Banco de Dados

serratec
Parque Tecnológico Região Serrana

Sumário

1. Conceitos Gerais sobre Bancos de Dados
2. Estudo de Caso: Serratec Music
3. Modelagem Entidade-Relacionamento
4. Normalização
5. Linguagem SQL
6. Linguagem SQL – Tópicos avançados

serratec
Parque Tecnológico Região Serrana

1. Conceitos Gerais sobre Bancos de Dados

Exemplos de Utilização

Bancos de Dados tradicionais

- Movimentações bancárias;
- Reservas de hotel, vôo, restaurantes;
- Compras on-line ou físicas;
- Etc.

Exemplos de Utilização

Novas áreas (Big Data / NOSQL)

- Redes sociais (Facebook, Twitter, Youtube, etc.);
- Mecanismos de Busca (Google, etc.);
- Informações geográficas (Google Maps, etc.);
- Aplicativos Mobile;
- Processamento Analítico Online – OLAP;
- Etc.

2. Estudo de Caso: Serratec Music

Serratec Music

O Serratec Music deseja organizar seu catálogo de álbuns de músicas. No contexto de seu negócio, um álbum musical é composto por várias músicas e possui autoria de um único artista. Por outro lado, uma mesma música pode estar contida em diferentes álbuns. Um álbum ainda possui uma única capa.

Além de organizar os álbuns, músicas e artistas, o Serratec Music deseja saber as datas de cadastro e atualização dos dados referentes a todos os registros a serem armazenados, assim como identificar o usuário responsável por tais ações. Por fim, é importante garantir que nenhum registro seja, de fato, deletado.

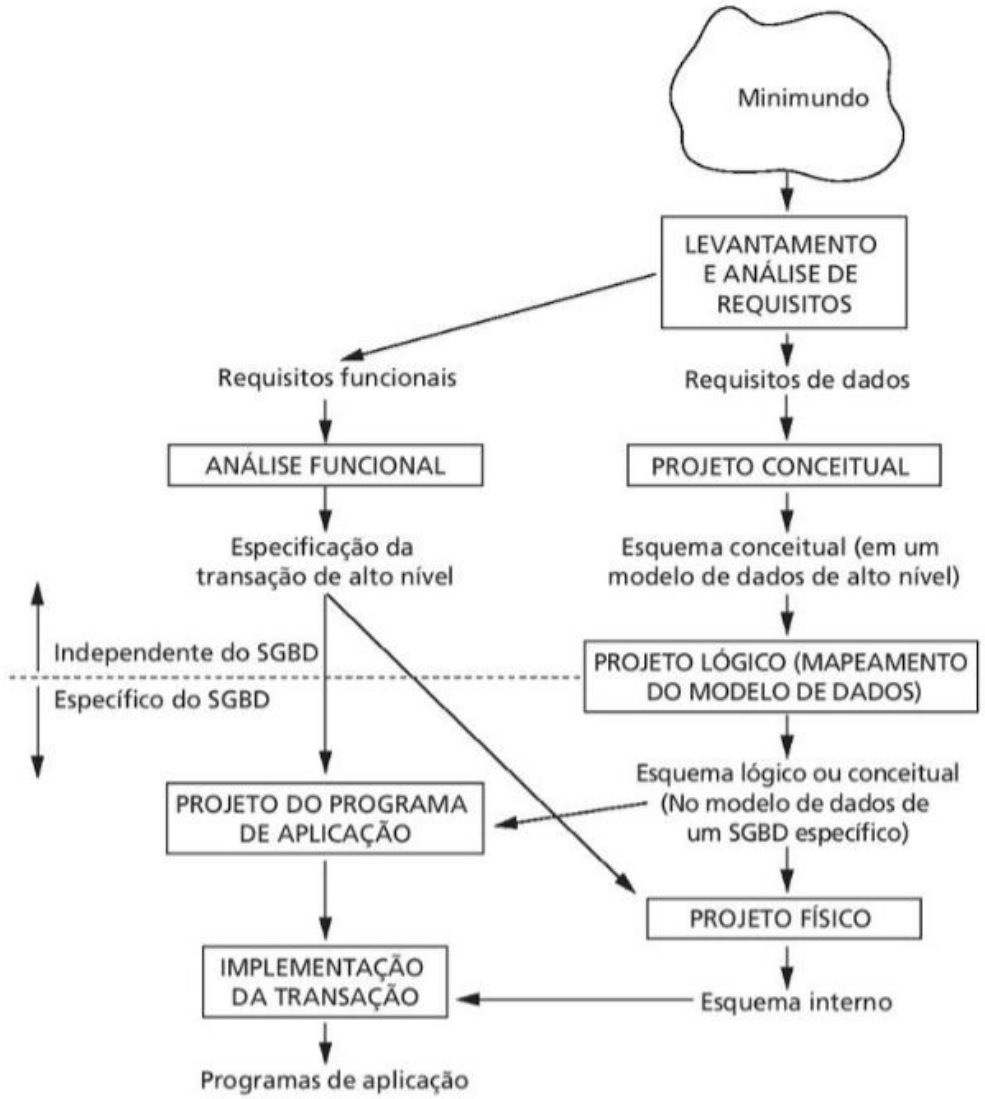
serratec
Parque Tecnológico Região Serrana

