

BD

Grupo A – Definições

Integrantes:

- Alisson Mayer Medeji;
- Pedro Pompeu;
- Rafael Lopes;
- Fernando Moraes dos Santos;
- Raí de Carvalho Juca;
- Nicolas Mingorance;
- Marcelo de Oliveira;
- Gean Ordonho Ataide;
- João Henrique Monteiro Alves;
- André Tipolt Lopes;
- Giovanni Vilarinho Barros;
- Otávio Augusto Azevedo Marçal;
- Tarcisio Pereira da Silva.

Conceitos sobre Banco de Dados

Conceitos Básicos

- Dado: fato do mundo real que está registrado

exemplos: endereço, data, nome etc.

- Informação: fato útil que pode ser extraído direta ou indiretamente a partir dos dados

exemplos: endereço de entrega, idade...

- Banco de Dados (BD): coleção de dados inter-relacionados e persistentes que representa um subconjunto dos fatos presentes em um domínio de aplicação (universo de discurso)

Antes de entender o que é um banco de dados, é importante saber a diferença entre as palavras “dados” e “informações”. Os dados são os fatos brutos, em sua forma primária, e podem não fazer nenhum sentido quando estão isolados; já as informações são o agrupamento de dados organizados, de forma que façam sentido e gerem algum conhecimento.

Segundo Korth, 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.”

Podemos exemplificar situações clássicas como uma lista telefônica, um catálogo de CDs ou um sistema de controle de RH de uma empresa.

SGBD

Já um sistema de gerenciamento de banco de dados (SGBD) é um software que possui recursos capazes de manipular as informações do banco de dados e interagir com o usuário. Exemplos de SGBDs são: Oracle, SQL Server, DB2, Postgre, MySQL, o próprio Access ou Paradox, entre outros.

A informação é muitas vezes a coisa mais valiosa das empresas, mantê-las e poder acessá-las sempre que necessário é primordial para tomar decisões importantes. Mas controlar o acesso a essas informações também é importantíssimo. Já pensou se elas caíssem em mãos erradas? E a perda de informações? Já imaginou se estragasse o HD do servidor onde está o banco de dados? Backup é uma forma de garantir que informações não serão perdidas.

Analizando os 5 primeiros SGBDs, temos Oracle em 1º lugar com um pequeno aumento no mês de março comparado com o mês de abril de 2020. O MySQL se mantém no 2º lugar e pode aparecer em 2020 como um concorrente forte para o primeiro lugar. Em 3º lugar está a Microsoft SQL Server, que vem caindo de pontuação em relação a março/2019. O PostgreSQL aparece em 4º lugar, já o MongoDB um banco de dados não SQL (NoSQL) permanece em 5º lugar.

Abstração de dados

O sistema de banco de dados deve garantir uma visão totalmente abstrata do banco de dados para o usuário, ou seja, para o usuário do banco de dados pouco importa qual unidade de armazenamento está sendo usada para guardar seus dados, contanto que os mesmos estejam disponíveis no momento necessário.

Esta abstração se dá em três níveis:

- 1- Nível de visão do usuário: as partes do banco de dados que o usuário tem acesso de acordo com a necessidade individual de cada usuário ou grupo de usuários;
- 2- Nível conceitual: define quais os dados que estão armazenados e qual o relacionamento entre eles;
- 3 - Nível físico: é o nível mais baixo de abstração, em que define efetivamente de que maneira os dados estão armazenados.

Modelo de Dados

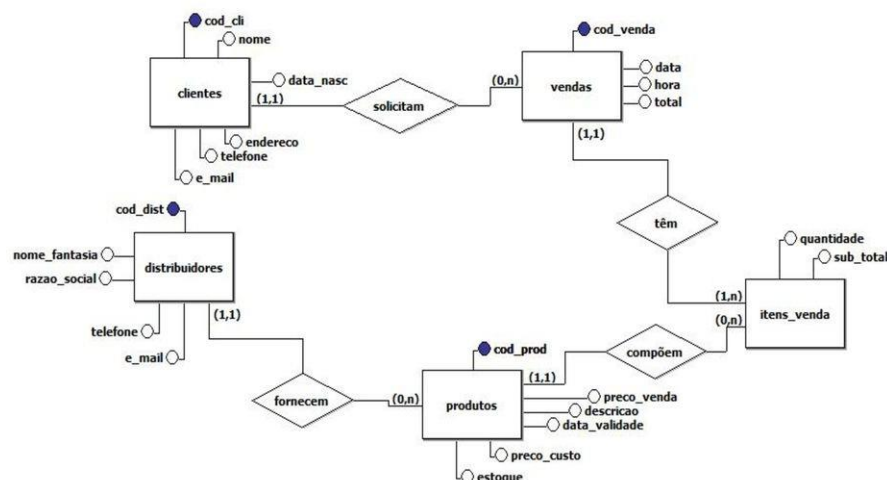
Um modelo de dados são as ferramentas que o profissional escolhe para a organização das informações e descrição de dados, relacionamentos...

