



Agenzia Spaziale Italiana

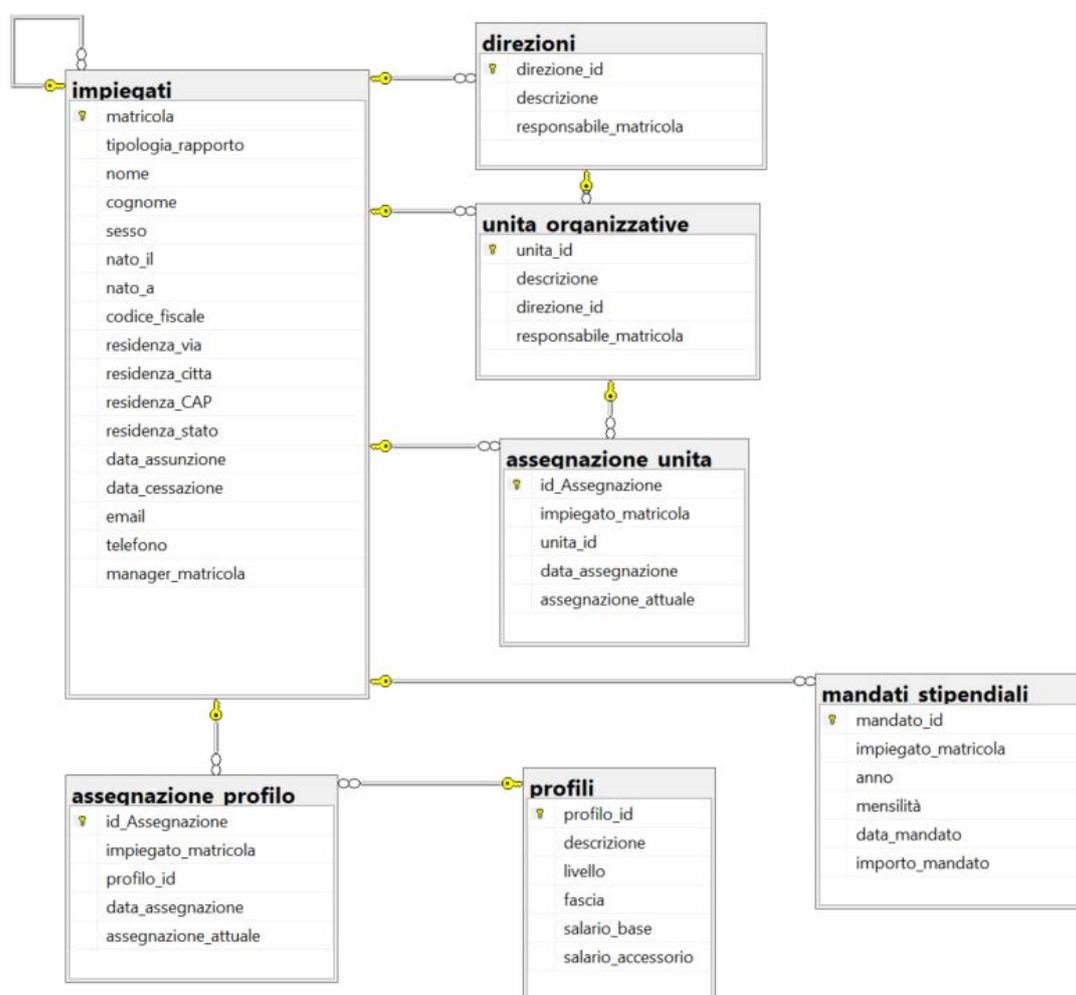
TRACCE PROVA SCRITTA

Bando n. 17/2021 – Selezione pubblica, per titoli ed esami, per la copertura di n. 2 posti a tempo pieno e indeterminato nell’Agenzia Spaziale Italiana, nel profilo di C.T.E.R., VI livello professionale VI - Area Processi e Digitalizzazione - **PROFILO: CTER 1**

Il primo titolo - in grassetto e sottolineato - indica la traccia sorteggiata

TRACCIA 1

Il candidato prenda visione del seguente grafico relazionale di entità provenienti da un sistema elaborativo per la gestione di Risorse Umane e da un sistema di gestione contabile (per la sola entità “mandati_stipendiali”):



