

UNIVERSITÀ DEGLI STUDI DI NAPOLI “PARTHENOPE”  
FACOLTÀ DI SCIENZE E TECNOLOGIE  
CORSO DI LAUREA IN INFORMATICA (PERCORSO GENERALE)



RELAZIONE PROGETTO DI BASI DI DATI  
Clinica Riabilitativa

DOCENTE  
Antonio Maratea

CANDIDATI  
Luigi Borchia  
0124/001259  
Luca Esposito  
0124/001698  
Marco Sautto  
0124/001737

Anno Accademico 2018-2019

# Indice

<b>1</b>	<b>Progettazione</b>	<b>1</b>
1.1	Sintesi dei Requisiti . . . . .	1
1.2	Diagrammi . . . . .	2
1.3	Utenti e Categorie . . . . .	2
1.4	Use Case . . . . .	2
1.5	Volumi . . . . .	7
1.6	Vincoli di Integrità . . . . .	8
1.7	Verifica di Normalità . . . . .	11
<b>2</b>	<b>Implementazione</b>	<b>13</b>
2.1	Creazione Utenti . . . . .	13
2.2	Gestione dei Permessi . . . . .	14
2.3	Data Definition Language . . . . .	16
2.4	Data Manipulation Language . . . . .	27
2.5	Trigger . . . . .	36
2.6	Procedure . . . . .	46
2.7	Funzioni . . . . .	56
2.8	Views . . . . .	57
2.9	Scheduler . . . . .	58

# Traccia 1

## Progettazione

Vengono descritti in questa sezione gli aspetti progettuali dell'elaborato che hanno preceduto l'implementazione vera e propria.

### 1.1 Sintesi dei Requisiti

Il database deve gestire i pazienti che effettuano un ricovero all'interno di una clinica riabilitativa. Tali pazienti vengono seguiti dalla loro accettazione in clinica fino alla loro dimissione, salvando i dati del ricovero nella relativa Cartella Clinica, con i relativi fogli che la compongono, e Cartella Infermieristica, anch'essa con i relativi fogli che la compongono. Tali dati devono essere inseriti ed aggiornati dagli addetti alla gestione del sistema informatico della clinica. Ogni persona che si trova nell'ambito della clinica è registrata con le sue generalità, per poi essere specializzata in medico, infermiere, fisioterapista o paziente. Quest'ultimo ha come unico attributo (al di fuori di quelli già associati dalla specializzazione come Persona) il numero di cellulare di un familiare. La Cartella Infermieristica associata al paziente può essere scritta da più infermieri ma non può essere scritta da un medico; più medici possono leggere la Cartella Infermieristica. Viceversa, la Cartella Clinica può essere scritta da più medici ma non può essere scritta da un infermiere; più infermieri possono leggere la Cartella Clinica. Si tiene traccia delle visite che ogni medico effettua ad un determinato paziente, gli interventi a cui egli è sottoposto, nonché la dieta da seguire durante il suo ricovero, fino alla dimissione. Si ricordi che un paziente che torna in clinica avrà una nuova Cartella Clinica ed una nuova Cartella Infermieristica, ma si terrà conto delle cartelle del precedente ricovero.

## 1.2 Diagrammi

Per una corretta lettura dei diagrammi EE/R e Relazionale, questi sono stati stampati su foglio A3 e allegati alla relazione.

## 1.3 Utenti e Categorie

Ad ogni paziente viene assegnata una Cartella Clinica scritta da uno o più medici e soltanto leggibile da uno o più infermieri; inoltre ogni paziente ha una Cartella Infermieristica scritta da uno o più infermieri e soltanto leggibile da uno o più medici. I medici possono creare diete per i pazienti, effettuare visite o interventi; a questi ultimi, gli infermieri possono solo assistere.

UTENTE	TIPO	VOLUME	PERMESSI
DB Clinica	Amministratore	1	ALL
Medico	Comune	M	
Infermiere	Comune	N	
Fisioterapista	Comune	J	

## 1.4 Use Case

La maggior parte delle operazioni vengono effettuate dal medico, dall'infermiere e dal fisioterapista, in quanto sono responsabili della buona riuscita del ricovero del paziente. Quest'ultimo è unicamente in grado di visualizzare i propri interventi, i propri esami e le proprie analisi.

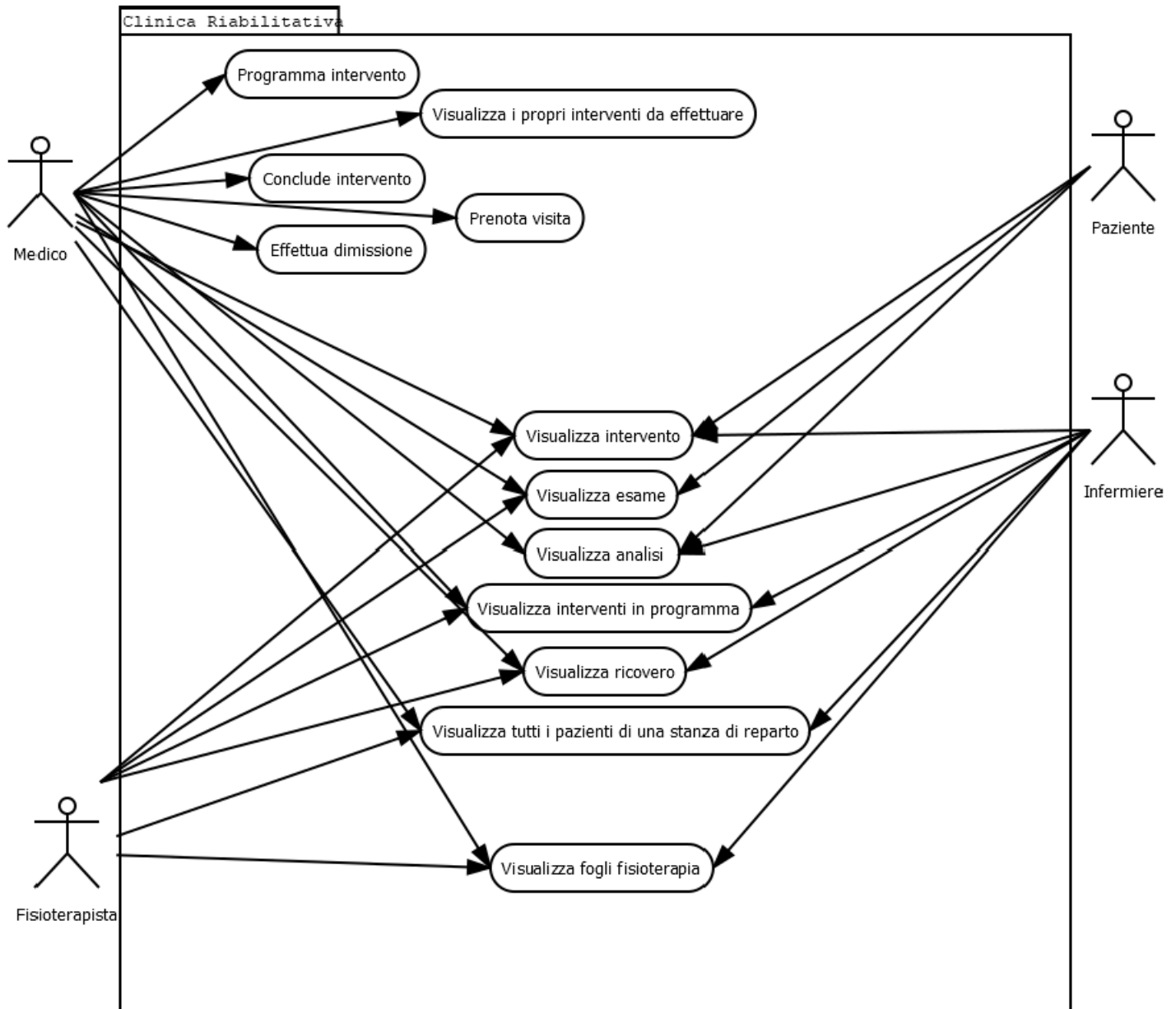


Figura 1.1: Diagramma use case

Sono riportate le schede descrittive delle principali operazioni degli utenti del DB. Si tenga conto che le visualizzazioni di intervento, analisi ed esami di un paziente dovrebbero essere diversi da quelli che possono essere visualizzati da un paziente stesso e dai medici, infermieri e fisioterapisti. Tuttavia, un medico deve essere in grado di visualizzare gli interventi di un paziente dato il suo CF

ed un paziente deve essere in grado di visualizzare i suoi interventi inserendo il suo CF. Il DB, ad ogni modo, non è in grado di riconoscere se il CF inserito dal paziente è un CF dell'utente-paziente o di un qualsiasi altro paziente, dunque se è in grado di ottenere un altro CF, potrà visualizzare gli interventi del paziente con l'ulteriore CF.

<b>OPERAZIONE</b>	Programma_Intervento
<b>SCOPO</b>	Programma un intervento
<b>ARGOMENTI</b>	CF Medico, CF Paziente, Sala Operatoria, Tipo intervento, Durata, Data Prevista
<b>RISULTATO</b>	Aggiorna intervento / errore
<b>ERRORI</b>	<ul style="list-style-type: none"> <li>-Medico e paziente sono la stessa persona</li> <li>-La data d'intervento precede quella di ricovero</li> <li>-La sala op. ha raggiunto il limite massimo giornaliero di operazioni</li> <li>-Il medico ha raggiunto il limite massimo di interventi giornalieri</li> <li>-Il paziente risulta dimesso</li> <li>-Non sono disponibili ulteriori medici per l'intervento</li> </ul>
<b>USA</b>	Intervento, Persona, Paziente, Medico, Infermiere, Ricovero, Dimissione, Effettua, Assiste
<b>MODIFICA</b>	Intervento, Ricovero, Effettua, Assiste
<b>PRIMA</b>	Il paziente non ha un intervento nei 15 giorni precedenti alla data d'intervento inserita
<b>POI</b>	I medici e gli infermieri non possono effettuare altri interventi finchè non è concluso

<b>OPERAZIONE</b>	Prenota_Visita
<b>SCOPO</b>	Effettuare la prenotazione di una visita per un paziente
<b>ARGOMENTI</b>	CF Paziente, Tipo visita
<b>RISULTATO</b>	Aggiunta visita da effettuare / Errore
<b>ERRORI</b>	-Nessun medico disponibile per effettuare la visita
<b>USA</b>	Visita, Medico, Effettua, Intervento
<b>MODIFICA</b>	Visita
<b>PRIMA</b>	Il paziente può avere o meno delle visite già prenotate
<b>POI</b>	Il paziente ha una visita prenotata in più ed il medico ha tale visita da effettuare nella data specificata.