Um projeto contém duas fases: Modelo Conceitual e Projeto Lógico.

Essas etapas preparam o banco de dados antes dele existir, evitando erros e facilitando possíveis manutenções futuras.

O modelo conceitual é a descrição do BD de maneira independente ao SGBD, ou seja, define quais os dados que aparecerão no BD, mas sem se importar com a implementação que se dará ao BD. Desta forma, há uma abstração em nível de SGBD. Uma das técnicas mais utilizadas dentre os profissionais da área é a abordagem entidade-relacionamento (ER), onde o modelo é representado graficamente através do diagrama entidade-relacionamento (DER).

Exemplo DER



O modelo lógico descreve o BD no nível do SGBD, ou seja, depende do tipo particular de SGBD que será usado. Não podemos confundir com o Software que será usado. O tipo de SGBD que o modelo lógico trata é se ele é relacional, orientado a objetos, hierárquico etc. O SGBD relacional é o mais conhecido. Nele, os dados são organizados em tabelas.

Modelagem de dados

Os modelos de dados são ferramentas que permitem demonstrar como serão construídas as estruturas de dados que darão suporte aos processos de negócios, como os dados estarão organizados e quais os relacionamentos que pretendemos estabelecer entre eles.

Os principais objetivos da modelagem de dados são: representar o ambiente observado, documentar e normalizar, fornecer processos de validação e observar processos de relacionamentos entre objetos.

Esses modelos consistem em 3 perspectivas:

1. **Conceitual:** Como representação de alto nível e com foco no ponto de vista do usuário criador dos dados. É sempre o primeiro modelo a ser desenvolvido e muito fácil de ser compreendido, pois não há limitações ou tecnologias específicas.
2. **Lógica:** agrega detalhes de implementação e leva em conta as regras e algumas limitações de recursos padronizados. Aqui é possível definir atributos que serão as chaves para a estrutura.
3. **Física:** demonstra os dados fisicamente e leva em consideração todas as regras e limitações do banco de dados. Obedece padrões e validações. E a partir daqui que o modelo deve estar espelhado para o seu banco de dados final.

Embora Modelo Lógicos e Modelos Físico parecerem similares, e eles de fato são, o nível de detalhes que eles modelam pode ser significativamente diferente. Isso porque o objetivo de cada diagrama é diferente – podemos usar um Modelo Lógicos para explorar conceitos do domínio com os envolvidos no projeto e Modelos Físico para definir o projeto do banco de dados.

Modelagem Transacional e Informacional

Os transacionais são aqueles projetados para atender à sistemas transacionais, ou seja, que tenham foco nos dados provenientes de interações de um sistema informatizado com suas fontes de dados. Ela prima armazenar e recuperar os dados referentes às transações. A preocupação nesse caso é com a normalização dos atributos, útil ao processamento de informações imediatas e não ao gerenciamento e tomada de decisões.

Por outro lado, a modelagem informacional – ou dimensional – tem objetivo oposto, ela visa atender a necessidade da tomada de decisão e para isso se baseia em grandes volumes de dados da base transacional. Nesse caso, o histórico é essencial e necessita de muita armazenagem.

Sistemas Cliente-Servidor

Na modelagem de dados também é importante saber sobre a estrutura fundamental dos sistemas, que normalmente são Cliente-Servidor, e consiste em estações de trabalho conectadas via rede aos servidores. Essas estações oferecem interfaces apropriadas para uso dos servidores, bem como processamento para aplicações.

Já no sistema cliente-servidor de três camadas, a indicação é para web, já que possui uma camada intermediária entre o cliente e o servidor de banco de dados. Essa camada é o servidor de aplicações ou servidor web e pode armazenar regras de negócio para acessar os dados do servidor.

Fundamentos

Certos conceitos são utilizados na modelagem conceitual para descrever os papéis de cada componente do domínio analisado. Dentre eles, podemos citar:

Instância: denomina uma ocorrência, um registro que existe ou passou a existir de uma entidade;

Entidade: define qualquer coisa que seja identificável, singular e tenha existência bem delimitada;

Atributo: é usado para referir-se à cada característica possuída pelas instâncias de uma entidade ou de um relacionamento e sua representação no DER se dá em forma de balões;

Relacionamento: descreve um evento significativo que ocorre entre instâncias de duas entidades e é representado por um losango;

Cardinalidade: conceito usado para dizer quantas vezes uma instância de uma entidade pode se relacionar com instâncias de outra entidade, também referenciado como “grau de relacionamento”;

Condicionalidade: responsável por comunicar se a existência de uma instância de uma entidade está condicionada à existência de uma instância de outra entidade.