3. Modelagem Entidade-Relacionamento

Alguns conceitos:

- Modelagem conceitual/ Modelagem física;
- Foco na estrutura e restrições;
- Modelo de dados de alto nível;
- Notação – Diagrama ER (vs UML – Diagrama de Classes)

Figura 1: Modelo Conceitual e Físico

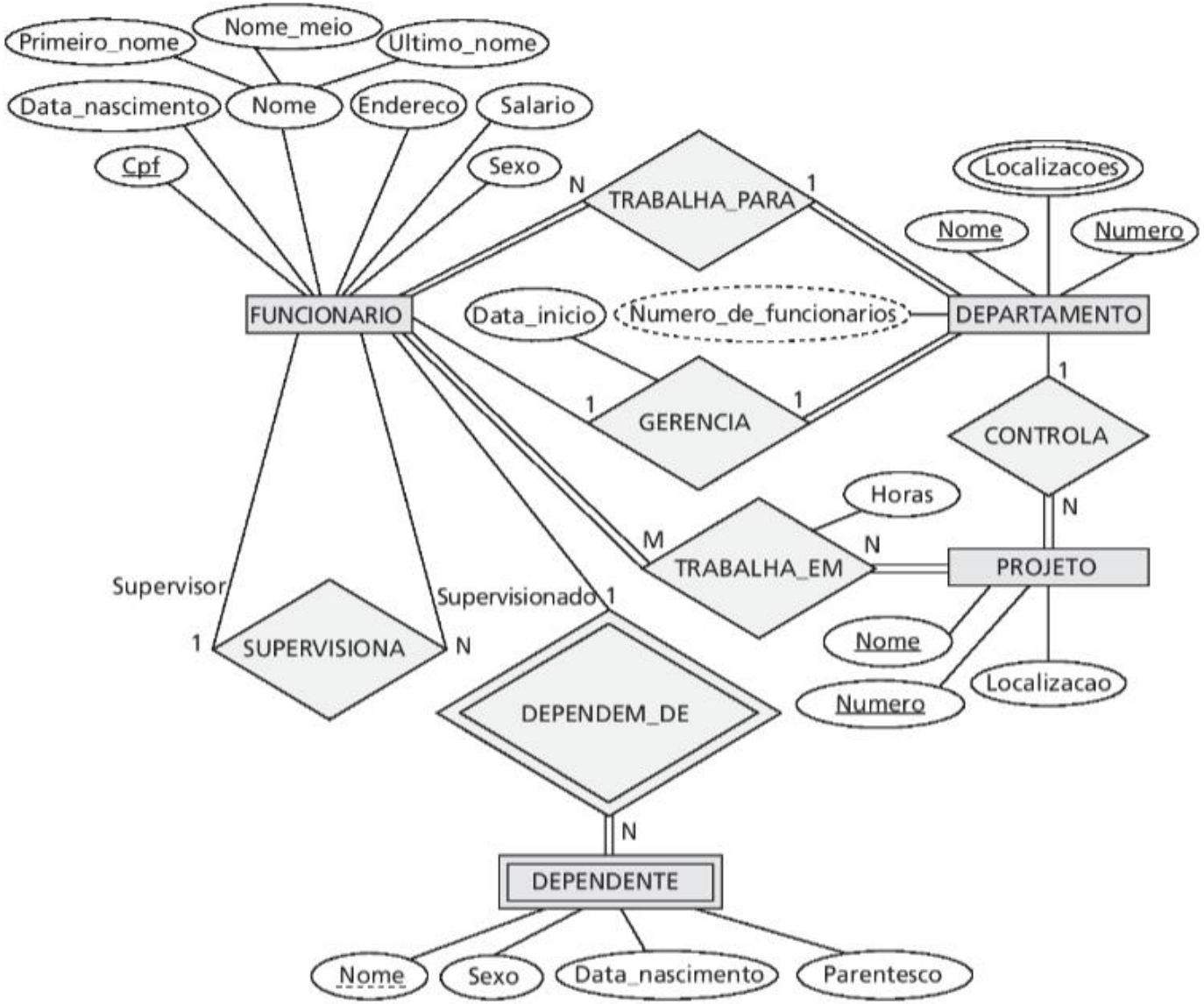


Fonte: Sistemas de banco de dados (Navathe, E.)