<b>OPERAZIONE</b>	Visualizza_Interventi_Programma_Medico
<b>SCOPO</b>	Visualizza gli interventi da effettuare da un medico nello specifico
<b>ARGOMENTI</b>	CF Medico
<b>RISULTATO</b>	Visualizza interventi
<b>ERRORI</b>	
<b>USA</b>	Intervento, Effettua, E_Tenuto
<b>MODIFICA</b>	
<b>PRIMA</b>	
<b>POI</b>	

<b>OPERAZIONE</b>	Concludi_Intervento
<b>SCOPO</b>	Segnare un intervento come concluso
<b>ARGOMENTI</b>	ID Intervento, Data Effettiva, Durata
<b>RISULTATO</b>	Conclusio intervento / Errore
<b>ERRORI</b>	-Intervento non esistente -La data effettiva precede la data prevista
<b>USA</b>	Intervento
<b>MODIFICA</b>	Intervento
<b>PRIMA</b>	L'intervento non è ancora concluso, quindi non ha una durata effettiva
<b>POI</b>	L'intervento è concluso e si ha una durata effettiva

<b>OPERAZIONE</b>	Visualizza_Esame_Paziente
<b>SCOPO</b>	Visualizza gli esami di un paziente
<b>ARGOMENTI</b>	CF Paziente
<b>RISULTATO</b>	Visualizza esame / Errore
<b>ERRORI</b>	- Il paziente non ha esami
<b>USA</b>	Esami, Cartella Clinica, Ricovero
<b>MODIFICA</b>	
<b>PRIMA</b>	
<b>POI</b>	

<b>OPERAZIONE</b>	Effettua_Dimissione
<b>SCOPO</b>	Effettua la dimissione di un paziente
<b>ARGOMENTI</b>	ID Ricovero, Codice Diagnosi, Modalità Dimissione
<b>RISULTATO</b>	Aggiunta dimissione / Errore
<b>ERRORI</b>	- Il paziente non ha effettuato esami - Il paziente è già stato dimesso
<b>USA</b>	Dimissione, Cartella Clinica
<b>MODIFICA</b>	Dimissione
<b>PRIMA</b>	Il paziente non è stato dimesso, quindi è ancora in clinica
<b>POI</b>	Il paziente è dimesso

<b>OPERAZIONE</b>	Visualizza_Interventi_Programma
<b>SCOPO</b>	Visualizzare tutti gli interventi non ancora effettuati
<b>ARGOMENTI</b>	
<b>RISULTATO</b>	Visualizzazione degli interventi programmati
<b>ERRORI</b>	
<b>USA</b>	Intervento
<b>MODIFICA</b>	
<b>PRIMA</b>	
<b>POI</b>	

<b>OPERAZIONE</b>	Visualizza_Interventi_Paziente
<b>SCOPO</b>	Visualizza gli interventi di un paziente
<b>ARGOMENTI</b>	CF Paziente
<b>RISULTATO</b>	Visualizza intervento / Errore
<b>ERRORI</b>	-Il paziente non ha interventi
<b>USA</b>	Intervento
<b>MODIFICA</b>	
<b>PRIMA</b>	
<b>POI</b>	

<b>OPERAZIONE</b>	Visualizza_Analisi_Paziente
<b>SCOPO</b>	Visualizza le analisi di un paziente
<b>ARGOMENTI</b>	CF Paziente
<b>RISULTATO</b>	Visualizza analisi / Errore
<b>ERRORI</b>	-Il paziente non ha analisi effettuate
<b>USA</b>	Analisi, Cartella Clinica, Ricovero
<b>MODIFICA</b>	
<b>PRIMA</b>	
<b>POI</b>	



<b>OPERAZIONE</b>	Visualizza_Ricovero
<b>SCOPO</b>	Visualizza i dati dei ricoveri dell'ultima settimana
<b>ARGOMENTI</b>	
<b>RISULTATO</b>	CF, nome, cognome e reparto dei pazienti
<b>ERRORI</b>	
<b>USA</b>	Ricovero, Persona
<b>MODIFICA</b>	
<b>PRIMA</b>	
<b>POI</b>	

<b>OPERAZIONE</b>	Visualizza_Pazienti_Reparto
<b>SCOPO</b>	Visualizza i dati dei pazienti ricoverati in una data stanza di reparto
<b>ARGOMENTI</b>	Numero stanza, Nome reparto
<b>RISULTATO</b>	CF, nome, cognome ed età dei pazienti in una stanza di reparto / Errore
<b>ERRORI</b>	- Non ci sono pazienti nella stanza di reparto
<b>USA</b>	Reparto, E_Tenuto, Ricovero, Persona
<b>MODIFICA</b>	
<b>PRIMA</b>	
<b>POI</b>	

<b>OPERAZIONE</b>	Visualizza_Fogli_Terapia
<b>SCOPO</b>	Visualizza i fogli terapia di tutti i pazienti non ancora dimessi
<b>ARGOMENTI</b>	
<b>RISULTATO</b>	Fogli terapia dei pazienti ancora in clinica
<b>ERRORI</b>	
<b>USA</b>	Foglio_Fisioterapia, Cartella_Clinica, Ricovero, Dimissione
<b>MODIFICA</b>	
<b>PRIMA</b>	
<b>POI</b>	

## 1.5 Volumi

La tabella dei volumi indica approssimativamente il size di ogni tabella, quindi la quantità di ogni entità, all'interno del DB una volta a regime.

TABELLA	TIPO	VOLUME	INCREMENTO	PERIODO
Persona	E	3510	613	Anno
Medico	E	200	2	Anno
Infermiere	E	300	10	Anno
Fisioterapista	E	10	1	Anno
Paziente	E	3000	50	Mese
Ha_I	A	10	5	Mese
Assiste	A	16	1	Mese
Effettua	A	25	5	Mese
Visita	E	10	15	Giorno
Ha_M	A	15	5	Mese
Cartella Inf.	E	3000	50	Mese
Intervento	E	5000	10	Mese
Dieta	E	2500	50	Mese
Cartella Clin.	E	3000	50	Mese
Foglio_Terapia	E	3000	50	Mese
Diario Inf.	ED	3000	50	Mese
Contiene	A	6000	0	Mese
Farmaco	E	5000	12000	Anno
Sintomatologia_DI	ED	12000	4.380.000	Anno
Ricovero	E	4500	55	Mese
E_Tenuto	A	225	55	Mese
Reparto	E	20	0	Anno
Dimissione	E	2990	50	Mese
Esami	ED	5000	100	Mese
Analisi	ED	5500	200	Mese
Diario Clin.	ED	3000	55	Mese
Foglio Fisioterapia	ED	3000	55	Mese
Sintomatologia_DC	ED	3000	1.095.000	Anno

## 1.6 Vincoli di Integrità

Sono detti statici i vincoli che limitano i valori assumibili da alcuni attributi indipendentemente dal tempo, mentre sono detti dinamici quelli che riguardano valori che cambiano nel tempo; anche molte regole di business si presentano come vincoli dinamici e la differenza con questi ultimi è molto sottile. Si riportano i vincoli più importanti per il DB\_Clinica. Non compaiono nell'elenco, in quanto ovvi, i vincoli di chiave primaria e chiave esterna.

**Statici**

- Per ogni Persona è necessario avere Codice Fiscale, Nome, Cognome, Data di Nascita, Sesso (che potrà essere M o F).
- Per ogni Infermiere è necessario avere l'iscrizione all'Albo.
- Per ogni Medico è necessario avere l'iscrizione all'Albo e la Specializzazione (che potrà essere Medico Generale, Medicina Interna, Cardiologia, Cardiochirurgia, Chirurgia Generale, Neurochirurgia, Neurologia, Geriatria, Fisiatria, Dietologia, Ortopedia e traumatologia).
- Per ogni Ricovero è necessario avere un Codice Diagnosi (che andrà da 101 a 110) ed indicherà la diagnosi in ingresso del paziente.
- Per ogni Foglio Terapia è necessario avere un Metodo di Somministrazione (che potrà essere Orale, Rettale, Endovena, Inalatoria, Transcutanea, Intravascolare, Intramuscolare).
- Per ogni Farmaco è necessario avere il Tipo (che sarà Antimicrobico, Catartico, Emetocatartico, Oressizzante, Stomachico).
- Per ogni Sintomatologia del Diario Infermieristico, verrà tenuta traccia del Sintomo nel tempo (che potrà essere Gonfiore, Febbre, Tosse, Vomito, Diarrea, Infiammazione, Dolori addominali, Nausea, Stanchezza, Stitichezza).
- Per ogni Intervento è necessario avere il Numero della Sala Operatoria (che potrà andare da 10 a 15) e il Tipo di Intervento (che potrà essere Amputazione, Anastomosi, Artrodesi, Artroplastica, Aspirazione, Asportazione, Cardioplegia, Cardiotomia, Cateterismo, Cistectomia, Drenaggio, Gastrotomia).
- Per ogni Reparto è necessario avere il Numero Piano (che andrà da 0 a 5), il Numero dei posti in ogni stanza (che saranno 5), il Numero di Stanze (che saranno 15 per ogni piano) e il Nome del Reparto (che potrà essere Cardiologia, Anatomia e Istologia patologica, Anestesia e Rianimazione, Chirurgia, Oncologia, Gastroenterologia, Medicina Generale, Ortopedia e Traumatologia, Radiologia, Diabetologia, Geriatria, Medicina Interna, Neurologia, Pediatria, Urologia, Psichiatria).

- Per ogni Dimissione è necessario avere la Data e Ora, il Codice Diagnosi in uscita (che avrà valore da 100 a 110) e la Modalità di Dimissione (che avrà valore da 0 a 10 dove 0 indica completamente autonomo e 10 completamente dipendente).
- Per ogni Dieta è necessario avere Tipo di dieta (che potrà essere Normoenergetica, Lattea integrata, Liquida, Semiliquida, Ipocalorica, Ipoglicidica IDDM, Ipoglicidica NIDDM, Lipolipidica, Ipoproteica, Ipopurinica, Ipernutritiva, Astringente), Kcal, Proteine e Grassi.
- Per ogni Visita è necessario avere il Tipo di Visita (che potrà essere Cardiologica, Dietologica, Fisiatrica, Geriatrica, Neurologica, Generica).
- Per ogni Cartella Clinica è necessario avere il Sintomo di ingresso del paziente (che potrà essere Gonfiore, Febbre, Tosse, Vomito, Diarrea, Infiammazione, Dolori addominali, Nausea, Stanchezza, Stitichezza).
- Per ogni Esame si avrà un Esito (che potrà essere 0, ossia positivo, 1, ossia negativo) e il Tipo di Esame (che potrà essere Radiografia, Risonanza Magnetica, TAC, Ecografia, PET).
- Per ogni Sintomatologia della Cartella Clinica viene tenuto traccia del Sintomo in relazione al giro medico effettuato ogni mattina. Il Sintomo potrà essere Gonfiore, Febbre, Tosse, Vomito, Diarrea, Infiammazione, Dolori addominali, Nausea, Stanchezza, Stitichezza.
- Per ogni Foglio Fisioterapia è necessario avere Rinforzo Muscolare, Elettrostimolazione Muscolare e Deambulazione. Ognuno potrà avere da 0 a 10, dove 0 indica che il paziente non collabora o non è autosufficiente e 10 indica che collabora o è autosufficiente.

### **Dinamici**

- Ogni intervento deve avvenire a distanza di 15 giorni per ogni paziente.
- La data di dimissione di un paziente non può essere precedente alla data di ricovero.
- In una sala operatoria è possibile effettuare massimo 3 interventi al giorno.

- Gli interventi in ogni sala operatoria non si possono accavallare e solo nella sala 14 si fanno interventi superiori alle 6 ore.
- Un medico può partecipare al massimo a 10 interventi in una settimana e di al più tre tipi diversi.
- Un infermiere non può effettuare un intervento.
- Un medico non può assistere ad un intervento.
- La durata degli interventi per i pazienti di età superiore a 75 anni non può essere superiore a tre ore.
- Un paziente non può essere tenuto in due reparti contemporaneamente.
- Il fisioterapista non può assistere interventi.
- Il fisioterapista non può effettuare interventi.
- Un medico non può effettuare visite nei giorni in cui opera e gli interventi hanno la priorità sulle visite. Se a un medico viene assegnato un intervento e nella stessa data quel medico ha una visita, la visita viene posticipata al giorno dopo.

## 1.7 Verifica di Normalità

La verifica di normalità viene effettuata con lo scopo di eliminare la ridondanza. Teniamo in considerazione soltanto le prime tre forme, di cui la terza è composta dalla forma BCNF (Boyce-Codd Normal Form) e dalla Terza Forma. Dopo aver effettuato le opportune verifiche, si è notato che lo schema si ritrova in Terza Forma e di seguito spiegheremo le motivazioni di tale considerazione.

