```sql
CREATE DATABASE easybuild_projetos;
go

USE easybuild_projetos;

CREATE TABLE alocacao(
    codigo_funcionario    int    NOT NULL,
    codigo_projeto        int    NOT NULL
);

ALTER TABLE alocacao ADD CONSTRAINT PK_alocacao
          PRIMARY KEY CLUSTERED (codigo_funcionario, codigo_projeto);

CREATE TABLE cargo(
    codigo_cargo    int             IDENTITY(1,1),
    nome_cargo      varchar(50)     NOT NULL
);

ALTER TABLE cargo ADD CONSTRAINT PK_cargo
          PRIMARY KEY CLUSTERED (codigo_cargo);

CREATE TABLE departamento(
    codigo_departamento             int             IDENTITY(1,1),
    nome_departamento               varchar(200)    NOT NULL,
    codigo_funcionario_gerente      int             NULL,
    codigo_departamento_superior    int             NULL
);

ALTER TABLE departamento ADD CONSTRAINT PK_departamento
          PRIMARY KEY CLUSTERED (codigo_departamento);

CREATE TABLE engenheiro(
    codigo_funcionario    int               NOT NULL,
    numero_crea           decimal(12, 0)    NOT NULL
);

ALTER TABLE engenheiro ADD CONSTRAINT PK_engenheiro
          PRIMARY KEY CLUSTERED (codigo_funcionario);

CREATE TABLE funcionario(
    codigo_funcionario     int               IDENTITY(1,1),
    nome_funcionario       varchar(200)      NOT NULL,
    cpf_funcionario        decimal(11, 0)    NOT NULL,
    salario_funcionario    decimal(10, 2)    NULL,
    codigo_departamento    int               NOT NULL,
    codigo_cargo           int               NOT NULL
);

ALTER TABLE funcionario ADD CONSTRAINT PK_funcionario
          PRIMARY KEY CLUSTERED (codigo_funcionario);

CREATE TABLE motorista(
    codigo_funcionario    int               NOT NULL,
    numero_cnh            decimal(11, 0)    NOT NULL,
    tipo_cnh              char(1)           NOT NULL
);

ALTER TABLE motorista ADD CONSTRAINT PK_motorista
          PRIMARY KEY CLUSTERED (codigo_funcionario);

CREATE TABLE projeto(
    codigo_projeto                     int             IDENTITY(1,1),
    nome_projeto                       varchar(200)    NOT NULL,
    codigo_departamento_responsavel    int             NOT NULL
);

ALTER TABLE projeto ADD CONSTRAINT PK_projeto
          PRIMARY KEY CLUSTERED (codigo_projeto);

CREATE INDEX IX_funcionario_projeto ON alocacao(codigo_funcionario);
CREATE INDEX IX_projeto ON alocacao(codigo_projeto);

CREATE UNIQUE INDEX AK_nome_cargo ON cargo(nome_cargo);

CREATE INDEX IX_gerente ON departamento(codigo_funcionario_gerente);
CREATE INDEX IX_superior ON departamento(codigo_departamento_superior);

CREATE INDEX IX_funcionario_engenheiro ON
engenheiro(codigo_funcionario);

CREATE INDEX IX_departamento ON funcionario(codigo_departamento);
CREATE INDEX IX_cargo ON funcionario(codigo_cargo);

CREATE INDEX IX_funcionario_motorista ON motorista(codigo_funcionario);

CREATE INDEX IX_responsavel ON projeto(codigo_departamento_responsavel);

ALTER TABLE departamento ADD CONSTRAINT AK_nome_departamento
          UNIQUE NONCLUSTERED (nome_departamento);


ALTER TABLE engenheiro ADD CONSTRAINT AK_crea
          UNIQUE NONCLUSTERED (numero_crea);

ALTER TABLE funcionario ADD CONSTRAINT AK_cpf
          UNIQUE NONCLUSTERED (cpf_funcionario);

ALTER TABLE motorista ADD CONSTRAINT AK_cnh
          UNIQUE NONCLUSTERED (numero_cnh);

ALTER TABLE projeto ADD CONSTRAINT AK_nome_projeto
          UNIQUE NONCLUSTERED (nome_projeto);

ALTER TABLE alocacao ADD CONSTRAINT FK_funcionario_alocacao
    FOREIGN KEY (codigo_funcionario)
    REFERENCES funcionario(codigo_funcionario);

ALTER TABLE alocacao ADD CONSTRAINT FK_projeto_alocacao
    FOREIGN KEY (codigo_projeto)
    REFERENCES projeto(codigo_projeto);

ALTER TABLE departamento ADD CONSTRAINT
FK_departamento_departamento_superior
    FOREIGN KEY (codigo_departamento_superior)
    REFERENCES departamento(codigo_departamento);

ALTER TABLE departamento ADD CONSTRAINT FK_funcionario_departamento
    FOREIGN KEY (codigo_funcionario_gerente)
    REFERENCES funcionario(codigo_funcionario);

ALTER TABLE engenheiro ADD CONSTRAINT FK_funcionario_engenheiro
    FOREIGN KEY (codigo_funcionario)
    REFERENCES funcionario(codigo_funcionario);

ALTER TABLE funcionario ADD CONSTRAINT FK_cargo_funcionario
    FOREIGN KEY (codigo_cargo)
    REFERENCES cargo(codigo_cargo);

ALTER TABLE funcionario ADD CONSTRAINT FK_departamento_funcionario
    FOREIGN KEY (codigo_departamento)
    REFERENCES departamento(codigo_departamento);

ALTER TABLE motorista ADD CONSTRAINT FK_funcionario_motorista
    FOREIGN KEY (codigo_funcionario)
    REFERENCES funcionario(codigo_funcionario);

ALTER TABLE projeto ADD CONSTRAINT FK_departamento_projeto
    FOREIGN KEY (codigo_departamento_responsavel)
    REFERENCES departamento(codigo_departamento);
```

