/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* CENTRO UNIVERSITÁRIO DE BRASÍLIA \*/

/\* CURSO SUPERIOR TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS \*/

/\* DISCIPLINA DE BANCO DE DADOS \*/

/\* LINGUAGEM SQL \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-- DDL

-- Criar as tabelas do MER\_Fisico\_Pjt\_BD\_Pedido.

-- a. Script DDL para criar as tabelas do BD de pedidos: CLIENTE, PF, PJ, PRODUTO, FONE\_CLIENTE,

-- FONE\_VENEDOR, PRATELEIRA e VENDEDOR. Colocar as regras de integridade no script de criação.

-- SOLUÇÃO:

/\*==============================================================\*/

/\* Table: CLIENTE \*/

/\*==============================================================\*/

DROP TABLE CLIENTE CASCADE CONSTRAINT;

create table CLIENTE (

CODIGO INTEGER not null,

NOME varchar2(60) not null,

RUA varchar2(80) not null,

NR NUMBER(4) not null,

BAIRRO varchar2(50) not null,

COMPLEMENTO varchar2(100),

CIDADE varchar2(50) not null,

UF varchar2(2) not null,

CEP varchar2(10) not null,

constraint PK\_CLIENTE primary key (CODIGO)

);

/\*==============================================================\*/

/\* Table: VENDEDOR \*/

/\*==============================================================\*/

DROP TABLE VENDEDOR CASCADE CONSTRAINT;

create table VENDEDOR (

MATRICULA INTEGER not null,

NOME varchar2(60) not null,

CNPF varchar2(14) not null,

constraint PK\_VENDEDOR primary key (MATRICULA)

);

/\*==============================================================\*/

/\* Table: FONE\_CLIENTE \*/

/\*==============================================================\*/

DROP TABLE FONE\_CLIENTE CASCADE CONSTRAINT;

create table FONE\_CLIENTE (

NUMERO varchar2(14) not null,

CODIGO\_CLI INTEGER not null,

constraint PK\_FONE\_CLIENTE primary key (NUMERO, CODIGO\_CLI),

constraint FK\_FONE\_CLIENTE foreign key (CODIGO\_CLI) references CLIENTE (CODIGO)

);

/\*==============================================================\*/

/\* Table: FONE\_VENDEDOR \*/

/\*==============================================================\*/

DROP TABLE FONE\_VENDEDOR CASCADE CONSTRAINT;

create table FONE\_VENDEDOR (

NUMERO varchar2(14) not null,

MATRICULA\_VEN INTEGER not null,

constraint PK\_FONE\_VENDEDOR primary key (NUMERO, MATRICULA\_VEN),

constraint FK\_FONE\_VENDEDOR foreign key (MATRICULA\_VEN) references VENDEDOR (MATRICULA)

);

/\*==============================================================\*/

/\* Table: PF \*/

/\*==============================================================\*/

DROP TABLE PF;

create table PF (

CODIGO\_CLI INTEGER not null,

CNPF varchar2(14) not null,

RG varchar2(10) not null,

DATA\_NASCIMENTO DATE not null,

constraint PK\_PF primary key (CODIGO\_CLI),

constraint FK\_PF\_CLIENTE foreign key (CODIGO\_CLI) references CLIENTE (CODIGO)

);

/\*==============================================================\*/

/\* Table: PJ \*/

/\*==============================================================\*/

DROP TABLE Pj;

create table PJ (

CODIGO\_CLI INTEGER not null,

CNPJ varchar2(19) not null,

IE varchar2(10) not null,

NOME\_FANTASIA varchar2(60) not null,

constraint PK\_PJ primary key (CODIGO\_CLI),

constraint FK\_PJ\_CLIENTE foreign key (CODIGO\_CLI) references CLIENTE (CODIGO)

);

/\*==============================================================\*/

/\* Table: PRATELEIRA \*/

/\*==============================================================\*/

DROP TABLE PRATELEIRA CASCADE CONSTRAINT;

create table PRATELEIRA (

CODIGO NUMBER(3) not null,

DESCRICAO varchar2(50) not null,

NUMERO\_SECOES NUMBER(2) not null,

constraint PK\_PRATELEIRA primary key (CODIGO)