- **Prima Forma**

Lo schema è in prima forma per definizione, in quanto nel Flat Relational Model ogni attributo non atomico viene tradotto in un attributo atomico singolo, ossia ogni attributo singolo che compone l'attributo non atomico è un attributo a se stante;

- **Seconda Forma**

Nello schema costruito, non sono presenti chiavi multi-attributo, quindi ogni attributo è dipendente da una chiave che è attributo atomico. Seppur nel Diagramma EE/R vi sono più chiavi deboli come data ed ora, nel Diagramma Relazionale vengono riportare come un'unica chiave debole.

- **BCNF (Boyce-Codd Normal Form)**

Il caso più particolare di dipendenza funzionale che comunque rispetta le proprietà della forma BCNF è quella che si ha all'interno della tabella "Intervento", riportata di seguito:

INTERVENTO	
+ID_Intervento:	CHAR(10) = <PK>
+Data_prevista:	DATE
+Data_effettiva:	DATE
+Tipo:	VARCHAR
+Durata_Prevista:	NUMBER
+Durata_Effettiva:	NUMBER
+Num_Sala_Operatoria:	NUMBER
+ID_Ricovero:	CHAR(10) = <FK>
+CF_PAZ:	CHAR(16) = <FK>

Nel modo in cui è stato studiato ed ideato tale DB, si ha che ogni intervento è definito univocamente da un ID\_Intervento, generato durante la programmazione dell'intervento. Seppur ID\_Intervento sia la chiave primaria, la coppia (Data\_prevista, Num\_Sala\_Operatoria) è in grado di determinare funzionalmente l>ID\_Intervento tale che

$$(Data\_prevista, Num\_Sala\_Operatoria) \rightarrow ID\_Intervento$$

Siccome il DB è stato strutturato in maniera tale da ottenere un solo intervento in una data sala operatoria in una specifica data, l'insieme della data prevista e del numero della sala operatoria rappresentano una superchiave, in quanto tale coppia è in grado di indicare univocamente una tupla. Essendo dunque una dipendenza da chiave all'interno della teoria della normalizzazione, possiamo concludere che lo schema è in forma BCNF.

# Traccia 2

## Implementazione

Per l'implementazione è stato utilizzato il DBMS Oracle 18c XE per eseguire il codice PL/SQL.

### 2.1 Creazione Utenti

La creazione degli utenti di tale database si basa su un utente amministratore, a cui sono garantiti tutti i permessi, e gli utenti medico, infermiere, fisioterapista e paziente.

```
CREATE USER DB_ADMIN IDENTIFIED BY admin;
GRANT ALL PRIVILEGES TO DB_ADMIN;

CREATE USER MEDICO IDENTIFIED BY medico;
CREATE USER INFERMIERE IDENTIFIED BY infermiere;
CREATE USER FISIOTERAPISTA IDENTIFIED BY fisioterapista;
CREATE USER PAZIENTE IDENTIFIED BY paziente;

CREATE ROLE MED_ROLE;
CREATE ROLE INF_ROLE;
CREATE ROLE FISIO_ROLE;
CREATE ROLE PAZ_ROLE;

GRANT MED_ROLE TO MEDICO;
GRANT PAZ_ROLE TO MEDICO;

GRANT INF_ROLE TO INFERMIERE;
GRANT PAZ_ROLE TO INFERMIERE;

GRANT FISIO_ROLE TO FISIOTERAPISTA;
GRANT PAZ_ROLE TO FISIOTERAPISTA;
GRANT PAZ_ROLE TO PAZIENTE;
```

## 2.2 Gestione dei Permessi

Di seguito vengono riportati i permessi garantiti agli utenti sulle tabelle del DB.

### Medico:

```
GRANT CONNECT, CREATE SESSION TO MED_ROLE;

GRANT SELECT,INSERT,UPDATE ON cartella_clinica TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE ON ricovero TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE ON ha_M TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE ON e_tenuto TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE ON dimissione TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE ON diario_clinico TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE ON sintomatologia_cc TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE ON intervento TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE ON foglio_terapia TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE ON contiene TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE ON farmaco TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE ON effettua TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE,DELETE ON visita TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE,DELETE ON dieta TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE,DELETE ON esami TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE,DELETE ON analisi TO MED_ROLE;
GRANT SELECT,INSERT,UPDATE,DELETE ON assiste TO MED_ROLE;

GRANT SELECT ON foglio_fisioterapia TO MED_ROLE;
GRANT SELECT ON reparto TO MED_ROLE;
GRANT SELECT ON cartella_infermieristica TO MED_ROLE;
GRANT SELECT ON diario_infermieristico TO MED_ROLE;
GRANT SELECT ON sintomatologia_di TO MED_ROLE;

GRANT EXECUTE ON PROGRAMMAZIONE_INTERVENTO TO MED_ROLE;
GRANT EXECUTE ON PRENOTA_VISITA TO MED_ROLE;
GRANT EXECUTE ON VISUALIZZA_PAZIENTE TO MED_ROLE;
GRANT EXECUTE ON VIS_PAZ_STANZ_REP TO MED_ROLE;
GRANT EXECUTE ON VIS_FOGLIO_FISIO_E_ESAMI_PAZ TO MED_ROLE;
GRANT EXECUTE ON TRY_DIM TO MED_ROLE;
GRANT EXECUTE ON PROG_INT_ANALISI TO MED_ROLE;
GRANT EXECUTE ON INTERVENTI_DA_FARE TO MED_ROLE;
GRANT EXECUTE ON END_INT TO MED_ROLE;
GRANT EXECUTE ON NUM_INT_EFF_PAZ TO MED_ROLE;
GRANT EXECUTE ON CF_PAZ_DA_NOME TO MED_ROLE;
```

### Infermiere:

```
GRANT CONNECT, CREATE SESSION TO INF_ROLE;
```



```

--INF_ROLE
GRANT SELECT,INSERT ON ha_I TO INF_ROLE;
GRANT SELECT,INSERT ON CARTELLA_INFERMIERISTICA TO INF_ROLE;

GRANT SELECT,INSERT,DELETE ON FOGLIO_TERAPIA TO INF_ROLE;

GRANT SELECT,INSERT,DELETE,UPDATE ON DIARIO_INFERMIERISTICO TO INF_ROLE;
GRANT SELECT,INSERT,DELETE,UPDATE ON sintomatologia_di TO INF_ROLE;

GRANT SELECT ON contiene TO INF_ROLE;
GRANT SELECT ON FARMACO TO INF_ROLE;
GRANT SELECT ON INTERVENTO TO INF_ROLE;
GRANT SELECT ON assiste TO INF_ROLE;
GRANT SELECT ON RICOVERO TO INF_ROLE;
GRANT SELECT ON REPARTO TO INF_ROLE;
GRANT SELECT ON e_tenuto TO INF_ROLE;
GRANT SELECT ON DIMISSIONE TO INF_ROLE;
GRANT SELECT ON effettua TO INF_ROLE;
GRANT SELECT ON DIETA TO INF_ROLE;
GRANT SELECT ON VISITA TO INF_ROLE;
GRANT SELECT ON CARTELLA_CLINICA TO INF_ROLE;
GRANT SELECT ON ESAMI TO INF_ROLE;
GRANT SELECT ON ANALISI TO INF_ROLE;
GRANT SELECT ON DIARIO_CLINICO TO INF_ROLE;
GRANT SELECT ON sintomatologia_cc TO INF_ROLE;
GRANT SELECT ON FOGLIO_FISIOTERAPIA TO INF_ROLE;

GRANT EXECUTE ON VISUALIZZA_PAZIENTE TO INF_ROLE;
GRANT EXECUTE ON VIS_PAZ_STANZ_REP TO INF_ROLE;
GRANT EXECUTE ON VIS_FOGLIO_FISIO_E_ESAMI_PAZ TO INF_ROLE;
GRANT EXECUTE ON NUM_INT_EFF_PAZ TO INF_ROLE;
GRANT EXECUTE ON CF_PAZ_DA_NOME TO INF_ROLE;

```

#### Fisioterapista:

```

GRANT CONNECT, CREATE SESSION TO FISIO_ROLE;

--FISIO_ROLE
GRANT SELECT ON ha_I TO FISIO_ROLE;
GRANT SELECT ON CARTELLA_INFERMIERISTICA TO FISIO_ROLE;
GRANT SELECT ON FOGLIO_TERAPIA TO FISIO_ROLE;
GRANT SELECT ON contiene TO FISIO_ROLE;
GRANT SELECT ON FARMACO TO FISIO_ROLE;
GRANT SELECT ON DIARIO_INFERMIERISTICO TO FISIO_ROLE; GRANT SELECT ON
    sintomatologia_di TO FISIO_ROLE;
GRANT SELECT ON INTERVENTO TO FISIO_ROLE;
GRANT SELECT ON assiste TO FISIO_ROLE;
GRANT SELECT ON RICOVERO TO FISIO_ROLE;
GRANT SELECT ON REPARTO TO FISIO_ROLE;
GRANT SELECT ON e_tenuto TO FISIO_ROLE;
GRANT SELECT ON DIMISSIONE TO FISIO_ROLE;
GRANT SELECT ON effettua TO FISIO_ROLE;
GRANT SELECT ON DIETA TO FISIO_ROLE;
GRANT SELECT ON VISITA TO FISIO_ROLE;
GRANT SELECT ON ha_M TO FISIO_ROLE;

```

```

GRANT SELECT ON ESAMI TO FISIO_ROLE;
GRANT SELECT ON ANALISI TO FISIO_ROLE;
GRANT SELECT ON DIARIO_CLINICO TO FISIO_ROLE;
GRANT SELECT ON sintomatologia_cc TO FISIO_ROLE;

GRANT SELECT,INSERT ON CARTELLA_CLINICA TO FISIO_ROLE;
GRANT SELECT,INSERT,UPDATE ON FOGLIO_FISIOTERAPIA TO FISIO_ROLE;

GRANT EXECUTE ON VISUALIZZA_PAZIENTE TO FISIO_ROLE;
GRANT EXECUTE ON VIS_PAZ_STANZ_REP TO FISIO_ROLE;
GRANT EXECUTE ON VIS_FOGLIO_FISIO_E_ESAMI_PAZ TO FISIO_ROLE;
GRANT EXECUTE ON NUM_INT_EFF_PAZ TO FISIO_ROLE;
GRANT EXECUTE ON CF_PAZ_DA_NOME TO FISIO_ROLE;

```

#### Paziente:

```

PAZIENTE GRANT CONNECT, CREATE SESSION TO PAZ_ROLE;

--PAZ_ROLE
GRANT EXECUTE ON VISUALIZZA_INT_PAZ TO paziente;
GRANT EXECUTE ON VIS_ANALISI_PAZ TO paziente;
GRANT EXECUTE ON VIS_ESAMI_PAZ TO paziente;

```

## 2.3 Data Definition Language

Verranno di seguito definite le tabelle create con i vincoli di integrità che sono stati ritenuti utili implementare. Per differenziare le entità dalle relazioni del diagramma EE/R illustrato, le prime sono identificate da un nome in maiuscolo all'interno del codice, mentre le relazioni sono identificate da nome in minuscolo. Si noti che, nonostante i campi "Data" e "Ora" siano due attributi separati per le entità, in Oracle vengono visti come un unico tipo DATE, in cui viene specificata la data nel formato dd/mm/aaaa e l'ora nel formato hh/mm (vengono trascurati i secondi) nel formato 24H.