String types:

Data type	Description	Storage
char(n)	Fixed width character string. Maximum 8,000 characters	Defined width
varchar(n)	Variable width character string. Maximum 8,000 characters	2 bytes + number of chars
varchar(max)	Variable width character string. Maximum 1,073,741,824 characters	2 bytes + number of chars
text	Variable width character string. Maximum 2GB of text data	4 bytes + number of chars
nchar	Fixed width Unicode string. Maximum 4,000 characters	Defined width x 2
nvarchar	Variable width Unicode string. Maximum 4,000 characters	
nvarchar(max)	Variable width Unicode string. Maximum 536,870,912 characters	
ntext	Variable width Unicode string. Maximum 2GB of text data	
bit	Allows 0, 1, or NULL	
binary(n)	Fixed width binary string. Maximum 8,000 bytes	
varbinary	Variable width binary string. Maximum 8,000 bytes	
varbinary(max)	Variable width binary string. Maximum 2GB	
image	Variable width binary string. Maximum 2GB	

Number types:

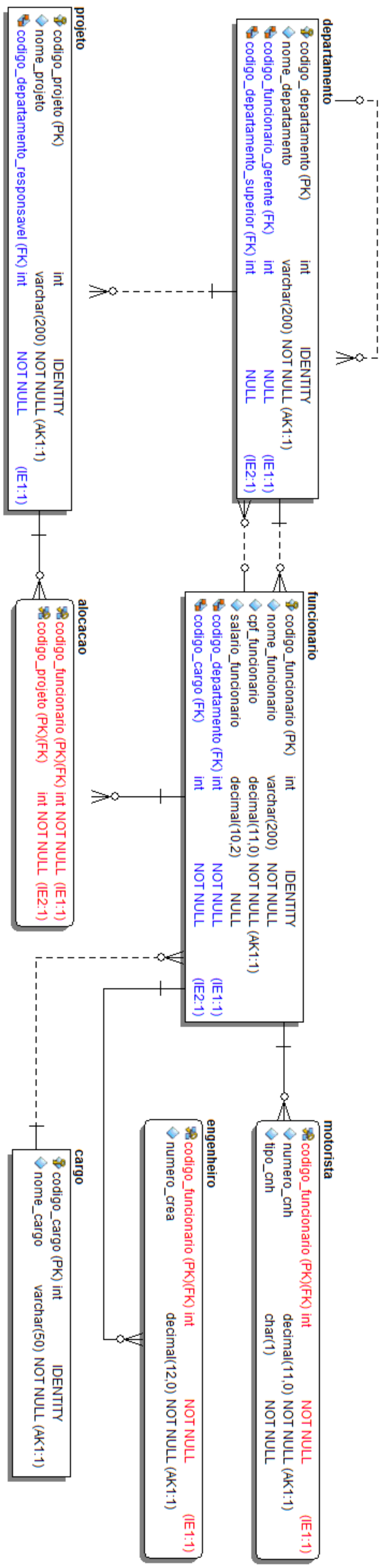
Data type	Description	Storage
tinyint	Allows whole numbers from 0 to 255	1 byte
smallint	Allows whole numbers between -32,768 and 32,767	2 bytes
int	Allows whole numbers between -2,147,483,648 and 2,147,483,647	4 bytes
bigint	Allows whole numbers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807	8 bytes
decimal(p,s)	Fixed precision and scale numbers. Allows numbers from -10^38 +1 to 10^38 –1. The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18. The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0	5-17 bytes
numeric(p,s)	Fixed precision and scale numbers. Allows numbers from -10^38 +1 to 10^38 –1. The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18. The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0	5-17 bytes
smallmoney	Monetary data from -214,748.3648 to 214,748.3647	4 bytes
money	Monetary data from -922,337,203,685,477.5808 to 922,337,203,685,477.5807	8 bytes
float(n)	Floating precision number data from -1.79E + 308 to 1.79E + 308. The n parameter indicates whether the field should hold 4 or 8 bytes. float(24) holds a 4-byte field and float(53) holds an 8-byte field. Default value of n is 53.	4 or 8 bytes
real	Floating precision number data from -3.40E + 38 to 3.40E + 38	4 bytes