);

/\*==============================================================\*/

/\* Table: PRODUTO \*/

/\*==============================================================\*/

DROP TABLE PRODUTO CASCADE CONSTRAINT;

create table PRODUTO (

CODIGO INTEGER not null,

DESCRICAO varchar2(80) not null,

UNID\_MEDIDA varchar2(10) not null,

constraint PK\_PRODUTO primary key (CODIGO)

);

-- b. Criar as tabelas associativas: PEDIDO, ESTOQUE, CAPACIDADE\_ESTOQUE E ITEM\_PRODUTO.

-- Não incluir na criação dessas tabelas as chaves primária (PK) e estrangeiras (FK).

-- SOLUÇÃO:

/\*==============================================================\*/

/\* Table: PEDIDO \*/

/\*==============================================================\*/

DROP TABLE PEDIDO CASCADE CONSTRAINT;

create table PEDIDO (

NUMERO NUMBER(6) not null,

DATA DATE not null,

PRAZO\_ENTREGA DATE,

RUA varchar2(80),

NR NUMBER(4),

BAIRRO varchar2(50),

COMPLEMENTO varchar2(100),

CIDADE varchar2(50),

UF varchar2(2),

CEP varchar2(10),

TOTAL\_FATURA NUMBER(8,2),

CODIGO\_CLI INTEGER not null,

MATRICULA\_VEN INTEGER not null

);

/\*==============================================================\*/

/\* Table: ITEM\_PRODUTO \*/

/\*==============================================================\*/

DROP TABLE ITEM\_PRODUTO;

create table ITEM\_PRODUTO (

CODIGO\_PRO INTEGER not null,

NUMERO\_PED NUMBER(6) not null,

QUANTIDADE NUMBER(5) not null,

PRECO\_UNITARIO NUMBER(5,2) not null,

VALOR\_ITEM NUMBER(7,2)

);

/\*==============================================================\*/

/\* Table: CAPACIDADE\_ESTOQUE \*/

/\*==============================================================\*/

DROP TABLE CAPACIDADE\_ESTOQUE;

create table CAPACIDADE\_ESTOQUE (

CODIGO\_PRO INTEGER not null,

CODIGO\_PRA NUMBER(3) not null,

QUANTIDADE NUMBER(5) not null

);

/\*==============================================================\*/

/\* Table: ESTOQUE \*/

/\*==============================================================\*/

DROP TABLE ESTOQUE;

create table ESTOQUE (

DATA\_ENTRADA DATE not null,

CODIGO\_PRO INTEGER not null,

CODIGO\_PRA NUMBER(3) not null,

QUANTIDADE NUMBER(5) not null,

DATA\_VALIDADE DATE

);

-- c. Criar as constraints de chave primaria (primary key) para as tabelas associativas.

-- RESOLUÇÃO:

alter table PEDIDO

ADD constraint PK\_PEDIDO primary key (NUMERO);

alter table ITEM\_PRODUTO

ADD constraint PK\_ITEM\_PRODUTO primary key (CODIGO\_PRO, NUMERO\_PED);

alter table CAPACIDADE\_ESTOQUE

ADD constraint PK\_CAPACIDADE\_ESTOQUE primary key (CODIGO\_PRO, CODIGO\_PRA);

alter table ESTOQUE

ADD constraint PK\_ESTOQUE primary key (CODIGO\_PRO, CODIGO\_PRA, DATA\_ENTRADA);

-- d. Criar as constraints das chaves estrangeiras (foreing key) para as tabelas associativas.