#### Persona

La tabella Persona rappresenta l'entità che viene specializzata successivamente in Medico, Infermiere, Fisioterapista e Paziente e viene popolata dall'amministratore del DB.

```

CREATE TABLE PERSONA(
CF                CHAR(16) PRIMARY KEY,
VIA                VARCHAR2(20),
CITTA              VARCHAR2(20),

```

```

CAP          CHAR(5),
NOME         VARCHAR2(20) NOT NULL,
COGNOME      VARCHAR2(20) NOT NULL,
DATA_DI_NASCITA  DATE NOT NULL,
CELLULARE    CHAR(10),
SESSO        CHAR(1) NOT NULL,
CONSTRAINT CH_SESSO CHECK (SESSO IN ('M','F'))
);

```

### Infermiere

La tabella Infermiere rappresenta l'entità che crea la Cartella Infermieristica e che assiste agli interventi. E' identificato da un codice fiscale e da un codice di registrazione all'Albo degli Infermieri.

```

CREATE TABLE INFERMIERE(
CF_INF       CHAR(16) PRIMARY KEY,
ALBO         CHAR(6) NOT NULL,
CONSTRAINT FK_CF_INF FOREIGN KEY (CF_INF) REFERENCES PERSONA(CF) ON
DELETE CASCADE
);

```

### Medico

La tabella Medico rappresenta l'entità che crea la Cartella Clinica, effettua gli interventi, prescrive una dieta ai pazienti ed effettua le visite ad essi. E' identificato oltre che dal codice fiscale e dal codice di registrazione all'Albo dei Medici, anche dalla specializzazione. La specializzazione è caratterizzata dal vincolo che definisce tutte le possibili specializzazioni che possono essere associate ad un medico; in mancanza di specializzazione, il medico avrà come specializzazione "Medico Generale".

```

CREATE TABLE MEDICO(
CF_MED       CHAR(16) PRIMARY KEY,
ALBO         CHAR(6) NOT NULL,
SPECIALIZZAZIONE VARCHAR2(30) DEFAULT 'Medico Generale' NOT NULL,
CONSTRAINT FK_CF_MED FOREIGN KEY (CF_MED) REFERENCES PERSONA(CF) ON
DELETE CASCADE,
CONSTRAINT CH_SPECIALIZZAZIONE CHECK (SPECIALIZZAZIONE IN ('Medicina
Interna', 'Cardiologia', 'Cardiochirurgia', 'Chirurgia
Generale', 'Neurochirurgia', 'Neurologia', 'Geriatrica', 'Fisiatria',
'Dietologia', 'Ortopedia e traumatologia'))
);

```

### Fisioterapista

La tabella Fisioterapista rappresenta l'entità che scrive il Foglio della Fisioterapia, contenuto nella Cartella Clinica del paziente. E' rappresentato da un codice fiscale e dal codice di registrazione all'Albo dei Medici.

```
CREATE TABLE FISIOTERAPISTA(
CF_FISIO      CHAR(16) PRIMARY KEY,
ALBO          CHAR(6) NOT NULL,
CONSTRAINT FK_CF_FISIO FOREIGN KEY (CF_FISIO) REFERENCES PERSONA(CF) ON
DELETE CASCADE
);
```

### Paziente

La tabella Paziente rappresenta l'entità del paziente tenuto in ricovero dalla clinica. E' identificato da un codice fiscale e dal numero di cellulare di un familiare.

```
CREATE TABLE PAZIENTE(
CF_PAZ        CHAR(16) PRIMARY KEY,
CELL_FAMILIARE CHAR(10),
CONSTRAINT FK_CF_PAZ FOREIGN KEY (CF_PAZ) REFERENCES PERSONA(CF) ON
DELETE CASCADE );
```

### Ricovero

La tabella Ricovero rappresenta l'entità del ricovero legato al paziente in clinica.

```
CREATE TABLE RICOVERO(
ID_RICOVERO    CHAR(10) PRIMARY KEY,
CODICE_DIAGNOSI NUMBER NOT NULL,
DATA_RICOVERO  DATE NOT NULL,
CF_PAZ         CHAR(16),
CONSTRAINT FK_R FOREIGN KEY (CF_PAZ) REFERENCES PAZIENTE(CF_PAZ) ON
DELETE CASCADE,
CONSTRAINT CH_CODICE_DIAGNOSI CHECK (CODICE_DIAGNOSI IN
(101,102,103,104,105,106,107,108,109,110))
);
```

### Cartella Infermieristica

La tabella Cartella Infermieristica rappresenta l'entità della cartella infermieristica scritta dagli infermieri che è legata al paziente.

```
CREATE TABLE CARTELLA_INFERMIERISTICA(
COD_CART_INF   CHAR(10) PRIMARY KEY,
ID_RICOVERO     CHAR(10) NOT NULL,
CONSTRAINT FK_ID_RICOVERO FOREIGN KEY (ID_RICOVERO) REFERENCES
RICOVERO(ID_RICOVERO) ON DELETE CASCADE
);
```

**ha\_I**

La tabella ha\_I rappresenta la relazione che vi è tra la cartella infermieristica e l'infermiere che l'ha scritta.

```
CREATE TABLE ha_I(
  COD_CART_INF      CHAR(10),
  CF_INF            CHAR(16),
  CONSTRAINT PK_ha PRIMARY KEY (COD_CART_INF, CF_INF),
  CONSTRAINT FK_ha_CF_INF FOREIGN KEY (CF_INF) REFERENCES
    INFERMIERE(CF_INF) ON DELETE CASCADE,
  CONSTRAINT FK_ha_COD_CART_INF FOREIGN KEY (COD_CART_INF) REFERENCES
    CARTELLA_INFERMIERISTICA(COD_CART_INF) ON DELETE CASCADE
);
```

**Foglio\_Terapia**

La tabella Foglio\_Terapia rappresenta l'entità del foglio della terapia che il paziente deve seguire ed è contenuto nella cartella clinica. I vincoli applicano un range di tipologie di somministrazione della terapia al paziente.

```
CREATE TABLE FOGLIO_TERAPIA(
  DATA_ORA          DATE,
  METODO_SOMMINISTRA VARCHAR2(20) NOT NULL,
  COD_CART_INF       CHAR(10) NOT NULL,
  CONSTRAINT PK_FT PRIMARY KEY (DATA_ORA, COD_CART_INF),
  CONSTRAINT CH_METODO_SOMMINISTRA CHECK (METODO_SOMMINISTRA
    IN('Orale', 'Rettale', 'Endovena', 'Inalatoria', 'Transcutanea',
    'Intravascolare', 'Intramuscolare')),
  CONSTRAINT FK_COD_CART_INF FOREIGN KEY (COD_CART_INF) REFERENCES
    CARTELLA_INFERMIERISTICA(COD_CART_INF) ON DELETE CASCADE
);
```

**Farmaco**

La tabella Farmaco rappresenta l'entità del farmaco contenuto nel foglio terapia. E' identificato dal codice farmaco, dal tipo (vincolato), dal nome del farmaco e dal numero di assunzioni ogni giorno.

```
CREATE TABLE FARMACO(
  COD_FARMACO        CHAR(10) PRIMARY KEY,
  TIPO               VARCHAR2(20),
  NOME_FARMACO       VARCHAR2(20) NOT NULL,
  ASSUNZ_GIORNALIERA NUMBER NOT NULL,
  CONSTRAINT CH_TIPO CHECK (TIPO IN ('Antimicrobico', 'Catartico',
    'Emetocartartico', 'Oressizzante', 'Stomachico'))
);
```

**contiene**

La tabella contiene rappresenta la relazione che vi è tra il foglio terapia ed il

farmaco.

```
CREATE TABLE contiene(
  DATA_ORA          DATE,
  COD_CART_INF       CHAR(10),
  COD_FARMACO        CHAR(10) NOT NULL,
  CONSTRAINT PL_contiene PRIMARY KEY (DATA_ORA, COD_CART_INF),
  CONSTRAINT FK_contiene_COD_FARMACO FOREIGN KEY (COD_FARMACO) REFERENCES
    FARMACO(COD_FARMACO),
  CONSTRAINT FK_contiene_DATA_ORA FOREIGN KEY (DATA_ORA, COD_CART_INF)
    REFERENCES FOGLIO_TERAPIA(DATA_ORA, COD_CART_INF) ON DELETE CASCADE
);
```

### Diario\_Infermieristico

La tabella Diario\_Infermieristico rappresenta l'entità Diario\_Infermieristico dell'EE/R contenuta nella cartella infermieristica e ha lo scopo di registrare i sintomi rilevati dagli infermieri.

```
CREATE TABLE DIARIO_INFERMIERISTICO(
  DATA_ORA          DATE,
  COD_CART_INF       CHAR(10),
  CONSTRAINT PK_DI PRIMARY KEY (DATA_ORA, COD_CART_INF),
  CONSTRAINT FK_DI FOREIGN KEY (COD_CART_INF) REFERENCES
    CARTELLA_INFERMIERISTICA(COD_CART_INF) ON DELETE CASCADE
);
```

### sintomatologia\_di

La tabella sintomatologia\_di rappresenta l'attributo multivalore "Sintomatologia" del diario infermieristico.

```
CREATE TABLE sintomatologia_di(
  DATA_ORA          DATE,
  SINTOMO            VARCHAR2(30) NOT NULL,
  COD_CART_INF       CHAR(10),
  CONSTRAINT PK_SINT PRIMARY KEY (DATA_ORA, COD_CART_INF),
  CONSTRAINT FK_SINT FOREIGN KEY (DATA_ORA, COD_CART_INF) REFERENCES
    DIARIO_INFERMIERISTICO(DATA_ORA, COD_CART_INF) ON DELETE CASCADE,
  CONSTRAINT CH_SINTOMAT CHECK (SINTOMO IN ('Gonfiore', 'Febbre', 'Tosse',
    'Vomito', 'Diarrea', 'Infiammazione', 'Dolori addominali', 'Nausea',
    'Stanchezza', 'Stitichezza'))
);
```

### Intervento

La tabella intervento rappresenta l'entità che è identificata dall'ID dell'intervento eseguito da uno o più medici ed assistito da uno o più infermieri ad un paziente. Come attributi sono presenti inoltre il tipo di intervento, il numero di sala operatoria in cui è eseguito tale intervento, la durata, la data prevista e la data

effettiva in cui viene effettuato tale intervento, che può non coincidere con la data prevista.

```
CREATE TABLE INTERVENTO(
  ID_INTERVENTO      CHAR(10) PRIMARY KEY,
  DATA_PREVISTA      DATE NOT NULL,
  DATA_EFFETTIVA     DATE,
  TIPO                VARCHAR2(20) NOT NULL,
  DURATA              NUMBER NOT NULL,
  NUM_SALA_OPERA      NUMBER NOT NULL,
  ID_RICOVERO         CHAR(10),
  CF_PAZ              CHAR(16),
  CONSTRAINT FK_INT_ID FOREIGN KEY (ID_RICOVERO) REFERENCES
    RICOVERO(ID_RICOVERO) ON DELETE CASCADE, CONSTRAINT FK_INT_CF FOREIGN
    KEY (CF_PAZ) REFERENCES PAZIENTE(CF_PAZ) ON DELETE CASCADE,
  CONSTRAINT CH_NUM_SALA_OPERA CHECK (NUM_SALA_OPERA IN
    (10,11,12,13,14,15)),
  CONSTRAINT CH_TIPO_INT CHECK (TIPO IN ('Amputazione', 'Anastomosi',
    'Artrodesi', 'Artroplastica', 'Aspirazione', 'Asportazione',
    'Cardioplegia', 'Cardiotomia', 'Cateterismo', 'Cistectomia',
    'Drenaggio', 'Gastrotomia'))
);
```

### assiste

La tabella assiste rappresenta la relazione tra l'infermiere e l'intervento.

```
CREATE TABLE assiste(
  CF_INF              CHAR(16),
  ID_INTERVENTO       CHAR(10) PRIMARY KEY,
  CONSTRAINT FK_A_ID FOREIGN KEY (ID_INTERVENTO) REFERENCES
    INTERVENTO(ID_INTERVENTO) ON DELETE CASCADE,
  CONSTRAINT FK_A_CF FOREIGN KEY (CF_INF) REFERENCES INFERMIERE(CF_INF) ON
    DELETE CASCADE
);
```