Il candidato analizzi le istruzioni TSQL che definiscono le entità sopra schematizzate:

```
CREATE TABLE direzioni (  
  direzione_id INT IDENTITY (1,1) PRIMARY KEY,  
  descrizione VARCHAR (100) NOT NULL,
```

```

    responsabile_matricola INT NULL
);
CREATE TABLE unita_organizzative (
    unita_id INT IDENTITY (1,1) PRIMARY KEY,
    descrizione VARCHAR (100) NOT NULL,
    direzione_id INT NOT NULL,
    responsabile_matricola INT NULL,
    CONSTRAINT FK_Unit_dir FOREIGN KEY (direzione_id) REFERENCES direzioni (direzione_id)
);
CREATE TABLE profili (
    profilo_id INT IDENTITY (1,1) PRIMARY KEY,
    descrizione VARCHAR (100) NOT NULL,
    livello VARCHAR (10) NULL,
    fascia VARCHAR (10) NULL,
    salario_base DECIMAL (10, 2) NULL,
    salario_accessorio DECIMAL (10, 2) NULL
);
CREATE TABLE impiegati (
    matricola INT CONSTRAINT PK_Imp PRIMARY KEY,
    tipologia_rapporto VARCHAR (50) NOT NULL,
    nome VARCHAR (100) NOT NULL,
    cognome VARCHAR (100) NOT NULL,
    sesso VARCHAR (1) NOT NULL,
    nato_il DATE NOT NULL,
    nato_a VARCHAR (100) NOT NULL,
    codice_fiscale VARCHAR (16) NOT NULL,
    residenza_via VARCHAR (100) NULL,
    residenza_citta VARCHAR (100) NULL,
    residenza_CAP VARCHAR (10) NULL,
    residenza_stato VARCHAR (100) NULL,
    data_assunzione DATE NOT NULL,
    data_cessazione DATE NULL, --[tale attributo è compilato all'atto della cessazione dal servizio]
    email VARCHAR (100) NULL,
    telefono VARCHAR (20) NULL,
    manager_matricola INT NULL,
    CONSTRAINT FK_Manager FOREIGN KEY (manager_matricola) REFERENCES impiegati (matricola),
    CONSTRAINT CK_sesso CHECK(sesso = 'F' or sesso = 'M')
);
CREATE TABLE assegnazione_profilo (
    id_Assegnazione int identity(1,1) PRIMARY KEY,
    impiegato_matricola INT NOT NULL,
    profilo_id INT NOT NULL,
    data_assegnazione date default (cast(getdate() as date)) NOT NULL,
    assegnazione_attuale BIT default 'true', --[valore 'false' per assegnazioni pregresse]
    CONSTRAINT FK_ap_profilo FOREIGN KEY (profilo_id) REFERENCES profili (profilo_id),
    CONSTRAINT FK_ap_impiegato FOREIGN KEY (impiegato_matricola) REFERENCES impiegati (matricola) ON DELETE
CASCADE ON UPDATE CASCADE
);
CREATE TABLE assegnazione_unita (
    id_Assegnazione int identity(1,1) PRIMARY KEY,
    impiegato_matricola INT NOT NULL,
    unita_id INT NOT NULL,
    data_assegnazione date default (cast(getdate() as date)) NOT NULL,
    assegnazione_attuale BIT default 'true', --[valore 'false' per assegnazioni pregresse]
    CONSTRAINT FK_au_unita FOREIGN KEY (unita_id) REFERENCES unita_organizzative (unita_id),
    CONSTRAINT FK_au_impiegato FOREIGN KEY (impiegato_matricola) REFERENCES impiegati (matricola) ON DELETE
CASCADE ON UPDATE CASCADE
);
CREATE TABLE mandati_stipendiali (
    mandato_id INT IDENTITY (1,1),
    impiegato_matricola INT NOT NULL,
    anno int NOT NULL,
    mensilità int NOT NULL,
    data_mandato date NOT NULL,
    importo_mandato DECIMAL (8, 2) NOT NULL,

```

```
CONSTRAINT PK_mandati PRIMARY KEY (mandato_id),
CONSTRAINT FK_mandati FOREIGN KEY (impiegato_matricola) REFERENCES impiegati (matricola) ON DELETE
CASCADE ON UPDATE CASCADE
);
```

