

Relazione dell'elaborato di Basi di Dati
Sistema di gestione penitenziario

Leonardo Grimaldi

23 agosto 2024

Indice

1	Analisi	3
1.1	Introduzione	3
1.2	Intervista	3
1.3	Estrazione dei concetti principali	4
2	Progettazione concettuale	6
2.1	Schema scheletro	6
2.2	Raffinamenti proposti	8
2.3	Schema concettuale finale	8
3	Progettazione logica	10
3.1	Stima del volume dei dati	10
3.2	Descrizione delle operazioni principali e stima della loro frequenza	12
3.3	Schemi di navigazione e tabelle degli accessi	12
3.3.1	Inserire un nuovo detenuto	12
3.3.2	Scambio posto letto con un altro detenuto	13
3.4	Raffinamento dello schema	14
3.4.1	Specializzazione Amministratore e Guardia	14
3.4.2	Specializzazione Letto, Medica e Solitaria	14
3.4.3	Scelta delle chiavi primarie	15
3.5	Analisi delle ridondanze	15
3.6	Traduzione di entità e associazioni in relazioni	16
3.7	Schema relazionale finale	16
3.8	Traduzione delle operazioni in query SQL	18
3.8.1	Inserimento nuovo detenuto	18
3.8.2	Scambio posto letto con un altro detenuto	18
3.8.3	Ricoverare un detenuto	19
3.8.4	Isolare un detenuto	19
3.8.5	Inserimento nuova guardia	20
3.8.6	Inserimento nuovo amministratore	20

3.8.7	Visualizzare il personale (amministratori, guardie) . . .	20
3.8.8	Visualizzare la lista dei detenuti presenti	20
3.8.9	Visualizzare l'andamento settimanale di nuovi detenuti	21
3.8.10	Visualizzare i primi cinque detenuti che sono stati tra- sferiti in celle solitarie più volte	21
3.8.11	Inserimento nuovo orario	21
3.9	Progettazione dell'applicazione	22
3.9.1	Screenshot interfaccia utente	23

Capitolo 1

Analisi

1.1 Introduzione

Viene commissionata da un ente governativo la realizzazione di un software gestionale per una casa circondariale che faciliti il tracciamento di detenuti e loro spostamenti.

1.2 Intervista

Si chiede di realizzare un portale che consenta di gestire e storicizzare varie operazioni comuni di un carcere. Per i **detenuti** in arrivo si vogliono memorizzare gli estremi della persona. I dati richiesti sono:

- Nome, cognome, data di nascita, il numero della carta d'identità, al-
tezza

Il carcere gestisce solamente detenuti italiani maggiorenni in possesso di carta d'identità quindi non occorre gestire il caso in cui essa non sia presente. Un detenuto può essere rilasciato e rientrare nel carcere, ma anche decedere durante la sua permanenza. Ai detenuti sono assegnate delle **celle** letto in
base alla disponibilità. Esse hanno una capacità e più prigionieri possono
risiedere al loro interno.

Nel corso della loro permanenza le assegnazioni possono subire variazioni e si dovrà quindi tenere traccia degli **spostamenti**. Questo include la data e ora di uscita e in quale cella è avvenuto lo spostamento. All'interno della
prigione sono presenti anche celle mediche e solitarie all'interno delle quali il
prigioniero può risiedere temporaneamente. Ogni cella appartiene a un **piano**
che viene pattugliato da una o più guardie. Ogni piano fa parte di un solo

blocco. I turni di pattuglia sono assegnati in base a un **orario** prestabilito in cui ogni giorno della settimana è formato da 3 turni:

- 20 • Mattina: 06:00 - 14:00
- Pomeriggio/sera: 14:00 - 22:00
- Notte: 22:00 - 06:00 (del giorno successivo)

La guardia lavorerà quindi per 8 ore al giorno con una pausa intermedia di 30 minuti e fine turno di 30 minuti. Le pause e i cambi di turno non verranno
25 gestiti dal database ai fini di copertura dell'orario, ma si suppone che vi sia una guardia di riserva che subentra temporaneamente.

Il **personale** del carcere è formato quindi da guardie, ma anche da amministratori e di entrambi si vuole memorizzare: il nome, cognome, data di nascita, sesso e codice fiscale. Gli amministratori sono le persone che
30 hanno accesso al sistema gestionale e possono essere anche le guardie stesse. Dovranno poter accedere al sistema con una password a loro assegnata. Sia le guardie che gli amministratori possiedono un badge che li identifica univocamente all'interno della struttura. Di loro si vuole memorizzare inoltre:

- 35 • Nome, cognome, codice fiscale e sesso.

1.3 Estrazione dei concetti principali

Dall'intervista si possono estrapolare diverse figure che consentiranno di modellare lo schema concettuale.

Detenuto

Sinonimi: prigioniero, carcerato

Persona rinchiusa nel carcere. Ha una cella letto assegnata per tutta la permanenza.

Operazioni

- Trasferimento cella letto
- Spostamento temporaneo in celle mediche o solitarie
- Dichiarazione di decesso

Cella

In generale, il luogo dove risiede il carcerato. Ha una capacità massima e può essere di tre tipi: letto, medica e solitaria. Può appartenere a un solo piano.

