Banco de Dados
Conceitos e Modelagem

Paulo Meirelles, IME-USP
paulormm@ime.usp.br

** Baseado nos slides da Profa. Kelly Rosa Braghetto (IME-USP)*
** Usa um exemplo do post: Entendendo Diagramas Entidade-Relação com exemplos carnavalescos*

O que é um **Dado**?

O que é um **Banco de Dados**?

Parte 1

Banco de Dados

Agenda de contatos: telefones e endereços pessoais

Catálogo de biblioteca: informações do acervo de livros

Dados da Receita Federal: declarações de imposto de renda

Registros acadêmicos: matrículas e notas de alunos

Controle de loja: informações de estoque e vendas

Prontuários médicos: histórico de pacientes em hospitais

Dados meteorológicos: registros coletados em São Paulo

Banco de Dados

Estão em **toda parte**, organizando informações essenciais para o funcionamento de serviços e instituições.

A internet atinge **5,52 bilhões de usuários em 2024.**

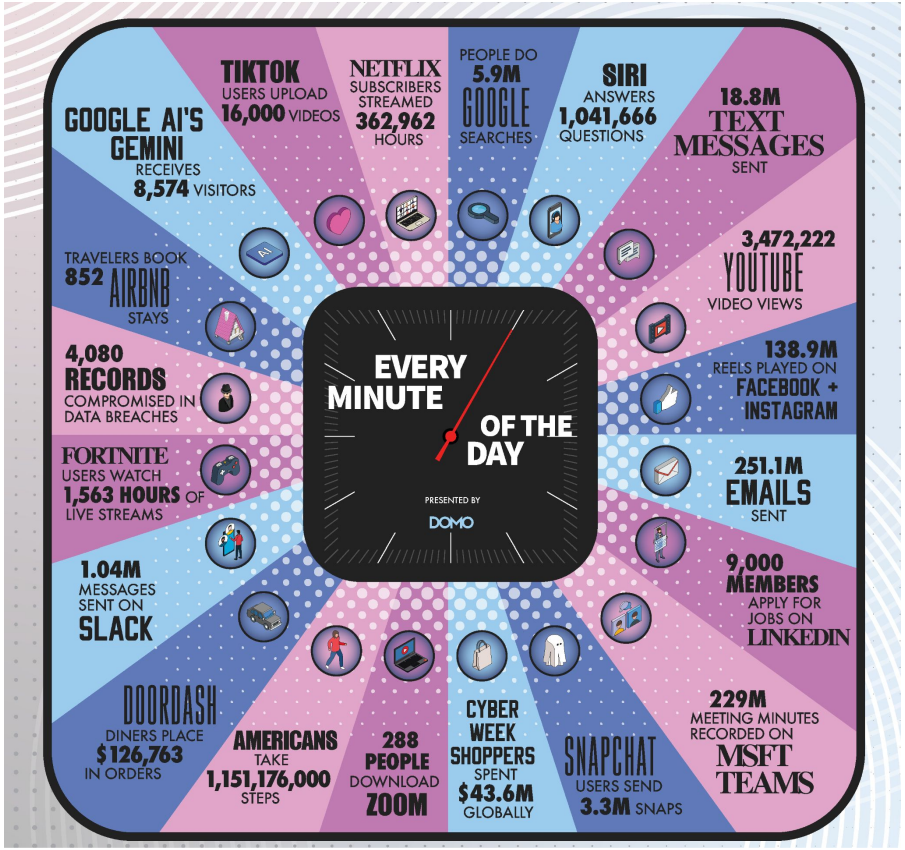
Volume massivo de dados gerados a cada 60 segundos.

IA generativa começa a superar plataformas digitais tradicionais.

Novas formas de engajamento online estão emergindo.

Transformações nas atividades digitais cotidianas.

Aumento de investimentos corporativos em dados e tecnologias orientadas à IA.



Fonte: https://www.domo.com/learn/infographic/data-never-sleeps-12

O que é um Banco de Dados?

Definição formal (Elmasri & Navathe, 2016):

- **Banco de Dados:** coleção de dados relacionados.
- **Dados:** fatos conhecidos que podem ser registrados e que possuem significado implícito.

O que é um Banco de Dados?

Um banco de dados armazena i**nformações sobre algum domínio de interesse** (ex.: biblioteca, hospital, universidade).

Os dados são **organizados de forma estruturada, com regras claras** para facilitar acesso e manipulação.

O objetivo é **reduzir redundância, garantir integridade, e disponibilizar os dados para múltiplos usos**.

Um banco de dados é geralmente acompanhado por um **SGBD**, que controla o acesso e a integridade dessas informações.

O que é um Banco de Dados?

Um banco de dados **não é apenas um repositório de dados**, mas uma estrutura planejada e significativa.

Propriedades Implícitas de um Banco de Dados

Representa um aspecto do mundo real (minimundo)

- O banco de dados modela um domínio específico, como uma escola, hospital ou loja.

Coleção lógica e coerente de dados com significado

- Dados organizados e relacionados entre si.
- Uma coleção aleatória de dados não é um banco de dados!

Construído com um propósito definido

- Projetado para atender a usuários e aplicações específicas.

BD não é, necessariamente, um SGBD

Um banco de dados informatizado pode ser criado e mantido sem o uso de um SGBD:

- Aplicações específicas: programas feitos sob medida para armazenar e acessar dados
  - Um sistema feito em Python que grava dados em arquivos
  - Planilhas ou arquivos de texto com estrutura própria
- Ainda que contenham dados organizados, **não oferecem as garantias e funcionalidades de um SGBD**

BD não é, necessariamente, um SGBD

Um banco de dados informatizado pode ser criado e mantido sem o uso de um SGBD:

- **Limitações dessas abordagens**
  - Falta de controle de acesso, segurança e concorrência
  - Difícil reutilização e manutenção dos dados
  - Redundância e inconsistência mais frequentes

Sistema de Gerenciamento de Banco de Dados

Um SGBD é um sistema de software de propósito geral:

- Define a estrutura dos dados (**modelagem e esquemas**)
  - Data Definition Language (DDL)
- Constrói e organiza os dados de acordo com essa estrutura
- Manipula os dados por meio de i**nserções, consultas, atualizações e remoções**
  - Data Manipulation Language (DML)
- Compartilha os dados de forma controlada entre múltiplos usuários e aplicações

Tipos de SGBDs

Relacionais [nosso foco!]

- Modelo baseado em tabelas (relações com linhas e colunas).
- Usa a linguagem **SQL** (Structured Query Language).

Tipos de SGBDs

Relacionais:

- PostgreSQL
- MySQL
- SQLite
- Oracle
- SQL Server
- H2 (embutido ou modo cliente-servidor)

Tipos de SGBDs

NoSQL ("Not Only SQL")

- Criados para alta escalabilidade e flexibilidade de esquema
- Usados em contextos como Big Data, aplicações web em larga escala
- Não substitui o modelo relacional: são abordagens complementares com diferentes finalidades

Tipos de SGBDs

NoSQL ("Not Only SQL")

- Documentos: MongoDB, Couchbase
- Chave-Valor: Redis, Riak
- Colunar: Cassandra, HBase
- Grafos: Neo4j, Amazon Neptune

Tipos de SGBDs

Tipo de Banco	Usa SQL?	Observação
Relacional (ex.: PostgreSQL)	✅ Sim	SQL é o padrão principal.
NoSQL Documento (ex.: MongoDB)	❌ Não (usa JSON queries)	Tem comandos próprios, não SQL.
NoSQL Colunar (ex.: Cassandra)	⚠️ Parcial (CQL)	Similar ao SQL, mas com restrições.
NoSQL Grafo (ex.: Neo4j)	❌ Não (usa Cypher)	Linguagem declarativa específica.

Sistema de Gerenciamento de Banco de Dados

Atua como um "**intermediário inteligente**" entre os usuários e os dados armazenados.