### Reparto

La tabella Reparto rappresenta il reparto in cui è tenuto il ricovero del paziente. E' identificato dal numero del piano, dal nome del reparto, dal numero dei posti ed il numero delle stanze. Si noti che il vincolo sul numero del piano vede un range di valori interi che vanno da 0 a 5; il range dei posti va da 1 a 6 ed il nome del reparto indica la tipologia del reparto in questione.

```
CREATE TABLE REPARTO(
  N_PIANO             NUMBER NOT NULL,
  NOME_REPARTO        VARCHAR2(30) NOT NULL,
  N_POSTI             NUMBER DEFAULT 5 NOT NULL,
  N_STANZE            NUMBER DEFAULT 15 NOT NULL,
  CONSTRAINT PK_REP PRIMARY KEY (N_PIANO, NOME_REPARTO),
  CONSTRAINT CH_N_PIANO CHECK (N_PIANO IN (0,1,2,3,4,5)),
  CONSTRAINT CH_N_POSTI CHECK (N_POSTI IN (1,2,3,4,5,6)),
```

```

CONSTRAINT CH_NOME_REPARTO CHECK (NOME_REPARTO IN ('Cardiologia',
'Anatomia e Istologia patologica', 'Anestesia e Rianimazione',
'Chirurgia', 'Oncologia', 'Gastroenterologia', 'Medicina Generale',
'Ortopedia e
Traumatologia', 'Radiologia', 'Diabetologia', 'Geriatria', 'Medicina
Interna', 'Neurologia', 'Pediatria', 'Urologia', 'Psichiatria'))
);

```

### e\_tenuto

La tabella e\_tenuto rappresenta la relazione tra il ricovero del paziente ed il reparto in cui è ospitato.

```

CREATE TABLE e_tenuto(
ID_RICOVERO      CHAR(10) PRIMARY KEY,
DATA_ORA         DATE NOT NULL,
N_PIANO          NUMBER NOT NULL,
NOME_REPARTO     VARCHAR2(20),
CONSTRAINT FK_ID_ET FOREIGN KEY (ID_RICOVERO) REFERENCES
RICOVERO(ID_RICOVERO) ON DELETE CASCADE, CONSTRAINT FK_N_PIANO
FOREIGN KEY (N_PIANO, NOME_REPARTO) REFERENCES REPARTO(N_PIANO,
NOME_REPARTO) ON DELETE CASCADE
);

```

### Dimissione

La tabella Dimissione rappresenta l'entità della dimissione del paziente dalla clinica. Si noti che, data una dimissione del paziente, egli può ritornare in clinica ma con un nuovo ID ricovero, ma i dati del suo precedente ricovero saranno conservati e tenuti in considerazione al momento del nuovo ricovero.

```

CREATE TABLE DIMISSIONE(
ID_DIMISSIONE    CHAR(10) PRIMARY KEY,
DATA_ORA         DATE NOT NULL,
CODICE_DIAGNOSI  NUMBER DEFAULT 100 NOT NULL,
MOD_DIMISSIONE   NUMBER NOT NULL,
ID_RICOVERO      CHAR(10),
CONSTRAINT FK_ID_DIM FOREIGN KEY (ID_RICOVERO) REFERENCES
RICOVERO(ID_RICOVERO) ON DELETE CASCADE,
--da 0 (completamente autonomo) a 10 (completamente dipendente)
CONSTRAINT CH_MOD_DIMISSIONE CHECK (MOD_DIMISSIONE IN
(0,1,2,3,4,5,6,7,8,9,10)),
--CODICE_DIAGNOSI: da 100 a 110 (dove 100 indica completamente guarito)
CONSTRAINT CH_CODICE_DIAGNOSI CHECK (CODICE_DIAGNOSI IN
(100,101,102,103,104,105,106,107,108,109,110))
);

```

### effettua

La tabella effettua rappresenta la relazione tra il medico che effettua l'intervento



e l'intervento stesso.

```
CREATE TABLE effettua(
CF_MED          CHAR(16),
ID_INTERVENTO   CHAR(10) PRIMARY KEY,
CONSTRAINT FK_EFF FOREIGN KEY (CF_MED) REFERENCES MEDICO(CF_MED) ON
    DELETE CASCADE,
CONSTRAINT FK_ID_EFF FOREIGN KEY (ID_INTERVENTO) REFERENCES
    INTERVENTO(ID_INTERVENTO) ON DELETE CASCADE
);
```

### Dieta

La tabella Dieta rappresenta l'entità della dieta scritta dal medico che il paziente deve seguire durante il ricovero. Viene inoltre stabilito un tipo ben preciso di dieta e ne viene effettuato il check dal vincolo.

```
CREATE TABLE DIETA(
CODICE_DIETA     CHAR(10) PRIMARY KEY,
TIPO             VARCHAR2(30) NOT NULL,
KCAL             NUMBER NOT NULL,
PROTEINE        NUMBER NOT NULL,
GRASSI          NUMBER NOT NULL,
CF_MED          CHAR(16),
ID_RICOVERO     CHAR(10),
CONSTRAINT FK_CF_MED_DIETA FOREIGN KEY (CF_MED) REFERENCES
    MEDICO(CF_MED) ON DELETE CASCADE, CONSTRAINT FK_ID_RICOVERO_DIETA
    FOREIGN KEY (ID_RICOVERO) REFERENCES RICOVERO(ID_RICOVERO) ON DELETE
    CASCADE,
CONSTRAINT CH_TIPO_DIETA CHECK (TIPO IN ('Normoenergetica', 'Lattea
    integrata', 'Liquida', 'Semiliquida', 'Ipocalorica', 'Ipoglicidica
    IDDM', 'Ipoglicidica NIDDM', 'Lipolipidica', 'Ipoproteica',
    'Ipopurinica', 'Ipernutritiva', 'Astringente'))
);
```

### Visita

La tabella Visita rappresenta l'entità della singola visita effettuata dal medico ad un paziente. Tenendo conto che per ovvi motivi un paziente riceva più visite, vengono salvate la data e l'ora prevista della visita e la data e l'ora della visita effettiva (in quanto possono non coincidere), in modo tale da differenziare una visita dall'altra.

```
CREATE TABLE VISITA(
DATA_ORA_PREVISTA DATE NOT NULL,
DATA_ORA_EFFETTIVA DATE,
TIPO_VISITA       VARCHAR2(20) DEFAULT 'Generica' NOT NULL,
CF_PAZ           CHAR(16) PRIMARY KEY,
CF_MED          CHAR(16),
ID_RICOVERO     CHAR(10),
CONSTRAINT FK_CF_PAZ_V FOREIGN KEY (CF_PAZ) REFERENCES PAZIENTE(CF_PAZ)
```

```

    ON DELETE CASCADE,
CONSTRAINT FK_CF_MED_V FOREIGN KEY (CF_MED) REFERENCES MEDICO(CF_MED) ON
DELETE CASCADE,
CONSTRAINT FK_ID_RICOVERO_V FOREIGN KEY (ID_RICOVERO) REFERENCES
RICOVERO(ID_RICOVERO) ON DELETE CASCADE,
CONSTRAINT CH_TIPO_VISITA CHECK (TIPO_VISITA IN ('Cardiologica',
'Dietologica', 'Fisiatrice', 'Geriatrica', 'Neurologica'))
);

```

### Cartella\_Clinica

La tabella Cartella\_Clinica rappresenta l'entità della cartella clinica scritta dai medici che è legata al paziente.

```

CREATE TABLE CARTELLA_CLINICA(
CODICE_CC          CHAR(10) PRIMARY KEY,
SINTOMO            VARCHAR2(30) NOT NULL,
ID_RICOVERO        CHAR(10),
CONSTRAINT FK_ID_RICOVERO_C FOREIGN KEY (ID_RICOVERO) REFERENCES
RICOVERO(ID_RICOVERO) ON DELETE CASCADE,
CONSTRAINT CH_SINTOMO CHECK (SINTOMO IN ('Gonfiore', 'Febbre', 'Tosse',
'Vomito', 'Diarrea', 'Infiammazione', 'Dolori addominali', 'Nausea',
'Stanchezza', 'Stitichezza'))
);

```

### ha\_m

La tabella ha\_M rappresenta la relazione che vi è tra la cartella clinica ed il medico che l'ha scritta.

```

CREATE TABLE ha_M(
CODICE_CC          CHAR(10) PRIMARY KEY,
CF_MED             CHAR(16),
CONSTRAINT FK_CODICE_CC_H FOREIGN KEY (CODICE_CC) REFERENCES
CARTELLA_CLINICA(CODICE_CC) ON DELETE CASCADE,
CONSTRAINT FK_CF_MED_H FOREIGN KEY (CF_MED) REFERENCES MEDICO(CF_MED) ON
DELETE CASCADE
);

```

### Esami

La tabella Esami rappresenta l'entità degli esami che effettua il paziente e che sono contenuti nella cartella clinica associata ad esso.

```

CREATE TABLE ESAMI(
DATA_ORA           DATE,
TIPO_ESAME         VARCHAR2(30) NOT NULL,
ESITO              NUMBER NOT NULL,
CODICE_CC          CHAR(10),
CONSTRAINT PK_ES PRIMARY KEY (DATA_ORA, CODICE_CC),
CONSTRAINT FK_CODICE_CC_ES FOREIGN KEY (CODICE_CC) REFERENCES

```

```

    CARTELLA_CLINICA(CODICE_CC) ON DELETE CASCADE,
--Esito -> 0 Positivo, 1 Negativo
CONSTRAINT CH_ESITO CHECK (ESITO IN (0, 1)),
CONSTRAINT CH_TIPO_ESAME CHECK (TIPO_ESAME IN ('Radiografia', 'Risonanza
    Magnetica', 'TAC', 'Ecografia', 'PET'))
);

```

### Analisi

La tabella Analisi rappresenta l'entità delle analisi che effettua il paziente e che sono contenute nella cartella clinica ad esso appartenente. I valori di glucosio, cloro, potassio e sodio sono valori che hanno uno scopo puramente dimostrativo.

```

CREATE TABLE ANALISI(
DATA_ORA_PRELIEVO    DATE,
GLUCOSIO              NUMBER NOT NULL,
CLORO                 NUMBER NOT NULL,
POTASSIO              NUMBER NOT NULL,
SODIO                NUMBER NOT NULL,
CODICE_CC             CHAR(10),
CONSTRAINT PK_PRIMARY KEY (DATA_ORA_PRELIEVO, CODICE_CC),
CONSTRAINT FK_CODICE_CC_AN FOREIGN KEY (CODICE_CC) REFERENCES
    CARTELLA_CLINICA(CODICE_CC) ON DELETE CASCADE
);

```

### Diario\_Clinico

La tabella Diario Clinico rappresenta l'entità che tiene traccia dei sintomi all'interno della cartella clinica scritta da uno o più medici.

```

CREATE TABLE DIARIO_CLINICO(
DATA_ORA             DATE,
CODICE_CC            CHAR(10),
CONSTRAINT PK_DC PRIMARY KEY (DATA_ORA, CODICE_CC), CONSTRAINT FK_DC
    FOREIGN KEY (CODICE_CC) REFERENCES CARTELLA_CLINICA(CODICE_CC) ON
    DELETE CASCADE
);

