CREAZIONE TIPI ENUMERAZIONI

CREATE TYPE tipologia AS ENUM ('Ricerca Base' , 'Ricerca Industriale' , 'Ricerca Sperimentale' , 'Sviluppo Sperimentale');

CREATE TYPE ruolo AS ENUM ('Project Manager' , 'Dipendente Comune');

CREATE TYPE ambito AS ENUM ('Economia' , 'Medicina');

CREATE TYPE modalita AS ENUM('Telematica', 'Fisica');

CREATE TYPE piattaforma AS ENUM ('Microsoft Teams' , 'Discord' ,'Google Meet' , 'Zoom' , 'Cisco Webex' , 'Skype');

CREAZIONE TABELLE

CREATE TABLE Progetto

(

CodProgetto int PRIMARY KEY ,

NomeProgetto VARCHAR(50) NOT NULL,

TipoProgetto tipologia NOT NULL,

DescrizioneProgetto VARCHAR(300) ,

DataCreazione DATE NOT NULL,

DataTerminazione DATE,

Scadenza DATE

);

CREATE TABLE RuoloDipendente

(

IDRuolo int PRIMARY KEY,

RuoloDip ruolo NOT NULL

);

CREATE TABLE Provincia

(

CodProvincia CHAR(2) PRIMARY KEY,

NomeProvincia VARCHAR(20) NOT NULL

);

CREATE TABLE Comune

(

CodComune CHAR(4) PRIMARY KEY,

NomeComune VARCHAR(30) NOT NULL,

Provincia CHAR(2),

FOREIGN KEY (Provincia) REFERENCES Provincia(CodProvincia)

);

CREATE TABLE Dipendente

(

CF CHAR(16) PRIMARY KEY,

Nome VARCHAR(30) NOT NULL,

Cognome VARCHAR(30) NOT NULL,

DataNascita DATE NOT NULL,

Indirizzo VARCHAR (100) NOT NULL,

Email VARCHAR(100) UNIQUE NOT NULL,

TelefonoCasa VARCHAR(15),

Cellulare VARCHAR(15),

Salario FLOAT NOT NULL,

Valutazione FLOAT NOT NULL,

Pass VARCHAR(50) NOT NULL,

ComuneNascita CHAR(4),

FOREIGN KEY (ComuneNascita) REFERENCES Comune(CodComune)

)

CREATE TABLE Skill

(

IDSkill int PRIMARY KEY,

NomeSkill VARCHAR(30) NOT NULL

);

CREATE TABLE AmbitoProgetto

(

IDAmbito INT PRIMARY KEY,

NomeAmbito ambito NOT NULL

);

CREATE TABLE Meeting

(

IDMeeting INT PRIMARY KEY,

DataInizio DATE NOT NULL,

OrarioInizio TIME NOT NULL,

DataFine DATE NOT NULL,

OrarioFine TIME NOT NULL,

Modalità modalita NOT NULL,

Piattaforma piattaforma NOT NULL

);

CREATE TABLE SalaRiunione

(

CodSala VARCHAR(5) PRIMARY KEY,

Capienza INT NOT NULL,

IndirizzoSede VARCHAR(50) NOT NULL,

Piano INT NOT NULL

);

CREAZIONE TABELLE COLLEGAMENTI

CREATE TABLE RuoloDipendenteLink

(

Ruolo int ,

Progetto int,

FOREIGN KEY(Ruolo) REFERENCES RuoloDipendente(IDRuolo),

FOREIGN KEY(Progetto) REFERENCES Progetto(CodProgetto)

);

CREATE TABLE SkillDipendenteLink

(

Skill int,

Dipendente CHAR(16),

FOREIGN KEY (Skill) REFERENCES Skill (IDSkill),

FOREIGN KEY (Dipendente) REFERENCES Dipendente (CF)

);

CREATE TABLE AmbitoProgettoLink

(

Ambito INT,

Progetto INT,

FOREIGN KEY (Ambito) REFERENCES AmbitoProgetto(IDAmbito),

FOREIGN KEY (Progetto) REFERENCES Progetto(CodProgetto)

);

CREATE TABLE MeetingDipendenteLink

(

Meeting INT,

Dipendente CHAR(16),

FOREIGN KEY (Meeting) REFERENCES Meeting(IDMeeting),

FOREIGN KEY (Dipendente) REFERENCES Dipendente(CF)

);

CREATE TABLE MeetingProgettoLink

(

Meeting INT,

Progetto INT,

FOREIGN KEY (Meeting) REFERENCES Meeting(IDMeeting),

FOREIGN KEY (Progetto) REFERENCES Progetto(CodProgetto)

);

CREATE TABLE MeetingSalaLink

(

Meeting INT,

Sala VARCHAR(5),

FOREIGN KEY (Meeting) REFERENCES Meeting(IDMeeting),

FOREIGN KEY (Sala) REFERENCES SalaRiunione(CodSala)

);

INSERIMENTO DEI VINCOLI NELLE TABELLE

VINCOLI SULLA TABELLA DIPENDENTE:

ALTER TABLE dipendente

ADD CONSTRAINT EmailLegit

CHECK (email ~\* '^[A-Za-z0-9.\_%-]+[@][A-Za-z0-9.-]+[.][A-Za-z]+$');

ALTER TABLE dipendente

ADD CONSTRAINT SalarioPositivo

CHECK (salario >= 0);

ALTER TABLE dipendente

ADD CONSTRAINT ValutazioneLimitata

CHECK (valutazione BETWEEN 0 AND 10 );

VINCOLI SULLA TABELLA PROGETTO:

ALTER TABLE progetto

ADD CONSTRAINT ScadenzaEsistente

CHECK (scadenza >= datacreazione);

ALTER TABLE progetto

ADD CONSTRAINT DataTerminazioneEsistente

CHECK (dataterminazione >= datacreazione);

VINCOLI SULLA TABELLA MEETING:

ALTER TABLE meeting

ADD CONSTRAINT OrarioFineMeetingEsistente

CHECK (orariofine >= orarioinizio);

ALTER TABLE meeting

ADD CONSTRAINT DataFineMeetingEsistente

CHECK (datafine >= datainizio);

VINCOLI SULLA TABELLA SALARIUNIONE:

ALTER TABLE salariunione

ADD CONSTRAINT PianoEsistente

CHECK (piano >=0);

ALTER TABLE salariunione

ADD CONSTRAINT CapienzaEsistente

CHECK (capienza >=0);