Relacionamentos entre instâncias de entidades

São comumente efetivados com estas 3 cardinalidades:

“Um” para “Um” (1:1): uma instância de uma entidade pode ou deve se relacionar com uma e apenas uma instância de outra entidade;

“Um” para “Muitos” (1:N): uma instância de uma entidade pode ou deve se relacionar com uma ou mais instâncias de outra entidade — é com toda certeza a cardinalidade mais encontrada em bancos de dados; e

“Muitos” para “Muitos” (N:N): uma ou mais instâncias de uma entidade podem ou devem se relacionar com uma ou mais instâncias de outra entidade — esse tipo de relacionamento transforma-se em uma entidade a partir da modelagem lógica.

Condição de relacionamentos

Podem ser regidos sob 2 condicionalidades (exemplificadas abaixo):

Parcial: quando instâncias de uma entidade Telefone podem estar relacionadas com instâncias de Pessoa em uma agenda, ou seja, onde o cadastro de números de telefone sem nome é permitido; e

Total: quando instâncias de uma entidade Carro devem estar relacionadas com instâncias de Cliente para entrar em uma oficina, ou seja, onde o cadastro de carros sem dono é proibido.

Entidades de uma relação 1:N

Duas categorias são usadas para defini-las:

“Fortes”: para entidades do lado 1 da relação; e

“Fracas”: para entidades do lado N.

Projeto de Banco de Dados

O projeto de banco de dados é fundamental para o sucesso na implementação da tecnologia de banco de dados em uma empresa. Para pequenos sistemas muitas vezes o responsável pelo projeto parte diretamente para a criação do banco de dados físico e em seguida a criação das tabelas, colunas e índices. Porém quando se pensa em um banco de dados para uma empresa grande, dependendo do tamanho do projeto é fundamental um bom processo de projeto de banco de dados para que se possa garantir que o usuário terá todos os seus requisitos de informação atendidos na forma de informações disponíveis no banco de dados, além de outras considerações já vistas anteriormente tais como disponibilidade, desempenho e confiabilidade do banco de dados.

Um projeto de banco de dados é caracterizado por um processo que possui fases distintas e com aspectos diferentes, mas que tem como objetivo final a implementação de um banco de dados que atenda às necessidades de informação do usuário e aos requisitos não funcionais de disponibilidade, desempenho e confiabilidade esperados.

As Três fases que fazem parte de um projeto de banco de dados:

1 - **Modelo conceitual** – Representa os conceitos do negócio e as associações existentes entre estes conceitos. Também são representados os atributos assim como as regras de negócio que regulam as associações e conceitos do negócio. Este modelo é independente da tecnologia de implementação usada para o banco de dados e por isto é a etapa mais adequada para o envolvimento do usuário que não precisa ter conhecimentos técnicos. As características principais deste modelo são:

Visão Geral do negócio

Facilidade de entendimento entre usuários e desenvolvedores

Possui somente as entidades, relacionamentos e atributos principais

Os principais produtos da fase de projeto conceitual são:

- O diagrama de entidade e relacionamentos, também conhecido por modelo de entidade relacionamentos;
- Lista de Regras de Restrição de Integridade.

2- **Modelo Lógico** – Representa as estruturas de dados a serem implementadas e suas características considerando os limites impostos pelo modelo de dados usado para implementação do banco de dados. (banco de dados hierárquico, banco de dados de rede, banco de dados relacional etc.). As características principais deste modelo são:

- É derivado do modelo conceitual
- Possui entidades associativas em lugar de relacionamentos n:m
- Define as chaves primárias das entidades
- Define as chaves estrangeiras entre as entidades
- Normalização até a 3ª forma normal
- Adequado ao padrão de nomenclatura adotado pela empresa
- As Entidades e atributos são documentados em um Dicionário de Dados

O principal produto da fase de projeto lógico é o modelo relacional.

3- **Modelo Físico** – Este modelo representa a implementação do modelo lógico considerando algum tipo particular de tecnologia de banco de dados e os requisitos não funcionais (desempenho, disponibilidade, segurança) que foram identificados pelo analista de requisitos. As características principais deste modelo são:

- Elaborado a partir do modelo lógico

- Pode variar segundo a tecnologia usada para implementação do banco de dados
- Possui tabelas físicas (log, líder etc.)
- Possui colunas físicas (replicação)

No modelo físico, a linguagem SQL (Structured Query Language), é a linguagem padrão para definição, manipulação e controle de uso das estruturas de dados.

Existem muitos Sistemas Gerenciadores de Banco de Dados disponíveis no mercado. Como exemplo, podemos citar o PostgreSQL e o MySQL, que tem código aberto e são gratuitos. Também existe o Oracle, DB2, Sybase SQL Server, Informix e Microsoft SQL Server, que são pagos e não possuem código aberto, sendo bastante usados em corporações.