```

### Foglio\_Fisioterapia

La tabella Foglio\_Fisioterapia rappresenta l'entità foglio fisioterapia che è scritto dal fisioterapista, è contenuto nella cartella clinica del paziente e contiene i risultati delle terapie effettuate dal fisioterapista. Tali terapie sono di tipo number ed i vincoli aggiunti su tali valori fanno sì che siano fissati in una scala che va da 0 a 10, dove 0 rappresenta un valore di insufficiente autonomia o la non-collaborazione del paziente, 10 rappresenta la piena autonomia del paziente o la sua completa collaborazione.

```

CREATE TABLE FOGLIO_FISIOTERAPIA(
DATA_ORA          DATE,
RINF_MUSC         NUMBER,
ELETTR_MUSC       NUMBER,
DEAMBULAZIONE     NUMBER,
CF_FISIOTERAPISTA CHAR(16),
CODICE_CC         CHAR(10),
CONSTRAINT PK_FF PRIMARY KEY (DATA_ORA, CF_FISIOTERAPISTA, CODICE_CC),
CONSTRAINT FK_FF1 FOREIGN KEY (CF_FISIOTERAPISTA) REFERENCES
    FISIOTERAPISTA(CF_FISIO) ON DELETE CASCADE,
CONSTRAINT FK_FF2 FOREIGN KEY (CODICE_CC) REFERENCES
    CARTELLA_CLINICA(CODICE_CC) ON DELETE CASCADE,
--Livello di autonomia del paziente. Da 0 (insufficiente o non
--collaborativo) a 10 (autonomo o collaborativo)
CONSTRAINT CH_RINF_MUSC CHECK (RINF_MUSC IN (0,1,2,3,4,5,6,7,8,9,10)),
CONSTRAINT CH_ELETTR_MUSC CHECK (ELETTR_MUSC IN
    (0,1,2,3,4,5,6,7,8,9,10)),
CONSTRAINT CH_DEAMBULAZIONE CHECK (DEAMBULAZIONE IN
    (0,1,2,3,4,5,6,7,8,9,10))
);

```

### Sintomatologia\_CC

La tabella sintomatologia\_cc rappresenta l'attributo multivalore della sintomatologia del diario clinico.

```

CREATE TABLE sintomatologia_cc(
DATA_ORA          DATE,
SINTOMO           VARCHAR2(30) NOT NULL,
CODICE_CC         CHAR(10),
CONSTRAINT PK_SINT_CC PRIMARY KEY(DATA_ORA, CODICE_CC),
CONSTRAINT FK_SINT_CC FOREIGN KEY(DATA_ORA, CODICE_CC) REFERENCES
    DIARIO_CLINICO(DATA_ORA, CODICE_CC) ON DELETE CASCADE,
CONSTRAINT CH_SINTOMAT_CC CHECK (SINTOMO IN ('Gonfiore', 'Febbre',
    'Tosse', 'Vomito', 'Diarrea', 'Infiammazione', 'Dolori addominali',
    'Nausea', 'Stanchezza', 'Stitichezza'))
);

```

## 2.4 Data Manipulation Language

Il popolamento dell'entità persona, medico, infermiere e fisioterapista viene effettuata dagli amministratori del DB. Resta da definire da chi avverrà il popolamento dei pazienti. Vengono riportate due insert per ogni tabella del DB.

```
--PERSONA
INSERT INTO
  PERSONA(CF,VIA,CITTA,CAP,NOME,COGNOME,DATA_DI_NASCITA,CELLULARE,SESSO)
VALUES ('AAA00000000000001','IV
NOVEMBRE','NAPOLI','80124','GIUSEPPE','CRISPINO',TO_DATE('14/10/1978',
'dd/mm/yyyy'),'3330928372','M');
INSERT INTO
  PERSONA(CF,VIA,CITTA,CAP,NOME,COGNOME,DATA_DI_NASCITA,CELLULARE,SESSO)
VALUES ('AAA00000000000002','GIUSEPPE
VERDI','MILANO','20019','MATTEO','RUOCCO',TO_DATE('10/06/1988',
'dd/mm/yyyy'),'3332389223','M');
INSERT INTO
  PERSONA(CF,VIA,CITTA,CAP,NOME,COGNOME,DATA_DI_NASCITA,CELLULARE,SESSO)
VALUES ('AAA00000000000003','CORSO
UMBERTO','CASORIA','80026','LUCA','VINCENZI',TO_DATE('20/01/1958',
'dd/mm/yyyy'),'3662834531','M');
INSERT INTO
  PERSONA(CF,VIA,CITTA,CAP,NOME,COGNOME,DATA_DI_NASCITA,CELLULARE,SESSO)
VALUES ('AAA00000000000004','GIUSEPPE
GARIBALDI','FRATTAMAGGIORE','80027','ANTONIO','CAPASSO',
TO_DATE('27/03/1967','dd/mm/yyyy'),'3442835831','M');
INSERT INTO
  PERSONA(CF,VIA,CITTA,CAP,NOME,COGNOME,DATA_DI_NASCITA,CELLULARE,SESSO)
VALUES
  ('AAA00000000000005','MARCONI','ROMA','00123','LAURA','MATROPIETRO',
TO_DATE('15/02/1977','dd/mm/yyyy'),'3552835831','F');
```

```
--MEDICO
INSERT INTO MEDICO(CF_MED,ALBO,SPECIALIZZAZIONE) VALUES
  ('AAA00000000000001','928401','Cardiologia');
INSERT INTO MEDICO(CF_MED,ALBO,SPECIALIZZAZIONE) VALUES
  ('AAA00000000000002','928402','Cardiologia');
INSERT INTO MEDICO(CF_MED,ALBO,SPECIALIZZAZIONE) VALUES
  ('AAA00000000000003','928403','Dietologia');
INSERT INTO MEDICO(CF_MED,ALBO,SPECIALIZZAZIONE) VALUES
  ('AAA00000000000004','928404','Dietologia');
INSERT INTO MEDICO(CF_MED,ALBO,SPECIALIZZAZIONE) VALUES
  ('AAA00000000000005','928405','Fisiatria');
```

```
--PAZIENTE
INSERT INTO PAZIENTE(CF_PAZ,CELL_FAMILIARE) VALUES
  ('AAA00000000000017','3330928372');
```

```

INSERT INTO PAZIENTE(CF_PAZ,CELL_FAMILIARE) VALUES
('AAA00000000000018','3330928373');
INSERT INTO PAZIENTE(CF_PAZ,CELL_FAMILIARE) VALUES
('AAA00000000000019','3330928374');
INSERT INTO PAZIENTE(CF_PAZ,CELL_FAMILIARE) VALUES
('AAA00000000000020','3330928375');
INSERT INTO PAZIENTE(CF_PAZ,CELL_FAMILIARE) VALUES
('AAA00000000000021','3330928376');

```

```

--FISIOTERAPISTA
INSERT INTO FISIOTERAPISTA (CF_FISIO, ALBO) VALUES ('BBB0000000000001',
'FIS001');
INSERT INTO FISIOTERAPISTA (CF_FISIO, ALBO) VALUES ('BBB0000000000002',
'FIS002');
INSERT INTO FISIOTERAPISTA (CF_FISIO, ALBO) VALUES ('BBB0000000000003',
'FIS003');
INSERT INTO FISIOTERAPISTA (CF_FISIO, ALBO) VALUES ('BBB0000000000004',
'FIS004');
INSERT INTO FISIOTERAPISTA (CF_FISIO, ALBO) VALUES ('BBB0000000000005',
'FIS005');

```

```

--INFERMIERE
INSERT INTO INFERMIERE(CF_INF, ALBO) VALUES ('CCC0000000000001',
'817301');
INSERT INTO INFERMIERE(CF_INF, ALBO) VALUES ('CCC0000000000002',
'817302');
INSERT INTO INFERMIERE(CF_INF, ALBO) VALUES ('CCC0000000000003',
'817303');
INSERT INTO INFERMIERE(CF_INF, ALBO) VALUES ('CCC0000000000004',
'817304');
INSERT INTO INFERMIERE(CF_INF, ALBO) VALUES ('CCC0000000000005',
'817305');

```

```

--RICOVERO
INSERT INTO RICOVERO(ID_RICOVERO,CODICE_DIAGNOSI,DATA_RICOVERO,CF_PAZ)
VALUES ('3000000001',101,TO_DATE('01/01/2019 09:00:00','dd/mm/yyyy
HH24:mi:ss'),'AAA00000000000017');
INSERT INTO RICOVERO(ID_RICOVERO,CODICE_DIAGNOSI,DATA_RICOVERO,CF_PAZ)
VALUES ('3000000002',102,TO_DATE('10/01/2019 11:00:00','dd/mm/yyyy
HH24:mi:ss'),'AAA00000000000018');
INSERT INTO RICOVERO(ID_RICOVERO,CODICE_DIAGNOSI,DATA_RICOVERO,CF_PAZ)
VALUES ('3000000003',103,TO_DATE('12/01/2019 10:00:00','dd/mm/yyyy
HH24:mi:ss'),'AAA00000000000019');
INSERT INTO RICOVERO(ID_RICOVERO,CODICE_DIAGNOSI,DATA_RICOVERO,CF_PAZ)
VALUES ('3000000004',104,TO_DATE('22/01/2019 13:00:00','dd/mm/yyyy
HH24:mi:ss'),'AAA00000000000020');
INSERT INTO RICOVERO(ID_RICOVERO,CODICE_DIAGNOSI,DATA_RICOVERO,CF_PAZ)
VALUES ('3000000005',105,TO_DATE('01/02/2019 09:00:00','dd/mm/yyyy
HH24:mi:ss'),'AAA00000000000021');

```

```

--INTERVENTO
INSERT INTO
  INTERVENTO(ID_INTERVENTO,DATA_PREVISTA,DATA_EFFETTIVA,TIPO,DURATA,
  NUM_SALA_OPERA,ID_RICOVERO,CF_PAZ) VALUES
  ('1000000001',TO_DATE('11/02/2019 11:00:00','dd/mm/yyyy
  HH24:mi:ss'),TO_DATE('11/02/2019 11:00:00','dd/mm/yyyy
  HH24:mi:ss'),'Amputazione',1,10,'3000000001','AAA0000000000017');
INSERT INTO
  INTERVENTO(ID_INTERVENTO,DATA_PREVISTA,DATA_EFFETTIVA,TIPO,DURATA,
  NUM_SALA_OPERA,ID_RICOVERO,CF_PAZ) VALUES
  ('1000000002',TO_DATE('13/02/2019 11:00:00','dd/mm/yyyy
  HH24:mi:ss'),TO_DATE('13/02/2019 12:00:00','dd/mm/yyyy
  HH24:mi:ss'),'Aspirazione',5,10,'3000000001','AAA0000000000017');
INSERT INTO
  INTERVENTO(ID_INTERVENTO,DATA_PREVISTA,DATA_EFFETTIVA,TIPO,DURATA,
  NUM_SALA_OPERA,ID_RICOVERO,CF_PAZ) VALUES
  ('1000000003',TO_DATE('14/02/2019 11:00:00','dd/mm/yyyy
  HH24:mi:ss'),TO_DATE('14/02/2019 11:00:00','dd/mm/yyyy
  HH24:mi:ss'),'Aspirazione',3,10,'3000000002','AAA0000000000018');
INSERT INTO
  INTERVENTO(ID_INTERVENTO,DATA_PREVISTA,DATA_EFFETTIVA,TIPO,DURATA,
  NUM_SALA_OPERA,ID_RICOVERO,CF_PAZ) VALUES
  ('1000000004',TO_DATE('15/02/2019 11:00:00','dd/mm/yyyy
  HH24:mi:ss'),TO_DATE('16/02/2019 11:00:00','dd/mm/yyyy
  HH24:mi:ss'),'Asportazione',6,11,'3000000002','AAA0000000000018');
INSERT INTO
  INTERVENTO(ID_INTERVENTO,DATA_PREVISTA,DATA_EFFETTIVA,TIPO,DURATA,
  NUM_SALA_OPERA,ID_RICOVERO,CF_PAZ) VALUES
  ('1000000005',TO_DATE('03/03/2019 11:00:00','dd/mm/yyyy
  HH24:mi:ss'),TO_DATE('05/03/2019 11:00:00','dd/mm/yyyy
  HH24:mi:ss'),'Asportazione',6,11,'3000000003','AAA0000000000019');

