

ISTITUTO PENITENZIARIO

BRAGAGGIA PIERA – Matricola 1102048

ZAMBON TOMMASO – Matricola 1103145

Account laboratorio utilizzato per la consegna: pbragagg

ABSTRACT

La Casa di Reclusione di Padova è una delle strutture penitenziarie più grandi del triveneto. È stata costruita negli anni '80 ed è entrata in funzione nel settembre del 1991. Denominata il "Nuovo Complesso", ora è costituita da 370 camere detentive ove sono allocati un numero oscillante nel tempo tra 600 e 900 detenuti. Il personale è costituito da 320 membri della polizia penitenziaria e 25 impiegati tra amministrativi ed educatori.

Nella struttura sono presenti varie strutture per il lavoro e il tempo libero dei detenuti. L'istituto offre ai detenuti una serie di attività di lavoro (manutenzione ordinaria, preparazione pasti, lavorazioni domestiche, pasticceria, gelateria, ecc...) culturali (teatro, biblioteca, coro, ecc...), sportive e scolastiche (dall'alfabetizzazione all'educazione universitaria).

Lo scopo del progetto è di modellare un sistema di gestione e organizzazione di un istituto penitenziario ispirato a questa struttura, concentrandosi sulle figure dei detenuti, del personale amministrativo e della polizia penitenziaria.

Per i primi sono rilevanti gli aspetti relativi al motivo della loro presenza nell'istituto, come il processo che ha portato alla detenzione, e della permanenza nella struttura, ad esempio l'organizzazione delle attività di lavoro e di studio, culturali o sportive.

Per l'organico sono invece rilevanti le aree di competenza, definendo le responsabilità all'interno della struttura, che siano per supervisione di reparti o per gestione di attività.

1. ANALISI DEI REQUISITI

La figura centrale dell'istituto penitenziario è ovviamente il detenuto, di questo ci interessano:

- La matricola assegnata
- I dati anagrafici (nome, cognome, data di nascita)
- Le attività a cui partecipa all'interno dell'istituto.
- Il lavoro che svolge, in quanto viene richiesto ai detenuti di svolgere obbligatoriamente un'attività di lavoro
- La stanza detentiva dove risiede
- Il conto corrente personale (aperto e gestito dall'Ufficio Conti Correnti dell'istituto, non modellato in questo progetto) dove vengono registrati i guadagni per le prestazioni di lavoro
- Il processo che lo ha portato alla reclusione

Il processo ha come caratteristiche

- La data
- Il reato commesso
- L'avvocato difensore
- La città del tribunale di competenza
- La durata della pena
- La data di inizio dell'incarcerazione

L'avvocato viene modellato con:

- Un codice identificativo assegnato dal sistema
- Nome
- Cognome
- Lo studio dove pratica

Per lo studio legale ci interessa

- Codice identificativo assegnato dal sistema
- Nome
- Città, dove si trova lo studio
- Recapito, numero telefonico dello studio
- Indirizzo, via della sede legale dello studio

L'istituto è diviso in reparti che vengono identificati da:

- Un codice identificativo
- Nome
- Il tipo: possono essere zone comuni (palestra, scuola, teatro, ecc) o blocchi di stanze detentive
- Il codice del responsabile del reparto

Ogni stanza ha:

- Un numero identificativo
- Un numero variabile di posti letto
- Il codice del reparto a cui appartiene

Il personale dell'istituto si divide in polizia penitenziaria, impiegati, educatori che svolgono funzioni amministrative, educative oppure supervisionano attività di lavoro o culturali, e per ognuno di loro è necessario conoscere:

- Codice identificativo univoco
- Dati anagrafici (nome, cognome, data di nascita)
- Mansione all'interno dell'istituto
- Stipendio annuo (calcolato in base alla mansione)

Per i turni di lavoro della polizia penitenziaria ci interessa:

- La data in cui ogni guardia svolge un turno
- Gli orari di inizio e fine
- Il reparto che la guardia supervisiona

Il personale amministrativo e gli educatori non ha una lista con i turni in quanto hanno un orario fisso (lun-ven 8-12 13 - 17).

I detenuti hanno l'obbligo di effettuare un lavoro retribuito durante la loro permanenza nell'istituto, per ogni lavoro viene specificato:

- Il nome dell'attività svolta
- La retribuzione oraria
- Se viene fatto all'interno dell'istituto (es lavori di pulizia) o alle dipendenze di terzi, gestite da imprese pubbliche o private (in particolare cooperative sociali)
- Per ogni lavoro si ha come referente un membro del personale amministrativo
- Il numero massimo di detenuti che possono svolgere il lavoro

I turni di lavoro vengono registrati in uno storico che specifica:

- La data
- L'ora di inizio e fine

Le attività all'interno sono invece gestite da un educatore e sono definite da:

- Nome identificativo
- Un massimo di partecipanti
- Reparto in cui viene svolta
- Descrizione
- Da chi viene supervisionata (educatore)

2. PROGETTAZIONE CONCETTUALE

2.1 LISTA DELLE CLASSI

2.1.1 Detenuto: entità che modella le informazioni anagrafiche dei detenuti all'interno dell'istituto detentivo:

- Mat: *int* "PK" - Matricola identificativa del detenuto
- Nome: *string*
- Cognome: *string*
- DataN: *date* - Data di nascita del detenuto
- Naz: *string* - Indica la nazionalità del detenuto

2.1.2 Processo: entità

- Data: *date* - Data del processo
- Reato: *string* - Reato commesso dall'imputato
- Durata: *int* - giorni di reclusione
- DataI: *date* - data di inizio della reclusione

2.1.3 Studio: entità che modella uno studio legale dove esercita un avvocato

- CodS: *int* "PK" - identifica univocamente lo studio legale
- Nome: *string* - nome dello studio legale
- Città: *string* - città dove si colloca lo studio
- Indirizzo: *string* - via e numero dello studio
- Recapito: *string* - recapito telefonico

2.1.4 Avvocato: entità che identifica un avvocato difensore per un processo

- CodA: *int* "PK" - identifica univocamente l'avvocato
- Nome: *string*
- Cognome: *string*

2.1.5 Reparto: è una superclasse per Blocco e Zona Comune

- CodR: *int* "PK" - codice che identifica univocamente il reparto
- Nome: *string* - nome del reparto (luogo all'interno dell'istituto)

