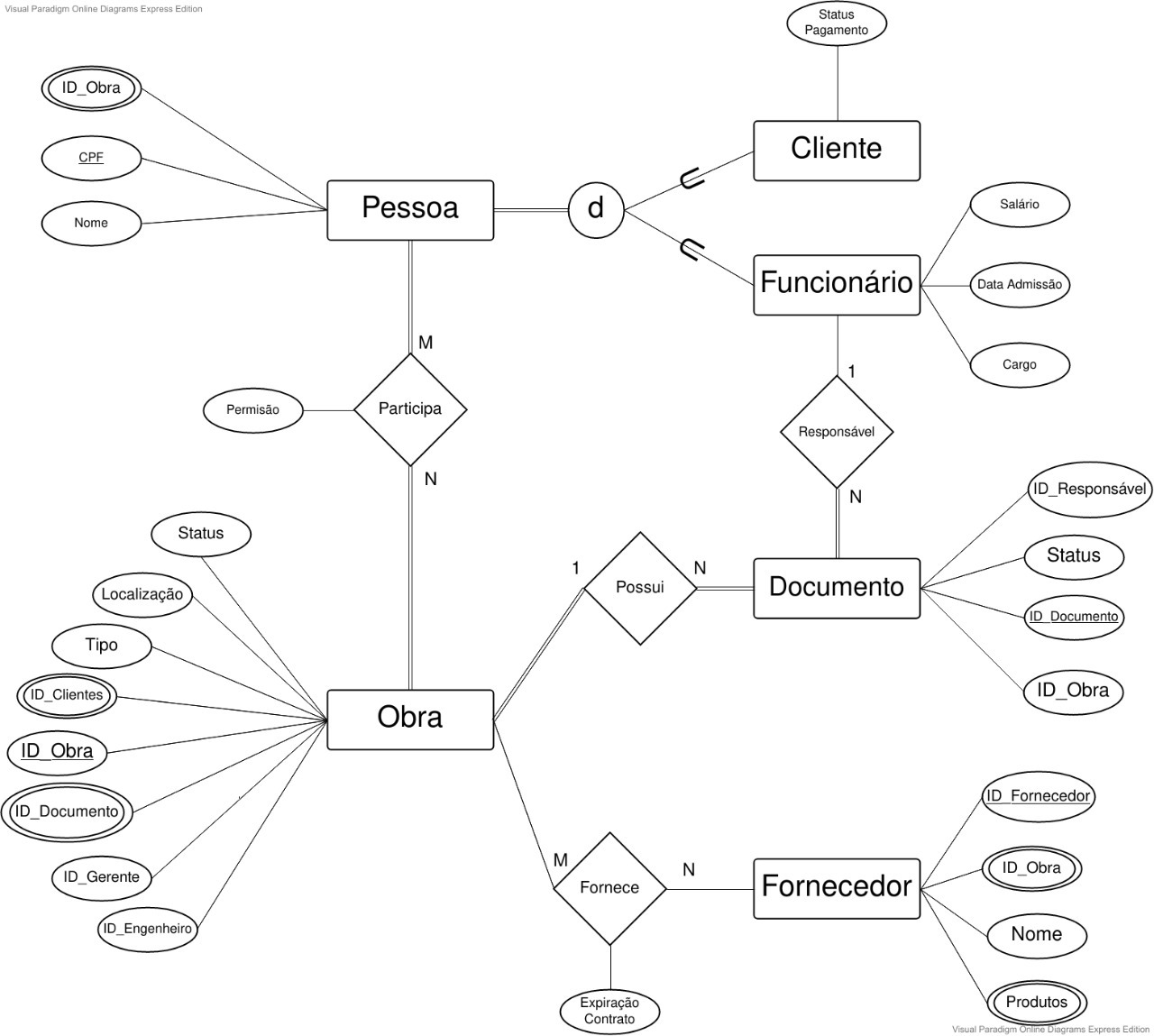
UNIVERSIDADE FEDERAL DE MINAS GERAIS

DISCIPLINA: INTRODUÇÃO A BANCO DE DADOS - DCC011

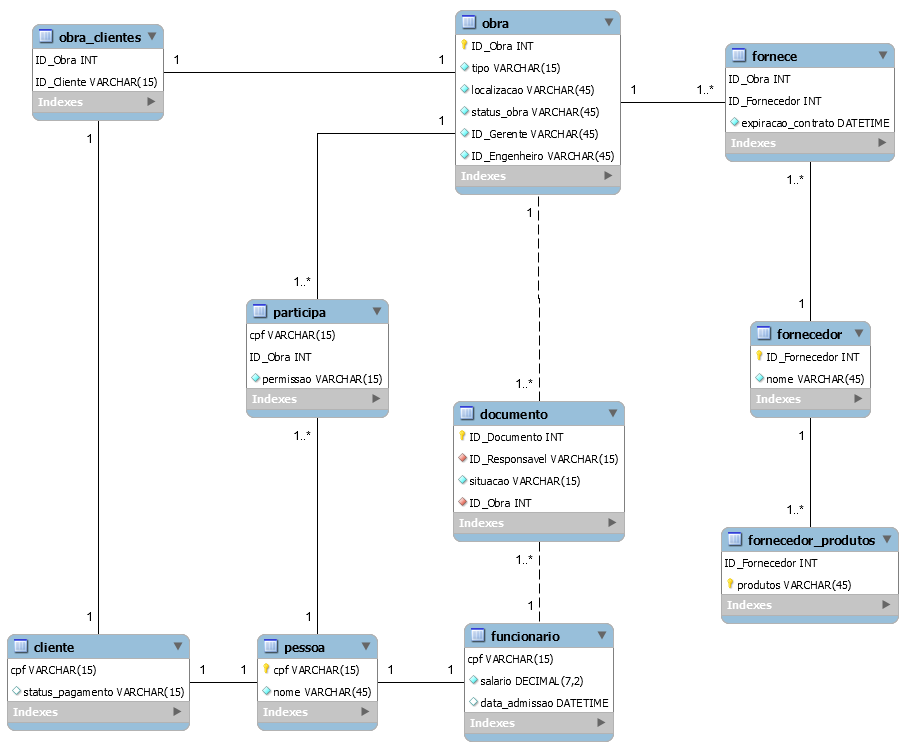
TRABALHO PRÁTICO 2

Integrantes:

Leandro Luiz Duarte Teixeira



Projeto Lógico



Este é o projeto lógico que fizemos a partir do Diagrama Entidade-Relacionamento apresentado acima. A modelagem do fornecedor foi a mais objetiva; ele possui um ID e um nome na própria tabela. Para implementar os produtos, foi usada uma tabela fornecedor-produtos. Foi cogitado também criar uma tabela só para os produtos e fazer disso uma relação M x N. Essa ideia, porém, foi descartada; fazer isso exigiria uma padronização da forma de apresentação dos produtos.

Da forma como implementamos, um fornecedor está livre para fornecer “cimento” enquanto um outro pode optar por fornecer apenas “cimento da marca Mauá”, sem exigir que o primeiro modifique os dados para apresentar todas as marcas e variedades que ele oferta ou que o segundo generalize o produto mesmo quando ele só vende uma marca específica que pode não ser a desejada para aquela obra específica.

A variável multivalorada “Id\_Obra” presente em Fornecedor pode ser obtida implicitamente através da tabela Fornece, que representa a relação entre o fornecedor e a obra. Ao inserir as restrições de modificação na tabela Fornece, foi decidido inserir bloqueio a mudança e cascata a remoção em ambas as chaves. A lógica é, uma vez firmado um contrato entre as duas partes e inserida a data de expiração, nenhuma outra relação pode vir a herdar essa data de expiração sem o firmamento de um novo contrato. Por fins de segurança, se for necessário trocar o ID de um fornecedor ou de uma obra, os contratos previamente definidos deverão ser cancelados e refirmados. Por outro lado, se um fornecedor ou obra deixa de existir, todos os contratos a eles relacionados podem ser removidos.