Parte 2

Arquitetura de Sistemas de Bancos de Dados

Segundo o modelo ANSI(American National Standards Institute)/SPARC (Standards Planning and Requirements Committee), é organizada em três níveis: **Externo, Conceitual e Interno**

- Uma forma de organizar a estrutura de um banco de dados de maneira independente da aplicação e dos usuários, promovendo **abstração, flexibilidade e segurança**

Níveis da Arquitetura

Nível Externo (Visão dos Usuários)

Cada usuário ou aplicação vê apenas **uma parte dos dados** relevante para suas necessidades

Cada um desses usuários interage com a mesma base de dados, mas vê apenas o que é relevante para seu papel, por meio de uma **visão externa personalizada**

Visões externas são como janelas filtradas para o banco de dados completo, adaptadas às **necessidades e permissões de cada usuário ou aplicação**

Níveis da Arquitetura

View SQL:

-- Visão de livros disponíveis para qualquer aluno: oculta todos os dados que não interessam ou que não devem ser acessados (como empréstimos de outros usuários ou status internos dos livros).

```
CREATE VIEW livros_disponiveis AS

SELECT id, titulo, autor, ano_publicacao

FROM livros

WHERE status = 'disponível';
```

Níveis da Arquitetura

View SQL:

– Uma bibliotecária tem uma visão mais ampla, com acesso a todos os dados, inclusive nomes dos usuários e todos os empréstimos.

```
CREATE VIEW todos_emprestimos AS
SELECT e.id, l.titulo, u.nome, e.data_emprestimo, e.data_devolucao
FROM emprestimos e
JOIN livros l ON e.id_livro = l.id
JOIN usuarios u ON e.id_usuario = u.id;
```

Níveis da Arquitetura

Nível Conceitual (Visão Lógica)

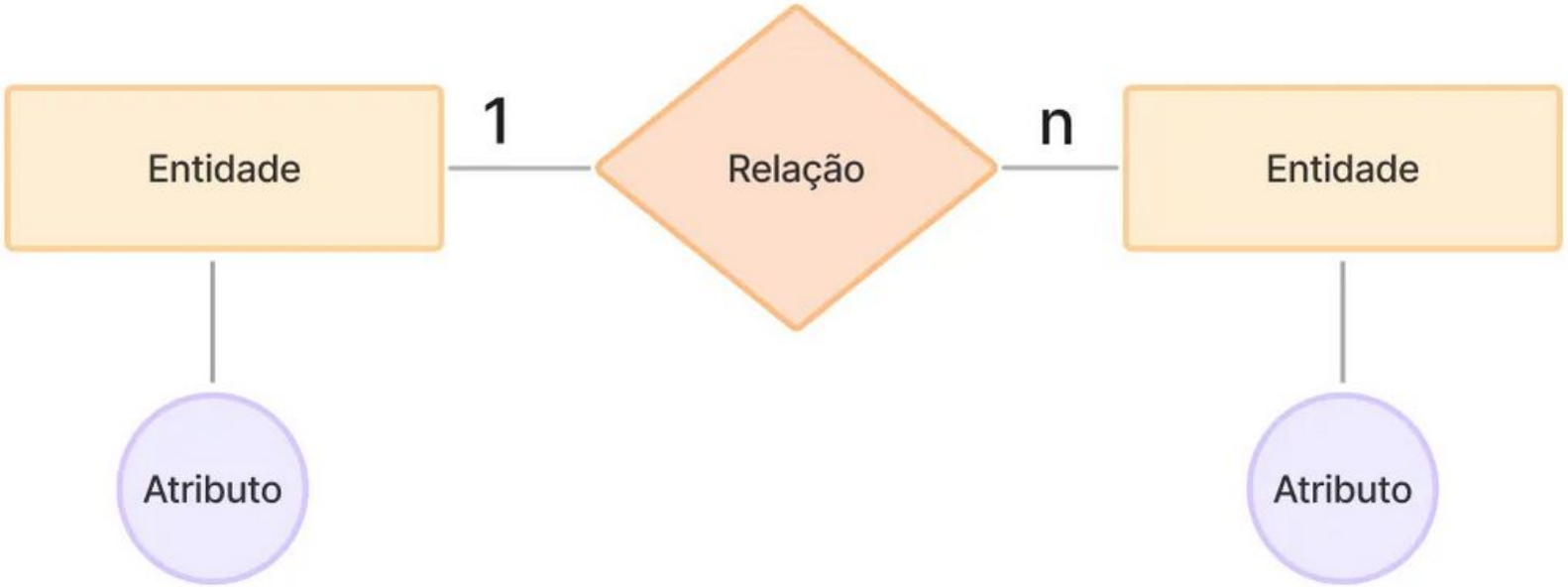
Descreve a estrutura lógica global do banco de dados

Define **entidades, atributos, relacionamentos e restrições**

Modelagem conceitual (Modelo ER) é o nosso foco

Níveis da Arquitetura

Nível Conceitual (Visão Lógica)



Níveis da Arquitetura

Nível Interno (Visão Física)

Define como os dados são armazenados fisicamente no sistema.

Envolve **índices, arquivos, endereços e estratégias de acesso**.

Níveis da Arquitetura

Nível Interno (Visão Física)

-- Indexação para acelerar consultas

```
CREATE INDEX idx_livros_titulo ON livros (titulo);
```

- SGBD **mantém uma estrutura de dados auxiliar** (como uma árvore B+) para localizar títulos rapidamente, sem fazer varredura linha a linha.
  - Árvore B+ organiza os dados de forma ordenada e balanceada.
  - Em vez de percorrer todos os registros, o SGBD navega pelos nós da árvore até encontrar rapidamente o valor desejado
  - As folhas da árvore B+ contêm os ponteiros reais para os dados, e estão todas ligadas sequencialmente, facilitando também consultas por faixa

Níveis da Arquitetura

Nível	Descrição	Exemplo
Externo	O que o usuário vê	Lista dos livros disponíveis
Conceitual	O que o banco representa logicamente	Tabelas `livros`, `usuários`, `empréstimos`, e os relacionamentos entre elas
Interno (Físico)	Como os dados são armazenados	Arquivos em disco, índices por título, cache, registros binários

Níveis da Arquitetura

A **modelagem começa no nível conceitual**, mas é projetada para ser adaptável às visões dos usuários (externo) e à estrutura física (interno)

Parte 3

Projeto de Banco de Dados

Projeto Conceitual

- Independente de SGBD
- Modelo Entidade-Relacionamento (MER)

Projeto Lógico

- Traduz o modelo conceitual para um modelo lógico (ex: relacional)
- Aqui já pensamos em tabelas, colunas, tipos

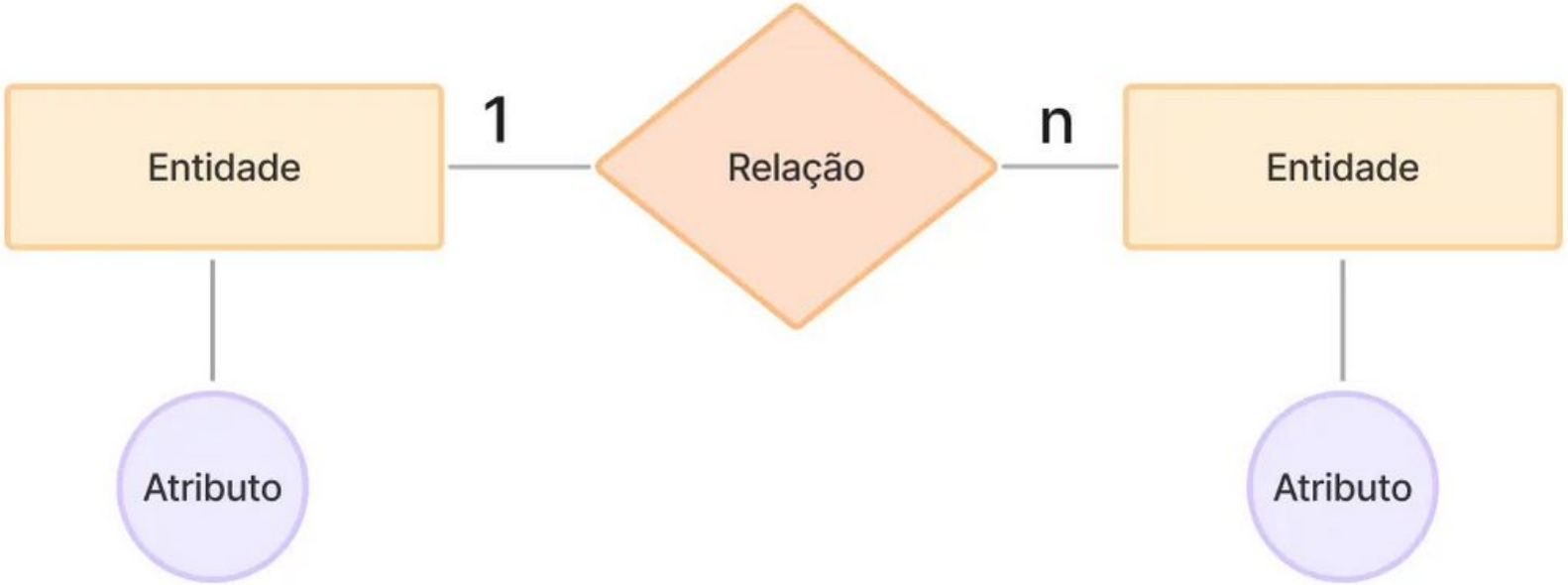
Projeto Físico

- Ajustes para desempenho e armazenamento
- Índices, partições, estruturas internas

Modelo Entidade-Relacionamento (MER)

- Criado por Peter Chen em 1976
- Representa **entidades, seus atributos e os relacionamentos** entre elas
- É a primeira etapa do projeto conceitual de um banco de dados
- Ajuda a transformar um **domínio real em uma estrutura lógica e clara**

Modelo Entidade-Relacionamento (MER)



Entidade

- Representa algo do **mundo real** que queremos registrar no banco de dados
- Pode ser uma pessoa, objeto, lugar, evento... ou até um bloco de Carnaval
- Toda entidade deve ter um **identificador único**:
  - Ex: Pessoa → CPF
  - Bloco de Carnaval → Nome do bloco