Si divide in due sottoclassi:

Blocco: blocco di stanze detentive

Zona Comune: zona comune all'interno dell'istituto dove si effettuano attività (palestra, mensa, ecc)

2.1.6 Stanza: luogo dove risiede il detenuto

- Numero: *int* "PK" - identifica la stanza detentiva
- Posti: *int* - numero di posti letto

2.1.7 Impiegato: è superclasse che modella i membri del personale dell'istituto

- CF: *string* "PK" - identifica univocamente l'impiegato
- Nome: *string*
- Cognome: *string*
- DataN: *date* - data di nascita
- Stipendio: *long int* - compenso annuo dell'impiegato

Si divide in tre sottoclassi:

Guardia: membro della polizia penitenziaria all'interno dell'istituto

- Specializzazione: *string* - unità in cui si è formato il membro della polizia
- Data di reclutamento: *date* - data in cui la guardia è diventata un membro della polizia penitenziaria

Amministrativo: impiegato amministrativo dell'istituto

- Ruolo: *string* - ruolo ricoperto in ufficio amministrativo. (segretario, ecc.)

Educatore:

- Laurea: *string* - tipo di formazione professionale conseguita

2.1.8 Attività: superclasse che modella un attività svolta all'interno dell'istituto

- Codice: *int* "PK" - codice dell'attività
- Max: *int* - numero massimo di partecipanti
- Descrizione: *string*

2.1.9 Lavoro: tipo di lavoro svolto dai detenuti

- Codice: *string* "PK" - identificativo per l'attività di lavoro
- Max: *int* - numero massimo di partecipanti
- Tipo: *enum*{"interno","esterno"} - identifica il tipo di lavoro, se interno o esterno
- Descrizione: *string* - descrive il tipo di attività
- Retribuzione: *int* - retribuzione oraria

2.1.10 Turno di lavoro: registra giorno per giorno i turni di lavoro dei detenuti

- Data: *date* "PK" - data in cui viene svolto il turno
- Ora Inizio: *time* "PK" - ora inizio del turno
- Ora Fine: *time* - ora fine del turno

2.1.11 Azienda appaltata: impresa esterna all'istituto che gestisce un lavoro

- P.IVA: *string* "PK" - identifica l'azienda appaltata
- Nome: *string* - nome dell'azienda
- Recapito: *string* - recapito per l'azienda
- Indirizzo: *string* - indirizzo dell'azienda

2.1.12 Conto corrente: conto corrente personale del detenuto per le spese interne all'istituto penitenziario

- CC: *int* "PK" - codice che identifica il conto corrente di ogni detenuto
- Saldo: *long int* - saldo del conto corrente personale

2.2 LISTA DELLE RELAZIONI

2.2.1 Stanza-Reparto: "appartiene a"

- *Molteplicità 1:N* Ogni stanza appartiene ad un solo reparto ed un reparto (se blocco) ha molte stanze
- Totalità: totale verso Stanza / parziale verso Reparto. Un reparto se non è blocco non ha stanze, una stanza detentiva appartiene sempre ad un reparto (blocco)

2.2.2 Detenuto-Stanza: "risiede in"

- *Molteplicità 1:N* Ogni detenuto risiede in una sola stanza, una stanza può contenere più di un detenuto
- Totalità: totale verso Detenuto / parziale verso Stanza. Una stanza può non essere ancora assegnata ad alcun detenuto. Un detenuto, appena viene registrato all'interno del sistema, deve avere una stanza dove risiedere

2.2.3 Guardia-Reparto: "ha turno assegnato in"

- *Molteplicità M:N* Ogni guardia può avere più turni in diversi giorni in diversi reparti, un reparto può avere più persone in turno nello stesso giorno
- Totalità: parziale verso Guardia / parziale verso Reparto. Può esserci una guardia a cui non è stato ancora assegnato un turno in un reparto o un reparto al quale non è stato assegnato nessuno in un turno.
- Attributi:
 - Data: date - data del turno
 - Ora inizio: time - Ora di inizio del turno
 - Ora fine: time - Ora di fine del turno

2.2.4 Guardia-Reparto: "è responsabile in"

- *Molteplicità: 1:1* - Ogni reparto deve avere un responsabile, e una guardia può essere responsabile al massimo di un reparto
- Totalità: totale verso Reparto / parziale verso Guardia. Tutti i reparti hanno un responsabile e non tutte le guardie sono responsabili di un reparto.

2.2.5 Attività-Reparto: "svolta in"

- *Molteplicità: 1:N* - Ogni attività viene svolta in un solo reparto e in un reparto possono essere svolte più attività
- Totalità: parziale verso Attività / parziale verso Reparto - Può esserci un'attività alla quale non è ancora stato assegnato un reparto dove può essere svolta. Un reparto (es blocco) può non avere attività svolte in esso.
- Attributi:
 - Ora inizio: *time* - Ora di inizio attività
 - Ora fine: *time* - Ora di fine attività
 - Giorno della settimana: *string* - giorno in cui viene svolta

2.2.6 Detenuto-Attività: "iscritto a"

- *Molteplicità: N:M* - Ogni detenuto può svolgere più attività ed un'attività viene svolta da più detenuti
- Totalità: parziale verso Detenuti / parziale verso Attività - Un detenuto può non svolgere alcuna attività all'interno dell'istituto ed un'attività può non avere detenuti iscritti ad essa

2.2.7 Educatore-Attività: “è responsabile”

- Molteplicità: 1:1 - Ogni educatore è responsabile di una sola attività, e ogni attività ha un solo educatore responsabile
- Totalità: totale verso Educatore / parziale verso Attività - Un educatore esterno deve avere un'attività di cui è responsabile (altrimenti non sarebbe impiegato), invece un'attività può essere proposta e quindi non avere ancora un responsabile.