serratec
Parque Tecnológico Região Serrana

3.1. Modelo/Projeto Conceitual

Figura 2: Modelo Conceitual



Fonte: Sistemas de banco de dados (Navathe, E.)

serratec
Parque Tecnológico Região Serrana

Na prática:

- Descrever dados como:
  - Entidades: coisa ou objeto no mundo real com existência (física ou conceitual) independente;
  - Atributos: propriedades específicas que descrevem uma Entidade;
  - Relacionamentos: ocorrem quando um atributo de uma entidade se refere a outro tipo de entidade.

serratec
Parque Tecnológico Região Serrana

Serratec Music:: Identificando Entidades

- Entidades identificadas a partir do MiniMundo:
  - Artista;
  - Álbum;
  - Capa do Álbum;
  - Música;
  - Usuario.

serratec
Parque Tecnológico Região Serrana

Serratec Music:: Identificando Atributos

- Atributos identificados por Entidades

- Artista:
  - Nome
  - Tipo
- Álbum:
  - Título
- Capa do Álbum:
  - Álbum
  - Tipo de Mídia
- Música:
  - Título
  - Minutos

serratec
Parque Tecnológico Região Serrana

Serratec Music:: Identificando Atributos

- Atributos identificados por Entidades

- Usuário:
  - Nome
  - Login
  - E-mail
  - Senha

serratec
Parque Tecnológico Região Serrana

Serratec Music:: Identificando Atributos

- Atributos-chave identificados por Entidades

- Artista:
  - Código
- Álbum:
  - Código
- Capa do Álbum:
  - Código do Álbum
- Música:
  - Código
- Usuário:
  - Código

serratec
Parque Tecnológico Região Serrana

Serratec Music:: Identificando Relacionamentos

- Relacionamentos identificados entre Entidades
  - Artista e Álbum
    - Um Artista pode produzir zero ou vários Álbuns;
  - Álbum e Capa de Álbum:
    - Um Álbum possui uma única Capa;
  - Álbum e Música:
    - Um Álbum possui várias Músicas;
    - Uma Música pode estar presente em vários Álbuns;

serratec
Parque Tecnológico Região Serrana

3.1.1. Cardinalidade, Chave Primária e Chave Estrangeira

- Cardinalidade:
  - Também é conhecida como Grau do Relacionamento;
  - Representa o número de ocorrências de uma entidade A que está associado com ocorrências de outra Entidade, B;
  - Existem três Graus de Relacionamento:
    - Um para Um
    - Um para Muitos / Muitos para Um
    - Muitos para Muitos