Entidade

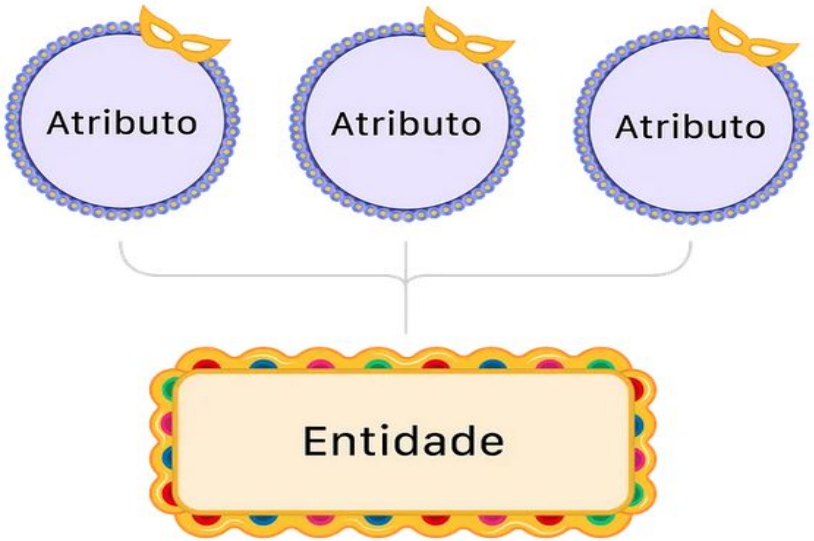
- Entidade é representada por um retângulo
  - “Folião” é um exemplo de entidade
  - “Paulo”, que é um folião, é uma ocorrência de entidade folião

Atributo

É uma característica ou informação que descreve uma entidade.

- Entidade CRIATURA
- Atributo: COR DO CABELO
- "Anjo moreno", "diabo loiro"

Atributo



- Na notação de Chen, um atributo fica dentro de uma elipse, ligado à entidade por um traço

Atributo

- Algumas ocorrências de criaturas podem ser “anjo”, “monstro”, ou “diabo”.
- Agora que temos um atributo COR DO CABELO para essas criaturas, podemos ter um “anjo moreno”, ou mesmo um diabo loiro.

Tipos de Atributos

Atributo chave

- Identificador único da entidade.
- Ex: Nome do bloco → “Trinca de Ás”

Atributo composto

- Pode ser decomposto em partes menores.
- Ex: Endereço da sede → Rua, Número, Bairro...

Tipos de Atributos

Atributo multivalorado

- Permite mais de um valor para a mesma entidade.
- Ex: Cores do bloco “Eu Acho é Pouco” → Vermelha e Amarela

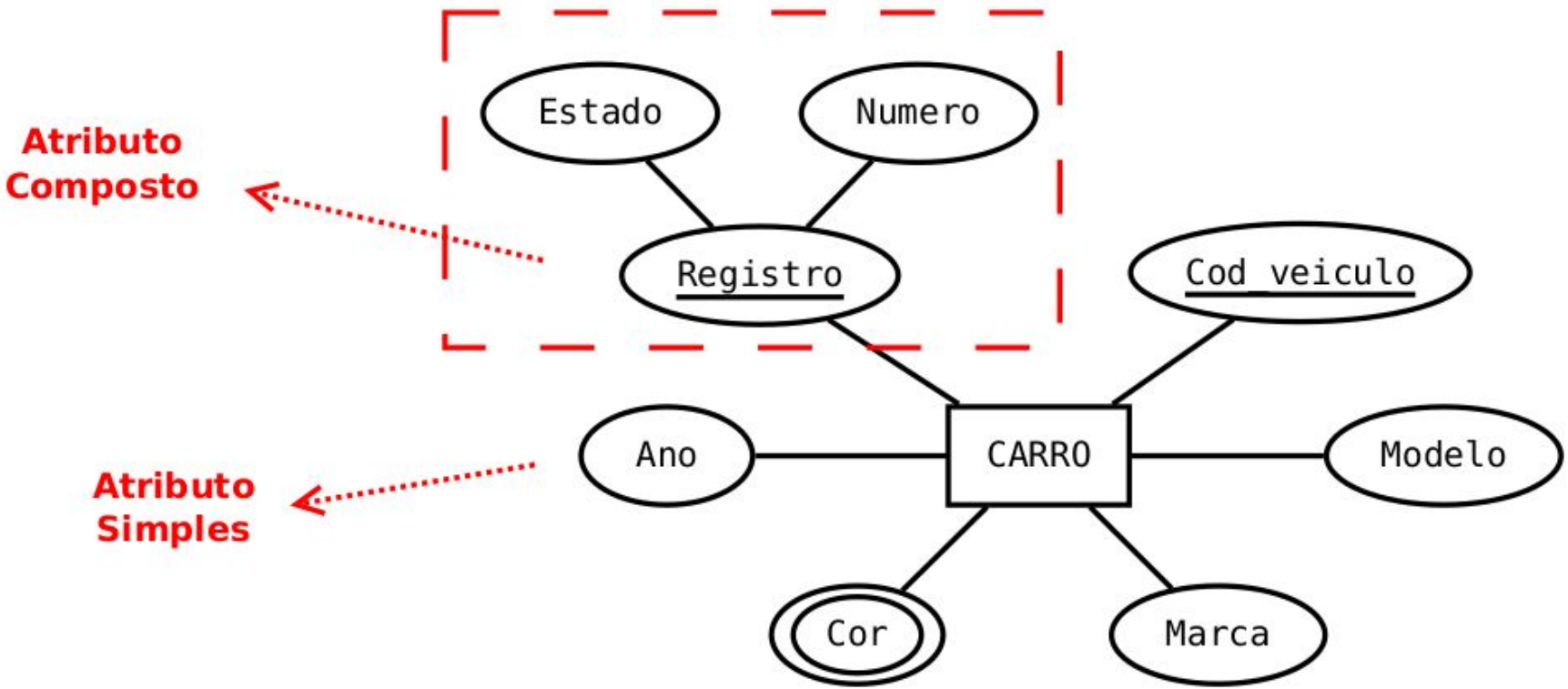
Atributo derivado

- Calculado a partir de outro atributo.
- Ex: Ano do bloco = 2023 − Data de Fundação (1952)

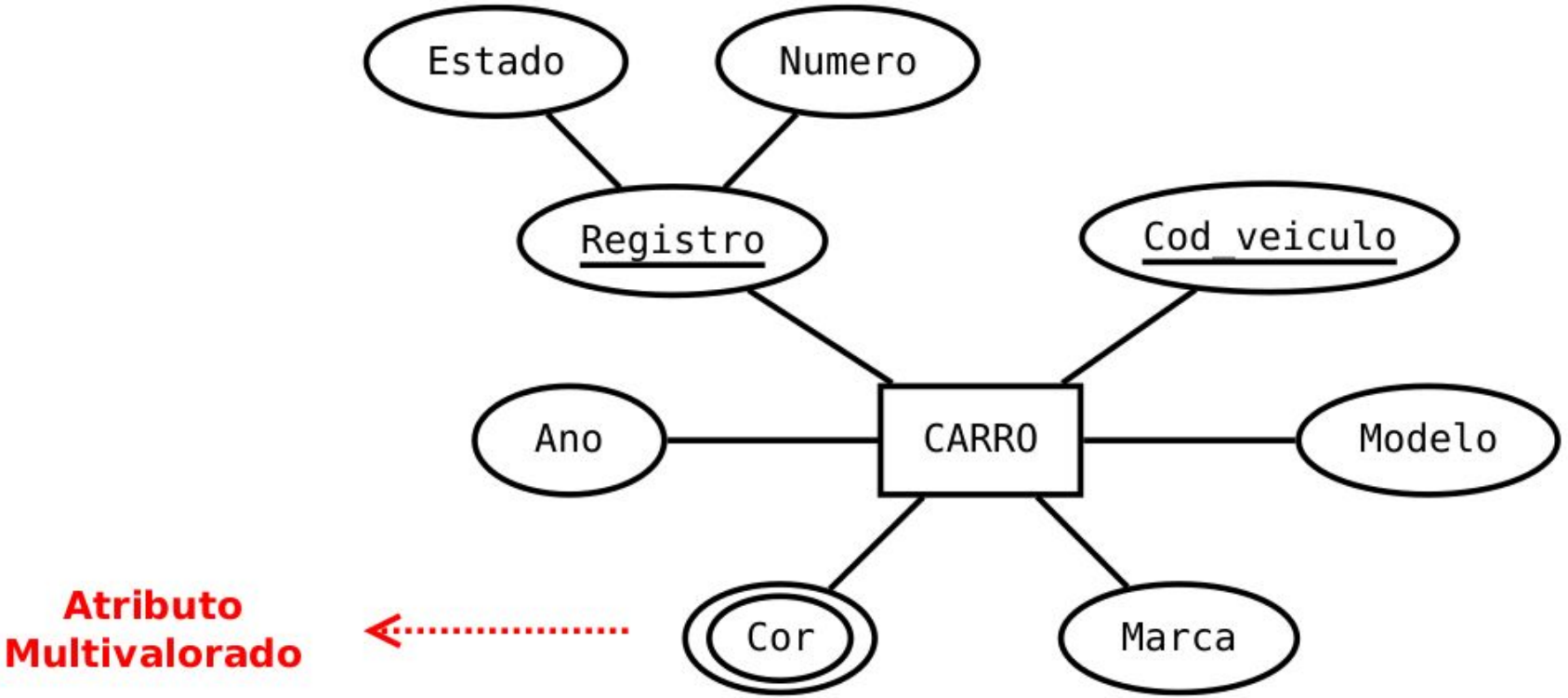
Atributo de atributo

- Um atributo que possui suas próprias propriedades.
- Ex: Estandarte (atributo do bloco) → Cores (atributo do estandarte)