Piano

Piano dell'edificio. In esso sono contenute molteplici celle. Esso può essere controllato da una o più guardie.

Blocco

Parte strutturale del carcere dove sono presenti un insieme di piani.

Personale

L'insieme di persone che non sono detenuti, ma lavorano nel carcere e garantiscono la sicurezza e il suo corretto funzionamento. Si dividono in guardie e amministratori e posseggono un badge.

Guardie

Personale carcerario a cui è affidato il compito di controllare i piani in un certo turno del giorno

Amministratori

Personale che può accedere al sistema software gestionale attraverso una password. Gli amministratori possono essere anche delle guardie.

Operazioni

- Inserimento guardie, assegnazione orario di lavoro
- Gestione detenuti: registrazione, trasferimento

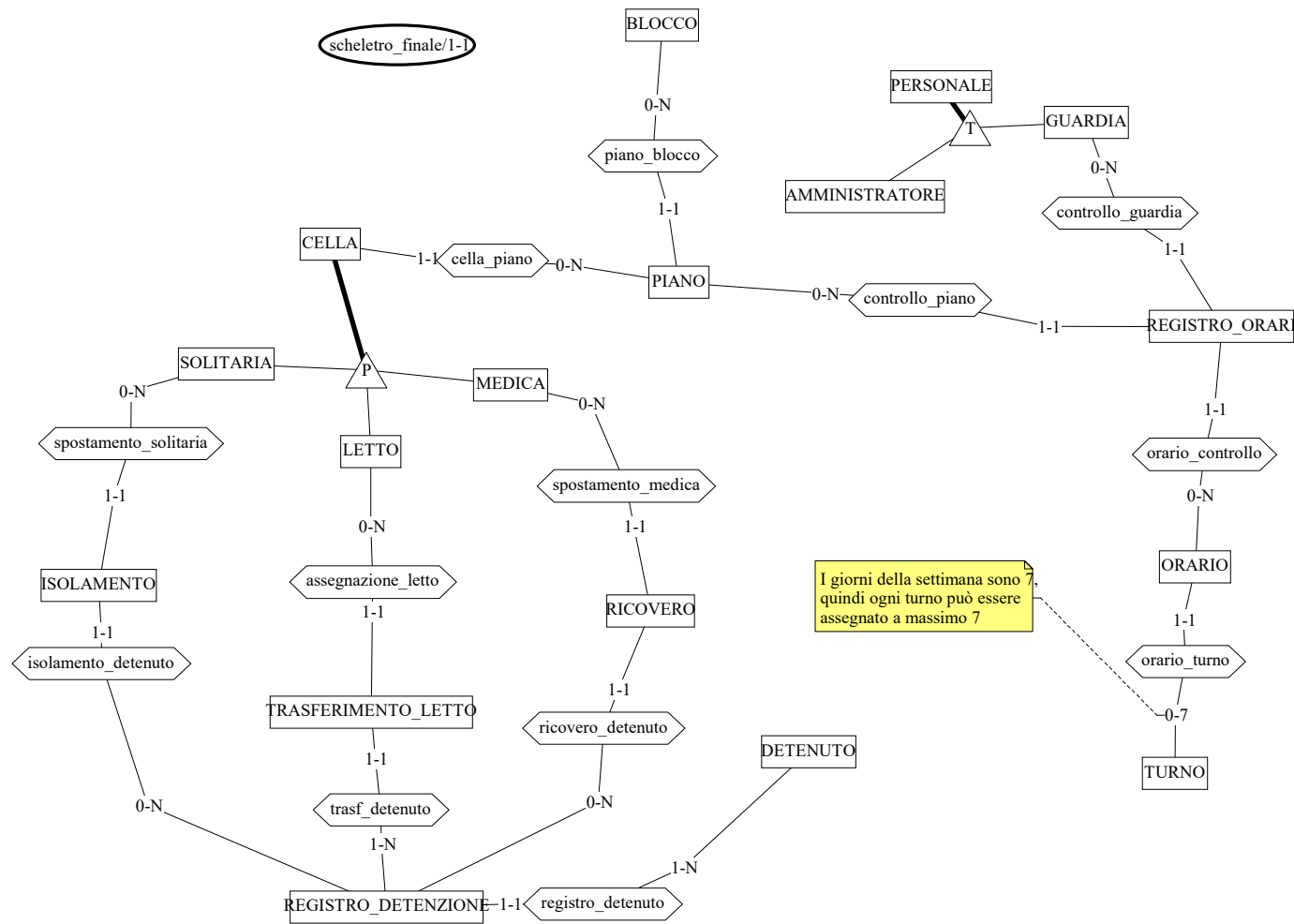
Orario

L'orario di lavoro che sarà assegnato alle guardie. Avrà tre turni: mattina (06:00 - 14:00), pomeriggio (14:00 - 22:00) e notte (22:00 - 06:00).

