dados do sistema. Apresenta uma descrição textual da estrutura lógica e física do banco de dados. Neste dicionário você pode simplesmente colocar a descrição por extenso de campos e tabela como também colocar outras características dos campos como , tipo de dado, se o mesmo é requerido ou não. Abaixo segue um exemplo de um pequeno dicionário de dados.

TB_FUNCIONARIO		
<u>Campo</u>	<u>Tipo de Dado</u>	<u>Observação</u>
id_funcionario	INT	Identificador numérico de cada funcionário cadastrado(código).
nm_funcionario	VARCHAR(80)	Nome completo do funcionário.
ds_email_funcionario	VARCHAR(50)	Email do funcionário.
ds_user_name	VARCHAR(30)	Login para o funcionario se logar no WebSuporte.
ds_senha_funcionario	VARCHAR(08)	Senha para o funcionário se logar no WebSuporte.
dt_admissao_funcionario	DATE	Data de admissão do funcionário.

Tipos de Dados

MySQL DATATYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 2147483647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LOBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

Normalização de Dados

- A regra de ouro que devemos observar no projeto de um banco de dados baseado no Modelo Relacional de Dados é a de "não misturar assuntos em uma mesma Tabela".
- Normalização de dados é o processo formal e passo a passo que examina os atributos de uma entidade, com o objetivo de evitar anomalias observadas na inclusão, exclusão e alteração de registros.

Normalização de Dados

- Por exemplo: na Tabela Cliente devemos colocar somente atributos relacionados com o assunto Cliente. Não devemos misturar atributos relacionados com outros assuntos, tais como Pedido, Produto, etc.
- Essa "Mistura de Assuntos" em uma mesma tabela acaba por gerar repetição desnecessária dos dados bem como inconsistência dos dados.

Tabela de Filmes

Entidade: Filmes							
idFilme	Nome	Gênero	idMidia	Idioma	Tipo	Seção	Preço
656565	Uma janela suspeita	Drama	1001	dub	DVD	25	3,50
656565	Uma janela suspeita	Drama	1002	dub	VHS	25	3,50
656565	Uma janela suspeita	Drama	1003	leg	DVD	25	3,50
656565	Uma janela suspeita	Drama	1004	leg	VHS	25	3,50
323232	Minority Report	Ficção	2550	dub	DVD	32	4,20
323232	Minority Report	Ficção	2550	dub	VHS	32	4,20
323232	Minority Report	Ficção	2550	leg	DVD	32	4,20
323232	Minority Report	Ficção	2550	leg	VHS	32	4,20

Normalização de Dados

- Normalmente após a aplicação das regras de normalização de dados, algumas tabelas acabam sendo divididas em duas ou mais tabelas, o que no final gera um número maior de tabelas do que o originalmente previsto. Este processo causa a simplificação dos atributos de uma tabela, colaborando significativamente para a estabilidade do modelo de dados, reduzindo-se consideravelmente as necessidades de manutenção.

Tabela de Filmes Normalizada

Entidade: Filmes

idFilme	Nome	Gênero
656565	Uma janela suspeita	Drama
323232	Minority Report	Ficção

Entidade: Mídias

idMidia	idFilme	Idioma	Tipo	Seção	Preço
1001	656565	dub	DVD	25	3,50
1002	656565	dub	VHS	25	3,50
1003	656565	leg	DVD	25	3,50
1004	656565	leg	VHS	25	3,50
2550	323232	dub	DVD	32	4,20
2550	323232	dub	VHS	32	4,20
2550	323232	leg	DVD	32	4,20
2550	323232	leg	VHS	32	4,20

Os objetivos da normalização

- Minimização de redundâncias e inconsistências;
- Facilidade de manipulações do banco de dados;
- Ganho de performance no SGBD;
- Facilidade de manutenção do sistema de Informação;
- Entre outros.

Formas Normais

- O Processo de normalização aplica uma série de regras sobre as tabelas de um banco de dados, para verificar se estão corretamente projetadas. Embora existam cinco formas normais (ou regras de normalização), na prática usamos um conjunto de três Formas Normais.
- Vejamos as três primeiras formas normais do processo de normalização de dados.
 - Primeira Forma Normal (1FN)
 - Segunda Forma Normal (2FN)
 - Terceira Forma Normal (3FN)

Normalização de Dados

- Para aplicar a normalização de dados é necessário considerar a sequência das formas normais, isto é, para aplicar a segunda forma normal por exemplo, é necessário que seja aplicado a primeira forma normal. Da mesma forma, para aplicar a terceira forma normal é necessário que já tenha sido feita a **normalização** na segunda forma normal.

Primeira Forma Normal

- **Primeira Forma Normal (1FN)**
- Uma relação estará na primeira forma normal 1FN, se não houver grupo de dados repetidos, isto é, se todos os valores forem únicos. Em outras palavras podemos definir que a primeira forma normal não admite repetições ou campos que tenha mais que um valor.
- Os procedimentos mais recomendados para aplicar a 1FN são os seguintes:
 - a) Identificar a chave primária da entidade;
 - b) Identificar o grupo repetitivo e removê-lo da entidade;
 - c) Criar uma nova entidade com a chave primária da entidade anterior e o grupo repetitivo.