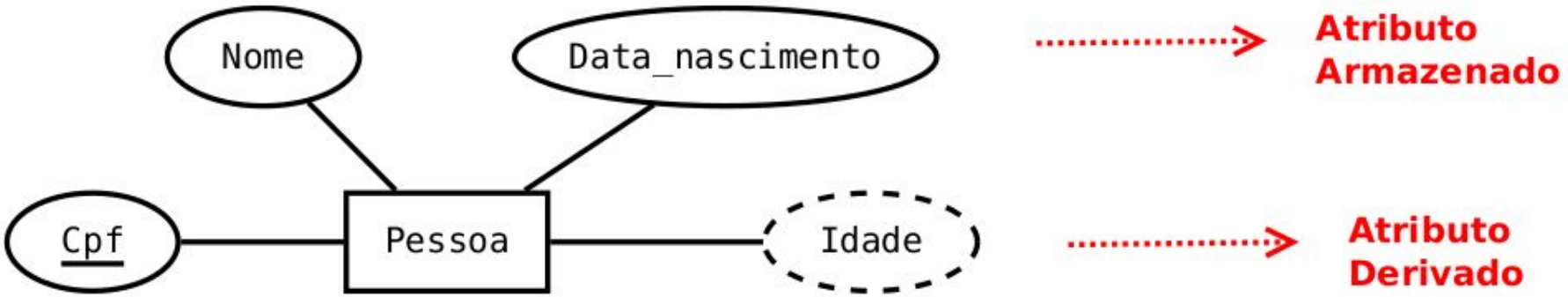
Tipos de Atributos



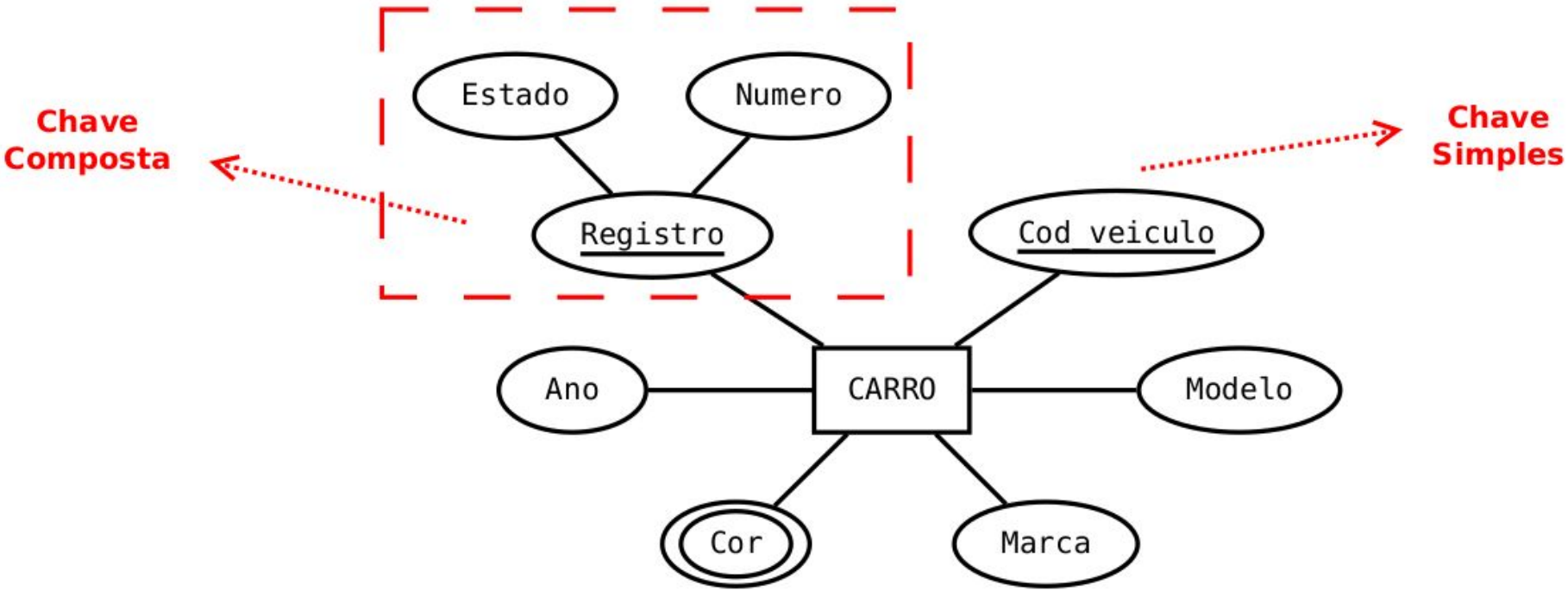
Tipos de Atributos



Tipos de Atributos



Tipos de Atributos



Relacionamento

Ação que conecta duas ou mais entidades.

- *Um FOLIÃO → FREQUENTA → um *BLOCO DE CARNAVAL*

A relação é mútua:

- O bloco também é frequentado pelo folião!

Relacionamento

- A relação é representada por um losango conectado às entidades.
  - Leitura: cada bloco tem um estandarte. E cada estandarte pertence a um bloco.

Cardinalidade nas Relações

1:1 – Um para Um

- Um BLOCO possui um único ESTANDARTE, e o ESTANDARTE pertence a um único BLOCO

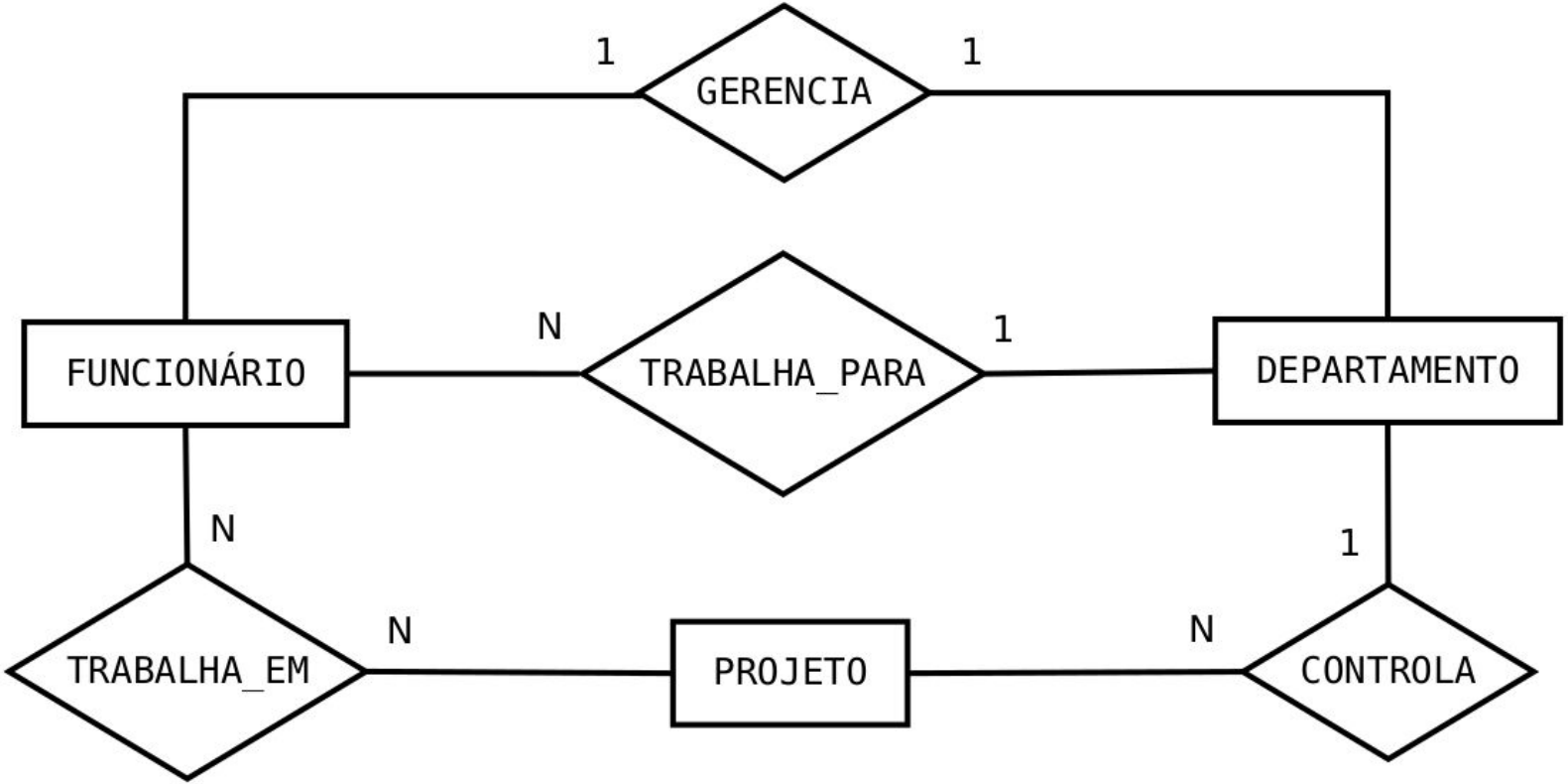
1:N – Um para Muitos

- Um BLOCO pode ter vários FOLIÕES, mas cada FOLIÃO só escolheu um BLOCO para seguir