Etapas no projeto de um Banco de dados:

- Determinar qual o objetivo do projeto de Banco de Dados: Isto ajuda na determinação de quais os dados devem ser armazenados. É fundamental ter bem claro qual o objetivo a ser alcançado com o banco de dados. É fazer o acompanhamento das despesas, pedidos dos clientes, multas de trânsito, a evolução das vendas ou outro objetivo qualquer.
- Determinar as Entidades necessárias: Após definirmos os objetivos do Banco de Dados, as informações devem ser definidas e separadas em assuntos diferentes, tais como "Clientes", "Empregados", "Pedidos", pois cada um irá compor uma tabela no banco de dados. É importante considerar que conceitos diferentes devem ser representados como entidades e relacionamentos diferentes. Isto significa que não devemos misturar conceitos, assuntos, representando todos como apenas uma única entidade.
- Determinar os Atributos de cada Entidade: Definir quais informações devem ser mantidas em cada entidade. Por exemplo, a entidade Clientes poderia ter um atributo para o CPF Do Cliente, outro para o Nome Do Cliente e assim por diante.
- Determinar o identificador de cada Entidade: Determinar, em cada entidade, qual(is) atributo(s) será(ão) utilizado(s) como identificador de cada ocorrência na entidade.
- Determinar os Relacionamentos: Determinar as associações existentes entre as entidades do modelo de dados, identificando de que forma as entidades se relacionam, e representando os relacionamentos entre as entidades, detalhando a cardinalidade entre os relacionamentos. A cardinalidade especifica o número de vezes que uma ocorrência de uma entidade pode estar associada a ocorrências da outra entidade. Por exemplo, Clientes podem Fazer

Vários Pedidos, então existe um relacionamento do tipo Um-para-vários entre a tabela Clientes (lado um) e a tabela Pedidos (lado vários).

- Normalizar a Estrutura do Banco de Dados: Antes de inserir muitos dados, ou até mesmo antes de inserir qualquer dado, verificar se a estrutura contém redundância de atributos nas entidades, eliminando a possibilidade de inconsistência nos dados causada por anomalias de atualização devido a redundância de atributos. Isto, normalmente, pode ser obtido através do processo de Normalização.

DER

DER, é a abreviação de Diagrama de Entidade Relacionamento. Esse diagrama nada mais é do que uma representação gráfica do MER (Modelo de entidade Relacionamentos). Em termos conceituais, podemos dizer que “Diagrama de Entidade Relacionamento” é um modelo diagramático que descreve um modelo de dados.

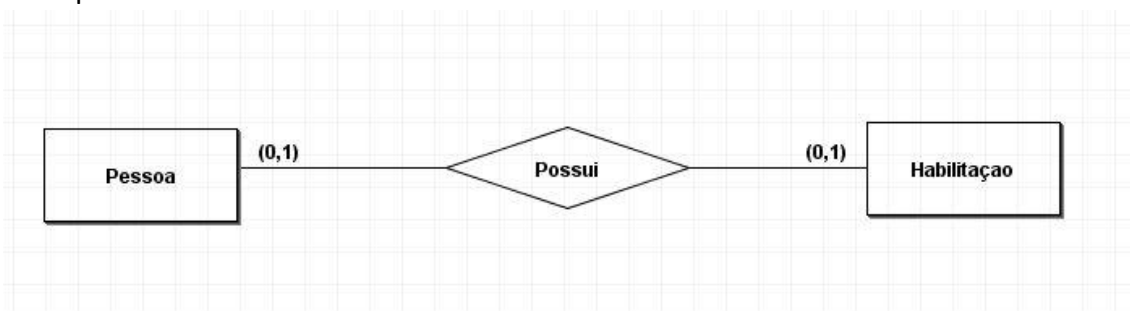
Na elaboração de um projeto de banco de dados, diagramas ER são usados para modelar e criar bancos de dados relacionais, na parte das regras lógicas desse negócio, em um modelo lógico de dados e no termo de tecnologia específica a ser colocada (no modelo físico de dados).

Em um modelo DER possuímos as entidades sendo substantivos concretos ou abstratos onde é possível guardar dados. Existem as entidades fortes, entidades fracas e entidades associativas.

Um modelo DER possui também os relacionamentos que são uma ação entre as entidades, representados por um losango. Cada relacionamento possui uma cardinalidade que representa o número de interações entre as entidades. As cardinalidades podem ser de: 1-1(um para um), N-1(um para muitos) ou N-N(muitos para muitos).

Cada entidade possui uma característica chamada de atributo. Eles podem ser descritivos, nominativos ou referenciais. Além disso existem os atributos Simples e Compostos. Quando um atributo representa um valor único da entidade é chamado de Atributo Identificador.