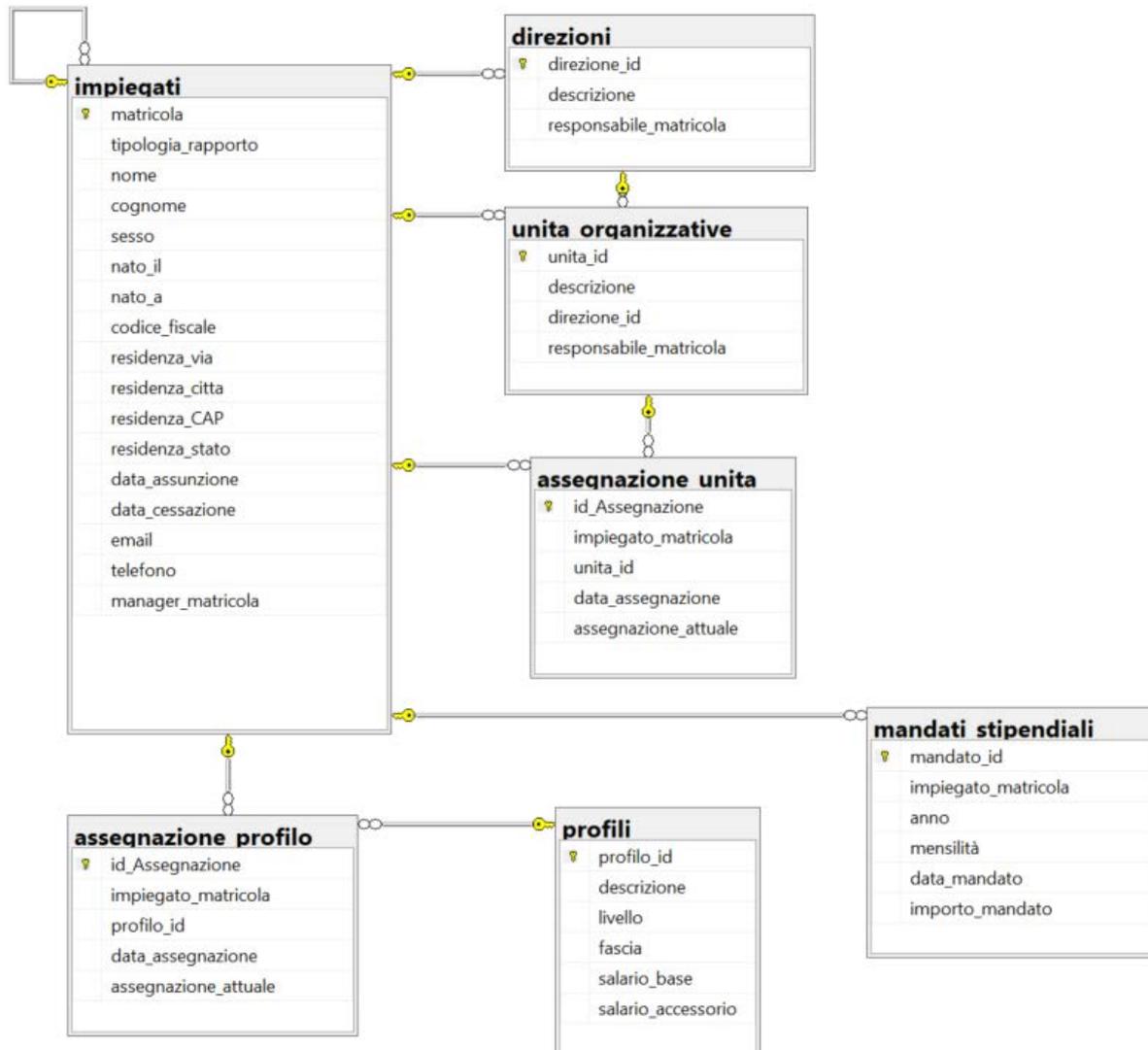
```
ALTER TABLE unita_organizzative
ADD CONSTRAINT FK_Unita_Resp FOREIGN KEY (responsabile_matricola)
REFERENCES impiegati (matricola) ON DELETE SET NULL ON UPDATE CASCADE;
```

```
ALTER TABLE direzioni
ADD CONSTRAINT FK_Direzione_Resp FOREIGN KEY (responsabile_matricola)
REFERENCES impiegati (matricola) ON DELETE SET NULL ON UPDATE CASCADE;
```

