

```
EMPRESA (
    cnpj    CHAR(14)  PRIMARY KEY,
```

```
    nome    VARCHAR(120) NOT NULL
);
```

## 2.2 FUNCIONARIO (e subclasses)

```
FUNCIONARIO (
    cpf      CHAR(11)  PRIMARY KEY,
    cnpj     (FK),      -- → EMPRESA
    nome     VARCHAR(120) NOT NULL,
    salario  DECIMAL(10,2) NOT NULL
);

TECNICO      (cpf (FK,PK));
ADMINISTRADOR (cpf (FK,PK));
OPERADOR     (cpf (FK,PK));

MOTORISTA (
    cpf      (FK,PK),
    cod_habilitacao CHAR(9) (NOT NULL, U)
);

COBRADOR (cpf (FK,PK));
```

## 2.3 ESCALA e CUMPRIDA\_POR

```
ESCALA (
    cod_escala INT PRIMARY KEY,
    hora_inicio TIMESTAMP,
    hora_fim    TIMESTAMP,
    dia_semana  INT CHECK (dia_semana BETWEEN 0 AND 6)
);

CUMPRIDA_POR (
    cpf      (FK),
    cod_escala (FK),
    PRIMARY KEY (cpf, cod_escala)
);
```

## 2.4 USUARIO • BILHETE • BILHETE\_COBRADOR • BILHETE\_USUARIO

```

USUARIO (
    cpf          CHAR(11) PRIMARY KEY,
    nome         VARCHAR(120) NOT NULL,
    data_nascimento DATE,
    genero       CHAR(1) CHECK (genero IN ('M','F','O'))
);

BILHETE (
    cod_cartao   INT PRIMARY KEY,
    saldo        DECIMAL(10,2) NOT NULL,
    tipo_cartao  INT NOT NULL,
);

BILHETE_COBRADOR (
    cod_cartao   (FK) (U),
    cpf (FK) PRIMARY KEY
);

BILHETE_USUARIO (
    cod_cartao   (FK) PRIMARY KEY,
    cpf          (FK)
);

```

## 2.5 LINHA • ROTA • PARTIDA\_PREVISTA

```

LINHA (
    cod_linha    INT PRIMARY KEY,
    nome_linha   VARCHAR(50) NOT NULL,
    data_criacao TIMESTAMP NOT NULL DEFAULT NOW()
);

ROTA (
    cod_rota     INT PRIMARY KEY,
    texto_letreiro VARCHAR(120) NOT NULL,
    cod_linha    (FK)
);

PARTIDA_PREVISTA (
    cod_rota     (FK),
    horario_previsto TIME NOT NULL,
    dia_semana   INT CHECK (dia_semana BETWEEN 0 AND 6),
    PRIMARY KEY (cod_rota, dia_semana, horario_previsto)
);

```

## 2.6 Rede de Pontos e Trechos

```
PONTO (  
    cod_ponto INT PRIMARY KEY,  
    latitude  FLOAT NOT NULL,  
    longitude FLOAT NOT NULL,  
    nome      VARCHAR(120),  
    (latitude, longitude) (U)  
);  
  
TRECHO (  
    cod_ponto_inicio (FK),    -- → PONTO  
    cod_ponto_fim     (FK),  
    comprimento       FLOAT,  
    tempo_medio       INTERVAL NOT NULL,  
    geometria         OBJECT NOT NULL,  
    PRIMARY KEY (cod_ponto_inicio, cod_ponto_fim)  
);  
  
PERCORRE (  
                                     -- rota × trecho  
    cod_rota      (FK),  
    ordem         INT NOT NULL,  
    cod_ponto_inicio (FK),  
    cod_ponto_fim   (FK),  
    PRIMARY KEY (cod_rota, ordem),  
    FOREIGN KEY (cod_ponto_inicio, cod_ponto_fim)  
        REFERENCES TRECHO(cod_ponto_inicio, cod_ponto_fim)  
);
```