Exemplo de DER:

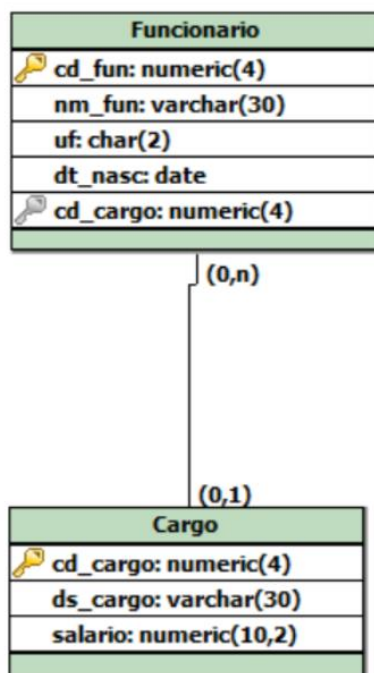


MLR

Modelo lógico relacional

Um modelo relacional é um modelo de dados representativo (ou de implementação), adequado a ser o modelo subjacente de um SGBD, um banco de dados construído puramente baseado no modelo relacional estará inteiramente normalizado. A ideia de modelo relacional se baseia no princípio de que as informações em uma base de dados podem ser consideradas como relações matemáticas e que podem ser representadas, de maneira uniforme, através do uso de tabelas onde as linhas representam as ocorrências de uma entidade e as colunas representam os atributos de uma entidade do modelo conceitual.

Exemplo de modelo lógico relacional:



Principais Vantagens do Modelo Relacional:

- Independência total dos dados;
- Visão múltipla dos dados;
- Redução acentuada na atividade de desenvolvimento. Particularmente para extração de dados para relatórios e consultas específicas do usuário;
- Maior segurança no acesso aos dados;

- Maior agilidade para consulta/atualização;
- Qualidade dos dados garantida por restrições de integridade.

História sobre a criação dos Bancos de Dados e Modelo Físico

Antigamente muitas empresas armazenavam seus dados em arquivos físicos, porém com o tempo a necessidade de haver alguma ferramenta para facilitar esse processo foi aumentando.

Por volta do final dos anos 1970 surgiu um sistema baseado nas ideias de Edgar Frank Codd, um pesquisador da IBM que no começo da década havia realizado um artigo intitulado “**A Relational Model of Data for Large Shared Data Banks**”. Com esse sistema chamado de “**Sistema R**” também foi criado a linguagem de consulta estruturada (SQL - Structured Query Language).

Com o passar do tempo muitos sistemas de banco de dados foram baseados no Sistema R. Hoje possuímos dois tipos de SGBDS (Sistema gerenciador de banco de dados), sendo eles:

- **Relacionais:** Modelam seus dados em formas de tabelas que se relacionam entre si, possuindo diversos atributos e tipos de dados diferentes.
- **Não-relacionais(NoSQL):** São sistemas onde não é utilizado SQL como linguagem de consulta, são bem conhecidos por sua facilidade no desenvolvimento, utilizando diversos modelos de dados, entre eles: documentos, gráficos, chave-valor e colunares.

Dentre os Principais SGBDS utilizados pelo mundo estão: OracleDB, MySQL, SQLServer, PostgreSQL e DB2. Cada um se destacando por diversos motivos, seja pela facilidade, segurança, recursos avançados, etc.

Regras Modelo Físico

- Restrições são regras aplicadas nas colunas de uma tabela.
- São usadas para limitar os tipos de dados que são inseridos.

Principais Constraints do SQL:

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- DEFAULT

Restrição NOT NULL:

- A constraint NOT NULL impõe a uma coluna a NÃO aceitar valores NULL.
- A constraint NOT NULL obriga um campo a sempre possuir um valor.
- Deste modo, não é possível inserir um registro (ou atualizar) sem entrar com um valor neste campo.

Restrição UNIQUE

- A restrição UNIQUE identifica de forma única cada registro em uma tabela de um banco de dados.
- As constraints UNIQUE e PRIMARY KEY garantem a unicidade em uma coluna ou conjunto de colunas.
- Uma constraint PRIMARY KEY automaticamente possui uma restrição UNIQUE definida.
- Você pode ter várias constraints UNIQUE em uma tabela, mas apenas uma Chave Primária por tabela.

Restrição PRIMARY KEY

- A constraint PRIMARY KEY identifica de forma única cada registro em uma tabela de banco de dados.
- Chaves Primárias devem conter valores únicos.
- Uma coluna de chave primária não pode conter valores NULL
- Cada tabela deve ter uma chave primária e apenas uma chave primária.

Restrição FOREIGN KEY

- Uma FOREIGN KEY (Chave Estrangeira) em uma tabela é um campo que aponta para uma chave primária em outra tabela.

Restrição CHECK

- A constraint CHECK é usada para limitar uma faixa de valores que podem ser colocados em uma coluna.
- Se uma constraint CHECK for definida em uma única coluna ela permitirá apenas determinados valores para a coluna.
- Se a constraint CHECK for definida para a tabela, ela poderá limitar os valores em algumas colunas com base nos valores de outras colunas do registro.