- 1.a** Sulla base del modello assegnato, il Candidato, salvaguardando l'integrità referenziale, scriva le istruzioni SQL per sostituire il manager degli **impiegati** (**manager_matricola**) identificato con matricola 114 con manager identificato da matricola n.120 solo se tale manager esiste;
- 1.b** La Direzione Risorse Umane ha la necessità di avere a disposizione un cruscotto (dashboard) interattivo che possa mostrare l'andamento delle progressioni di carriera di ogni dipendente nel corso del tempo. Il candidato descriva le tabelle, gli attributi ed eventuali accorgimenti tecnici definiti nello schema proposto, che ritiene idonei ad un'analisi che possa rispondere alle esigenze descritte. Il candidato inoltre descriva quali strumenti e/o servizi e/o tecnologie ritiene utili impiegare per rappresentare in maniera dinamica le informazioni.

TRACCIA 2

Il candidato prenda visione del seguente grafico relazionale di entità provenienti da un sistema elaborativo per la gestione di Risorse Umane e da un sistema di gestione contabile (per la sola entità "mandati stipendiali"):



Il candidato analizzi le istruzioni TSQL che definiscono le entità sopra schematizzate:

```
CREATE TABLE direzioni (  
  direzione_id INT IDENTITY (1,1) PRIMARY KEY,  
  descrizione VARCHAR (100) NOT NULL,  
  responsabile_matricola INT NULL  
);
```

```
CREATE TABLE unita_organizzative (  
  unita_id INT IDENTITY (1,1) PRIMARY KEY,  
  descrizione VARCHAR (100) NOT NULL,  
  direzione_id INT NOT NULL,  
  responsabile_matricola INT NULL,  
  CONSTRAINT FK_Unit_dir FOREIGN KEY (direzione_id) REFERENCES direzioni (direzione_id)  
);
```

```
CREATE TABLE profili (  
  profilo_id INT IDENTITY (1,1) PRIMARY KEY,  
  descrizione VARCHAR (100) NOT NULL,  
  livello VARCHAR (10) NULL,  
  fascia VARCHAR (10) NULL,  
  salario_base  
  salario_accessorio
```

```

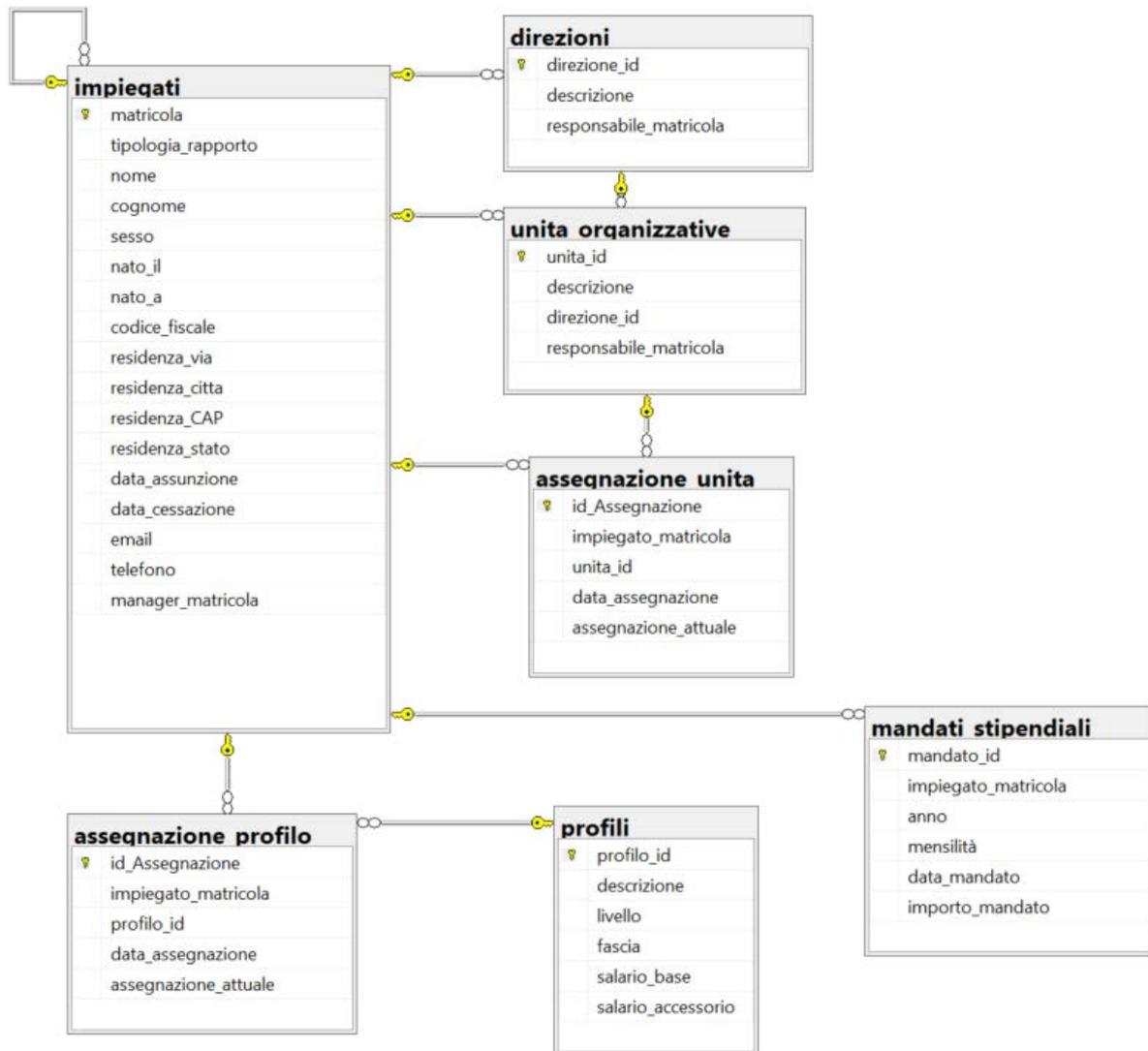
    salario_base DECIMAL (10, 2) NULL,
    salario_accessorio DECIMAL (10, 2) NULL
);
CREATE TABLE impiegati (
    matricola INT CONSTRAINT PK_Imp PRIMARY KEY,
    tipologia_rapporto VARCHAR (50) NOT NULL,
    nome VARCHAR (100) NOT NULL,
    cognome VARCHAR (100) NOT NULL,
    sesso VARCHAR (1) NOT NULL,
    nato_il DATE NOT NULL,
    nato_a VARCHAR (100) NOT NULL,
    codice_fiscale VARCHAR (16) NOT NULL,
    residenza_via VARCHAR (100) NULL,