-- RESOLUÇÃO:

alter table PEDIDO

add constraint FK\_PED\_VENDEDOR foreign key (MATRICULA\_VEN)

references VENDEDOR (MATRICULA);

alter table PEDIDO

add constraint FK\_PED\_CLIENTE foreign key (CODIGO\_CLI)

references CLIENTE (CODIGO);

alter table ITEM\_PRODUTO

add constraint FK\_ITEM\_PEDIDO foreign key (NUMERO\_PED)

references PEDIDO (NUMERO);

alter table ITEM\_PRODUTO

add constraint FK\_ITEM\_PRODUTO foreign key (CODIGO\_PRO)

references PRODUTO (CODIGO);

alter table CAPACIDADE\_ESTOQUE

add constraint FK\_CAP\_EST\_PRATELEIRA foreign key (CODIGO\_PRA)

references PRATELEIRA (CODIGO);

alter table CAPACIDADE\_ESTOQUE

add constraint FK\_CAP\_EST\_PRODUTO foreign key (CODIGO\_PRO)

references PRODUTO (CODIGO);

alter table ESTOQUE

add constraint FK\_EST\_PRATELEIRA foreign key (CODIGO\_PRA)

references PRATELEIRA (CODIGO);

alter table ESTOQUE

add constraint FK\_EST\_PRODUTO foreign key (CODIGO\_PRO)

references PRODUTO (CODIGO);

-- e. Criar uma constraint que garanta que o valor do NUMERO da tabela PEDIDO esteja entre, inclusive,

-- 1 e 99999.

-- RESOLUÇÃO:

alter table PEDIDO

add constraint CK\_NUMERO\_PEDIDO check (NUMERO BETWEEN 1 AND 99999);

-- f. Juntar as tabelas FONE\_CLIENTE e FONE\_VENDEDOR em uma única tabela, relacionando-a com CLIENTE e

-- VENDEDOR. Criar as regras de integridade básica necessárias para garantir a qualidade

-- dos dados. A nova tabela se chamará FONE, com as colunas CODIGO\_CLI, MATRICULA\_VEN, NUMERO.

-- Quando a coluna CODIGO\_CLI estivel preenchida a MATRICULA\_VEN estará vazia e vice-versa.

-- RESOLUÇÃO:

/\*==============================================================\*/

/\* Table: FONE \*/

/\*==============================================================\*/

DROP TABLE FONE CASCADE CONSTRAINT;

create table FONE (

CODIGO INTEGER not null,

NUMERO varchar2(14) not null,

CODIGO\_CLI INTEGER not null,

MATRICULA\_VEN INTEGER not null,

constraint PK\_FONE primary key (CODIGO),

constraint FK\_FONE\_CLI foreign key (CODIGO\_CLI) references CLIENTE (CODIGO),

constraint FK\_FONE\_VEN foreign key (MATRICULA\_VEN) references VENDEDOR (MATRICULA),

constraint UK\_FONE unique key (NUMERO,CODIGO\_CLI,MATRICULA\_VEN)

);

-- g. Criar os indexes das tabelas PF e PJ. Estes indexes serão únicos e utilizados

-- como listas invertidas.

-- RESOLUÇÃO:

CREATE UNIQUE INDEX IND\_CNPF\_PF

ON PF (CNPF DESC);

CREATE UNIQUE INDEX IND\_CNPF\_PJ

ON PJ (CNPJ DESC);

-- h. Criar comentários para a tabela ITEM\_PRODUTO e suas respectivas colunas.

-- RESOLUÇÃO:

comment on table ITEM\_PRODUTO is

'Mantém os dados dos itens de produtos dos pedidos no negócio do sistema de pedido.';

comment on column ITEM\_PRODUTO.CODIGO\_PRO is

'Código que identifica cada produto do negócio. PEx: 12345.';

comment on column ITEM\_PRODUTO.NUMERO\_PED is

'Número que identifica cada pedido efetuado no sistema. Seu valor é sequencial e é gerado por uma sequence de nome SEQ\_NUNMERO\_PEDIDO. PEx: 123456.';

-- i. Criar uma sequence de nome SEQ\_ITEM\_ID para o campo CODIGO de ITEM\_PRODUTO, com

-- incremento de 2.

-- RESOLUÇÃO:

create sequence SEQ\_ITEM\_ID increment by 2;