N:N – Muitos para Muitos

- Um FOLIÃO pode frequentar vários BLOCOS, e um BLOCO pode ser frequentado por vários FOLIÕES

Cardinalidade nas Relações



Variações na Representação de Relacionamentos

Exemplo: Pessoa → Local de Nascimento

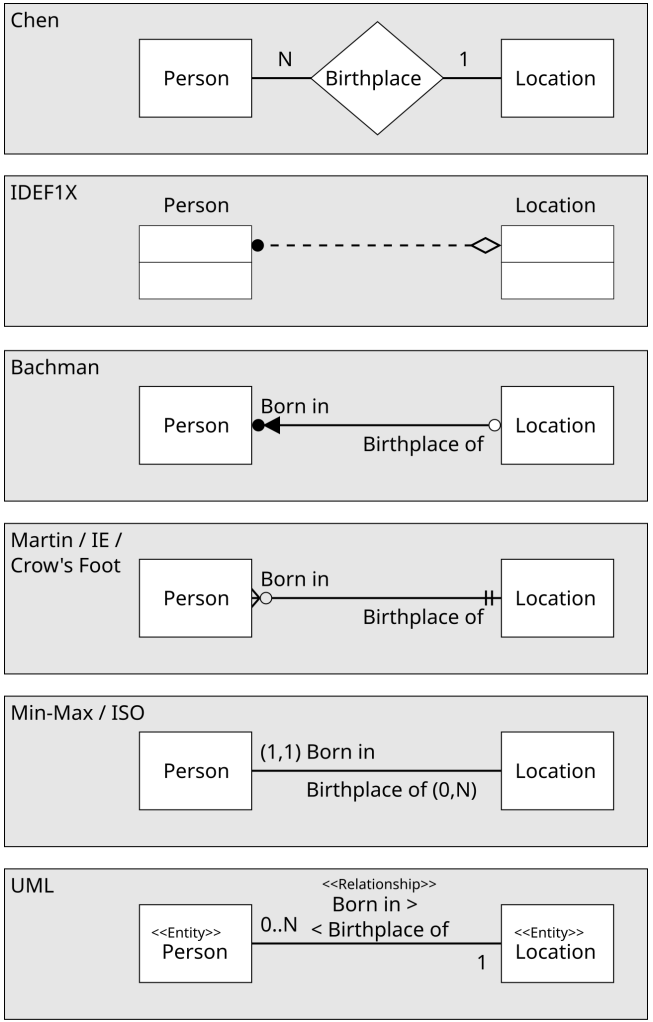
- Relação 1:N obrigatória do lado "Pessoa"
- Cada pessoa deve ter exatamente um local de nascimento.
- Cada local pode ter zero ou mais pessoas associadas.

Possibilidades de Representação

- Através de uma relação explícita em diagrama ER (losango entre Pessoa e Local).
- Atributo estrangeiro: chave do local incluída na entidade Pessoa.
- Tabela associativa (em casos mais complexos ou para uniformizar o modelo).

Importância do Modelo

- A escolha da representação depende do nível de abstração desejado.
- Todas as representações transmitem a mesma semântica do relacionamento 1:N.



Participação Total e Parcial em Relações

Participação Total

- Uma ocorrência da entidade sempre participa da relação.
- Representada por uma **linha dupla** no diagrama
  - Todo BONECO GIGANTE precisa estar associado a um BLOCO
    - Participação total do lado de BONECO GIGANTE

Participação Total e Parcial em Relações

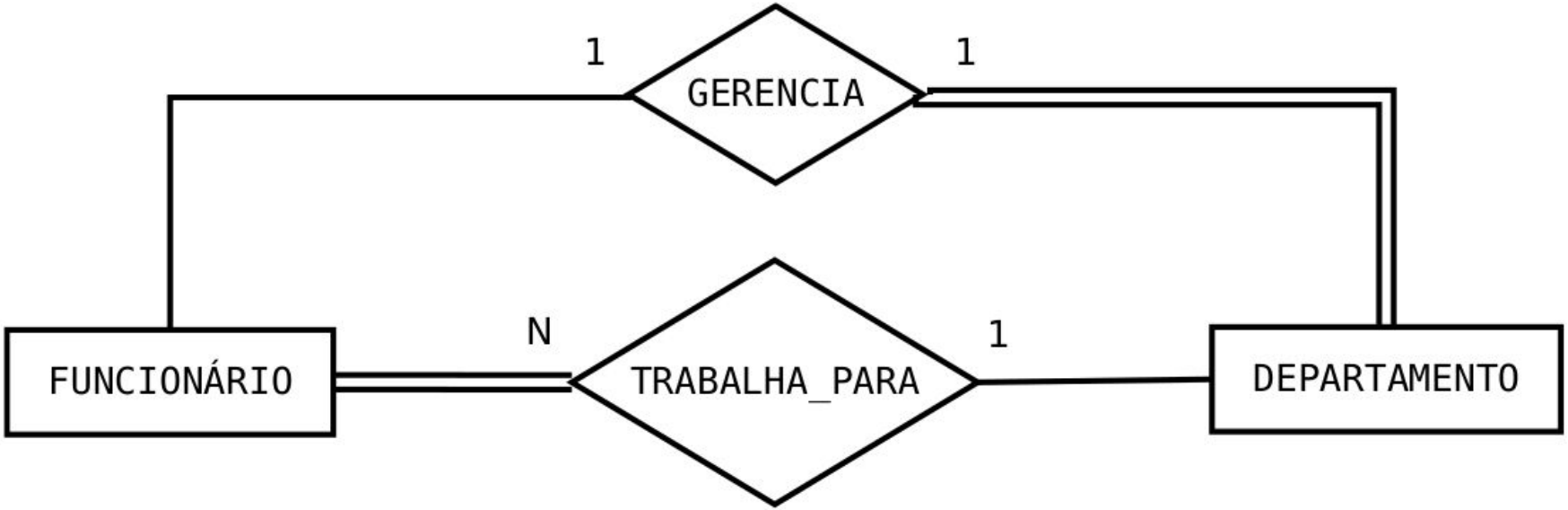
Participação Parcial

- A ocorrência da entidade pode ou não participar da relação.
- Representada por uma **linha simples** no diagrama.
  - Nem todo BLOCO tem um BONECO GIGANTE
    - Participação parcial do lado de BLOCO.

Participação Total e Parcial em Relações

- O boneco do “Homem da Meia” noite só existe porque há o bloco do “Homem da Meia Noite”.

Participação Total e Parcial em Relações



Mapeamento ER → Modelo Relacional

Entidade → Tabela

- Cada entidade vira uma tabela.
- Os atributos da entidade se tornam colunas da tabela.
- O atributo-chave vira a chave primária da tabela.

Mapeamento ER → Modelo Relacional

Atributos compostos

- São decompostos em seus atributos simples.
  - Ex.: ENDEREÇO → RUA, NÚMERO, BAIRRO, etc.

Atributos multivalorados

- Criam uma nova tabela com chave estrangeira referenciando a entidade original.

  Ex.: CORES_DO_BLOCO(cor, bloco_id)

Mapeamento ER → Modelo Relacional

Relacionamentos

- 1:1 → chave estrangeira em um dos lados.
- 1:N → chave estrangeira no lado N.
- N:N → cria uma nova tabela intermediária com chaves estrangeiras dos dois lados.
  - Ex.: FOLIAO_BLOCO(foliao_id, bloco_id)

Mapeamento ER → Modelo Relacional

Relacionamentos com atributos

- Também viram tabelas, com colunas para os atributos e as chaves estrangeiras das entidades associadas.
- Se for um relacionamento 1:N, não precisaria da tabela. Os atributos do relacionamento ficam junto com a chave estrangeira.

Participação total

- Pode influenciar na obrigatoriedade de chave estrangeira (constraints NOT NULL).