    residenza_citta VARCHAR (100) NULL,
    residenza_CAP VARCHAR (10) NULL,
    residenza_stato VARCHAR (100) NULL,
    data_assunzione DATE NOT NULL,
    data_cessazione DATE NULL, --[tale attributo è compilato all'atto della cessazione dal servizio]
    email VARCHAR (100) NULL,
    telefono VARCHAR (20) NULL,
    manager_matricola INT NULL,
    CONSTRAINT FK_Manager FOREIGN KEY (manager_matricola) REFERENCES impiegati (matricola),
    CONSTRAINT CK_sesso CHECK(sesso = 'F' or sesso = 'M')
);
CREATE TABLE assegnazione_profilo (
    id_Assegnazione int identity(1,1) PRIMARY KEY,
    impiegato_matricola INT NOT NULL,
    profilo_id INT NOT NULL,
    data_assegnazione date default (cast(getdate() as date)) NOT NULL,
    assegnazione_attuale BIT default 'true', --[valore 'false' per assegnazioni pregresse]
    CONSTRAINT FK_ap_profilo FOREIGN KEY (profilo_id) REFERENCES profili (profilo_id),
    CONSTRAINT FK_ap_impiegato FOREIGN KEY (impiegato_matricola) REFERENCES impiegati (matricola) ON DELETE
CASCADE ON UPDATE CASCADE
);
CREATE TABLE assegnazione_unita (
    id_Assegnazione int identity(1,1) PRIMARY KEY,
    impiegato_matricola INT NOT NULL,
    unita_id INT NOT NULL,
    data_assegnazione date default (cast(getdate() as date)) NOT NULL,
    assegnazione_attuale BIT default 'true', --[valore 'false' per assegnazioni pregresse]
    CONSTRAINT FK_au_unita FOREIGN KEY (unita_id) REFERENCES unita_organizzative (unita_id),
    CONSTRAINT FK_au_impiegato FOREIGN KEY (impiegato_matricola) REFERENCES impiegati (matricola) ON DELETE
CASCADE ON UPDATE CASCADE
);
CREATE TABLE mandati_stipendiali (
    mandato_id INT IDENTITY (1,1),
    impiegato_matricola INT NOT NULL,
    anno int NOT NULL,
    mensilità int NOT NULL,
    data_mandato date NOT NULL,
    importo_mandato DECIMAL (8, 2) NOT NULL,
    CONSTRAINT PK_mandati PRIMARY KEY (mandato_id),
    CONSTRAINT FK_mandati FOREIGN KEY (impiegato_matricola) REFERENCES impiegati (matricola) ON DELETE
CASCADE ON UPDATE CASCADE
);
ALTER TABLE unita_organizzative
ADD CONSTRAINT FK_Unita_Resp FOREIGN KEY (responsabile_matricola)
REFERENCES impiegati (matricola) ON DELETE SET NULL ON UPDATE CASCADE;
ALTER TABLE direzioni
ADD CONSTRAINT FK_Direzione_Resp FOREIGN KEY (responsabile_matricola)
REFERENCES impiegati (matricola) ON DELETE SET NULL ON UPDATE CASCADE;