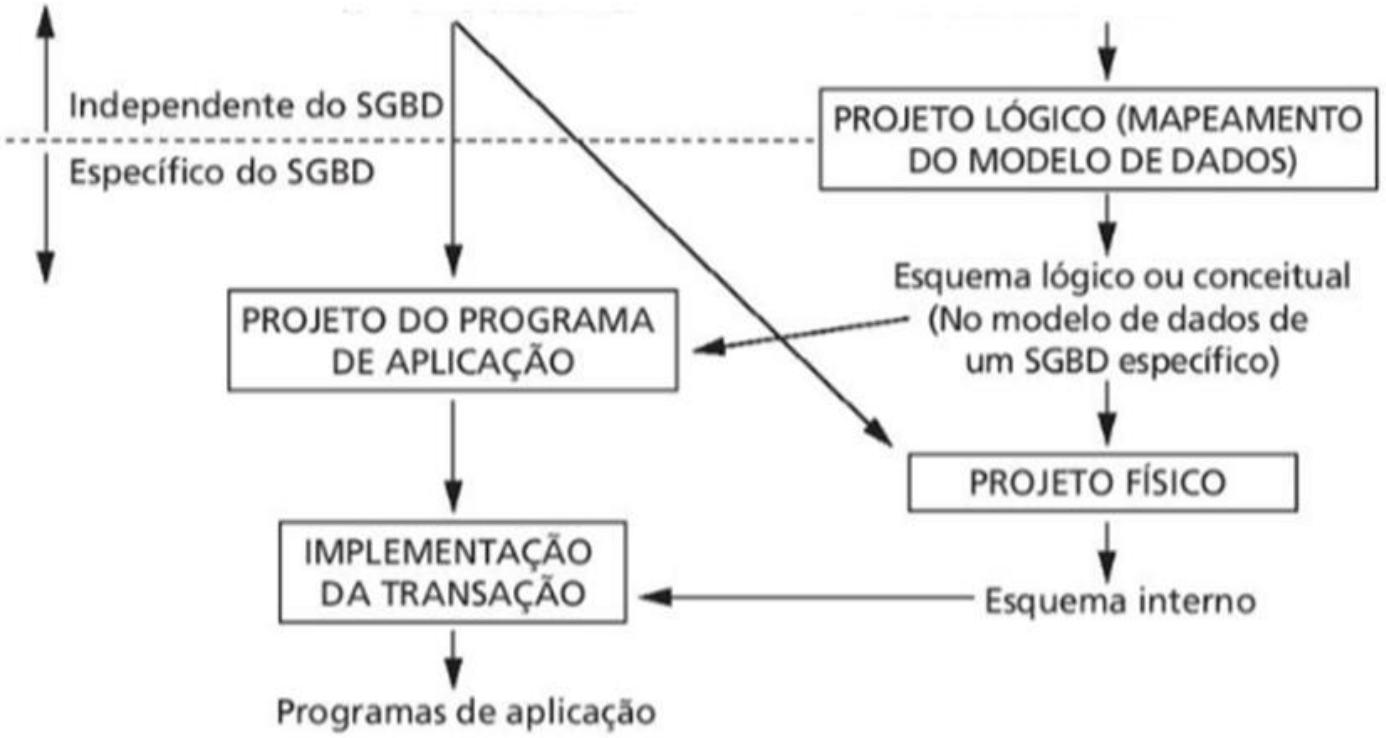
- Cardinalidade:
  - Exemplos de Graus de Relacionamento:
    - Um para Um: Álbum e Capa de Álbum
    - Um para Muitos / Muitos para Um: Artista e Álbum; Álbum e Artista
    - Muitos para Muitos: Álbum e Música; Música e Álbum

- Chave Primária:
  - Atributo de uma Entidade usado para identificar tuplas (conjunto de registros);
  - Uma entidade pode ter mais de uma chave que cumpra tal papel – nesse caso, tais atributos são chamados de chaves candidatas;
  - Dentre as chaves candidatas escolhe-se uma para ser a chave primária (PK – primary key);
  - Exemplos de chave-primária: número aleatório gerado e gerenciado pelo SGBD; CPF; CNPJ; Placa (de veículo); etc.

- Chave Estrangeira:
  - Atributo utilizado para definir a integridade referencial entre duas entidades;
  - Estabelece uma relação entre duas Entidades;
  - A Chave Primária de uma Entidade, quando inserida em outra Entidade para representar a relação entre ambas recebe o nome de Chave Estrangeira (FK – Foreign Key);
  - Exemplos de Chave Estrangeira: O identificador do Artista na Entidade Álbum; O identificador dá Entidade Álbum na Entidade Capa do Álbum (nesse caso, ela cumprirá dois papéis: Chave Primária e Chave Estrangeira).