Mapeamento ER → Modelo Relacional

Esse processo é chamado de projeto lógico do banco de dados, preparando o caminho para a criação em SQL …

Tabelas resultantes em SQL

```
-- Entidade BLOCO

CREATE TABLE BLOCO (

        nome VARCHAR(100) PRIMARY KEY,

        data_fundacao DATE,

        bairro VARCHAR(100)

);
```

Tabelas resultantes em SQL

```
-- Entidade FOLIAO

CREATE TABLE FOLIAO (
        cpf CHAR(11) PRIMARY KEY,
        nome VARCHAR(100),
        idade INT
);
```

Tabelas resultantes em SQL

-- Entidade ESTANDARTE

```
CREATE TABLE ESTANDARTE (
    id_estandarte INT PRIMARY KEY,
    ano_criacao INT,
    bloco_nome VARCHAR(100) UNIQUE NOT NULL,  -- Participação total (NOT NULL)
    FOREIGN KEY (bloco_nome) REFERENCES BLOCO(nome)
);
```

Tabelas resultantes em SQL

-- Atributo multivalorado: CORES do estandarte

```
CREATE TABLE CORES_ESTD (
    id_estandarte INT,
    cor VARCHAR(30),
    PRIMARY KEY (id_estandarte, cor),
    FOREIGN KEY (id_estandarte) REFERENCES
ESTANDARTE(id_estandarte)
);
```

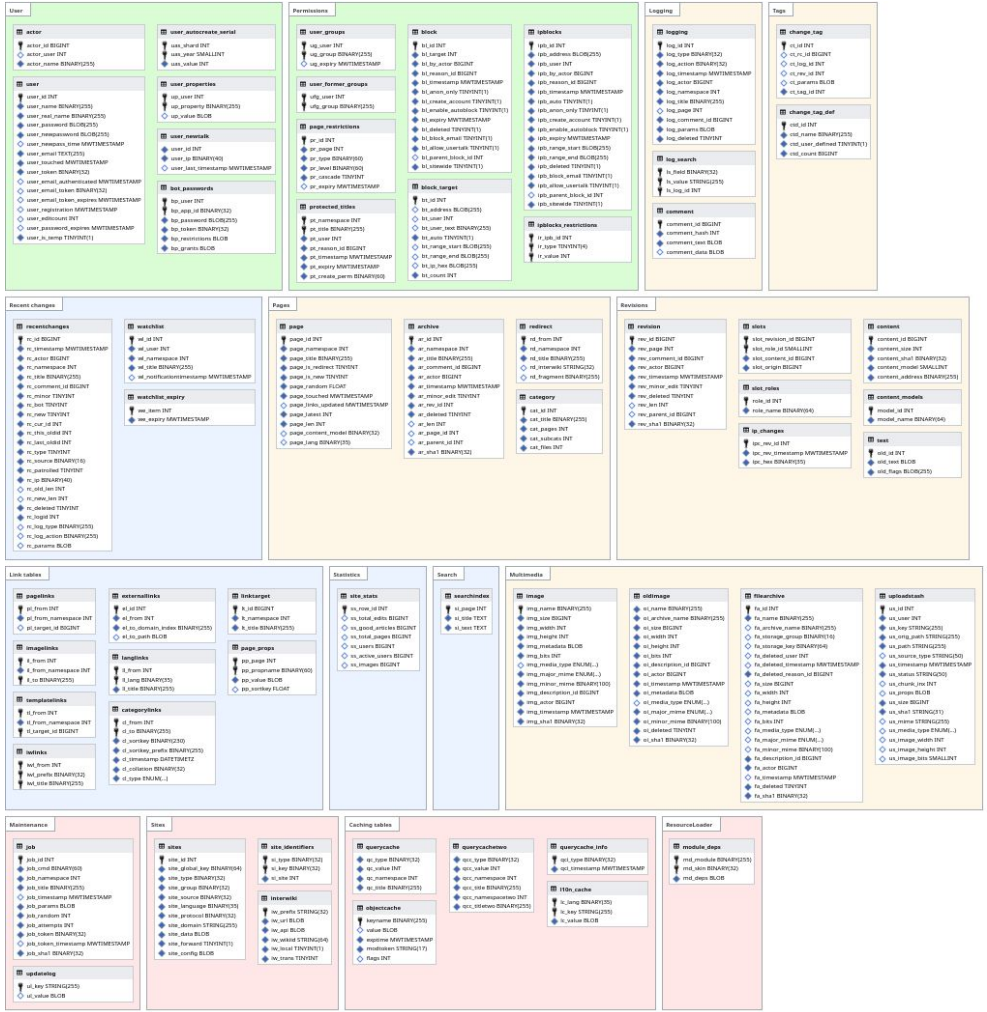
Tabelas resultantes em SQL

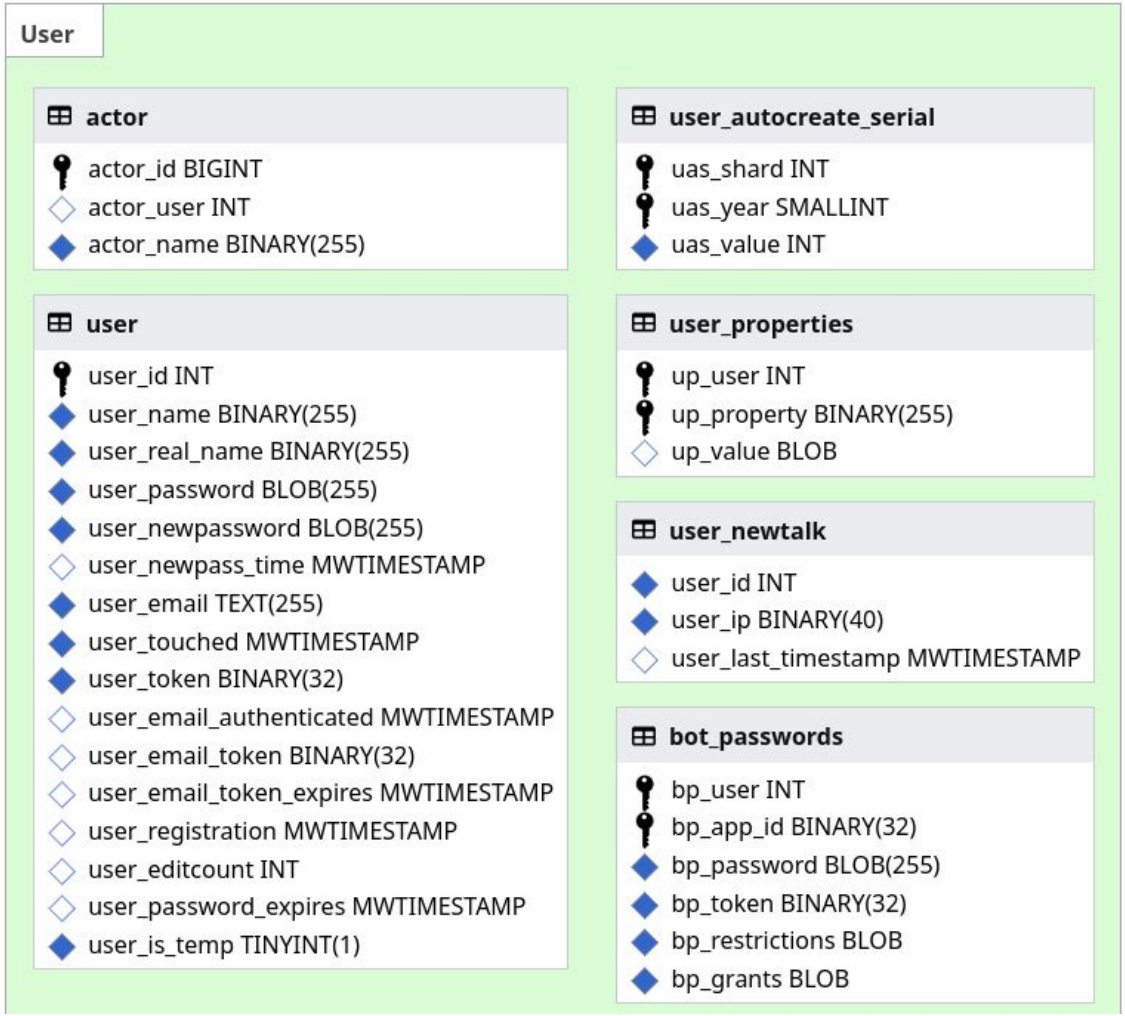
```
-- Relacionamento N:N: FREQUENTA (com atributos)
CREATE TABLE FREQUENTA (
        cpf CHAR(11),
        bloco_nome VARCHAR(100),
        ano INT,
        fantasia_usada VARCHAR(100),
        PRIMARY KEY (cpf, bloco_nome),
        FOREIGN KEY (cpf) REFERENCES FOLIAO(cpf),
        FOREIGN KEY (bloco_nome) REFERENCES BLOCO(nome)
);
```