Capitolo 2

Progettazione concettuale

2.1 Schema scheletro



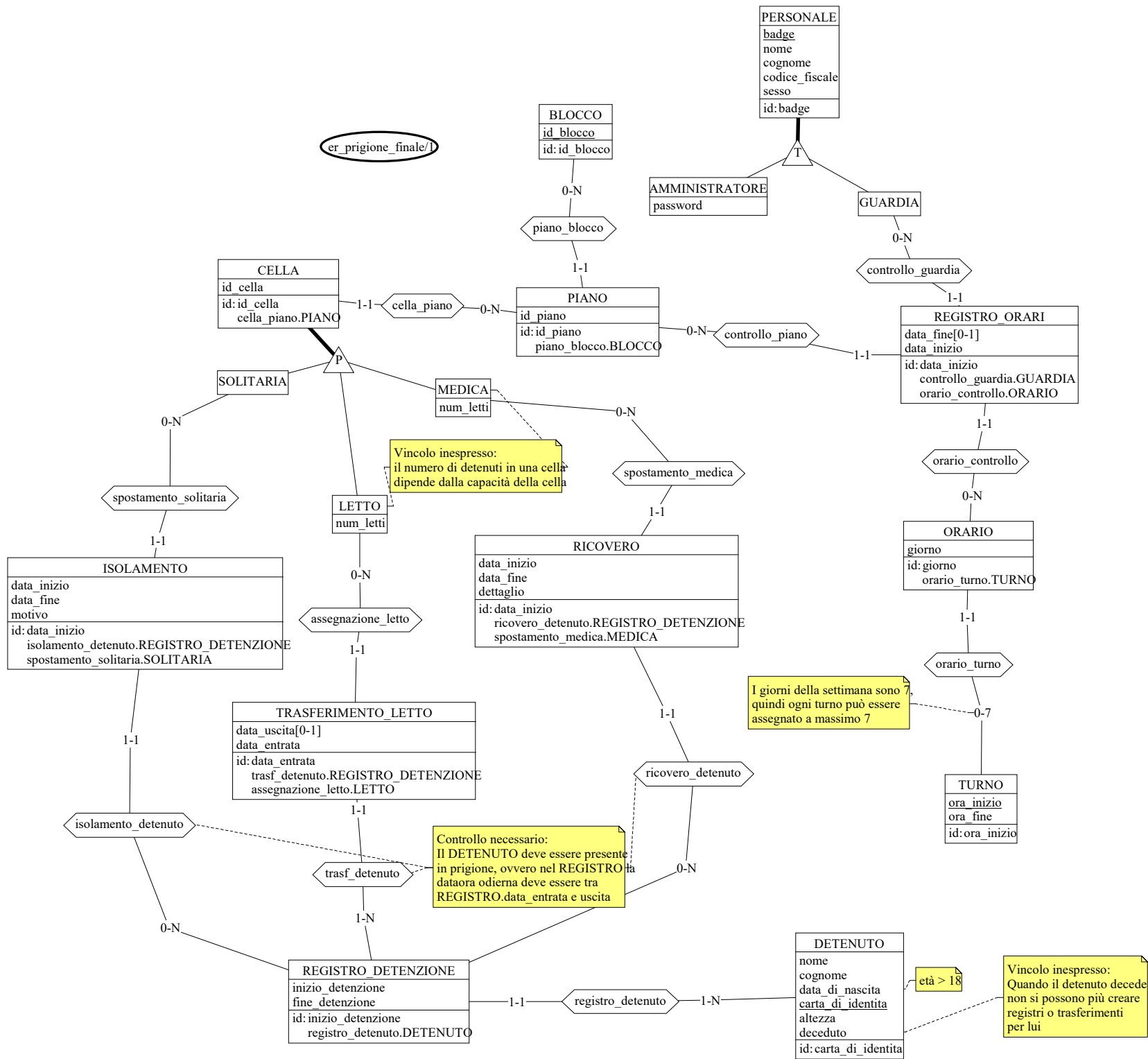
2.2 Raffinamenti proposti

Lo schema scheletro è una rappresentazione fedele ai concetti principali estratti nella sezione precedente, ma contiene anche un paio di elementi aggiuntivi che è stato necessario definire per modellare correttamente il dominio. Innanzitutto si possono notare le tre nuove entità TRASFERIMENTO_LETTO, RICOVERO e ISOLAMENTO che consentiranno di conservare le informazioni sui cambi di celle dei detenuti e ulteriori informazioni come la prognosi oppure il motivo. Tutte queste hanno una cardinalità 1-1 sia dalla parte delle corrispondenti specializzazioni che da quella del REGISTRO_DETENZIONE (introdotto nel seguente paragrafo) perché un TRASFERIMENTO non può esistere se manca un riferimento a chi e dove è stato spostato.

Un'altra aggiunta importante è il concetto di REGISTRO_DETENUTO; dall'intervista si è analizzato che un prigioniero potrebbe rientrare nel sistema e quindi si è creata la necessità di tenere traccia di questi. Modellando così, un detenuto può essere reinserito nel sistema senza perdere informazioni sui suoi incarceramenti passati. I TURNI, invece, sono in associazione 0-7 con l'ORARIO per esprimere il vincolo sul numero di giorni di una settimana (che infatti sono sette)

L'entità REGISTRO_ORARIO consente di storicizzare gli orari di lavoro delle GUARDIE per un certo PIANO. Le cardinalità delle associazioni riferite a questa entità sono state ideate in modo tale da consentire a più GUARDIE di pattugliare un piano e una GUARDIA avere un solo PIANO da controllare in un ORARIO.