## 2.7 Frota, Modelos e Garagens

```
FABRICANTE (  
    fabricante VARCHAR(60),  
    cod_modelo (FK)  
    (fabricante, cod_modelo) PRIMARY KEY  
);  
  
MODELO (  
    cod_modelo INT PRIMARY KEY,  
    tipo       VARCHAR(30) NOT NULL,  
    capacidade INT,  
    fabricante (FK)  
);  
  
GARAGEM (  
    cod_garagem INT PRIMARY KEY,  
    nome         VARCHAR(60) NOT NULL,  
    capacidade   INT,  
    endereco     VARCHAR(120) NOT NULL,  
    cidade        VARCHAR(60) NOT NULL,  
    estado        VARCHAR(2) NOT NULL,  
    pais          VARCHAR(3) NOT NULL,  
    PRIMARY KEY (cod_garagem),  
    FOREIGN KEY (cidade, estado, pais)  
        REFERENCES CIDADE(estado, cidade, pais)
```

```

        cod_garagem    INT PRIMARY KEY,
        logradouro     VARCHAR(120) NOT NULL,
        numero         INT NOT NULL,
        cidade         VARCHAR(80) NOT NULL,
        cep            CHAR(8) NOT NULL,
        capacidade      INT,
        estoque_diesel  FLOAT,
        num_eletropostos INT
    );

    VEICULO (
        cod_veiculo      INT PRIMARY KEY,
        latitude          FLOAT NOT NULL,
        longitude         FLOAT NOT NULL,
        data_inicio_operacao DATE NOT NULL,
        placa             CHAR(7) (U) NOT NULL,
        status            INT NOT NULL,
        ultima_atualizacao TIMESTAMP NOT NULL,
        cod_garagem       (FK),
        cod_modelo        (FK)
    );

```

## 2.8 Operações

```

    MANUTENCAO (
        cod_servico INT PRIMARY KEY,
        data        TIMESTAMP NOT NULL,
        descricao   CLOB NOT NULL,
        cpf         (FK),      -- técnico
        cod_veiculo (FK)
    );

    VIAGEM (
        cod_veiculo    (FK),
        datahora_inicio TIMESTAMP NOT NULL,
        cod_rota       (FK),
        cpf            (FK),   -- motorista
        datahora_fim   TIMESTAMP,
        PRIMARY KEY (cod_veiculo, cod_rota, datahora_inicio)
    );

    OCORRENCIA (
        cod_ocorrencia INT PRIMARY KEY,
        tipo           INT NOT NULL,
        descricao       CLOB NOT NULL,
        data_hora       TIMESTAMP NOT NULL,

```

```

        datahora_inicio (FK),
        cod_rota         (FK),
        cod_veiculo      (FK)
    );

    ENTRADA (
        horario          TIMESTAMP,
        cod_cartao        (FK),
        tarifa           DECIMAL(6,2) NOT NULL,
        datahora_inicio  TIMESTAMP,
        cod_rota          (FK),
        cod_veiculo       (FK),
        PRIMARY KEY (horario, datahora_inicio, cod_veiculo, cod_rota,
cod_cartao)
    );

```

### 3. Normalização (até FNBC)

Vamos avaliar a adequação do modelo a cada Forma Normal.

- **1FN** – todos os atributos são atômicos e não repetitivos. Neste caso, precisamos observar especialmente atributos multivalorados. No esquema EER, foram especificados três atributos compostos (**VEICULO.localizacao**, **GARAGEM.endereco** e **PONTO.localizacao**), e outros dois atributos multivalorados (**MODELO.fabricante** e **ROTA.viagem\_prevista**). Os atributos compostos foram representados de forma atômica no esquema relacional, e os atributos multivalorados foram convertidos em relações separadas. Assim, o esquema está na 1FN.
- **2FN** – nenhuma coluna não-chave depende somente de parte de uma PK composta. Neste caso, vamos observar as tabelas com chaves primárias compostas.
  - ENTRADA: O atributo **tarifa** depende dos dados da viagem (**datahora\_inicio**, **cod\_rota**, **cod\_veiculo**), do horário (**horario**) e do cartão utilizado (**cod\_cartao**). Portanto, não há dependência parcial.
  - VIAGEM: Um motorista (representado pelo atributo **cpf**) conduz uma viagem em um veículo (**cod\_veiculo**) num horário (**datahora\_inicio**) e rota (**cod\_rota**) específicos. A data de término de uma viagem também tem dependência funcional da combinação desses atributos. Assim, não há dependência parcial.
  - PARTIDA\_PREVISTA: Todos os atributos compõem a chave, de forma que não pode haver dependência parcial de um atributo não principal.
  - TRECHO: Todos os atributos dependem funcionalmente do ponto de início e fim (**cod\_ponto\_inicio**, **cod\_ponto\_fim**) simultaneamente, pois o trecho é definido por um intervalo entre pontos. Assim, não há dependência parcial.

- FABRICANTE\_MODELO: Todos os atributos compõem a chave e, portanto, não há dependência parcial de um atributo não principal. Portanto, o esquema está na 2FN.
- **3FN e FNBC** Detalhamos a justificativa sobre a adequação de cada tabela à FNBC abaixo. Ao garantir que o esquema está na FNBC, garantimos que está na 3FN e nas formas normais anteriores.