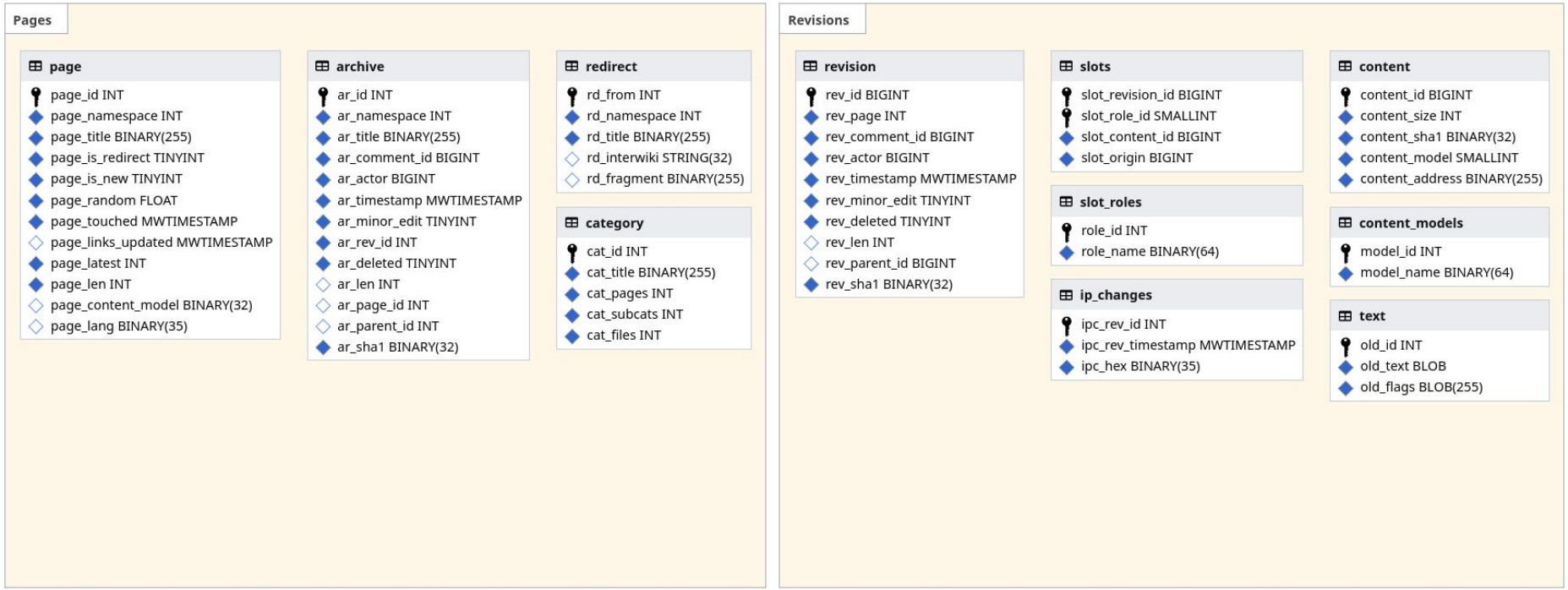
Extra

Database schema of MediaWiki 1.43.0 (December 2024)

https://www.mediawiki.org/wiki/DB


User
actor
actor_id BIGINT
actor_user INT
actor_name BINARY(255)
user
user_id INT
user_name BINARY(255)
user_real_name BINARY(255)
user_password BLOB(255)
user_newpassword BLOB(255)
user_newpass_time MWTIMESTAMP
user_email TEXT(255)
user_touched MWTIMESTAMP
user_token BINARY(32)
user_email_authenticated MWTIMESTAMP
user_email_token BINARY(32)
user_email_token_expires MWTIMESTAMP
user_registration MWTIMESTAMP
user_editcount INT
user_password_expires MWTIMESTAMP
user_is_temp TINYINT(1)
user_autocreate_serial
uas_shard INT
uas_year SMALLINT
uas_value INT
user_properties
up_user INT
up_property BINARY(255)
up_value BLOB
user_newtalk
user_id INT
user_ip BINARY(40)
user_last_timestamp MWTIMESTAMP
bot_passwords
bp_user INT
bp_app_id BINARY(32)
bp_password BLOB(255)
bp_token BINARY(32)
bp_restrictions BLOB
bp_grants BLOB
Permissions
user_groups
ug_user INT
ug_group BINARY(255)
ug_expiry MWTIMESTAMP
user_former_groups
ufg_user INT
ufg_group BINARY(255)
page_restrictions
pr_id INT
pr_page INT
pr_type BINARY(60)
pr_level BINARY(60)
pr_cascade TINYINT
pr_expiry MWTIMESTAMP
protected_titles
pt_namespace INT
pt_title BINARY(255)
pt_user INT
pt_reason_id BIGINT
pt_timestamp MWTIMESTAMP
pt_expiry MWTIMESTAMP
pt_create_perm BINARY(60)
block
bl_id INT
bl_target INT
bl_by_actor BIGINT
bl_reason_id BIGINT
bl_timestamp MWTIMESTAMP
bl_anon_only TINYINT(1)
bl_create_account TINYINT(1)
bl_enable_autoblock TINYINT(1)
bl_expiry MWTIMESTAMP
bl_deleted TINYINT(1)
bl_block_email TINYINT(1)
bl_allow_usertalk TINYINT(1)
bl_parent_block_id INT
bl_sitewide TINYINT(1)
block_target
bt_id INT
bt_address BLOB(255)
bt_user INT
bt_user_text BINARY(255)
bt_auto TINYINT(1)
bt_range_start BLOB(255)
bt_range_end BLOB(255)
bt_ip_hex BLOB(255)
bt_count INT
ipblocks
ipb_id INT
ipb_address BLOB(255)
ipb_user INT
ipb_by_actor BIGINT
ipb_reason_id BIGINT
ipb_timestamp MWTIMESTAMP
ipb_auto TINYINT(1)
ipb_anon_only TINYINT(1)
ipb_create_account TINYINT(1)
ipb_enable_autoblock TINYINT(1)
ipb_expiry MWTIMESTAMP
ipb_range_start BLOB(255)
ipb_range_end BLOB(255)
ipb_deleted TINYINT(1)
ipb_block_email TINYINT(1)
ipb_allow_usertalk TINYINT(1)
ipb_parent_block_id INT
ipb_sitewide TINYINT(1)
ipblocks_restrictions
ir_ipb_id INT
ir_type TINYINT(4)
ir_value INT
Logging
logging
log_id INT
log_type BINARY(32)
log_action BINARY(32)
log_timestamp MWTIMESTAMP
log_actor BIGINT
log_namespace INT
log_title BINARY(255)
log_page INT
log_comment_id BIGINT
log_params BLOB
log_deleted TINYINT
log_search
ls_field BINARY(32)
ls_value STRING(255)
ls_log_id INT
comment
comment_id BIGINT
comment_hash INT
comment_text BLOB
comment_data BLOB
Tags
change_tag
ct_id INT
ct_rc_id BIGINT
ct_log_id INT
ct_rev_id INT
ct_params BLOB
ct_tag_id INT
change_tag_def
ctd_id INT
ctd_name BINARY(255)
ctd_user_defined TINYINT(1)
ctd_count BIGINT
Recent changes
recentchanges
rc_id BIGINT
rc_timestamp MWTIMESTAMP
rc_actor BIGINT
rc_namespace INT
rc_title BINARY(255)
rc_comment_id BIGINT
rc_minor TINYINT
rc_bot TINYINT
rc_new TINYINT
rc_cur_id INT
rc_this_oldid INT
rc_last_oldid INT
rc_type TINYINT
rc_source BINARY(16)
rc_patrolled TINYINT
rc_ip BINARY(40)
rc_old_len INT
rc_new_len INT
rc_deleted TINYINT
rc_logid INT
rc_log_type BINARY(255)
rc_log_action BINARY(255)
rc_params BLOB
watchlist
wl_id INT
wl_user INT
wl_namespace INT
wl_title BINARY(255)
wl_notificationtimestamp MWTIMESTAMP
watchlist_expiry
we_item INT
we_expiry MWTIMESTAMP
Pages
page
page_id INT
page_namespace INT
page_title BINARY(255)
page_is_redirect TINYINT
page_is_new TINYINT
page_random FLOAT
page_touched MWTIMESTAMP
page_links_updated MWTIMESTAMP
page_latest INT
page_len INT
page_content_model BINARY(32)
page_lang BINARY(35)
archive
ar_id INT
ar_namespace INT
ar_title BINARY(255)
ar_comment_id BIGINT
ar_actor BIGINT
ar_timestamp MWTIMESTAMP
ar_minor_edit TINYINT
ar_rev_id INT
ar_deleted TINYINT
ar_len INT
ar_page_id INT
ar_parent_id INT
ar_sha1 BINARY(32)
redirect
rd_from INT
rd_namespace INT
rd_title BINARY(255)
rd_interwiki STRING(32)
rd_fragment BINARY(255)
category
cat_id INT
cat_title BINARY(255)
cat_pages INT
cat_subcats INT
cat_files INT
Revisions
revision
rev_id BIGINT
rev_page INT
rev_comment_id BIGINT
rev_actor BIGINT
rev_timestamp MWTIMESTAMP
rev_minor_edit TINYINT
rev_deleted TINYINT
rev_len INT
rev_parent_id BIGINT
rev_sha1 BINARY(32)
slots
slot_revision_id BIGINT
slot_role_id SMALLINT
slot_content_id BIGINT
slot_origin BIGINT
slot_roles
role_id INT
role_name BINARY(64)
sp_changes
sp_rev_id INT
sp_rev_timestamp MWTIMESTAMP
sp_hex BINARY(35)
content
content_id BIGINT
content_size INT
content_sha1 BINARY(32)
content_model SMALLINT
content_address BINARY(255)
content_models
model_id INT
model_name BINARY(64)
text
old_id INT
old_text BLOB
old_flags BLOB(255)
Link tables
pagelinks
pl_from INT
pl_from_namespace INT
pl_target_id BIGINT
imagelinks
il_from INT
il_from_namespace INT
il_to BINARY(255)
templatelinks
tl_from INT
tl_from_namespace INT
tl_target_id BIGINT
iwlinks
iwl_from INT
iwl_prefix BINARY(32)
iwl_title BINARY(255)
externallinks
el_id INT
el_from INT
el_to_domain_index BINARY(255)
el_to_path BLOB
langlinks
ll_from INT
ll_lang BINARY(35)
ll_title BINARY(255)
categorylinks
cl_from INT
cl_to BINARY(255)
cl_sortkey BINARY(230)
cl_sortkey_prefix BINARY(255)
cl_timestamp DATETIME2
cl_collation BINARY(32)
cl_type ENUM(...)
linktarget
lt_id BIGINT
lt_namespace INT
lt_title BINARY(255)
page_props
pp_page INT
pp_propname BINARY(60)
pp_value BLOB
pp_sortkey FLOAT
Statistics
site_stats
ss_row_id INT
ss_total_edits BIGINT
ss_good_articles BIGINT
ss_total_pages BIGINT
ss_users BIGINT
ss_active_users BIGINT
ss_images BIGINT
Search
searchindex
si_page INT
si_title TEXT
si_text TEXT
Multimedia
image
img_name BINARY(255)
img_size BIGINT
img_width INT
img_height INT
img_metadata BLOB
img_bits INT
img_media_type ENUM(...)
img_major_mime ENUM(...)
img_minor_mime BINARY(100)
img_description_id BIGINT
img_actor BIGINT
img_timestamp MWTIMESTAMP
img_sha1 BINARY(32)
oldimage
oi_name BINARY(255)
oi_archive_name BINARY(255)
oi_size BIGINT
oi_width INT
oi_height INT
oi_bits INT
oi_description_id BIGINT
oi_actor BIGINT
oi_timestamp MWTIMESTAMP
oi_metadata BLOB
oi_media_type ENUM(...)
oi_major_mime ENUM(...)
oi_minor_mime BINARY(100)
oi_deleted TINYINT
oi_sha1 BINARY(32)
filearchive
fa_id INT
fa_name BINARY(255)
fa_archive_name BINARY(255)
fa_storage_group BINARY(16)
fa_storage_key BINARY(64)
fa_deleted_user INT
fa_deleted_timestamp MWTIMESTAMP
fa_deleted_reason_id BIGINT
fa_size BIGINT
fa_width INT
fa_height INT
fa_metadata BLOB
fa_bits INT
fa_media_type ENUM(...)
fa_major_mime ENUM(...)
fa_minor_mime BINARY(100)
fa_description_id BIGINT
fa_actor BIGINT
fa_timestamp MWTIMESTAMP
fa_deleted TINYINT
fa_sha1 BINARY(32)
uploadstash
us_id INT
us_user INT
us_key STRING(255)
us_orig_path STRING(255)
us_path STRING(255)
us_source_type STRING(50)
us_timestamp MWTIMESTAMP
us_status STRING(50)
us_chunk_inx INT
us_props BLOB
us_size BIGINT
us_sha1 STRING(31)
us_mime STRING(255)
us_media_type ENUM(...)
us_image_width INT
us_image_height INT
us_image_bits SMALLINT
Maintenance
job
job_id INT
job_cmd BINARY(60)
job_namespace INT
job_title BINARY(255)
job_timestamp MWTIMESTAMP
job_params BLOB
job_random INT
job_attempts INT
job_token BINARY(32)
job_token_timestamp MWTIMESTAMP
job_sha1 BINARY(32)
updatelog
ul_key STRING(255)
ul_value BLOB
Sites
sites
site_id INT
site_global_key BINARY(64)
site_type BINARY(32)
site_group BINARY(32)
site_source BINARY(32)
site_language BINARY(35)
site_protocol BINARY(32)
site_domain STRING(255)
site_data BLOB
site_forward TINYINT(1)
site_config BLOB
site_identifiers
si_type BINARY(32)
si_key BINARY(32)
si_site INT
interwiki
iw_prefix STRING(32)
iw_url BLOB
iw_api BLOB
iw_wikid STRING(64)
iw_local TINYINT(1)
iw_trans TINYINT
Caching tables
querycache
qc_type BINARY(32)
qc_value INT
qc_namespace INT
qc_title BINARY(255)
objectcache
keyname BINARY(255)
value BLOB
exptime MWTIMESTAMP
modtoken STRING(17)
flags INT
querycachetwo
qcc_type BINARY(32)
qcc_value INT
qcc_namespace INT
qcc_title BINARY(255)
qcc_namespacetwo INT
qcc_titletwo BINARY(255)
querycache_info
qci_type BINARY(32)
qci_timestamp MWTIMESTAMP
l10n_cache
lc_lang BINARY(35)
lc_key STRING(255)
lc_value BLOB
ResourceLoader
module_deps
md_module BINARY(255)
md_skin BINARY(32)
md_deps BLOB