2.3 Schema concettuale finale



Capitolo 3

Progettazione logica

3.1 Stima del volume dei dati

Per avere una stima più del volume più precisa si è chiesto al committente la struttura del carcere. La capacità è di 400 persone e vi sono la sezione A e B con due piani e 50 celle ciascuna, abilitate a ospitare massimo due persone. Inoltre, vi è anche la sezione C composta anch'essa da due piani: il primo piano 10 celle solitarie, il secondo 10 celle mediche.

Concetto	Costrutto	Volume
DETENUTO	E	500
assegnazione_letto	A	1000
TRASFERIMENTO_LETTO	E	1000
trasf_detenuto	A	1000
assegnazione_letto	A	1000
ISOLAMENTO	E	200
spostamento_solitaria	A	200
isolamento_detenuto	A	200
RICOVERO	E	350
spostamento_medica	A	350
ricovero_detenuto	A	350
registro_detenuto	A	700
REGISTRO_DETENZIONE	E	700
CELLA	E	220
MEDICA	E	10
LETTO	E	200
SOLITARIA	E	10
cella_piano	A	220
PIANO	E	6
piano_blocco	A	6
BLOCCO	E	3
controllo_piano	A	175
REGISTRO_ORARI	E	175
controllo_guardia	A	175
GUARDIA	E	25
AMMINISTRATORE	E	10
PERSONALE	E	35
orario_controllo	A	175
ORARIO	E	21
orario_turno	A	21
TURNO	E	3

3.2 Descrizione delle operazioni principali e stima della loro frequenza

Operazione	Frequenza	Tipo
Inserimento nuovo detenuto	2/giorno	I
Scambio posto letto con un altro detenuto	5/mese	I
Ricoverare un detenuto	5/mese	I
Isolare un detenuto	3/mese	I
Inserimento nuova guardia	5/anno	I
Inserimento nuovo amministratore	2/anno	I
Visualizzare il personale (amministratori, guardie)	20/giorno	B
Visualizzare la lista dei detenuti presenti	50/giorno	B
Visualizzare l'andamento settimanale di nuovi detenuti	60/giorno	B
Visualizzare i primi cinque detenuti che sono stati trasferiti in celle solitarie più volte	50/mese	B
Inserimento nuovo orario	5/mese	I

3.3 Schemi di navigazione e tabelle degli accessi

In questa parte si elencano le tabelle degli accessi per le operazioni principali e più complesse. Le scritture costano il doppio.

3.3.1 Inserire un nuovo detenuto

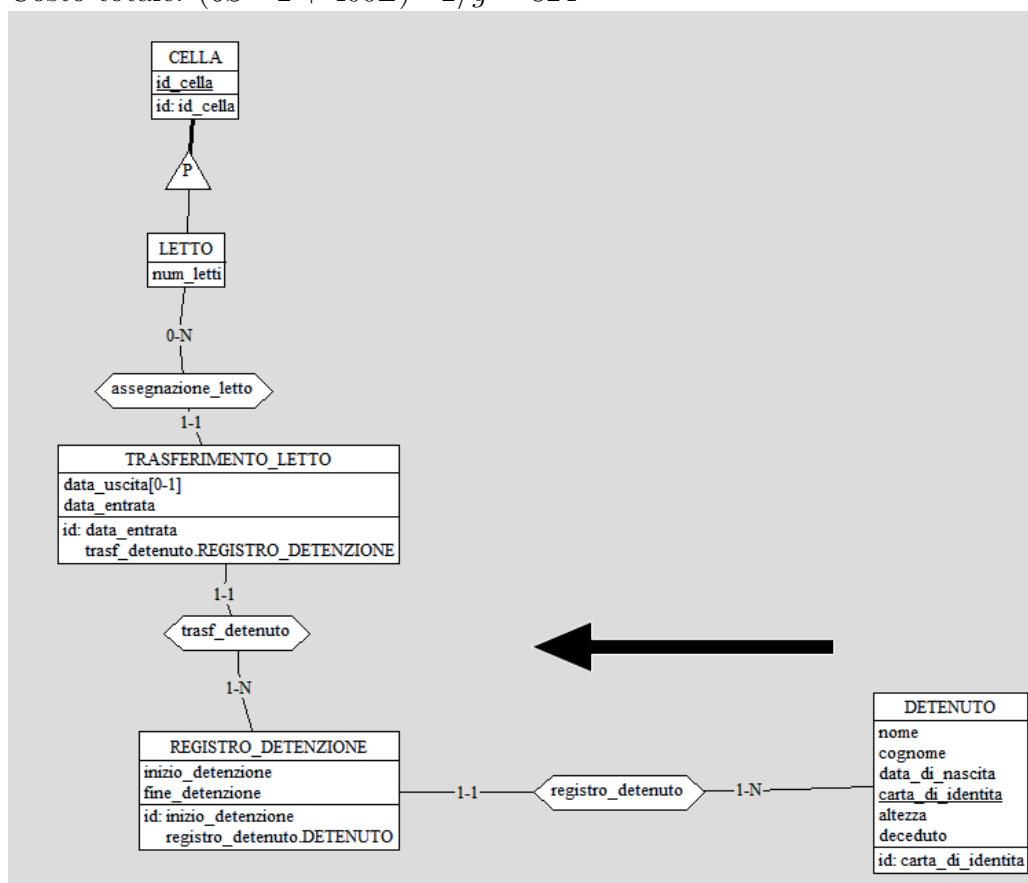
1. Inserire il detenuto
2. Inserirlo nel registro
3. Leggere le celle letto libere