2.2.8 Amministrativo-Lavoro: “supervisiona”

- Molteplicità: N:1 - Un amministrativo può supervisionare più lavori, ogni lavoro viene supervisionato da un solo impiegato amministrativo
- Totalità: parziale verso Amministrativo / totale verso Lavoro - un amministrativo può non avere incarichi di supervisione di lavoro, ogni lavoro deve essere supervisionato da un impiegato amministrativo

2.2.9 Lavoro-Detenuto: “impieghi precedenti”

- Molteplicità: N:M - Ad ogni detenuto viene assegnato almeno un lavoro, un lavoro può essere assegnato a più di un detenuto
- Totalità: totale verso Detenuto / parziale verso Lavoro - ad ogni detenuto deve venire assegnato un lavoro (che può cambiare durante la permanenza nell'istituto), ad un lavoro può non essere ancora stato assegnato alcun detenuto
- Attributi:
 - Data di inizio: date - Data di inizio impiego del detenuto in quel lavoro
 - Data di fine: date - Data di fine impiego del detenuto in quel lavoro

2.2.10 Detenuto-Lavoro: “impiego attuale”

- Molteplicità: 1:N - Ogni detenuto ha un solo lavoro attuale, un lavoro può essere impiego attuale di più detenuti
- Totalità: totale verso Detenuto / parziale verso Lavoro - ogni detenuto deve avere un impiego attuale, non tutti i lavori hanno impiegati attualmente dei detenuti
- Attributo:
 - Data di inizio: date - Data di inizio dell'impiego attuale

2.2.11 Lavoro-Turno: “svolto in”

- Molteplicità: N:1 - Ogni lavoro può essere svolto in più turni, mentre un turno è relativo ad un unico lavoro
- Totalità: parziale verso Lavoro / totale verso turno - un lavoro può non essere ancora stato eseguito, mentre il turno è sempre relativo ad un lavoro

2.2.12 Lavoro-Azienda Appaltata: “gestito da”

- Molteplicità: 1:N - Ogni lavoro può essere gestito al massimo da un'azienda. Un'azienda può gestire più lavori
- Totalità: parziale verso Lavoro / totale verso Azienda Appaltata - un lavoro può essere interno all'istituto e quindi avere una gestione interna e non di un'azienda appaltata, mentre un'azienda se registrata nel sistema deve gestire almeno un lavoro

2.2.13 Detenuto-Conto Corrente: “titolare di”

- Molteplicità: 1:1 - Ogni detenuto ha al massimo un conto corrente e un conto corrente appartiene ad un solo detenuto
- Totalità: totale verso Detenuto / totale verso Conto Corrente - Ogni detenuto è titolare di un conto corrente e ogni conto corrente è relativo ad un detenuto.

2.2.14 Processo-Detenuto: “relativo a”

- Molteplicità: 1:1 - Ogni processo è relativo ad un solo detenuto e ogni detenuto compare in un solo processo
- Totalità: totale verso Processo / totale verso Detenuto - Per ogni detenuto esiste un processo e ogni processo è relativo ad un detenuto

2.2.15 Avvocato-Processo: “difensore in”

- Molteplicità: N:1 - Ogni processo ha un solo avvocato difensore e un avvocato può fare il difensore in più di un processo
- Totalità: totale verso Avvocato / totale verso Processo - Non può esserci un processo senza avvocato, un avvocato partecipa sempre ad almeno un processo

2.2.16 Avvocato-Studio Legale: “pratica in”

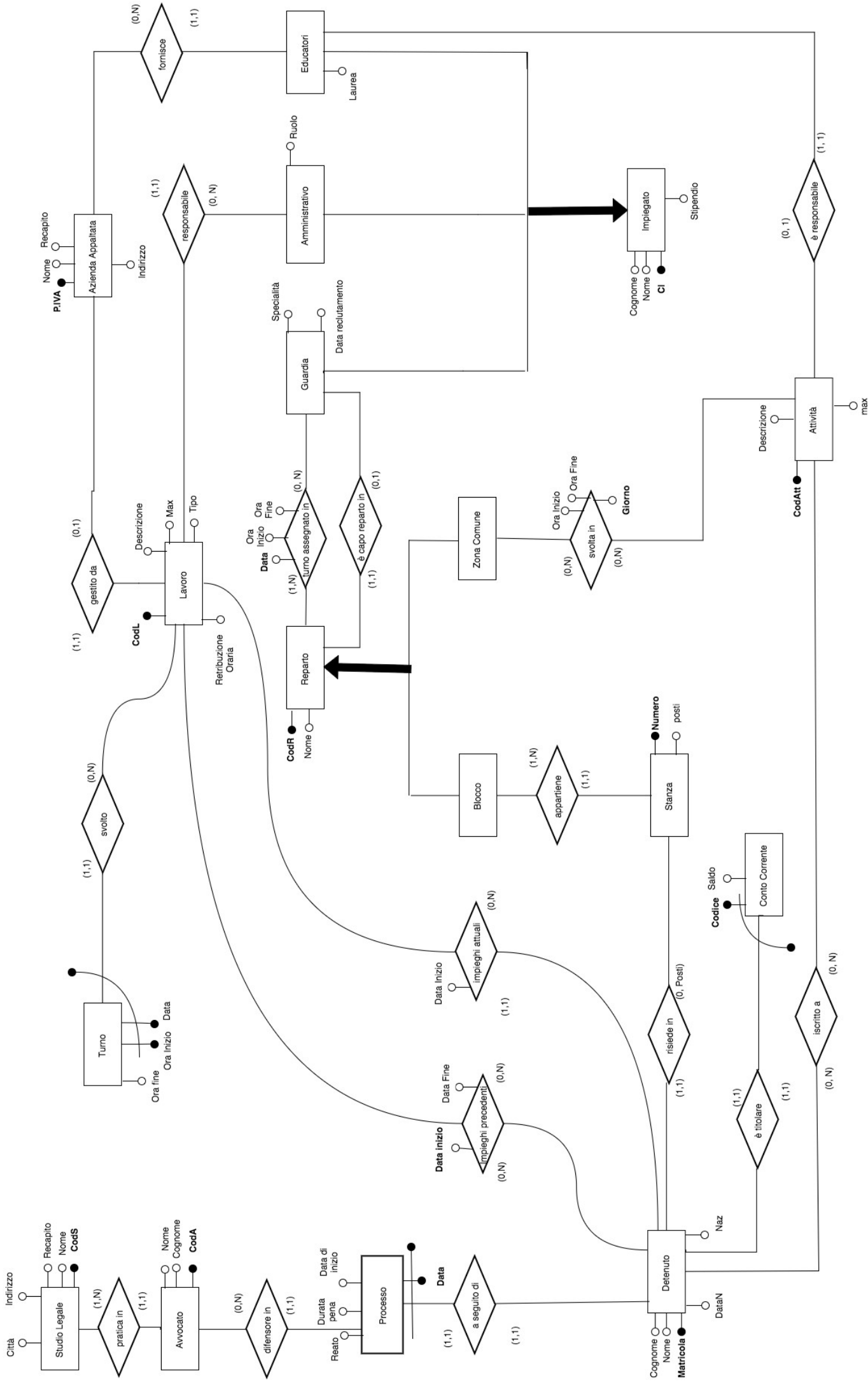
- Molteplicità: 1:N - Ogni avvocato pratica in un unico studio legale e uno studio legale ha molti avvocati
- Totalità: totale verso Avvocato / totale verso Studio Legale - Un avvocato appartiene sempre ad uno studio legale e uno studio legale ha sempre almeno un avvocato.