```

```

--DIETA
INSERT INTO
  DIETA(CODICE_DIETA,TIPO,KCAL,PROTEINE,GRASSI,CF_MED,ID_RICOVERO)
  VALUES ('2000000001','Normoenergetica',1800,92,60,'AAA0000000000003',
  '3000000001');
INSERT INTO
  DIETA(CODICE_DIETA,TIPO,KCAL,PROTEINE,GRASSI,CF_MED,ID_RICOVERO)
  VALUES ('2000000002','Liquida',1800,92,60,'AAA0000000000003',
  '3000000002');
INSERT INTO
  DIETA(CODICE_DIETA,TIPO,KCAL,PROTEINE,GRASSI,CF_MED,ID_RICOVERO)
  VALUES ('2000000003','Liquida',1800,92,60,'AAA0000000000003',
  '3000000003');
INSERT INTO
  DIETA(CODICE_DIETA,TIPO,KCAL,PROTEINE,GRASSI,CF_MED,ID_RICOVERO)
  VALUES ('2000000004','Normoenergetica',1800,92,60,'AAA0000000000003',
  '3000000004');
INSERT INTO
  DIETA(CODICE_DIETA,TIPO,KCAL,PROTEINE,GRASSI,CF_MED,ID_RICOVERO)
  VALUES ('2000000005','Normoenergetica',1800,92,60,'AAA0000000000003',
  '3000000005');

```

```
--assiste
INSERT INTO assiste(CF_INF,ID_INTERVENTO) VALUES
('CCC0000000000001','1000000001');
INSERT INTO assiste(CF_INF,ID_INTERVENTO) VALUES
('CCC0000000000001','1000000002');
INSERT INTO assiste(CF_INF,ID_INTERVENTO) VALUES
('CCC0000000000002','1000000003');
INSERT INTO assiste(CF_INF,ID_INTERVENTO) VALUES
('CCC0000000000002','1000000004');
INSERT INTO assiste(CF_INF,ID_INTERVENTO) VALUES
('CCC0000000000003','1000000005');
```

```
--effettua
INSERT INTO effettua(CF_MED,ID_INTERVENTO) VALUES
('AAA0000000000009','1000000001');
INSERT INTO effettua(CF_MED,ID_INTERVENTO) VALUES
('AAA0000000000010','1000000002');
INSERT INTO effettua(CF_MED,ID_INTERVENTO) VALUES
('AAA0000000000010','1000000003');
INSERT INTO effettua(CF_MED,ID_INTERVENTO) VALUES
('AAA0000000000010','1000000004');
INSERT INTO effettua(CF_MED,ID_INTERVENTO) VALUES
('AAA0000000000010','1000000005');
```

```
--REPARTO
INSERT INTO REPARTO(N_PIANO,NOME_REPARTO,N_POSTI,N_STANZE) VALUES
(0,'Cardiologia',3,6);
INSERT INTO REPARTO(N_PIANO,NOME_REPARTO,N_POSTI,N_STANZE) VALUES
(0,'Anatomia e Istologia patologica',5,6);
INSERT INTO REPARTO(N_PIANO,NOME_REPARTO,N_POSTI,N_STANZE) VALUES
(1,'Anestesia e Rianimazione',6,6);
INSERT INTO REPARTO(N_PIANO,NOME_REPARTO,N_POSTI,N_STANZE) VALUES
(1,'Chirurgia',1,5);
INSERT INTO REPARTO(N_PIANO,NOME_REPARTO,N_POSTI,N_STANZE) VALUES
(1,'Oncologia',1,4);
```

```
--è tenuto
INSERT INTO e_tenuto(ID_RICOVERO,DATA_ORA,N_PIANO,NOME_REPARTO) VALUES
('3000000001',TO_DATE('01/01/2019 09:00:00','dd/mm/yyyy
HH24:mi:ss'),0,'Cardiologia');
INSERT INTO e_tenuto(ID_RICOVERO,DATA_ORA,N_PIANO,NOME_REPARTO) VALUES
('3000000002',TO_DATE('10/01/2019 11:00:00','dd/mm/yyyy
HH24:mi:ss'),0,'Anatomia e Istologia patologica');
INSERT INTO e_tenuto(ID_RICOVERO,DATA_ORA,N_PIANO,NOME_REPARTO) VALUES
('3000000003',TO_DATE('12/01/2019 10:00:00','dd/mm/yyyy
HH24:mi:ss'),1,'Anestesia e Rianimazione');
INSERT INTO e_tenuto(ID_RICOVERO,DATA_ORA,N_PIANO,NOME_REPARTO) VALUES
('3000000004',TO_DATE('22/01/2019 13:00:00','dd/mm/yyyy
HH24:mi:ss'),1,'Chirurgia');
INSERT INTO e_tenuto(ID_RICOVERO,DATA_ORA,N_PIANO,NOME_REPARTO) VALUES
```



```
( '3000000005', TO_DATE( '01/02/2019 09:00:00', 'dd/mm/yyyy
HH24:mi:ss'), 1, 'Oncologia');
```

```
--DIMISSIONE
INSERT INTO
  DIMISSIONE(ID_DIMISSIONE, DATA_ORA, CODICE_DIAGNOSI, MOD_DIMISSIONE,
ID_RICOVERO) VALUES ( '4000000001', TO_DATE( '01/03/2019
11:00:00', 'dd/mm/yyyy HH24:mi:ss'), 100, 0, '3000000001');
INSERT INTO
  DIMISSIONE(ID_DIMISSIONE, DATA_ORA, CODICE_DIAGNOSI, MOD_DIMISSIONE,
ID_RICOVERO) VALUES ( '4000000002', TO_DATE( '11/03/2019
11:00:00', 'dd/mm/yyyy HH24:mi:ss'), 100, 1, '3000000002');
INSERT INTO
  DIMISSIONE(ID_DIMISSIONE, DATA_ORA, CODICE_DIAGNOSI, MOD_DIMISSIONE,
ID_RICOVERO) VALUES ( '4000000003', TO_DATE( '11/03/2019
11:10:00', 'dd/mm/yyyy HH24:mi:ss'), 100, 2, '3000000003');
INSERT INTO
  DIMISSIONE(ID_DIMISSIONE, DATA_ORA, CODICE_DIAGNOSI, MOD_DIMISSIONE,
ID_RICOVERO) VALUES ( '4000000004', TO_DATE( '22/04/2019
13:00:00', 'dd/mm/yyyy HH24:mi:ss'), 101, 3, '3000000004');
INSERT INTO
  DIMISSIONE(ID_DIMISSIONE, DATA_ORA, CODICE_DIAGNOSI, MOD_DIMISSIONE,
ID_RICOVERO) VALUES ( '4000000005', TO_DATE( '25/04/2019
12:00:00', 'dd/mm/yyyy HH24:mi:ss'), 102, 4, '3000000005');
```

```
--CARTELLA CLINICA
INSERT INTO CARTELLA_CLINICA (CODICE_CC, SINTOMO, ID_RICOVERO) VALUES
( 'CC00000001', 'Febbre', '3000000001');
INSERT INTO CARTELLA_CLINICA (CODICE_CC, SINTOMO, ID_RICOVERO) VALUES
( 'CC00000002', 'Stitichezza', '3000000002');
INSERT INTO CARTELLA_CLINICA (CODICE_CC, SINTOMO, ID_RICOVERO) VALUES
( 'CC00000003', 'Nausea', '3000000003');
INSERT INTO CARTELLA_CLINICA (CODICE_CC, SINTOMO, ID_RICOVERO) VALUES
( 'CC00000004', 'Dolori addominali', '3000000004');
INSERT INTO CARTELLA_CLINICA (CODICE_CC, SINTOMO, ID_RICOVERO) VALUES
( 'CC00000005', 'Infiammazione', '3000000005');
```

```
--CARTELLA INFERMIERISTICA
INSERT INTO CARTELLA_INFERMIERISTICA(COD_CART_INF, ID_RICOVERO) VALUES
( 'CI77277340', '3000000001');
INSERT INTO CARTELLA_INFERMIERISTICA(COD_CART_INF, ID_RICOVERO) VALUES
( 'CI61623100', '3000000002');
INSERT INTO CARTELLA_INFERMIERISTICA(COD_CART_INF, ID_RICOVERO) VALUES
( 'CI72962004', '3000000003');
INSERT INTO CARTELLA_INFERMIERISTICA(COD_CART_INF, ID_RICOVERO) VALUES
( 'CI32604765', '3000000004');
INSERT INTO CARTELLA_INFERMIERISTICA(COD_CART_INF, ID_RICOVERO) VALUES
( 'CI96389296', '3000000005');
```

```

--VISITA
INSERT INTO VISITA (DATA_ORA_PREVISTA, DATA_ORA_EFFETTIVA, TIPO_VISITA,
CF_PAZ, CF_MED, ID_RICOVERO) VALUES (TO_DATE('10/07/2019
10:00','dd/mm/yyyy hh24:mi'), TO_DATE('10/07/2019 10:12','dd/mm/yyyy
hh24:mi'), 'Neurologica','AAA00000000000021', 'AAA00000000000012',
'30000000001');
INSERT INTO VISITA (DATA_ORA_PREVISTA, DATA_ORA_EFFETTIVA, TIPO_VISITA,
CF_PAZ, CF_MED, ID_RICOVERO) VALUES (TO_DATE('10/07/2019
11:00','dd/mm/yyyy hh24:mi'), TO_DATE('10/07/2019 11:20','dd/mm/yyyy
hh24:mi'), 'Cardiologica','AAA00000000000023',
'AAA00000000000005', '30000000003');
INSERT INTO VISITA (DATA_ORA_PREVISTA, DATA_ORA_EFFETTIVA, TIPO_VISITA,
CF_PAZ, CF_MED, ID_RICOVERO) VALUES (TO_DATE('10/07/2019
12:00','dd/mm/yyyy hh24:mi'), TO_DATE('10/07/2019 12:44','dd/mm/yyyy
hh24:mi'), 'Neurologica','AAA00000000000021',
'AAA00000000000008', '30000000001');
INSERT INTO VISITA (DATA_ORA_PREVISTA, DATA_ORA_EFFETTIVA, TIPO_VISITA,
CF_PAZ, CF_MED, ID_RICOVERO) VALUES (TO_DATE('10/07/2019
13:10','dd/mm/yyyy hh24:mi'), TO_DATE('10/07/2019 14:39','dd/mm/yyyy
hh24:mi'), 'Fisiatrica','AAA00000000000022',
'AAA00000000000008', '30000000002');
INSERT INTO VISITA (DATA_ORA_PREVISTA, DATA_ORA_EFFETTIVA, TIPO_VISITA,
CF_PAZ, CF_MED, ID_RICOVERO) VALUES (TO_DATE('11/07/2019
10:00','dd/mm/yyyy hh24:mi'), TO_DATE('10/07/2019 10:12','dd/mm/yyyy
hh24:mi'), 'Neurologica','AAA00000000000018',
'AAA00000000000001', '30000000008');

```

```

--HA_M
INSERT INTO ha_M (CODICE_CC, CF_MED) VALUES ('CC00000001',
'AAA000000000000003');
INSERT INTO ha_M (CODICE_CC, CF_MED) VALUES ('CC00000002',
'AAA000000000000003');
INSERT INTO ha_M (CODICE_CC, CF_MED) VALUES ('CC00000003',
'AAA000000000000005');
INSERT INTO ha_M (CODICE_CC, CF_MED) VALUES ('CC00000004',
'AAA000000000000008');
INSERT INTO ha_M (CODICE_CC, CF_MED) VALUES ('CC00000005',
'AAA000000000000008');