Richiede: Contare i trasferimenti con data uscita NULL per ogni singola cella e verificare che siano minori del numero letti

4. Inserirlo in una cella letto

Concetto	Costrutto	Accessi	Tipo
DETENUTO	E	1	S
registro_detenuto	A	1	S
REGISTRO_DETENZIONE	E	1	S
TRASFERIMENTO_LETTO	E	$200 * 2 = 400$	L
trasf_detenuto	A	1	S
TRASFERIMENTO_LETTO	E	1	S
assegnazione_letto	A	1	S

Nota: $200 * 2 = 400$ perché 200 sono le celle letto e due sono i posti letto.
Costo totale: $(6S * 2 + 400L) * 2/g = 824$



3.3.2 Scambio posto letto con un altro detenuto

- Visualizzare le celle letto
- Visualizzare gli occupanti della cella selezionata

- Aggiornare la data uscita del TRASFERIMENTO_LETTO del primo detenuto
- Inserire un nuovo trasferimento per la cella di destinazione
- Aggiornare la data uscita del TRASFERIMENTO_LETTO dell'altro detenuto
- Inserire un nuovo trasferimento per l'altro detenuto nella cella del primo

Concetto	Costrutto	Accessi Tipo	
LETTO	E	200	L
TRASFERIMENTO_LETTO	E	2	L
DETENUTO	E	2	L
trasf_detenuto	A	2	S
TRASFERIMENTO_LETTO	E	4	S
assegnazione_letto	A	2	S

Costo totale: $(8S * 2 + 204L) * 5/30g = 36.7$

3.4 Raffinamento dello schema

3.4.1 Specializzazione Amministratore e Guardia

Per mappare la specializzazione si è deciso di creare una relazione per ogni sottoclasse (in questo caso due) e inserire gli attributi nelle corrispondenti, nonché una chiave esterna che si riferisca alla chiave primaria badge della relazione PERSONALE. Era anche possibile creare una sola relazione PERSONALE con gli attributi delle sotto-entità, ma questo avrebbe portato a maggiori controlli applicativi e valori NULL dato il numero di guardie vs amministratori.

3.4.2 Specializzazione Letto, Medica e Solitaria

Le tre specializzazioni che conferiscono nell'entità CELLA sono totali ed esclusive. Si possono quindi eliminare e creare una unica relazione CELLA che avrà tutti gli attributi delle sottoclassi eliminate: in questo caso solo num_letti. Viene inoltre aggiunto l'attributo "tipo" che potrà avere come valori ENUM: "Solitaria", "Letto", "Medica" per poter differenziare le singole celle. Sarebbe stato anche possibile eliminare l'entità CELLA e creare tre

relazioni, ma così facendo bisognava creare altre tre nuove relazioni da collegare con l'entità PIANO il che avrebbe complicato di molto le query SQL. Facendo in questo modo si ha un unico collegamento con l'entità PIANO, con l'unico svantaggio quello di dover esprimere a livello applicativo vincoli sul tipo.

3.4.3 Scelta delle chiavi primarie

Le chiavi di ogni entità sono già state scelte ed evidenziate nello schema E/R. Per la chiave primaria "badge" di PERSONALE si è fatto riferimento all'intervista del committente nel quale ha espressamente indicato che il personale è in possesso di un codice univoco.

3.5 Analisi delle ridondanze

Data la complessità e il costo delle operazioni di inserimento di un detenuto nonché l'assegnazione di una cella, si è deciso di inserire l'attributo ridondante "posti_occupati" che verrà incrementato e decrementato quando un detenuto rispettivamente entra o lascia una cella. Ovviamente, questo numero non dovrà mai superare il limite massimo dato da num_letti, quindi si dovrà gestire questo vincolo a livello applicativo. Dopo l'aggiunta dell'attributo ridondante, la tabella della sezione 3.3.1 Inserire un nuovo detenuto diventerà:

Concetto	Costrutto	Accessi Tipo	
DETENUTO	E	1	S
registro_detenuto	A	1	S
REGISTRO_DETENZIONE	E	1	S
trasf_detenuto	A	1	S
TRASFERIMENTO_LETTO	E	1	S
assegnazione_letto	A	1	S
CELLA	E	200	L
TRASFERIMENTO_LETTO	E	2	L
CELLA	E	1	S

Come si può notare, si aggiungono due letture di TRASFERIMENTO_LETTO per contare i nuovi occupanti e poi si aggiunge una scrittura a CELLA (in precedenza LETTO) per aggiornare l'attributo "posti_occupati". Si semplificano così anche le operazione d'isolamento di un detenuto e ricovero che non necessitano la lettura di tutti i record ISOLAMENTO e RICOVERO corrispondenti.