2.3 DESCRIZIONE DELLE GERARCHIE

Lo schema concettuale della base di dati prevede due gerarchie:

- La superclasse Impiegato che ha come sottoclassi Guardia, Amministrativo ed Educatore, ognuna delle quali ha attributi e relazioni proprie. L’ereditarietà definisce una partizione in quanto sono soddisfatte le proprietà di disgiunzione, un impiegato non può essere contemporaneamente una Guardia ed un Amministrativo (o Educatore), e di copertura, tutti gli impiegati dell’istituto appartengono ad una delle tre categorie.
- La superclasse Reparto che ha come sottoclassi Blocco e Zona Comune. Questa gerarchia soddisfa la proprietà di disgiunzione in quanto un reparto non può essere contemporaneamente un Blocco detentivo e una Zona Comune. Inoltre soddisfa anche la proprietà di copertura poiché non esiste un Reparto che non appartenga ad una delle due categorie. Le due sottoclassi non prevedono attributi rilevanti per la loro memorizzazione, ma la differenziazione è necessaria in quanto hanno due ruoli differenti (i Blocchi hanno una relazione con le stanze detentive, mentre le Zone Comuni hanno una relazione con le Attività)

Nella pagina che segue viene presentato lo schema E-R completo di generalizzazioni:



3. PROGETTAZIONE LOGICA

3.1 Descrizione testuale dello schema relazionale

Detenuto (Mat, Nome, Cognome, DataN, Naz, Numero)

Mat(PK)

Numero FK (Stanza)

ContoCorrente(CC, Mat*, Saldo)

PK(CC, Mat)

Mat FK(Detenuto)

Processo (Mat*, DataP, Reato, CodA*, Durata, DataI, Città)

PK (Mat, DataP)

Mat FK(Detenuto)

CodA FK(Avvocato)

Avvocato (CodA, Nome, Cognome, CodS*)

PK(CodA)

CodS FK(Studio)

Studio(CodS, Nome, Città, Indirizzo, Recapito)

PK(CodS)

Reparto(CodR, Nome, Tipo : enum {'blocco di stanze', 'luogo comune'}, ID_Guardia)

PK(CodR)

ID_Guardia FK(Guardia)

Stanza(Numero, CodR*, Posti)

PK(Numero)

CodR FK(Reparto)

Guardia(ID_Guardia, Nome, Cognome, DataN, Stipendio, Specialità, DataR)

PK(ID_Guardia)

RegistroTurniGuardie(ID_Guardia, Data, CodR, Oral, OraF)

PK(Data, CFG)

ID_Guardia FK(ID_Guardia)

CodR FK(Reparto)

Amministrativo(ID_Amm, Nome, Cognome, DataN, Stipendio, Ruolo)

PK(ID_Ammministrativo)

Educatore(ID_Educatore, Nome, Cognome, DataN, Stipendio, Laurea, CodAtt*, pIVA*)

PK(ID_Educatore)

CodAtt FK(Attività)

pIVA FK(AziendaAppaltata)

Attività(CodAtt, Descrizione, Max)

PK(CodAtt)

Iscritti(Mat, CodAtt)

PK(Mat, CodAtt)

Mat FK(Detenuto)

CodAtt FK(Attività)

RegistroAttività(CodAtt, CodR, GiornoSett, Oral, OraF)

PK(CodAtt, GiornoSett, Oral)

CodAtt FK(Attività)

CodR FK(Reparto)

AziendaAppaltata(pIVA, Nome, Recapito, Indirizzo)

PK(pIVA)

Lavoro(CodL, Descrizione, RetrOraria, Max, Tipo: enum{"esterno", "interno"}, pIVA*,
Responsabile)

PK(CodL)

pIVA FK(AziendaAppaltata)

Responsabile FK(Ammministrativo)

Turno(Data, Codice*, Oral, OraF)

PK(Data, Codice, Oral)
Codice FK(Lavoro)

RegistrolImpieghi(Mat*, CodL*, DataI, DataF)

PK(Mat, DataI)
Mat FK(Detenuto)
CodL FK(Lavoro)

ImpieghiAttuali(Mat*, CodL*, DatI)

PK(Mat)
Mat FK(Detenuto)
CodL FK(Lavoro)