Date types:

Data type	Description	Storage
datetime	From January 1, 1753 to December 31, 9999 with an accuracy of 3.33 milliseconds	8 bytes
datetime2	From January 1, 0001 to December 31, 9999 with an accuracy of 100 nanoseconds	6-8 bytes
smalldatetime	From January 1, 1900 to June 6, 2079 with an accuracy of 1 minute	4 bytes
date	Store a date only. From January 1, 0001 to December 31, 9999	3 bytes
time	Store a time only to an accuracy of 100 nanoseconds	3-5 bytes
datetimeoffset	The same as datetime2 with the addition of a time zone offset	8-10 bytes
timestamp	Stores a unique number that gets updated every time a row gets created or modified. The timestamp value is based upon an internal clock and does not correspond to real time. Each table may have only one timestamp variable	

Other data types:

Data type	Description
sql_variant	Stores up to 8,000 bytes of data of various data types, except text, ntext, and timestamp
uniqueidentifier	Stores a globally unique identifier (GUID)
xml	Stores XML formatted data. Maximum 2GB
cursor	Stores a reference to a cursor used for database operations
table	Stores a result-set for later processing



departamento
codigo_departamento (PK) int IDENTITY
nome_departamento varchar(200) NOT NULL (AK1:1)
codigo_funcionario_gerente (FK) int NULL (IE1:1)
codigo_departamento_superior (FK) int NULL (IE2:1)
projeto
codigo_projeto (PK) int IDENTITY
nome_projeto varchar(200) NOT NULL (AK1:1)
codigo_departamento_responsavel (FK) int NOT NULL (IE1:1)
funcionario
codigo_funcionario (PK) int IDENTITY
nome_funcionario varchar(200) NOT NULL
cpf_funcionario decimal(11,0) NOT NULL (AK1:1)
salario_funcionario decimal(10,2) NULL
codigo_departamento (FK) int NOT NULL (IE1:1)
codigo_cargo (FK) int NOT NULL (IE2:1)
alocacao
codigo_funcionario (PK)(FK) int NOT NULL (IE1:1)
codigo_projeto (PK)(FK) int NOT NULL (IE2:1)
cargo
codigo_cargo (PK) int IDENTITY
nome_cargo varchar(50) NOT NULL (AK1:1)
engenheiro
codigo_funcionario (PK)(FK) int NOT NULL (IE1:1)
numero_crea decimal(12,0) NOT NULL (AK1:1)
motorista
codigo_funcionario (PK)(FK) int NOT NULL (IE1:1)
numero_cnh decimal(11,0) NOT NULL (AK1:1)
tipo_cnh char(1) NOT NULL