3.6 Traduzione di entità e associazioni in relazioni

detenuto(carta_di_identita, nome, cognome, data_di_nascita, altezza, deceduto)

registro_detenzione(inizio_detenzione, fine_detenzione, carta_di_identita: detenuto)

trasferimento_letto(data_entrata, data_uscita*, inizio_detenzione: registro_detenzione, id_blocco: cella, id_piano: cella, id_cella: cella)

isolamento(data_inizio, data_fine, inizio_detenzione: registro_detenzione, id_blocco: cella, id_piano: cella, id_cella: cella, motivo)

ricovero(data_inizio, data_fine, inizio_detenzione: registro_detenzione, id_blocco: cella, id_piano: cella, id_cella: cella, dettaglio)

cella(id_blocco: piano, id_piano: piano, id_cella, tipo, num_letti, posti_occupati)

piano(id_piano, id_blocco: blocco)

blocco(id_blocco)

personale(badge, nome, cognome, codice_fiscale, sesso)

amministratore(badge: personale, password)

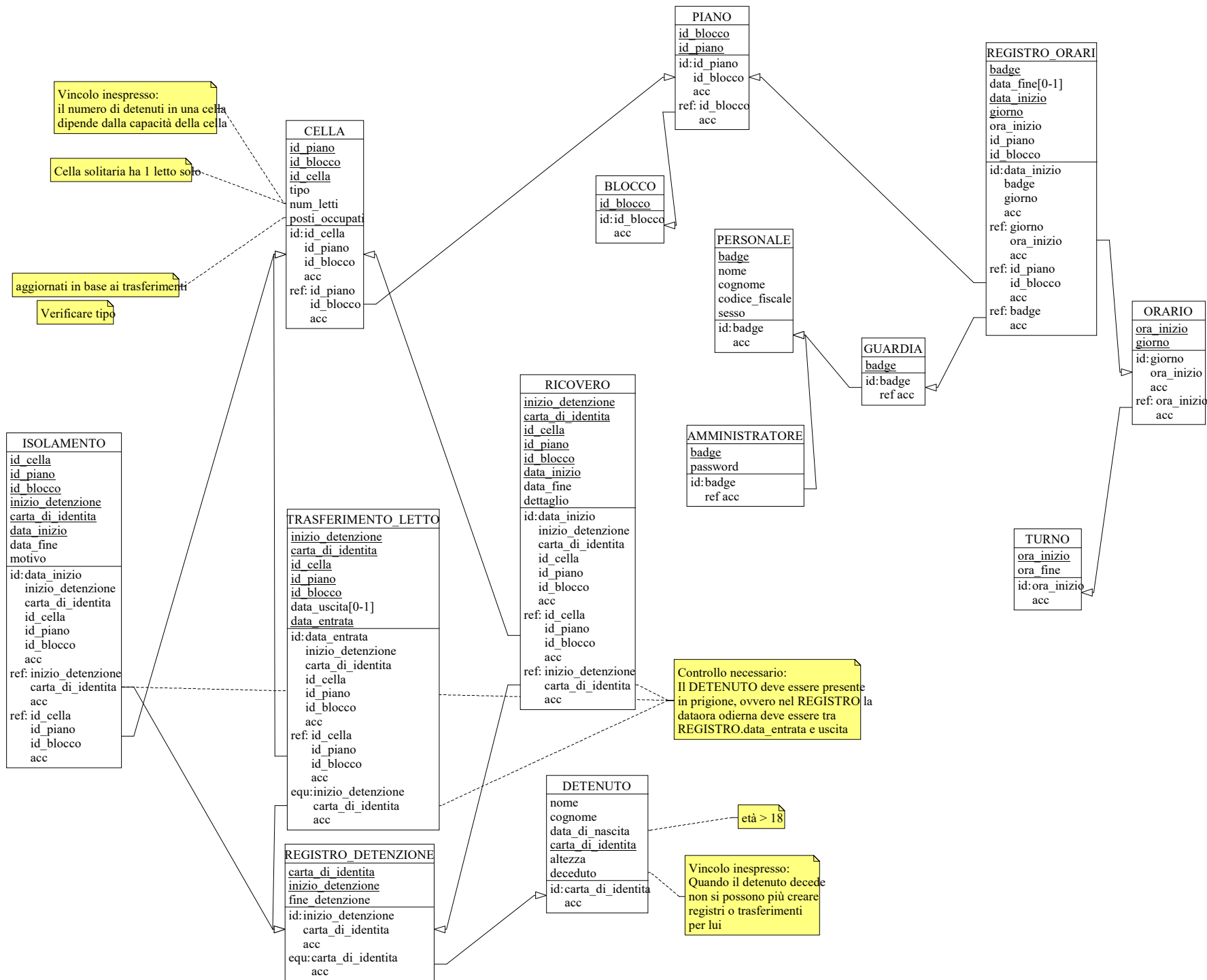
guardia(badge: personale)

registro_orari(data_inizio, badge: guardia, giorno: orario, data_fine*)

orario(giorno, ora_inizio: turno)

turno(ora_inizio, ora_fine)