3.2 Trasformazione della gerarchia

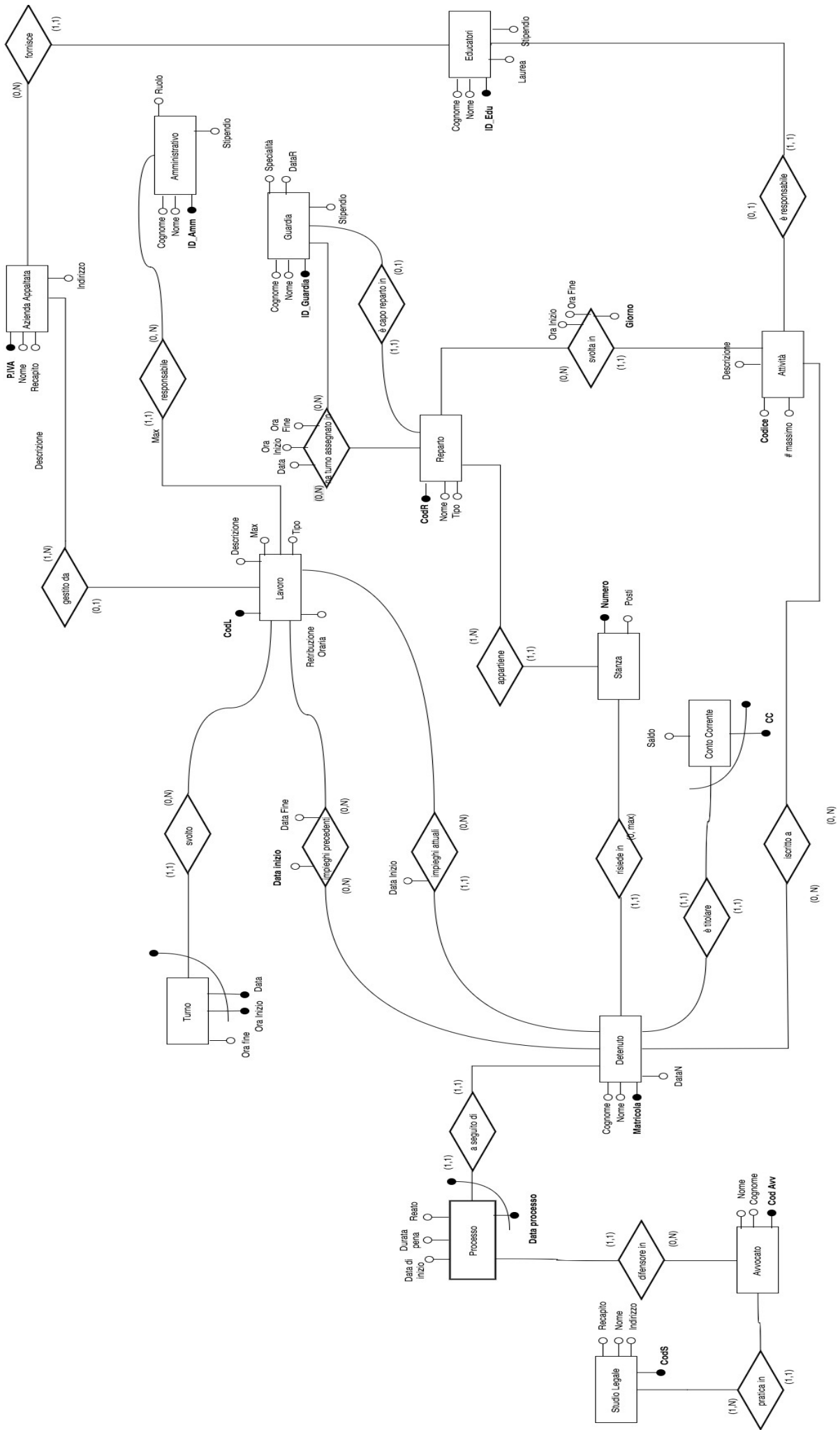
La gerarchia della superclasse *Impiegato* è stata implementata tramite partizionamento orizzontale poiché la classe radice non ha delle associazioni proprie e ogni entità figlia ha delle relazioni e degli attributi autonomi.

La gerarchia viene implementata accorpendo l'entità padre nelle seguenti entità figlie, che diventano:

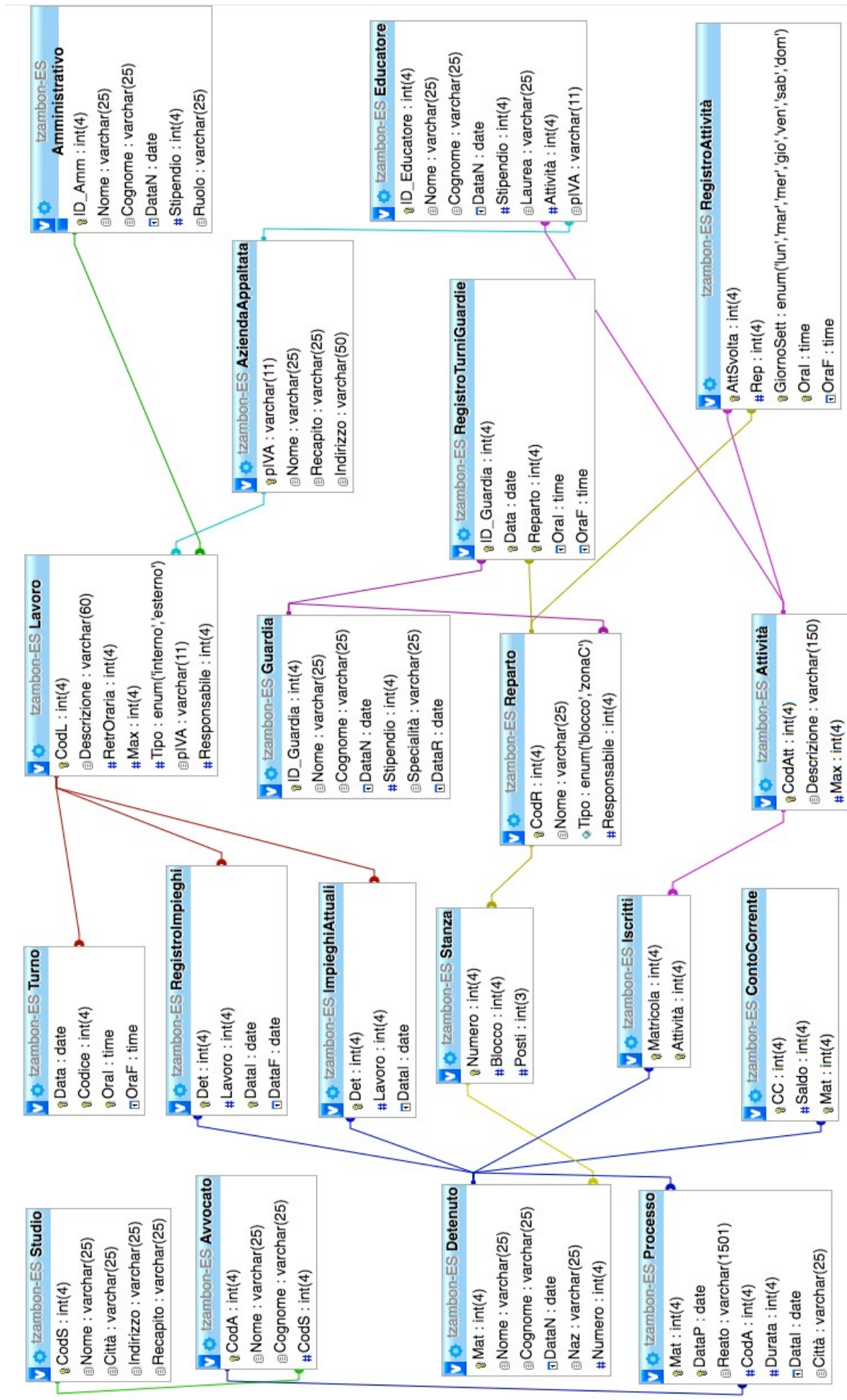
1. Guardia
 - ID_Guardia: *int(4)*;
 - Nome: *varchar(25)*;
 - Cognome: *varchar(25)*;
 - DataN: *date*;
 - Stipendio: *int(4)*;
 - Specialità: *varchar(25)*;
 - DataR: *date*;
2. Amministrativo
 - ID_Ammin: *int(4)*;
 - Nome: *varchar(25)*;
 - Cognome: *varchar(25)*;
 - DataN: *date*;
 - Stipendio: *int(4)*;
 - Ruolo: *varchar(25)*;
3. Educatore
 - ID_Educatore: *int(4)*;
 - Nome: *varchar(25)*;
 - Cognome: *varchar(25)*;
 - Laurea: *varchar(25)*;
 - Stipendio: *int(4)*;
 - DataN: *date*;