```

- 2.a Sulla base del modello assegnato, il Candidato, salvaguardando l'integrità referenziale, scriva le istruzioni SQL per assegnare nella tabella "assegnazione_unita" l'unità id n.28 all'impiegato con matricola numero 120, solo se tale impiegato esiste, rendendo tale assegnazione attuale (assegnazione_attuale = 'true')
- 2.b La Direzione Risorse Umane ha la necessità di avere a disposizione un cruscotto (dashboard) interattivo che possa mostrare l'andamento dei mandati stipendiali di ogni dipendente nel corso del tempo. Il Candidato descriva le tabelle, gli attributi ed eventuali accorgimenti tecnici definiti nello schema proposto, che ritiene idonei ad un'analisi che possa rispondere alle esigenze descritte. Il candidato inoltre descriva quali strumenti e/o servizi e/o tecnologie ritiene utili impiegare per rappresentare in maniera dinamica le informazioni.

TRACCIA 3

Il candidato prenda visione del seguente grafico relazionale di entità provenienti da un sistema elaborativo per la gestione di Risorse Umane e da un sistema di gestione contabile (per la sola entità “mandati_stipendiali”):



Il candidato analizzi le istruzioni TSQL che definiscono le entità sopra schematizzate:

```
CREATE TABLE direzioni (  
    direzione_id INT IDENTITY (1,1) PRIMARY KEY,  
    descrizione VARCHAR (100) NOT NULL,  
    responsabile_matricola INT NULL  
);  
CREATE TABLE unita_organizzative (  
    unita_id INT IDENTITY (1,1) PRIMARY KEY,  
    descrizione VARCHAR (100) NOT NULL,  
    direzione_id INT NOT NULL,  
    responsabile_matricola INT NULL,  
    CONSTRAINT FK_Unit_dir FOREIGN KEY (direzione_id) REFERENCES direzioni (direzione_id)  
);  
CREATE TABLE profili (  
    profilo_id INT IDENTITY (1,1) PRIMARY KEY,  
    descrizione VARCHAR (100) NOT NULL,  
    livello VARCHAR (10) NULL,
```

```

fascia VARCHAR (10) NULL,
salario_base DECIMAL (10, 2) NULL,
salario_accessorio DECIMAL (10, 2) NULL
);
CREATE TABLE impiegati (
matricola INT CONSTRAINT PK_Imp PRIMARY KEY,
tipologia_rapporto VARCHAR (50) NOT NULL,
nome VARCHAR (100) NOT NULL,
cognome VARCHAR (100) NOT NULL,
sesso VARCHAR (1) NOT NULL,
nato_il DATE NOT NULL,
nato_a VARCHAR (100) NOT NULL,
codice_fiscale VARCHAR (16) NOT NULL,
residenza_via VARCHAR (100) NULL,
residenza_citta VARCHAR (100) NULL,
residenza_CAP VARCHAR (10) NULL,
residenza_stato VARCHAR (100) NULL,
data_assunzione DATE NOT NULL,
data_cessazione DATE NULL,--[tale attributo è compilato all'atto della cessazione dal servizio]
email VARCHAR (100) NULL,
telefono VARCHAR (20) NULL,
manager_matricola INT NULL,
CONSTRAINT FK_Manager FOREIGN KEY (manager_matricola) REFERENCES impiegati (matricola),