Restrição DEFAULT

- A restrição DEFAULT é usada para inserir um valor padrão em uma coluna.
- O valor padrão será adicionado a todos os novos registros caso nenhum outro valor seja especificado.

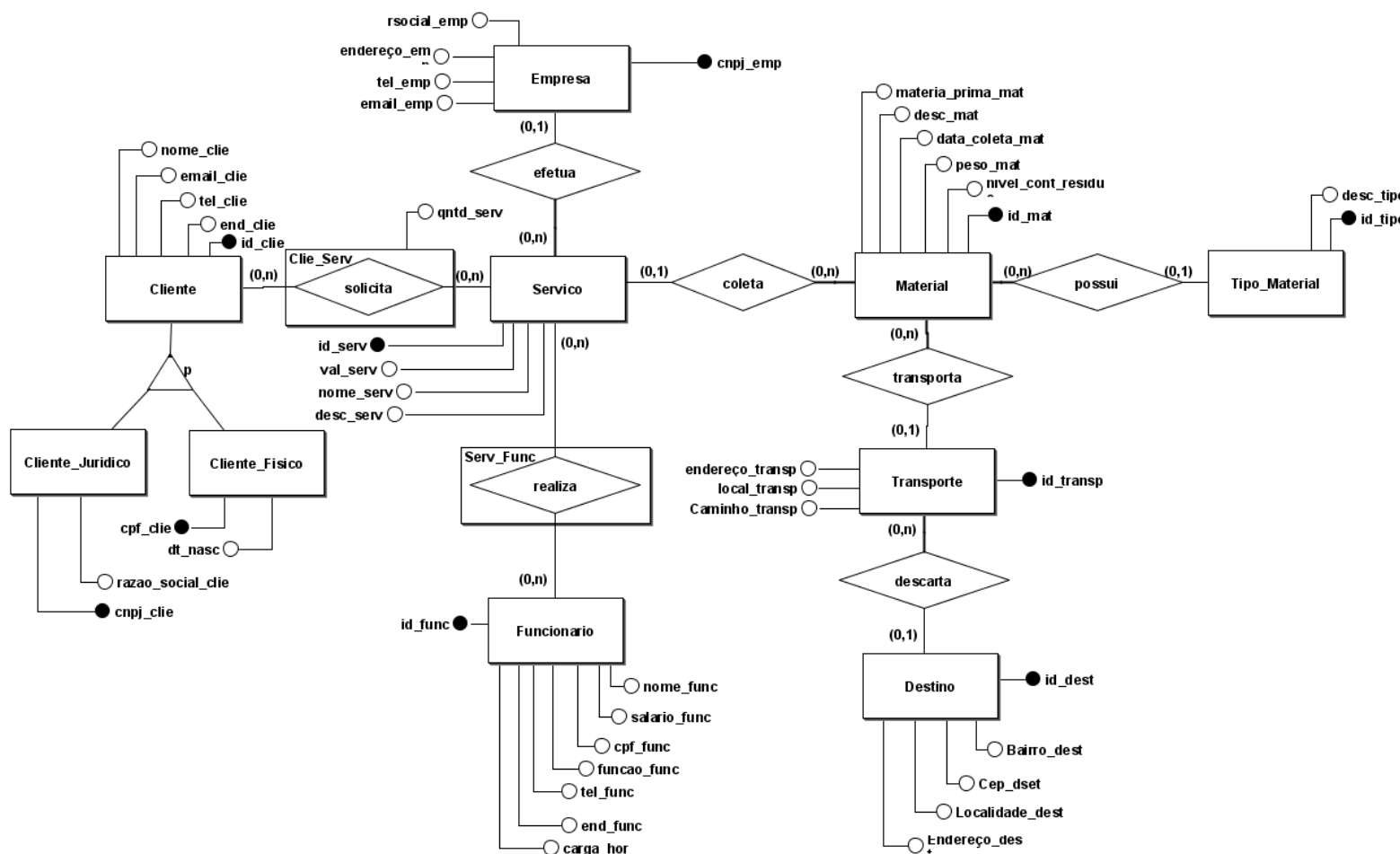
Modelo Físico

O modelo físico é um modelo de baixo nível onde se leva em consideração as limitações do SGBD e é formado por comandos SGBD para criar tabelas usando como referência o modelo lógico.

Cada SGBD tem seus tipos de dados mas os principais são: numérico, data e tempo e string podendo ter a quantidade de caracteres fixa ou variável.