La gerarchia della superclasse *Reparto* è stata implementata tramite accorpamento delle classi figlie nella classe genitore attraverso l'uso di un attributo *Tipo* di tipo `enum{'blocco', 'zonaC'}` che distingue le due tipologie di *Reparto*.

Nella pagina che segue si trova lo **schema E-R ristrutturato** dopo aver eliminato e trasformato le gerarchie:



4. SCHEMA LOGICO



5. DEFINIZIONE DELLO SCHEMA LOGICO TRAMITE DDL DI MySQL

```
/*Crea la tabella per Guardia*/
CREATE TABLE Guardia(
    ID_Guardia INT(4) PRIMARY KEY NOT NULL,
    Nome VARCHAR(25) NOT NULL,
    Cognome VARCHAR(25) NOT NULL,
    DataN DATE,
    Stipendio INT(4),
    Specialità VARCHAR(25),
    DataR DATE
);

/*Crea la tabella per Reparto*/
CREATE TABLE Reparto (
    CodR INT(4) PRIMARY KEY NOT NULL,
    Nome VARCHAR(25) NOT NULL,
    Tipo ENUM ('blocco','zonaC') NOT NULL,
    Responsabile INT(4),
    FOREIGN KEY (Responsabile) REFERENCES Guardia(ID_Guardia)
ON DELETE SET NULL ON UPDATE CASCADE
);

/*Crea la tabella per Stanza*/
CREATE TABLE Stanza (
    Numero INT(4) PRIMARY KEY NOT NULL,
    Blocco INT(4),
    Posti INT(3) NOT NULL,
    FOREIGN KEY (Blocco) REFERENCES Reparto(CodR) ON DELETE CASCADE ON UPDATE CASCADE
);

/*Crea la tabella per Detenuto*/
CREATE TABLE Detenuto(
    Mat INT(4) PRIMARY KEY NOT NULL,
    Nome VARCHAR(25) NOT NULL,
    Cognome VARCHAR(25) NOT NULL,
    DataN DATE NOT NULL,
    Naz VARCHAR(25),
    Numero INT(4),
    FOREIGN KEY (Numero) REFERENCES Stanza(Numero) ON DELETE SET NULL ON UPDATE
CASCADE
);

/*Crea la tabella per Studio*/
CREATE TABLE Studio(
    CodS INT(4) PRIMARY KEY NOT NULL,
    Nome VARCHAR(25) NOT NULL,
    Città VARCHAR(25) NOT NULL,
    Indirizzo VARCHAR(25) NOT NULL,
    Recapito VARCHAR(25)
);

/*Crea la tabella per Avvocato*/
```

```

CREATE TABLE Avvocato(
    CodA INT(4) PRIMARY KEY NOT NULL,
    Nome VARCHAR(25) NOT NULL,
    Cognome VARCHAR(25) NOT NULL,
    CodS INT(4),
    FOREIGN KEY (CodS) REFERENCES Studio(CodS) ON DELETE SET NULL ON UPDATE CASCADE
);

/*Crea la tabella per Processo*/
CREATE TABLE Processo(
    Mat INT(4) NOT NULL,
    DataP DATE NOT NULL,
    Reato VARCHAR(1501),
    CodA INT(4) ,
    Durata INT(4),
    DataI DATE,
    Città VARCHAR(25),
    PRIMARY KEY(Mat, DataP),
    FOREIGN KEY (Mat) REFERENCES Detenuto(Mat) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (CodA) REFERENCES Avvocato(CodA) ON DELETE SET NULL ON UPDATE CASCADE
);

/*Crea la tabella per ContoCorrente*/
CREATE TABLE ContoCorrente(
    CC INT(4) NOT NULL,
    Saldo INT(4),
    Mat INT(4) NOT NULL,
    PRIMARY KEY (CC, Mat),
    FOREIGN KEY (Mat) REFERENCES Detenuto(Mat) ON DELETE CASCADE ON UPDATE CASCADE
);

/*Crea la tabella per RegistroTurniGuardie*/
CREATE TABLE RegistroTurniGuardie (
    ID_Guardia INT(4) NOT NULL,
    Data DATE NOT NULL,
    Reparto INT(4),
    OraI TIME,
    OraF TIME,
    PRIMARY KEY (ID_Guardia, Data, Reparto),
    FOREIGN KEY (Reparto) REFERENCES Reparto(CodR) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (ID_Guardia) REFERENCES Guardia(ID_GUARDIA)
    ON DELETE CASCADE ON UPDATE CASCADE
);

/*Crea la tabella per Amministrativo*/
CREATE TABLE Amministrativo (
    ID_Amm INT(4) PRIMARY KEY NOT NULL,
    Nome VARCHAR(25) NOT NULL,
    Cognome VARCHAR(25) NOT NULL,
    DataN DATE,
    Stipendio INT(4),
    Ruolo VARCHAR(25)
);

/*Crea la tabella per AziendaAppaltata*/
CREATE TABLE AziendaAppaltata(
    pIVA VARCHAR(11) PRIMARY KEY NOT NULL,
    Nome VARCHAR(25) NOT NULL,
    Recapito VARCHAR(25),