3.1. Modelo/Projeto Físico

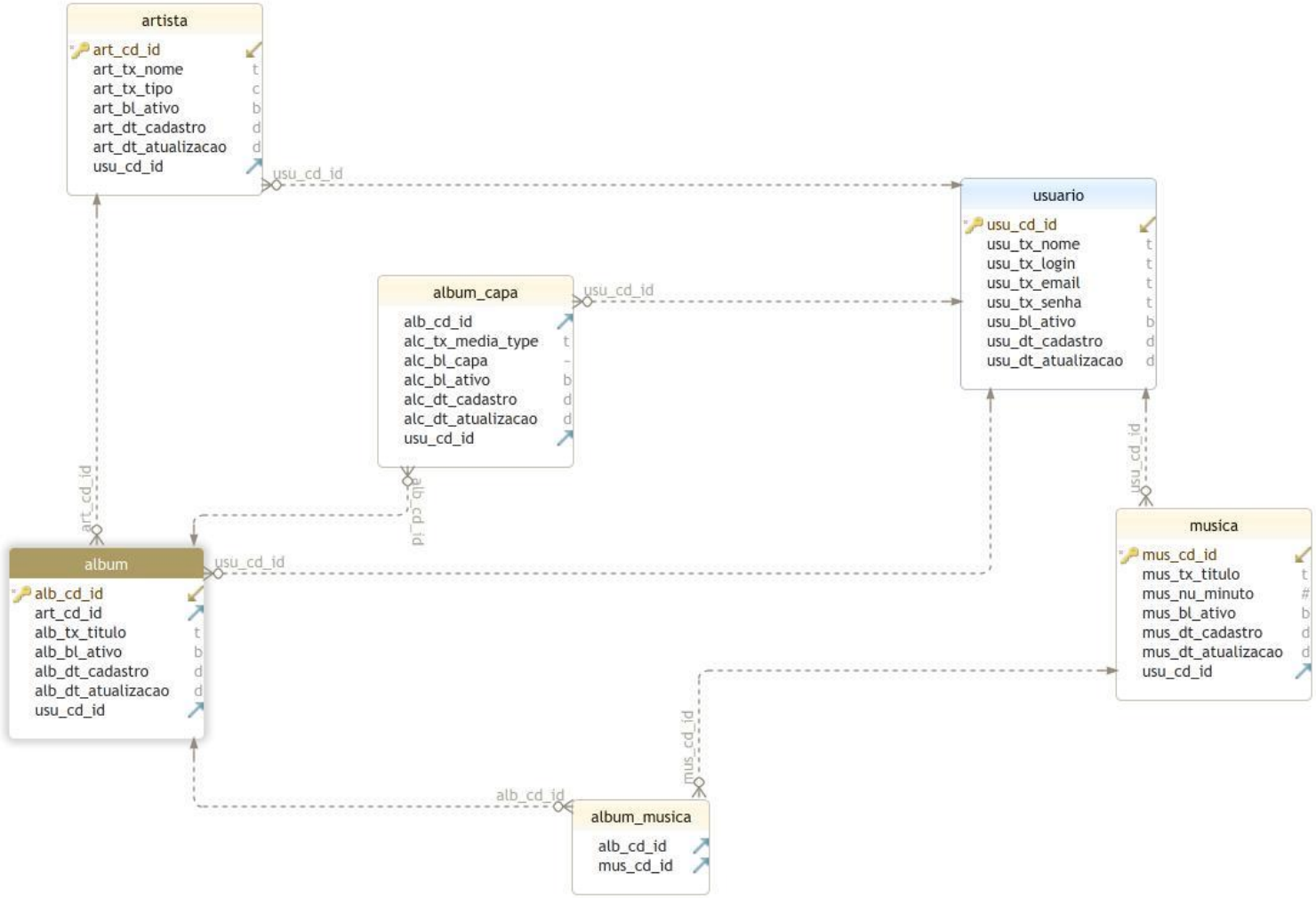
Figura 3: Modelo Físico



Fonte: Sistemas de banco de dados (Navathe, E.)

- Características do Modelo Físico:
  - Depende do SGBD;
  - Visão de baixo nível, com linguagem mais técnica;
  - Apresenta a estrutura – Tabelas, atributos e relacionamentos – que será, de fato, implementada no SGBD;
  - DER – Diagrama Entidade-Relacionamentos;
  - Diferentes tipos de notação.

Figura 4: Modelo Físico - SerratecMusic



Fonte: Elaboração própria

serratec
Parque Tecnológico Região Serrana

4. Normalização

- Motivação:
  - O processo de modelagem, visto até aqui, pode ser feito de diferentes formas:
    - com base na experiência de quem está modelando;
    - a partir de conhecimento prévio do negócio;
    - através do agrupamento lógico dos atributos, de acordo com suas características e particularidades;
    - Todas as alternativas anteriores.
  - Para validar um modelo, é necessário um modo mais formal:
    - que nos permita avaliar o modelo criado;
    - que nos permita analisar a qualidade do modelo;
    - etc.

- Pra que serve:
  - Analisar, de maneira formal, a modelagem de um banco de dados;
  - Garantir, sempre que possível, a preservação de informações juntamente com uma redundância mínima;
  - Evitar anomalias que possam ocorrer na inclusão, exclusão ou alteração de registros de uma entidade;