Tabela	Dependências Funcionais e Observações
EMPRESA	<code>cnpj</code> é determinante e superchave.
FUNCIONÁRIO	<code>cpf</code> → <code>cnpj</code> , <code>nome</code> , <code>salário</code> . Não há outras DF não triviais. Portanto, <code>cpf</code> é determinante e superchave.
TECNICO	Não existem DF não-triviais além da identidade; logo todo determinante é superchave.
ADMINISTRADOR	Não existem DF não-triviais além da identidade; logo todo determinante é superchave.
OPERADOR	Não existem DF não-triviais além da identidade; logo todo determinante é superchave.
MOTORISTA	Como <code>cod_habilitacao</code> foi declarado como UNIQUE NOT NULL, é considerado chave candidata. Dessa forma, as chaves candidatas são <code>cpf</code> e <code>cod_habilitacao</code> . Ambas são determinantes.
COBRADOR	Não existem DF não-triviais além da identidade; logo todo determinante é superchave.
ESCALA	<code>cod_escala</code> → <code>hora_inicio</code> , <code>hora_fim</code> , <code>dia_semana</code> . <code>cod_escala</code> é determinante e superchave.
CUMPRIDA_POR	Não existem DF não-triviais além da identidade; logo todo determinante é superchave.
USUARIO	Única DF não trivial é <code>cpf</code> → <code>nome</code> , <code>data_nascimento</code> , <code>genero</code> . <code>cpf</code> é determinante e superchave.
BILHETE	Chave candidata: <code>cod_cartao</code> . DF: <code>cod_cartao</code> → <code>saldo</code> , <code>tipo_cartao</code> . O determinante da tabela é a chave.
BILHETE_COBRADOR	Chaves candidatas: <code>cod_cartao</code> e <code>cpf</code> . DF: <code>cod_cartao</code> → <code>cpf</code> ou <code>cpf</code> → <code>cod_cartao</code> . Um dos determinantes da tabela é a chave candidata. Escolhemos um deles para garantir que a chave primária é minimal.
BILHETE_USUARIO	Chaves candidatas: <code>cod_cartao</code> . DF: <code>cod_cartao</code> → <code>cpf</code> . O determinante da tabela é a chave.

Tabela	Dependências Funcionais e Observações
LINHA	$\text{cod\_linha} \rightarrow \text{nome\_linha}, \text{data\_criacao}$ . Não existem outras DF não triviais. Logo, $\text{cod\_linha}$ é determinante e superchave.
ROTA	Como o $\text{texto\_letreiro}$ é único apenas dentro da linha, pode repetir em linhas diferentes. DF: $\text{cod\_rota} \rightarrow \text{texto\_letreiro}, \text{cod\_linha}$ . Logo, $\text{cod\_rota}$ é determinante e superchave.
PARTIDA_PREVISTA	A única DF não trivial é a própria PK ( $\text{cod\_rota}, \text{horario\_previsto}, \text{dia\_semana}$ ).
PONTO	Chaves candidatas: $\text{cod\_ponto}$ e ( $\text{latitude}, \text{longitude}$ ). Não há outras DF não triviais sem chaves candidatas como determinantes.
TRECHO	Como comprimento, tempo médio e geometria são computados individualmente para cada par de pontos, não há dependência funcional entre os atributos. ( $\text{cod\_ponto\_inicio}, \text{cod\_ponto\_fim}$ ) são chave e determinantes.
PERCORRE	( $\text{cod\_rota}, \text{ordem}$ ) é superchave e determinante.
FABRICANTE	Não existem DF não-triviais além da identidade; logo todo determinante é superchave.
MODELO	Considerando que " $\text{tipo}$ " não determina capacidade, $\text{cod\_modelo}$ é determinante e superchave.
GARAGEM	$\text{cod\_garagem}$ é determinante dos outros atributos (endereço e capacidade). Atributos como componentes do endereço ou capacidades não são principais, já que podem se repetir em garagens diferentes.
VEICULO	Não existem outras DF além das determinadas por $\text{placa}$ (ou $\text{cod\_veiculo}$ ).
MANUTENCAO	Não há outra DF não trivial. Então $\text{cod\_servico}$ é determinante e superchave.
VIAGEM	A única DF não trivial é a própria chave composta ( $\text{cod\_veiculo}, \text{datahora\_inicio}, \text{cod\_rota}$ ). Não existem outras DF em que o determinante não seja superchave.
OCORRENCIA	Não há outras DF não triviais além das determinadas pela superchave $\text{cod\_ocorrencia}$ .
ENTRADA	Todo determinante de DF não trivial é a própria chave composta.