CONSTRAINT CK_sesso CHECK(sesso = 'F' or sesso = 'M')
);
CREATE TABLE assegnazione_profilo (
id_Assegnazione int identity(1,1) PRIMARY KEY,
impiegato_matricola INT NOT NULL,
profilo_id INT NOT NULL,
data_assegnazione date default (cast(getdate() as date)) NOT NULL,
assegnazione_attuale BIT default 'true',--[valore 'false' per assegnazioni pregresse]
CONSTRAINT FK_ap_profilo FOREIGN KEY (profilo_id) REFERENCES profili (profilo_id),
CONSTRAINT FK_ap_impiegato FOREIGN KEY (impiegato_matricola) REFERENCES impiegati (matricola) ON DELETE
CASCADE ON UPDATE CASCADE
);
CREATE TABLE assegnazione_unita (
id_Assegnazione int identity(1,1) PRIMARY KEY,
impiegato_matricola INT NOT NULL,
unita_id INT NOT NULL,
data_assegnazione date default (cast(getdate() as date)) NOT NULL,
assegnazione_attuale BIT default 'true',--[valore 'false' per assegnazioni pregresse]
CONSTRAINT FK_au_unita FOREIGN KEY (unita_id) REFERENCES unita_organizzative (unita_id),
CONSTRAINT FK_au_impiegato FOREIGN KEY (impiegato_matricola) REFERENCES impiegati (matricola) ON DELETE
CASCADE ON UPDATE CASCADE
);
CREATE TABLE mandati_stipendiali (
mandato_id INT IDENTITY (1,1),
impiegato_matricola INT NOT NULL,
anno int NOT NULL,
mensilità int NOT NULL,
data_mandato date NOT NULL,
importo_mandato DECIMAL (8, 2) NOT NULL,
CONSTRAINT PK_mandati PRIMARY KEY (mandato_id),
CONSTRAINT FK_mandati FOREIGN KEY (impiegato_matricola) REFERENCES impiegati (matricola) ON DELETE
CASCADE ON UPDATE CASCADE
);
ALTER TABLE unita_organizzative
ADD CONSTRAINT FK_Unita_Resp FOREIGN KEY (responsabile_matricola)
REFERENCES impiegati (matricola) ON DELETE SET NULL ON UPDATE CASCADE;
ALTER TABLE direzioni
ADD CONSTRAINT FK_Direzione_Resp FOREIGN KEY (responsabile_matricola)
REFERENCES impiegati (matricola) ON DELETE SET NULL ON UPDATE CASCADE;

```

- 3.a** Sulla base del modello assegnato, il Candidato, salvaguardando l'integrità referenziale, scriva le istruzioni SQL per assegnare nella tabella "*assegnazione_profili*" il **profilo_id** n.6 all'impiegato con matricola numero 126, solo se tale impiegato esiste, rendendo tale assegnazione attuale (**assegnazione_attuale = 'true'**)
- 3.b** La Direzione Risorse Umane ha la necessità di avere a disposizione un cruscotto (dashboard) interattivo che possa mostrare l'assegnazione alle unità organizzative di ogni dipendente nel corso del tempo. Il Candidato descriva le tabelle, gli attributi ed eventuali accorgimenti tecnici definiti nello schema proposto, che ritiene idonei ad un'analisi che possa rispondere alle esigenze descritte. Il candidato inoltre descriva quali strumenti e/o servizi e/o tecnologie ritiene utili impiegare per rappresentare in maniera dinamica le informazioni.