GRUPO B – BANCO DE DADOS – DER

Gabriel Vinícius _ 20143	Maria Eduarda _ 20007
Rafaela Araújo _ 20136	Alex Pereira Corte _ 20059
Alice Primo _ 20012	Mariana _ 20065
Matheus Garcia _ 20024	Leticia _ 20008
Beatriz Fonseca _ 20050	Gabriel Abramo _ 20442
Gabriel Salomão _ 20056	



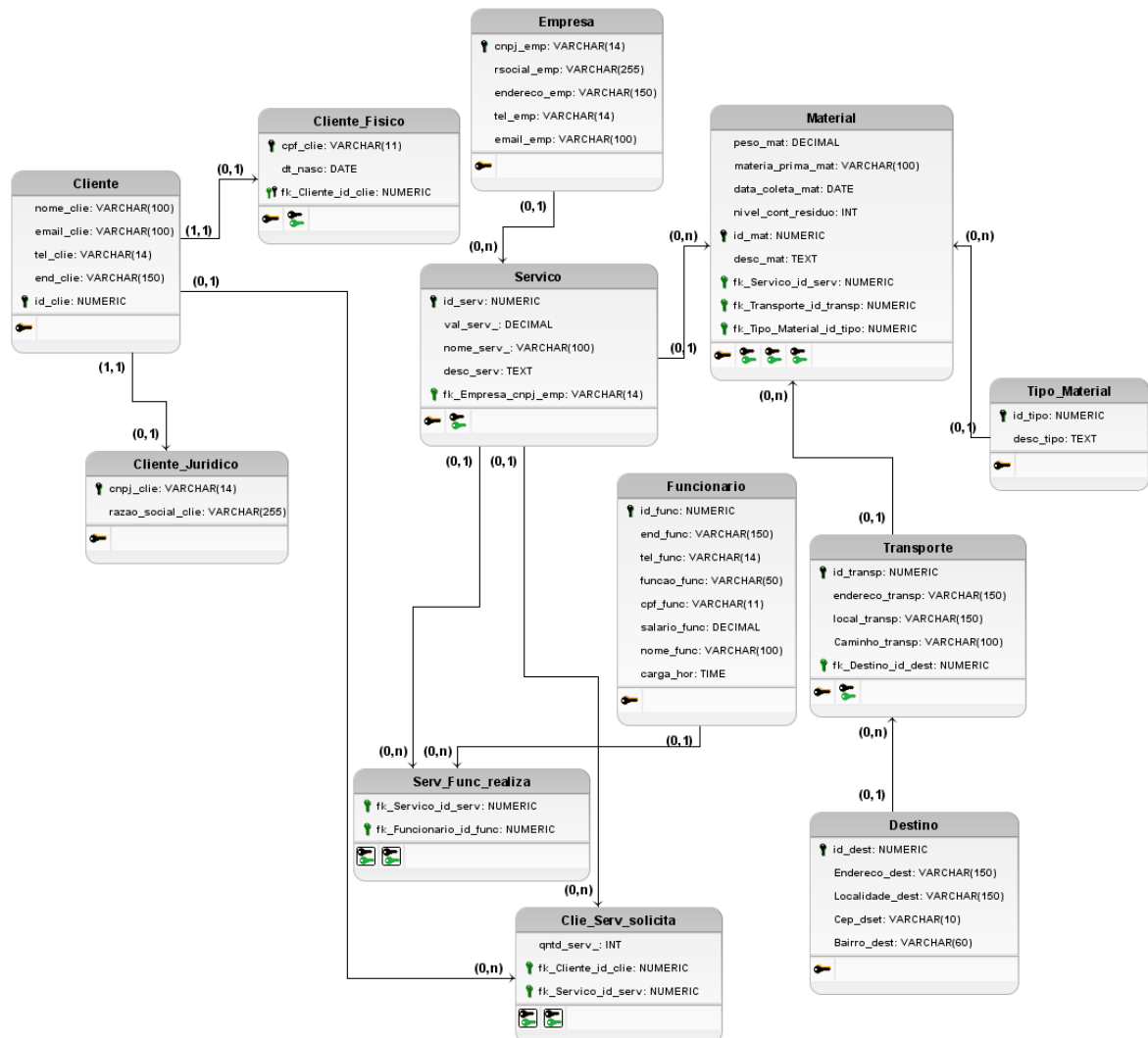
Projeto Interdisciplinar – Banco de Dados
ETEC de Guarulhos
1º M-TEC - Desenvolvimento de Sistemas

Grupo C – BD Interdisciplinar

Integrantes –

Renan Rithelli Correia Rocha;
Hihê de Lima Oliveira;
Gabriel Lima de Souza;
Kauã Andrade Pacheco;
Nicolas Rodrigues Sanchez;
Bryan Ferreira Barboza;
João Pedro Castro Barboza;
Kauã Sabino;
Wendell de Sousa Ferreira;
Pedro Henrique Marques de Oliveira;
Enrico Ribeiro Bueno;
João Pedro Alencar.