- Como é realizado:
  - Com base em relações matemáticas (teoria dos conjuntos);
  - Aplicando-se uma série de testes às relações existentes em um modelo a fim de verificar se as mesmas atendem às “formas normais”;
  - Decompondo as relações existentes sempre que necessário;

serratec

4.1. Formas Normais

- 1FN (1ª Forma Normal):
  - Todos os atributos de uma tabela devem ser atômicos, ou, em outras palavras, uma tabela não deverá conter nem grupos repetidos e nem atributos com mais de um valor.
  - Exemplo de Violação da 1FN:
    - ARTISTA [ID, NOME, TELEFONES]
  - Explicação da Violação:
    - A coluna TELEFONES é um atributo multivalorado e, portanto, não deve permanecer na tabela ARTISTA.
  - Normalizando a tabela:
    - ARTISTA[ID, NOME]
    - TELEFONE[ID, ARTISTA_ID, NUMERO_TELEFONE]

- 2FN (2ª Forma Normal):
  - A 1FN deve ser atendida;
  - Os atributos “não-chave” devem depender da chave-primária (que, nesse caso, será sempre composta – ou seja, entidade com mais de um atributo-chave, que garanta a unicidade das ocorrências de seus registros);
  - Os atributos “não-chave” não podem depender apenas de parte da chave-primária;
  - Exemplo de Violação da 2FN:
    - ALBUM_MUSICA [ALBUM_ID(PK), MUSICA_ID(PK), MUSICA_NOME]
  - Explicação da Violação:
    - A coluna MUSICA_NOME depende apenas da PK MUSICA_ID, o que consiste uma dependência parcial da chave primária da tabela (nesse caso, uma chave composta).
  - Normalizando a tabela:
    - ALBUM_MUSICA [ALBUM_ID(PK), MUSICA_ID(PK)]
    - MUSICA[MUSICA_ID(PK), MUSICA_NOME]

- 3FN (3ª Forma Normal):
  - A 2FN (e, consequentemente, a 1FN) deve ser atendida;
  - Não deve existir dependência transitiva entre os atributos, ou conjunto de atributos, não pertencentes à chave-primária;
  - Todos os atributos que não pertencem à chave-primária (que não a compõem) devem depender exclusivamente dela;
  - Exemplo de Violação da 3FN:
    - ALBUM [ALBUM_ID(PK), ARTISTA_ID, ARTISTA_NOME]
  - Explicação da Violação:
    - A coluna ARTISTA_NOME não depende da chave-primária ALBUM_ID.
  - Normalizando a tabela:
    - ALBUM [ALBUM_ID(PK), ARTISTA_ID]
    - ARTISTA[ARTISTA_ID(PK), ARTISTA_NOME]

5. Linguagem SQL

- O que é?
  - Linguagem de Consulta Estruturada;
  - Linguagem voltada para bancos de dados relacionais;
  - Segue padrões e convenções(ANSI / ISO), embora existam também implementações proprietárias;

- Sub-Conjuntos (variam de acordo com as operações realizadas)
  - DDL – Linguagem de Definição de Dados;
  - DML – Linguagem de Manipulação de Dados;
  - DQL – Linguagem de Consulta de Dados;

serratec

5.1. DDL

- Conjunto de instruções utilizadas para a Definição de Dados, ou seja, criação e manipulação das estruturas (tabelas, colunas, etc.) que compõem o projeto físico de Banco de Dados.

  - CREATE;
  - ALTER;
  - DROP;

- Exemplo de aplicação no Projeto Serratec Music

```
CREATE  TABLE usuario (
                usu_cd_id           serial  NOT NULL  ,
                usu_tx_nome         varchar(255)   ,
                usu_tx_login        varchar(100)   ,
                usu_tx_email        varchar(100)   ,
                usu_tx_senha        varchar(100)   ,
                usu_bl_ativo        boolean DEFAULT true   ,
                usu_dt_cadastro     timestamp DEFAULT CURRENT_TIMESTAMP   ,
                usu_dt_atualizacao  timestamp  DEFAULT CURRENT_TIMESTAMP ,
                PRIMARY KEY ( usu_cd_id )
);

CREATE  TABLE artista (
                art_cd_id           serial  NOT NULL  ,
                art_tx_nome         varchar(255)   ,
                art_tx_tipo         char(1)   ,
                art_bl_ativo        boolean DEFAULT true   ,
                art_dt_cadastro     timestamp DEFAULT CURRENT_TIMESTAMP   ,
                art_dt_atualizacao  timestamp DEFAULT CURRENT_TIMESTAMP   ,
                usu_cd_id           integer   ,
                PRIMARY KEY ( art_cd_id )
);
```

serratec
Parque Tecnológico Região Serrana

- Exemplo de aplicação no Projeto Serratec Music

```
ALTER TABLE artista ADD FOREIGN KEY (usu_cd_id) REFERENCES usuario(usu_cd_id);

DROP TABLE artista;
DROP TABLE usuario;
```

serratec
Parque Tecnológico Região Serrana

serratec

5.2. DML

- Conjunto de instruções utilizadas para a Manipulação de Dados (exceto consultas).