3.7 Schema relazionale finale



3.8 Traduzione delle operazioni in query SQL

3.8.1 Inserimento nuovo detenuto

```
1 INSERT INTO detenuto (nome, cognome, data_di_nascita,  
  ↪ carta_di_identita, altezza)  
2 VALUES ($1, $2, $3, $4, $5);  
3 INSERT INTO registro_detenzione (carta_di_identita,  
  ↪ inizio_detenzione, fine_detenzione)  
4 VALUES ($1, $2, $3);  
5 INSERT INTO trasferimento_letto (data_entrata, id_blocco,  
  ↪ id_piano, id_cella, inizio_detenzione, carta_di_identita)  
6 VALUES ($1, $2, $3, $4, $5, $6);
```

Aggiornare i posti occupati

```
7 SELECT COUNT(data_entrata) as occupanti  
8 FROM trasferimento_letto  
9 WHERE id_blocco = $1 AND id_piano = $2 AND id_cella = $3 AND  
  ↪ data_uscita IS NULL;  
10  
11 UPDATE cella  
12 SET posti_occupati = $1  
13 WHERE id_blocco = $2 AND id_piano = $3 AND id_cella = $4;
```

3.8.2 Scambio posto letto con un altro detenuto

```
1 -- Imposto la data uscita a ora  
2 UPDATE trasferimento_letto t  
3 SET data_uscita = NOW()  
4 WHERE t.carta_di_identita = $1 AND data_uscita IS NULL;  
5  
6 INSERT INTO trasferimento_letto (data_entrata, id_blocco,  
  ↪ id_piano, id_cella, inizio_detenzione, carta_di_identita)  
7 VALUES (NOW(), $1, $2, $3, (SELECT inizio_detenzione  
8 FROM registro_detenzione r  
9 WHERE r.carta_di_identita = $4  
10 ORDER BY r.inizio_detenzione DESC LIMIT 1), $4);  
11 -- Aggiorno i posti occupati  
12 SELECT COUNT(data_entrata) as occupanti  
13 FROM trasferimento_letto
```

```

14 WHERE id_blocco = $1 AND id_piano = $2 AND id_cella = $3 AND
    ↪ data_uscita IS NULL;
15
16 UPDATE cella
17 SET posti_occupati = $1
18 WHERE id_blocco = $2 AND id_piano = $3 AND id_cella = $4;

```

3.8.3 Ricoverare un detenuto

```

1 INSERT INTO ricovero (id_blocco, id_piano, id_cella,
    ↪ inizio_detenzione, carta_di_identita, data_inizio,
    ↪ data_fine, dettaglio)
2 VALUES($1, $2, $3, $4, $5, NOW(), $6, $7);
3 -- Conto i posti occupati
4 SELECT COUNT(data_inizio) as occupanti
5 FROM medica
6 WHERE id_blocco = $1 AND id_piano = $2 AND id_cella = $3 AND
    ↪ NOW() BETWEEN data_inizio AND data_fine;
7
8 UPDATE cella
9 SET posti_occupati = $1
10 WHERE id_blocco = $2 AND id_piano = $3 AND id_cella = $4;

```

3.8.4 Isolare un detenuto

```

1 INSERT INTO isolamento (id_blocco, id_piano, id_cella,
    ↪ inizio_detenzione, carta_di_identita, data_inizio,
    ↪ data_fine, motivo)
2 VALUES($1, $2, $3, $4, $5, NOW(), $6, $7)
3 -- Conto i posti occupati
4 SELECT COUNT(data_inizio) as occupanti
5 FROM isolamento
6 WHERE id_blocco = $1 AND id_piano = $2 AND id_cella = $3 AND
    ↪ NOW() BETWEEN data_inizio AND data_fine;
7
8 UPDATE cella
9 SET posti_occupati = $1
10 WHERE id_blocco = $2 AND id_piano = $3 AND id_cella = $4;

```

3.8.5 Inserimento nuova guardia

```
1 INSERT INTO personale (badge, nome, cognome, codice_fiscale,  
  ↪ sesso)  
2 VALUES ($1, $2, $3, $4, $5);  
3 INSERT INTO guardia (badge)  
4 VALUES ($1);
```

3.8.6 Inserimento nuovo amministratore

```
1 INSERT INTO personale (badge, nome, cognome, codice_fiscale,  
  ↪ sesso)  
2 VALUES ($1, $2, $3, $4, $5);  
3 INSERT INTO amministratore (badge, password)  
4 VALUES ($1, $2);
```

3.8.7 Visualizzare il personale (amministratori, guardie)

```
1 SELECT p.badge as "Badge", p.nome as "Nome", p.cognome as  
  ↪ "Cognome", p.codice_fiscale as "Codice fiscale", p.sesso  
  ↪ as "Sesso",  
2 (SELECT COUNT(*) AS num  
3 FROM guardia g  
4 WHERE p.badge = g.badge) AS "Guardia",  
5 (SELECT COUNT(*) AS num  
6 FROM amministratore a  
7 WHERE p.badge = a.badge) AS "Amministratore"  
8 FROM personale p;
```