```

```

    Indirizzo VARCHAR(25)
);

/*Crea la tabella per Lavoro*/
CREATE TABLE Lavoro(
    CodL INT(4) PRIMARY KEY NOT NULL,
    Descrizione VARCHAR(60),
    RetrOraria INT(4) NOT NULL,
    Max INT(4) NOT NULL,
    Tipo ENUM ('interno','esterno'),
    pIVA VARCHAR(11),
    Responsabile INT(4),
    FOREIGN KEY (pIVA) REFERENCES AziendaAppaltata(pIVA)
ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Responsabile) REFERENCES Amministrativo(ID_Amm)
ON DELETE SET NULL ON UPDATE CASCADE
);

/*Crea la tabella per Turno*/
CREATE TABLE Turno(
    Data DATE NOT NULL,
    Codice INT(4) NOT NULL,
    OraI TIME NOT NULL,
    OraF TIME NOT NULL,
    PRIMARY KEY (Data, Codice, OraI),
    FOREIGN KEY (Codice) REFERENCES Lavoro(CodL) ON DELETE CASCADE ON UPDATE CASCADE
);

/*Crea la tabella per RegistroImpieghi*/
CREATE TABLE RegistroImpieghi(
    Det INT(4) NOT NULL,
    Lavoro INT(4) NOT NULL,
    DataI DATE NOT NULL,
    DataF DATE NOT NULL,
    PRIMARY KEY (Det, DataI),
    FOREIGN KEY (Det) REFERENCES Detenuto(Mat) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Lavoro) REFERENCES Lavoro(CodL) ON DELETE NO ACTION ON UPDATE CASCADE
);

/*Crea la tabella per ImpieghiAttuali*/
CREATE TABLE ImpieghiAttuali(
    Det INT(4) PRIMARY KEY NOT NULL,
    Lavoro INT(4) NOT NULL,
    DataI DATE NOT NULL,
    FOREIGN KEY (Det) REFERENCES Detenuto(Mat) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Lavoro) REFERENCES Lavoro(CodL) ON DELETE CASCADE ON UPDATE CASCADE
);

/*Crea la tabella per Attività*/
CREATE TABLE Attività (
    CodAtt INT(4) PRIMARY KEY NOT NULL,
    Descrizione VARCHAR(150) NOT NULL,
    Max INT(4)
);

/*Crea la tabella per RegistroAttività*/
CREATE TABLE RegistroAttività (
    AttSvolta INT(4) NOT NULL,
    Rep INT(4),
    GiornoSett ENUM('lun','mar','mer','gio','ven','sab','dom') NOT NULL,
    OraI TIME,

```

```

OraF TIME,
PRIMARY KEY(AttSvolta, GIORNOSett, OraI),
FOREIGN KEY (AttSvolta) REFERENCES Attività(CodAtt) ON DELETE CASCADE ON UPDATE
CASCADE,
FOREIGN KEY (Rep) REFERENCES Reparto(CodR) ON DELETE SET NULL ON UPDATE CASCADE
);

/*Crea la tabella per Iscritti*/
CREATE TABLE Iscritti (
    Matricola INT(4) NOT NULL,
    Attività INT(4) NOT NULL,
    PRIMARY KEY (Matricola, Attività),
    FOREIGN KEY (Matricola) REFERENCES Detenuto(Mat) ON DELETE CASCADE ON UPDATE
CASCADE,
    FOREIGN KEY (Attività) REFERENCES Attività(CodAtt) ON DELETE CASCADE ON UPDATE
CASCADE
);

/*Crea la tabella per Educatore*/
CREATE TABLE Educatore(
    ID_Educatore INT(4) PRIMARY KEY NOT NULL,
    Nome VARCHAR(25) NOT NULL,
    Cognome VARCHAR(25) NOT NULL,
    DataN DATE,
    Stipendio INT(4),
    Laurea VARCHAR(25),
    Attività INT(4),
    pIVA VARCHAR(11),
    FOREIGN KEY (Attività) REFERENCES Attività(CodAtt)
ON DELETE SET NULL ON UPDATE CASCADE,
    FOREIGN KEY (pIVA) REFERENCES AziendaAppaltata(pIVA)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

6. QUERY

1. Selezionare nome, cognome e lavoro svolto da tutti i detenuti con più di 25 anni che svolgono un lavoro fuori dall'istituto

```
SELECT D.Nome AS Nome, D.Cognome AS Cognome, L.Descrizione  
FROM Detenuto D, ImpieghiAttuali I, Lavoro L  
WHERE D.Mat = I.Det AND I.Lavoro = L.CodL AND round(datediff(now(),D.DataN)/365)>25 AND  
L.Tipo = 'esterno'
```

Nome	Cognome	Descrizione
Paolo	Rossi	Costruzioni
Massimo	Esposito	Costruzioni
Tommaso	Romano	Giardinaggio
Adriano	Costa	Giardinaggio
Alberto	Giordano	Pittura
Fabio	Greco	Lavoro nei campi
Cesare	Moretti	Lavoro nei campi

2. Selezionare nome, cognome e stanza dei detenuti che fanno parte del Blocco1

```
SELECT D.Nome, D.Cognome, S.Numero  
FROM Detenuto D, Stanza S  
WHERE D.Numero = S.Numero AND S.Blocco = 501
```

Nome	Cognome	Numero
Paolo	Rossi	601
Davide	Gallo	601
Riccardo	Lombardi	601
Matteo	Russo	602
Enrico	De Luca	602
Luigi	Barbieri	602
Massimo	Esposito	609

3. Selezionare il nome e il cognome dei responsabili che supervisionano le attività che si svolgono di lunedì, indicando anche che attività si svolge e in quale reparto

```
SELECT DISTINCT G.Nome, G.Cognome, R.Nome as 'Nome Reparto', A.Descrizione
FROM Guardia G JOIN Reparto R ON G.ID_Guardia = R.Responsabile JOIN RegistroAttività RA ON
R.CodR = RA.Rep JOIN Attività A ON RA.AttSvolta = A.CodAtt
WHERE RA.GiornoSett = 'lun'
```

Nome	Cognome	Nome Reparto	Descrizione
Mario	Bianchi	Teatro	Tennis
Simone	Di Maggio	Aula Scolastica	Istruzione elementare
Simone	Di Maggio	Aula Scolastica	Istruzione superiore

4. Per ogni attività, seleziona la descrizione, il nome e cognome dell'educatore e il numero di iscritti.