Ao modelar a obra, foi levado muito em consideração as potenciais exigências do mundo real também. Além dos dados univalorados que são mais objetivos, toda obra deve ter ao menos um documento. Porém, inserir uma restrição de integridade forçaria os usuários do banco de dados a só inserirem uma obra no sistema uma vez que seu documento tenha sido registrado. Nós achamos melhor permitir que a obra apareça no sistema no momento em que ela tem um engenheiro e um gerente já registrados. Por outro lado, os documentos só podem ser inseridos no sistema se possuírem uma obra correspondida. E, seguindo a mesma linha do fornecedor-obra, não é possível modificar a chave de um responsável ou obra enquanto estes tiverem documentos atribuídos a eles. É possível remover uma obra com documentos, isso gerará uma reação em cascata, porém não é possível remover um funcionário responsável por documento enquanto ele ainda possuir documentos atribuídos

Situação parecida acontece com a relação obra-cliente. Uma forma alternativa de implementar seria inserir o Id da obra como chave estrangeira na tabela cliente, visto que todo cliente deve possuir uma obra relacionada (caso contrário, não deve ser considerado cliente ainda). Todavia, optamos por criar uma tabela obra-clientes justamente porque ela permite aos usuários do sistema inserir um cliente que ainda está escolhendo a obra que deseja comprar. Outra coisa importante para se notar sobre a relação obra-cliente é que há um bloqueio a remoção inserido apenas no ID\_Obra. Um cliente pode decidir desistir de uma obra a qualquer momento sem dar uma satisfação direta às respectivas obras. Porém, antes de deletar uma obra, é importante garantir que todos os seus clientes tenham sido removidos ou atribuídos a outras obras.