  - INSERT;
  - UPDATE;
  - DELETE;

- Exemplo de aplicação no Projeto Serratec Music

```
INSERT
        INTO
        usuario_teste
(usu_tx_nome,
        usu_tx_login,
        usu_tx_email,
        usu_tx_senha,
        usu_bl_ativo,
        usu_dt_cadastro,
        usu_dt_atualizacao)
VALUES('Aluno',
'aluno',
'aluno@email.com',
'senha@123',
TRUE,
now(),
now());
```

```
INSERT
        INTO
        artista
(art_tx_nome,
        art_tx_tipo,
        art_bl_ativo,
        art_dt_cadastro,
        art_dt_atualizacao,
        usu_cd_id)
VALUES('Milton Mascimento',
'1',
TRUE,
now(),
now(),
1);
```

serratec
Parque Tecnológico Região Serrana

- Exemplo de aplicação no Projeto Serratec Music

```
UPDATE
        artista
SET
        art_tx_nome = 'Milton Nascimento',
        art_tx_tipo = '2',
        art_bl_ativo = TRUE,
        art_dt_atualizacao = now(),
        usu_cd_id = 1
WHERE
        art_cd_id = 1;
```

```
DELETE
FROM
        public.artista
WHERE
        art_cd_id = 1;
```

serratec
Parque Tecnológico Região Serrana

serratec

5.3. DQL

- Conjunto de instruções utilizadas para a Consulta de Dados.
  - SELECT

- Exemplo de aplicação no Projeto Serratec Music

```
SELECT
        art_cd_id,
        art_tx_nome,
        art_tx_tipo,
        art_bl_ativo,
        art_dt_cadastro,
        art_dt_atualizacao,
        usu_cd_id
FROM
        artista;
```

```
SELECT
        art_cd_id,
        art_tx_nome,
        art_tx_tipo
FROM
        artista;
```

serratec
Parque Tecnológico Região Serrana

Figura 5: Sintaxe do Select (Postgresql)

```
[ WITH [ RECURSIVE ] with_query [, ...] ]
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
    [ * | expression [ [ AS ] output_name ] [, ...] ]
    [ FROM from_item [, ...] ]
    [ WHERE condition ]
    [ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
    [ HAVING condition ]
    [ WINDOW window_name AS ( window_definition ) [, ...] ]
    [ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select ]
    [ ORDER BY expression [ ASC | DESC | USING operator ] [ NULLS { FIRST | LAST } ] [, ...] ]
    [ LIMIT { count | ALL } ]
    [ OFFSET start [ ROW | ROWS ] ]
    [ FETCH { FIRST | NEXT } [ count ] { ROW | ROWS } { ONLY | WITH TIES } ]
    [ FOR { UPDATE | NO KEY UPDATE | SHARE | KEY SHARE } [ OF table_name [, ...] ] [ NOWAIT | SKIP LOCKED ] [...] ]

where from_item can be one of:

    [ ONLY ] table_name [ * ] [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
                [ TABLESAMPLE sampling_method ( argument [, ...] ) [ REPEATABLE ( seed ) ] ]
    [ LATERAL ] ( select ) [ AS ] alias [ ( column_alias [, ...] ) ]
    with_query_name [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
    [ LATERAL ] function_name ( [ argument [, ...] ] )
                [ WITH ORDINALITY ] [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
    [ LATERAL ] function_name ( [ argument [, ...] ] ) [ AS ] alias ( column_definition [, ...] )
    [ LATERAL ] function_name ( [ argument [, ...] ] ) AS ( column_definition [, ...] )
    [ LATERAL ] ROWS FROM( function_name ( [ argument [, ...] ] ) [ AS ( column_definition [, ...] ) ] [, ...] )
                [ WITH ORDINALITY ] [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
    from_item [ NATURAL ] join_type from_item [ ON join_condition | USING ( join_column [, ...] ) [ AS join_using_alias ] ]

and grouping_element can be one of:

    ( )
    expression
    ( expression [, ...] )
    ROLLUP ( { expression | ( expression [, ...] ) } [, ...] )
    CUBE ( { expression | ( expression [, ...] ) } [, ...] )
    GROUPING SETS ( grouping_element [, ...] )

and with_query is:

    with_query_name [ ( column_name [, ...] ) ] AS [ [ NOT ] MATERIALIZED ] ( select | values | insert | update | delete )
        [ SEARCH { BREADTH | DEPTH } FIRST BY column_name [, ...] SET search_seq_col_name ]
        [ CYCLE column_name [, ...] SET cycle_mark_col_name [ TO cycle_mark_value DEFAULT cycle_mark_default ] USING cycle_path_col_name ]

TABLE [ ONLY ] table_name [ * ]
```