User
actor
actor_id BIGINT
actor_user INT
actor_name BINARY(255)
user
user_id INT
user_name BINARY(255)
user_real_name BINARY(255)
user_password BLOB(255)
user_newpassword BLOB(255)
user_newpass_time MWTIMESTAMP
user_email TEXT(255)
user_touched MWTIMESTAMP
user_token BINARY(32)
user_email_authenticated MWTIMESTAMP
user_email_token BINARY(32)
user_email_token_expires MWTIMESTAMP
user_registration MWTIMESTAMP
user_editcount INT
user_password_expires MWTIMESTAMP
user_is_temp TINYINT(1)
user_autocreate_serial
uas_shard INT
uas_year SMALLINT
uas_value INT
user_properties
up_user INT
up_property BINARY(255)
up_value BLOB
user_newtalk
user_id INT
user_ip BINARY(40)
user_last_timestamp MWTIMESTAMP
bot_passwords
bp_user INT
bp_app_id BINARY(32)
bp_password BLOB(255)
bp_token BINARY(32)
bp_restrictions BLOB
bp_grants BLOB


Pages
page
page_id INT
page_namespace INT
page_title BINARY(255)
page_is_redirect TINYINT
page_is_new TINYINT
page_random FLOAT
page_touched MWTIMESTAMP
page_links_updated MWTIMESTAMP
page_latest INT
page_len INT
page_content_model BINARY(32)
page_lang BINARY(35)
archive
ar_id INT
ar_namespace INT
ar_title BINARY(255)
ar_comment_id BIGINT
ar_actor BIGINT
ar_timestamp MWTIMESTAMP
ar_minor_edit TINYINT
ar_rev_id INT
ar_deleted TINYINT
ar_len INT
ar_page_id INT
ar_parent_id INT
ar_sha1 BINARY(32)
redirect
rd_from INT
rd_namespace INT
rd_title BINARY(255)
rd_interwiki STRING(32)
rd_fragment BINARY(255)
category
cat_id INT
cat_title BINARY(255)
cat_pages INT
cat_subcats INT
cat_files INT
Revisions
revision
rev_id BIGINT
rev_page INT
rev_comment_id BIGINT
rev_actor BIGINT
rev_timestamp MWTIMESTAMP
rev_minor_edit TINYINT
rev_deleted TINYINT
rev_len INT
rev_parent_id BIGINT
rev_sha1 BINARY(32)
slots
slot_revision_id BIGINT
slot_role_id SMALLINT
slot_content_id BIGINT
slot_origin BIGINT
slot_roles
role_id INT
role_name BINARY(64)
ip_changes
ipc_rev_id INT
ipc_rev_timestamp MWTIMESTAMP
ipc_hex BINARY(35)
content
content_id BIGINT
content_size INT
content_sha1 BINARY(32)
content_model SMALLINT
content_address BINARY(255)
content_models
model_id INT
model_name BINARY(64)
text
old_id INT
old_text BLOB
old_flags BLOB(255)

Banco de Dados
Conceitos e Modelagem

Paulo Meirelles, IME-USP
paulormm@ime.usp.br

** Baseado nos slides da Profa. Kelly Rosa Braghetto (IME-USP)*
** Usa um exemplo do post: Entendendo Diagramas Entidade-Relação com exemplos carnavalescos*