3.8.8 Visualizzare la lista dei detenuti presenti

```
1 SELECT d.carta_di_identita AS "CDI", TRIM(d.nome) AS "Nome",  
  ↪ TRIM(d.cognome) as "Cognome", r.inizio_detenzione AS  
  ↪ "Inizio", r.fine_detenzione AS "Fine", CONCAT(t.id_blocco,  
  ↪ t.id_piano, '-', t.id_cella) AS "Cella"  
2 FROM registro_detenzione r  
3 JOIN detenuto d ON r.carta_di_identita = d.carta_di_identita  
4 JOIN trasferimento_letto t ON r.inizio_detenzione =  
  ↪ t.inizio_detenzione AND r.carta_di_identita =  
  ↪ t.carta_di_identita
```

```

5 WHERE NOW() BETWEEN r.inizio_detenzione AND r.fine_detenzione
   ↪ AND d.deceduto IS NOT TRUE AND t.data_uscita IS NULL
6 ORDER BY t.data_entrata DESC;

```

3.8.9 Visualizzare l'andamento settimanale di nuovi detenuti

```

1 SELECT inizio_detenzione AS "date", COUNT(carta_di_identita) AS
   ↪ "numero"
2 FROM registro_detenzione
3 WHERE inizio_detenzione BETWEEN DATE(NOW()) - 7 AND DATE(NOW())
4 GROUP BY inizio_detenzione
5 ORDER BY inizio_detenzione ASC

```

3.8.10 Visualizzare i primi cinque detenuti che sono stati trasferiti in celle solitarie più volte

```

1 SELECT d.carta_di_identita AS "CDI", TRIM(d.nome) AS "Nome",
   ↪ TRIM(d.cognome) AS "Cognome", COUNT(i.carta_di_identita) AS
   ↪ num
2 FROM isolamento i
3 JOIN detenuto d ON i.carta_di_identita = d.carta_di_identita
4 GROUP BY i.carta_di_identita
5 ORDER BY num DESC
6 LIMIT 5

```

3.8.11 Inserimento nuovo orario

```

1 INSERT INTO registro_orari (badge, data_inizio, data_fine,
   ↪ giorno, ora_inizio, id_blocco, id_piano)
2 VALUES ($1, $2, $3, $4, $5, $6, $7);

```

3.9 Progettazione dell'applicazione

Figura 3.1: Disegno della pagina detenuto e le finestre di ricovero e trasferimento

The figure displays three UI mockups for a detention management application.

Top Left: Transfer Form

Mario Rossi

Da: Cella B1 A: Cella L4

Con: Pietro L.

Conferma Cancella

Top Right: Hospitalization Form

Mario Rossi

A: Med C5 Data inizio: 2024-08-03 Data fine: 2024-09-20

Prognosi: Il paziente ha...

Conferma Cancella

Bottom: Main Dashboard

DB-Prigione

Home

Detenuti

Guardie

Nuovo Elimina Trasferisci Ricovera

Blocco Visualizza:Rientrati Isolamento

Nome	Cognome	Inizio	Fine	Cella	Deceduto
------	---------	--------	------	-------	----------

L'applicazione è stata creata utilizzando il framework NextJS, che ha consentito di produrre una web-app full stack. Per connettersi al database ed effettuare le query si è utilizzato il package chiamato 'node-postgres'. Si è fatto anche uso di framework per il frontend come TailwindCSS per il layout e lo stile e Mui X charts per i grafici. In alcuni casi è stata utilizzata anche la libreria 'zod' per validare l'input dell'utente attraverso i form. Il linguaggio utilizzato nel codice è TypeScript, versione tipizzata di JavaScript

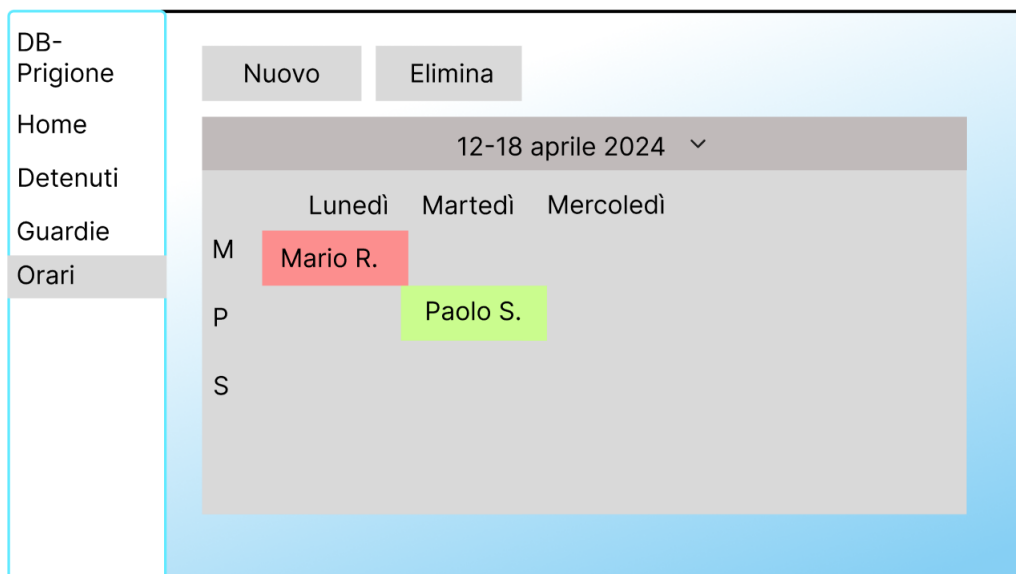


Figura 3.2: Disegno della pagina orario

3.9.1 Screenshot interfaccia utente