MLR – Modelo Lógico Relacional



cliente

id_clie	N	5	PK
nome_clie	A	30	NN
email_clie	A	40	
tel_clie	C	14	
endereco_clie	A	30	

cliente_juridico

cnpj_clie	C	14	PK
rsocial_clie	A	50	

cliente_fisico

cpf_clie	C	11	PK
id_clie			FK

empresa

cnpj_emp	C	14	PK
rsocial_emp	A	50	NN
email_emp	A	40	
tel_emp	C	14	
endereco_emp	A	30	

serviço

id_serv	N	5	PK
cnpj_emp			FK
nome_serv	A	20	NN
val_serv	N	12,2	
desc_serv	A		

solicita_clie_serv

id_clie			FK
id_serv			FK
quantidade	N	8	NN

funcionario

id_func	N	5	PK
nome_func	A	30	NN
cpf_func	C	11	NN
email_func	A	30	

MF – Modelo Físico – Dados

tel_func	C	14	
endereco_func	A	30	
salario_func	N	10,2	
funcao_func	A	20	
carga_hor	A	15	

realiza_serv_func

id_serv			FK
id_func			FK

destino

id_dest	N	5	PK
endereco_dest	A	30	
cep_dest	C	8	NN
localidade_dest	A	30	
bairro_dest	A	20	

transporte

id_transp	N	5	PK
id_dest			FK
endereco_transp	A	30	
local_transp	A	30	
caminho_transp	A	50	

tipo_material

id_tipo	N	5	PK
desc_tipo	A		

material

id_mat	N	5	PK
id_serv			FK
id_transp			FK
id_tipo			FK
peso_mat	N	5,2	
materia_prima	A	25	
data_coleta	D		
desc_mat	A		

```
create table cliente ( id_clie numeric(5)
                        constraint clie_id_pk primary key,
                        nome_clie varchar(30)
                        constraint clie_nome_nn not null,
                        email_clie varchar(40),
                        tel_clie char(14),
                        endereco_clie varchar(30)
                        );
```

```
create table cliente_juridico ( cnpj_clie char(14)
                             constraint clientejuridico_cnpj_pk primary key,
                             rsocial_clie varchar(50)
                             );
```

```
create table cliente_fisico ( cpf_clie char(11)
                           constraint clientefisico_cpf_pk primary key,
                           id_clie
                           clientefisico_id_fk references cliente
                           );
```

```
create table empresa ( cnpj_emp char(14)
                        constraint emp_cnpj_pk primary key,
                        rsocial_emp varchar(50)
                        constraint emp_rsocial_nn not null,
                        email_emp varchar(40),
                        tel_emp char(14),
                        endereco_emp varchar(30)
                        );
```

[illegible][illegible]

);

```
create table funcionario ( id_func numeric(5)
                        constraint func_id_pk primary key,
                        nome_func varchar(30)
                        constraint func_nome_nn not null,
                        cpf_func char(11)
                        constraint func_cpf_nn not null,
                        email_func varchar(30),
                        tel_func char(14),
                        endereco_func varchar(30),
                        salario_func numeric(10,2),
                        funcao_func varchar(20),
                        carga_hor varchar(15)
                        );
```

```
create table realiza_serv_func ( id_serv
                        constraint real_idserv_fk references servico,
                        id_func
                        constraint real_idfunc_fk references funcionario
                        );
```

```
create table destino ( id_dest numeric(5)
                        constraint dest_id_pk primary key,
                        endereco_dest varchar(30),
                        cep_dest char(8)
                        constraint dest_cep_nn not null,
                        localidade_dest varchar(30),
                        bairro_dest varchar(20)
                        );
```

```
create table transporte ( id_transp numeric(5)
                        constraint transp_id_pk primary key,
                        id_dest
                        constraint transp_iddest_fk references destino,
                        endereco_transp varchar(30),
                        local_transp varchar(30),
                        caminho_transp varchar(50)
                        );
```

```
create table tipo_material ( id_tipo numeric(5)
                        constraint tipo_id_pk primary key,
                        desc_tipo text
                        );
```

```
create table material ( id_mat numeric(5)
                        constraint mat_id_pk primary key,
                        id_serv
                        constraint mat_idserv_fk references servico,
                        id_transp
                        constraint mat_idtransp_fk references transporte,
                        id_tipo
                        constraint mat_idtipo_fk references tipo_material,
                        peso_mat numeric(5,2),
                        materia_prima varchar(25),
                        data_coleta date,
                        desc_mat text
                        );
```

MF – Modelo Físico – Query SQL

```
select * from cliente;  
select * from cliente_juridico;  
select * from cliente_fisico;  
select * from empresa;  
select * from servico;  
select * from solicita_clie_serv;  
select * from funcionario;  
select * from realiza_serv_func;  
select * from destino;  
select * from transporte;  
select * from tipo_material;  
select * from material;
```