serratec
Parque Tecnológico Região Serrana

serratec

6. Linguagem SQL – Tópicos Avançados

- Cláusulas

  - From : especifica a tabela da qual os registros serão selecionados
  - Where: especifica as condições de seleção dos registros
  - Group By: separa os registros selecionados em grupos específicos
  - Having: usada em conjunto com a Group By, expressa a condição a ser atendida no agrupamento
  - Order By: ordena os registros conforme critérios definidos
  - Distinct: seleciona registros distintos, sem repetição
  - Union: combina uma ou mais consultas num único resultado

- Agregação

  - AVG : calcula a média
  - COUNT: conta a quantidade de registros
  - SUM: soma os valores
  - MAX: retorna o valor mais alto
  - MIN: retorna o valor mais baixo

- Junção / Join
  - Combina colunas de uma ou mais tabelas

Figura 6: Tabelas Empregado e Departamento

Tabela Empregado

ÚltimoNome	IDDepartamento
Rafferty	31
Jones	33
Heisenberg	33
Robinson	34
Smith	34
Williams	NULO

Tabela Departamento

IDDepartamento	NomeDepartamento
31	Vendas
33	Engenharia
34	Administrativo
35	Marketing

Fonte: Wikipedia

- Junção
  - Inner Join (Junção Interna): Intersecção entre os registros das tabelas

```
SELECT empregado.ÚltimoNome, empregado.IDDepartamento, departamento.NomeDepartamento
FROM empregado
INNER JOIN departamento ON
empregado.IDDepartamento = departamento.IDDepartamento;
```

Empregado.ÚltimoNome	Empregado.Departamento	Departamento.NomeDepartamento
Robinson	34	Administrativo
Jones	33	Engenharia
Smith	34	Administrativo
Heisenberg	33	Engenharia
Rafferty	31	Vendas

- Junção
  - Outer Join (Junção Externa) : Retorna os registros da junção, mesmo sem correspondentes na outra tabela
    - Left Outer Join

```
SELECT *
FROM empregado
LEFT OUTER JOIN departamento ON empregado.IDDepartamento = departamento.IDDepartamento;
```

Empregado.ÚltimoNome	Empregado.IDDepartamento	Departamento.NomeDepartamento	Departamento.IDDepartamento
Jones	33	Engenharia	33
Rafferty	31	Vendas	31
Robinson	34	Administrativo	34
Smith	34	Administrativo	34
Williams	NULO	NULO	NULO
Heisenberg	33	Engenharia	33

- Junção
  - Outer Join (Junção Externa) : Retorna os registros da junção, mesmo sem correspondentes na outra tabela
    - Right Outer Join

```
SELECT *
FROM empregado RIGHT OUTER JOIN departamento
  ON empregado.IDDepartamento = departamento.IDDepartamento;
```

Empregado.ÚltimoNome	Empregado.IDDepartamento	Departamento.NomeDepartamento	Departamento.IDDepartamento
Smith	34	Administrativo	34
Jones	33	Engenharia	33
Robinson	34	Administrativo	34
Heisenberg	33	Engenharia	33
Rafferty	31	Vendas	31
NULO	NULO	*Marketing*	*35*

serratec
Parque Tecnológico Região Serrana

- Junção
  - Saiba mais:
    - Cross Join
    - Full Outer Join

- Junção
  - Resumo:

Figura 7: Resumo das funções de Junção

Junção de produto cartesiano é uma junção entre duas tabelas que origina uma terceira tabela constituída por todos os elementos da primeira combinadas com todos os elementos da segunda.

Junção Interna todas linhas de uma tabela se relacionam com todas as linhas de outras tabelas se elas tiverem ao menos 1 campo em comum

Junção Externa é uma seleção que não requer que os registros de uma tabela possuam registros equivalentes em outras

- **Left Outer Join** todos os registros da tabela esquerda mesmo quando não exista registros correspondentes na tabela direita.
- **Right Outer Join** todos os registros da tabela direita mesmo quando não exista registros correspondentes na tabela esquerda.
- **Full Outer Join** Esta operação apresenta todos os dados das tabelas à esquerda e à direita, mesmo que não possuam correspondência em outra tabela

Fonte: www.wikiversity.org (2022)

serratec