Código

CREATE TABLE `cliente` (

`cpf` varchar(15) NOT NULL,

`status\_pagamento` varchar(15) DEFAULT NULL,

PRIMARY KEY (`cpf`),

UNIQUE KEY `cpf\_UNIQUE` (`cpf`),

CONSTRAINT `cpf` FOREIGN KEY (`cpf`) REFERENCES `pessoa` (`cpf`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `documento` (

`ID\_Documento` int NOT NULL,

`ID\_Responsavel` varchar(15) NOT NULL,

`situacao` varchar(15) NOT NULL,

`ID\_Obra` int NOT NULL,

PRIMARY KEY (`ID\_Documento`),

UNIQUE KEY `ID\_Documento\_UNIQUE` (`ID\_Documento`),

UNIQUE KEY `ID\_Obra\_UNIQUE` (`ID\_Obra`),

KEY `Responsavel\_idx` (`ID\_Responsavel`),

CONSTRAINT `Obra` FOREIGN KEY (`ID\_Obra`) REFERENCES `obra` (`ID\_Obra`) ON DELETE CASCADE ON UPDATE RESTRICT,

CONSTRAINT `Responsavel` FOREIGN KEY (`ID\_Responsavel`) REFERENCES `funcionario` (`cpf`) ON DELETE RESTRICT ON UPDATE RESTRICT

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `fornece` (

`ID\_Obra` int NOT NULL,

`ID\_Fornecedor` int NOT NULL,

`expiracao\_contrato` datetime NOT NULL,

PRIMARY KEY (`ID\_Obra`,`ID\_Fornecedor`),

KEY `Forneceddor\_idx` (`ID\_Fornecedor`),

CONSTRAINT `Fornecedor\_For` FOREIGN KEY (`ID\_Fornecedor`) REFERENCES `fornecedor` (`ID\_Fornecedor`) ON DELETE CASCADE ON UPDATE RESTRICT,

CONSTRAINT `Obra\_For` FOREIGN KEY (`ID\_Obra`) REFERENCES `obra` (`ID\_Obra`) ON DELETE CASCADE ON UPDATE RESTRICT

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `fornecedor` (

`ID\_Fornecedor` int NOT NULL,

`nome` varchar(45) NOT NULL,

PRIMARY KEY (`ID\_Fornecedor`),

UNIQUE KEY `ID\_Fornecedor\_UNIQUE` (`ID\_Fornecedor`),

UNIQUE KEY `nome\_UNIQUE` (`nome`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `fornecedor\_produtos` (

`ID\_Fornecedor` int NOT NULL,

`produtos` varchar(45) NOT NULL,

PRIMARY KEY (`ID\_Fornecedor`,`produtos`),

CONSTRAINT `Fornecedor\_ForPro` FOREIGN KEY (`ID\_Fornecedor`) REFERENCES `fornecedor` (`ID\_Fornecedor`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `funcionario` (

`cpf` varchar(15) NOT NULL,

`salario` decimal(7,2) NOT NULL,

`data\_admissao` datetime DEFAULT NULL,

PRIMARY KEY (`cpf`),

UNIQUE KEY `cpf\_UNIQUE` (`cpf`),

CONSTRAINT `cpf\_Func` FOREIGN KEY (`cpf`) REFERENCES `pessoa` (`cpf`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `obra` (

`ID\_Obra` int NOT NULL,

`tipo` varchar(15) NOT NULL,

`localizacao` varchar(45) NOT NULL,

`status\_obra` varchar(45) NOT NULL,

`ID\_Gerente` varchar(45) NOT NULL,

`ID\_Engenheiro` varchar(45) NOT NULL,

PRIMARY KEY (`ID\_Obra`),

UNIQUE KEY `ID\_Obra\_UNIQUE` (`ID\_Obra`),

UNIQUE KEY `ID\_Gerente\_UNIQUE` (`ID\_Gerente`),

UNIQUE KEY `ID\_Engenheiro\_UNIQUE` (`ID\_Engenheiro`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `obra\_clientes` (

`ID\_Obra` int NOT NULL,

`ID\_Cliente` varchar(15) NOT NULL,

PRIMARY KEY (`ID\_Obra`,`ID\_Cliente`),

UNIQUE KEY `ID\_Cliente\_UNIQUE` (`ID\_Cliente`),

UNIQUE KEY `ID\_Obra\_UNIQUE` (`ID\_Obra`),

CONSTRAINT `ID\_Cliente\_ObraCli` FOREIGN KEY (`ID\_Cliente`) REFERENCES `cliente` (`cpf`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `ID\_Obra\_ObraCli` FOREIGN KEY (`ID\_Obra`) REFERENCES `obra` (`ID\_Obra`) ON DELETE RESTRICT ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `participa` (

`cpf` varchar(15) NOT NULL,

`ID\_Obra` int NOT NULL,

`permissao` varchar(15) NOT NULL,

PRIMARY KEY (`cpf`,`ID\_Obra`),

KEY `ID\_Obra\_idx` (`ID\_Obra`),

CONSTRAINT `cpf\_Part` FOREIGN KEY (`cpf`) REFERENCES `pessoa` (`cpf`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `ID\_Obra\_Part` FOREIGN KEY (`ID\_Obra`) REFERENCES `obra` (`ID\_Obra`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `pessoa` (

`cpf` varchar(15) NOT NULL,

`nome` varchar(45) NOT NULL,

PRIMARY KEY (`cpf`),

UNIQUE KEY `cpf\_UNIQUE` (`cpf`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;