Exemplo Tabela de Clientes

Código_cliente	Nome	Telefone	Endereço
C001	José	9563-6352 9847-2501	Rua Seis, 85 Morumbi 12536-965
C002	Maria	3265-8596	Rua Onze, 64 Moema 65985-963
C003	<u>Janio</u>	8545-8956 9598-6301	Praça ramos Liberdade 68858-633

Normalização de Dados

- Analisando teremos:
- Todos os clientes possuem Rua, CEP e Bairro, e essas informações estão na mesma célula da tabela, logo ela não está na **primeira forma normal**. Para normalizar, deveremos colocar cada informação em uma coluna diferente, como no exemplo a seguir:

Código_cliente	Nome	Telefone	Rua	Bairro	Cep
C001	José	9563-6352 9847-2501	Rua Seis, 85	Morumbi	12536-965
C002	Maria	3265-8596	Rua Onze, 64	Moema	65985-963
C003	Janio	8545-8956 9598-6301	Praça ramos	Liberdade	68858-633

Tabela desnormalizada, ou seja, não está na 1ª forma normal

Normalização de Dados

- Mesmo com o ajuste acima, a tabela ainda não está na **primeira forma normal**, pois há clientes com mais de um telefone e os valores estão em uma mesma célula. Para normalizar será necessário criar uma nova tabela para armazenar os números dos telefones e o campo-chave da tabela cliente. Veja o resultado a seguir:

Código_cliente	Nome	Rua	Bairro	Cep
C001	José	Rua Seis, 85	Morumbi	12536-965
C002	Maria	Rua Onze, 64	Moema	65985-963
C003	Janio	Praça ramos	Liberdade	68858-633

Tabela na primeira forma normal

Codigo_cliente	Telefone
C001	9563-6352
C001	9847-2501
C002	3265-8596
C003	8545-8956
C003	9598-6301

Tabela na 1ª forma normal

Segunda Forma Normal

- **Segunda Forma Normal (2FN)**
- Uma tabela está na **Segunda Forma Normal 2FN** se ela estiver na 1FN e todos os atributos não chave forem totalmente dependentes da chave primária (dependente de toda a chave e não apenas de parte dela).
- Se o nome do produto já existe na tabela produtos, então não é necessário que ele exista na tabela de produtos. A **segunda forma normal** trata destas anomalias e evita que valores fiquem em redundância no banco de dados.

Normalização de Dados

- Procedimentos:
- a) Identificar os atributos que não são funcionalmente dependentes de toda a chave primária;
- b) Remover da entidade todos esses atributos identificados e criar uma nova entidade com eles.
- A chave primária da nova entidade será o atributo do qual os atributos do qual os atributos removidos são funcionalmente dependentes.

Normalização de Dados

N_pedido	Codigo_produto	Produto	Quant	Valor_unit	Subtotal
1005	1-934	Impressora laser	5	1.500,00	7.500,00
1006	1-956	Impressora desjet	3	350,00	1.050,00
1007	1-923	Impressora matricial	1	190,00	190,00
1008	1-908	Impressora mobile	6	980,00	5.880,00

Tabela não está na segunda forma normal

Normalização de Dados

- Analisando teremos:
- O nome do produto depende do código do produto, porém não depende de N_pedido que é a chave primária da tabela, portanto não está na **segunda forma normal**. Isto gera problemas com a manutenção dos dados, pois se houver alteração no nome do produto teremos que alterar em todos os registros da tabela venda.
- Para normalizar esta tabela teremos de criar a tabela Produto que ficará com os atributos Código_produto e produto e na tabela Venda manteremos somente os atributos N_pedido, código_produto, quant, valor_unit e subtotal.

Normalização de Dados

Venda

N_pedido	Codigo_produto	Quant	Valor_unit	Subtotal
1005	1-934	5	1.500,00	7.500,00
1006	1-956	3	350,00	1.050,00
1007	1-923	1	190,00	190,00
1008	1-908	6	980,00	5.880,00

Tabela na 2ª forma normal

Produto

Codigo_produto	Produto
1-934	Impressora laser
1-956	Impressora desjet
1-923	Impressora matricial
1-908	Impressora mobile

Tabela na segunda forma normal

Conforme visto na Primeira Forma Normal, quando aplicamos normalização é comum gerar novas tabelas a fim de satisfazer as formas normais que estão sendo aplicadas.

Terceira Forma Normal

- Uma tabela está na **Terceira Forma Normal** 3FN se ela estiver na 2FN e se nenhuma coluna não-chave depender de outra coluna não-chave.
- Na **terceira forma normal** temos de eliminar aqueles campos que podem ser obtidos pela equação de outros campos da mesma tabela.
- Procedimentos:
 - Identificar todos os atributos que são funcionalmente dependentes de outros atributos não chave;
 - Removê-los.

Normalização de Dados

Considere a tabela abaixo:

N_pedido	Codigo_produto	Quant	Valor_unit	Subtotal
1005	1-934	5	1.500,00	7.500,00
1006	1-956	3	350,00	1.050,00
1007	1-923	1	190,00	190,00
1008	1-908	6	980,00	5.880,00

Tabela não está na terceira forma normal

Considerando ainda a nossa tabela Venda, veremos que a mesma não está na **terceira forma normal**, pois o subtotal é o resultado da multiplicação Quant X Valor_unit, desta forma a coluna subtotal depende de outras colunas não-chave.

FIM !!!!!

Banco de Dados