```
SELECT A.Descrizione, E.Nome, E.Cognome, H0.Partecipanti
FROM Attività A JOIN Educatore E ON A.CodAtt = E.Attività LEFT JOIN (SELECT I.Attività,
COUNT(I.Matricola) as Partecipanti FROM Iscritti I GROUP BY I.Attività) H0 ON A.CodAtt =
H0.Attività
```

Descrizione	Nome	Cognome	Partecipanti
Corso cucito	Marco	Rossi	4
Istruzione elementare	Paola	Bianchi	3
Istruzione superiore	Luca	Rossato	3
Calcio	Giovanni	Lora	3
Teatro	Claudio	Rossi	2
Pallavolo	Michele	Paolini	5
Tennis	Tommaso	De Marchi	3
Rugby	Francesca	Giusti	2

5. Per ogni reparto si vuole sapere la media dei soldi in CC dei detenuti che risiedono nel blocco

```
SELECT R.Nome, AVG(CC.Saldo) as SaldoMedio
FROM Reparto R JOIN Stanza S ON R.CodR = S.Blocco JOIN Detenuto D ON S.Numero = D.Numero
JOIN ContoCorrente CC ON D.Mat = CC.Mat
GROUP BY R.Nome
```

Nome	SaldoMedio
Blocco1	130.0000
Blocco2	123.3333
Blocco3	100.0000

6. Procedura che aumenta il saldo del conto corrente di un detenuto in base al lavoro che si seleziona e in base a quanto lo si vuole aumentare

```
CREATE PROCEDURE AccreditoCC (ValoreA INT(4), Lavoro INT(4))
UPDATE ContoCorrente CC JOIN ImpieghiAttuali IA ON CC.Mat = IA.Det
SET CC.Saldo = CC.Saldo + ValoreA
WHERE IA.Lavoro = Lavoro
```


7. TRIGGER e FUNZIONI

7.1 TRIGGERS

7.1.1 Trigger 'AssegnaStanza':

Prima dell'inserimento di un Detenuto nella base di dati il trigger cerca la prima stanza libera e la assegna al detenuto.

```
DROP TRIGGER IF EXISTS AssegnaStanza;
DELIMITER $$
CREATE TRIGGER AssegnaStanza BEFORE INSERT ON Detenuto
FOR EACH ROW
BEGIN
    DECLARE stanzaAssegnata INT;

    SELECT MIN(S1.Numero) as NumeroStanza
    FROM Stanza S1 left join (SELECT COUNT(D.Mat) as Occupanti, D.Numero
    FROM Detenuto D
    GROUP BY D.Numero) SL on SL.Numero = S1.Numero
    WHERE S1.Posti > SL.Occupanti or SL.Occupanti is null INTO stanzaAssegnata;

    SET NEW.Numero = stanzaAssegnata;

END
$$
DELIMITER ;
```

7.1.2 Trigger "RegistraLavoro":

Quando viene modificato il lavoro attuale di un detenuto verifica la disponibilità del nuovo lavoro assegnato, ma se è già stato raggiunto il massimo numero di posti di lavoro disponibili gli si assegna il primo lavoro con posti ancora non occupati. Inoltre, se il detenuto aveva già un lavoro assegnato, questo viene inserito nel registro impieghi specificando in quale periodo è stato effettuato.

```
DELIMITER &&
CREATE TRIGGER RegistraLavoro BEFORE UPDATE ON ImpieghiAttuali
FOR EACH ROW
BEGIN

    DECLARE DetImp INT;
    DECLARE MaxImp INT;
    DECLARE NewL INT;

    SELECT COUNT(IA.Det)
    FROM ImpieghiAttuali IA
    WHERE Lavoro = NEW.Lavoro INTO DetImp;
```

```

SELECT Max
FROM Lavoro L
WHERE L.CodL = NEW.Lavoro INTO MaxImp;

```

```

IF(DetImp > MaxImp) THEN
SELECT MIN(L.CodL)
FROM Lavoro L LEFT JOIN (SELECT COUNT(Det) as Imp, Lavoro FROM ImpieghiAttuali GROUP BY
Lavoro) IA ON L.CodL = IA.Lavoro
WHERE L.Max > IA.Imp OR IA.Imp IS NULL INTO NewL;
SET NEW.Lavoro = NewL;
END IF;

```

```

INSERT INTO RegistroImpieghi(Det, Lavoro, DataI, DataF) VALUES
(NEW.Det, OLD.Lavoro, OLD.DataI, NEW.DataI);

```

```

END
&&
DELIMITER ;

```

7.2 FUNCTIONS

7.2.1 Funzione “ContaMassimo”:

Con questa funzione viene calcolato il lavoro con il massimo numero di detenuti impiegati in una determinata data inserita dall’utente del sistema.

```

DROP FUNCTION IF EXISTS ContaMassimo;
DELIMITER $$
CREATE FUNCTION ContaMassimo (Datalavoro DATE)
RETURNS VARCHAR(25)
BEGIN
DECLARE LavoroMassimo VARCHAR(25);

```

```

SELECT H.Descrizione
FROM
(SELECT COUNT(D.Mat) as numL, L.Descrizione
FROM Detenuto D JOIN ImpieghiAttuali IA ON D.Mat = IA.Det JOIN Lavoro L ON IA.Lavoro =
L.CodL JOIN RegistroImpieghi RI ON L.CodL = RI.Lavoro
WHERE (RI.DataI <= Datalavoro AND RI.DataF >= Datalavoro) OR IA.DataI <= Datalavoro
GROUP BY L.Descrizione) H
ORDER BY H.numL DESC
LIMIT 1 INTO LavoroMassimo;

```

```

RETURN LavoroMassimo;

```

```

END

```

```

$$
DELIMITER ;

```

7.2.2 Funzione "DataRilascioDet":

Con questa funzione si calcola in che data un determinato detenuto verrà rilasciato.

```
DELIMITER $$
CREATE FUNCTION DataRilascio(Det INT(4))
RETURNS DATE
BEGIN
DECLARE DataF DATE;
DECLARE GiorniR INT(4);

SELECT DATE(DATE_ADD(P.DataI, interval P.Durata DAY))
FROM Detenuto D JOIN Processo P ON D.Mat = P.Mat
WHERE D.Mat = Det INTO DataF;

RETURN DataF;
END
$$
DELIMITER ;
```