```

```

--DIARIO CLINICO
INSERT INTO DIARIO_CLINICO (DATA_ORA, CODICE_CC) VALUES
(TO_DATE('01/01/2019 1:00','dd/mm/yyyy hh24:mi'), 'CC00000001');
INSERT INTO DIARIO_CLINICO (DATA_ORA, CODICE_CC) VALUES
(TO_DATE('01/01/2019 1:30','dd/mm/yyyy hh24:mi'), 'CC00000002');
INSERT INTO DIARIO_CLINICO (DATA_ORA, CODICE_CC) VALUES
(TO_DATE('02/01/2019 10:00','dd/mm/yyyy hh24:mi'), 'CC00000003');
INSERT INTO DIARIO_CLINICO (DATA_ORA, CODICE_CC) VALUES
(TO_DATE('02/01/2019 14:19','dd/mm/yyyy hh24:mi'), 'CC00000004');
INSERT INTO DIARIO_CLINICO (DATA_ORA, CODICE_CC) VALUES
(TO_DATE('03/01/2019 17:20','dd/mm/yyyy hh24:mi'), 'CC00000005');

```

**--FOGLIO FISIOTERAPIA**

```

INSERT INTO FOGLIO_FISIOTERAPIA (DATA_ORA, RINF_MUSC, ELETTR_MUSC,
DEAMBULAZIONE, CF_FISIOTERAPISTA, CODICE_CC)
VALUES(TO_DATE('01/01/2019 9:00','dd/mm/yyyy hh24:mi'), 8, 7, 9,
'BBB0000000000001', 'CC00000001');
INSERT INTO FOGLIO_FISIOTERAPIA (DATA_ORA, RINF_MUSC, ELETTR_MUSC,
DEAMBULAZIONE, CF_FISIOTERAPISTA, CODICE_CC)
VALUES(TO_DATE('01/01/2019 9:30','dd/mm/yyyy hh24:mi'), 5, 1, 2,
'BBB0000000000002', 'CC00000002');
INSERT INTO FOGLIO_FISIOTERAPIA (DATA_ORA, RINF_MUSC, ELETTR_MUSC,
DEAMBULAZIONE, CF_FISIOTERAPISTA, CODICE_CC)
VALUES(TO_DATE('02/01/2019 12:00','dd/mm/yyyy hh24:mi'), 6, 6, 6,
'BBB0000000000003', 'CC00000003');
INSERT INTO FOGLIO_FISIOTERAPIA (DATA_ORA, RINF_MUSC, ELETTR_MUSC,
DEAMBULAZIONE, CF_FISIOTERAPISTA, CODICE_CC)
VALUES(TO_DATE('02/01/2019 16:22','dd/mm/yyyy hh24:mi'), 3, 10, 9,
'BBB0000000000004', 'CC00000004');
INSERT INTO FOGLIO_FISIOTERAPIA (DATA_ORA, RINF_MUSC, ELETTR_MUSC,
DEAMBULAZIONE, CF_FISIOTERAPISTA, CODICE_CC)
VALUES(TO_DATE('03/01/2019 21:52','dd/mm/yyyy hh24:mi'), 1, 1, 4,
'BBB0000000000005', 'CC00000005');

```

**--ANALISI**

```

INSERT INTO ANALISI (DATA_ORA_PRELIEVO, GLUCOSIO, CLORO, POTASSIO,
SODIO, CODICE_CC) VALUES(TO_DATE('01/01/2019 9:30','dd/mm/yyyy
hh24:mi'), 30, 45, 62, 99, 'CC00000001');
INSERT INTO ANALISI (DATA_ORA_PRELIEVO, GLUCOSIO, CLORO, POTASSIO,
SODIO, CODICE_CC) VALUES(TO_DATE('01/01/2019 10:00','dd/mm/yyyy
hh24:mi'), 77, 25, 56, 23, 'CC00000002');
INSERT INTO ANALISI (DATA_ORA_PRELIEVO, GLUCOSIO, CLORO, POTASSIO,
SODIO, CODICE_CC) VALUES(TO_DATE('02/01/2019 12:30','dd/mm/yyyy
hh24:mi'), 57, 70, 44, 21, 'CC00000003');
INSERT INTO ANALISI (DATA_ORA_PRELIEVO, GLUCOSIO, CLORO, POTASSIO,
SODIO, CODICE_CC) VALUES(TO_DATE('02/01/2019 16:50','dd/mm/yyyy
hh24:mi'), 83, 62, 37, 95, 'CC00000004');
INSERT INTO ANALISI (DATA_ORA_PRELIEVO, GLUCOSIO, CLORO, POTASSIO,
SODIO, CODICE_CC) VALUES(TO_DATE('03/01/2019 22:10','dd/mm/yyyy
hh24:mi'), 69, 87, 15, 42, 'CC00000005');

```

**--ESAMI**

```

INSERT INTO ESAMI (DATA_ORA, TIPO_ESAME, ESITO, CODICE_CC)
VALUES(TO_DATE('01/01/2019 9:30','dd/mm/yyyy hh24:mi'),
'Radiografia', 0, 'CC00000001');
INSERT INTO ESAMI (DATA_ORA, TIPO_ESAME, ESITO, CODICE_CC)
VALUES(TO_DATE('01/01/2019 13:30','dd/mm/yyyy hh24:mi'), 'Risonanza
Magnetica', 1, 'CC00000002');
INSERT INTO ESAMI (DATA_ORA, TIPO_ESAME, ESITO, CODICE_CC)
VALUES(TO_DATE('02/01/2019 14:00','dd/mm/yyyy hh24:mi'),
'Radiografia', 0, 'CC00000003');
INSERT INTO ESAMI (DATA_ORA, TIPO_ESAME, ESITO, CODICE_CC)
VALUES(TO_DATE('02/01/2019 17:55','dd/mm/yyyy hh24:mi'), 'PET', 0,
'CC00000004');
INSERT INTO ESAMI (DATA_ORA, TIPO_ESAME, ESITO, CODICE_CC)

```

```
VALUES(TO_DATE('04/01/2019 7:30','dd/mm/yyyy hh24:mi'), 'Ecografia',
0, 'CC00000005');
```

#### --SINTOMATOLOGIA

```
INSERT INTO sintomatologia_cc (DATA_ORA, SINTOMO, CODICE_CC)
VALUES(TO_DATE('01/01/2019 1:00','dd/mm/yyyy hh24:mi'), 'Diarrea',
'CC00000001');
INSERT INTO sintomatologia_cc (DATA_ORA, SINTOMO, CODICE_CC)
VALUES(TO_DATE('01/01/2019 1:30','dd/mm/yyyy hh24:mi'),
'Infiammazione', 'CC00000002');
INSERT INTO sintomatologia_cc (DATA_ORA, SINTOMO, CODICE_CC)
VALUES(TO_DATE('02/01/2019 10:00','dd/mm/yyyy hh24:mi'), 'Nausea',
'CC00000003');
INSERT INTO sintomatologia_cc (DATA_ORA, SINTOMO, CODICE_CC)
VALUES(TO_DATE('02/01/2019 14:19','dd/mm/yyyy hh24:mi'), 'Febbre',
'CC00000004');
INSERT INTO sintomatologia_cc (DATA_ORA, SINTOMO, CODICE_CC)
VALUES(TO_DATE('03/01/2019 17:20','dd/mm/yyyy hh24:mi'), 'Febbre',
'CC00000005');
```

#### --HA\_I

```
INSERT INTO HA_I(COD_CART_INF, CF_INF) VALUES ('CI77277340',
'CCC0000000000001');
INSERT INTO HA_I(COD_CART_INF, CF_INF) VALUES ('CI61623100',
'CCC0000000000002');
INSERT INTO HA_I(COD_CART_INF, CF_INF) VALUES ('CI72962004',
'CCC0000000000003');
INSERT INTO HA_I(COD_CART_INF, CF_INF) VALUES ('CI32604765',
'CCC0000000000003');
INSERT INTO HA_I(COD_CART_INF, CF_INF) VALUES ('CI78478268',
'CCC0000000000003');
```

#### --DIARIO\_INFERMIERISTICO

```
INSERT INTO DIARIO_INFERMIERISTICO(DATA_ORA, COD_CART_INF)
VALUES(TO_DATE('01/01/2019 10:03:00','dd/mm/yyyy
HH24:mi:ss'), 'CI77277340');
INSERT INTO DIARIO_INFERMIERISTICO(DATA_ORA, COD_CART_INF)
VALUES(TO_DATE('12/01/2019 12:12:00','dd/mm/yyyy
HH24:mi:ss'), 'CI61623100');
INSERT INTO DIARIO_INFERMIERISTICO(DATA_ORA, COD_CART_INF)
VALUES(TO_DATE('24/01/2019 17:36:00','dd/mm/yyyy
HH24:mi:ss'), 'CI61623100');
INSERT INTO DIARIO_INFERMIERISTICO(DATA_ORA, COD_CART_INF)
VALUES(TO_DATE('12/01/2019 14:41:00','dd/mm/yyyy
HH24:mi:ss'), 'CI72962004');
INSERT INTO DIARIO_INFERMIERISTICO(DATA_ORA, COD_CART_INF)
VALUES(TO_DATE('22/01/2019 15:00:00','dd/mm/yyyy
HH24:mi:ss'), 'CI32604765');
```

```
--sintomatologia_di
INSERT INTO sintomatologia_di(DATA_ORA, SINTOMO, COD_CART_INF)
  VALUES(TO_DATE('01/01/2019 10:03:00','dd/mm/yyyy
    HH24:mi:ss'),'Febbre', 'CI77277340');
INSERT INTO sintomatologia_di(DATA_ORA, SINTOMO, COD_CART_INF)
  VALUES(TO_DATE('12/01/2019 12:12:00','dd/mm/yyyy HH24:mi:ss'),'Dolori
    addominali', 'CI61623100');
INSERT INTO sintomatologia_di(DATA_ORA, SINTOMO, COD_CART_INF)
  VALUES(TO_DATE('24/01/2019 17:36:00','dd/mm/yyyy
    HH24:mi:ss'),'Nausea', 'CI61623100');
INSERT INTO sintomatologia_di(DATA_ORA, SINTOMO, COD_CART_INF)
  VALUES(TO_DATE('12/01/2019 14:41:00','dd/mm/yyyy
    HH24:mi:ss'),'Gonfiore', 'CI72962004');
INSERT INTO sintomatologia_di(DATA_ORA, SINTOMO, COD_CART_INF)
  VALUES(TO_DATE('22/01/2019 15:00:00','dd/mm/yyyy
    HH24:mi:ss'),'Diarrea', 'CI32604765');
```

```
--FOGLIO_TERAPIA
INSERT INTO FOGLIO_TERAPIA(DATA_ORA, METODO_SOMMINISTRA, COD_CART_INF)
  VALUES(TO_DATE('02/01/2019 11:20:00','dd/mm/yyyy HH24:mi:ss'),
    'Transcutanea', 'CI77277340');
INSERT INTO FOGLIO_TERAPIA(DATA_ORA, METODO_SOMMINISTRA, COD_CART_INF)
  VALUES(TO_DATE('26/01/2019 10:30:00','dd/mm/yyyy HH24:mi:ss'),
    'Transcutanea', 'CI77277340');
INSERT INTO FOGLIO_TERAPIA(DATA_ORA, METODO_SOMMINISTRA, COD_CART_INF)
  VALUES(TO_DATE('14/02/2019 17:15:00','dd/mm/yyyy HH24:mi:ss'),
    'Endovena', 'CI77277340');
INSERT INTO FOGLIO_TERAPIA(DATA_ORA, METODO_SOMMINISTRA, COD_CART_INF)
  VALUES(TO_DATE('18/01/2019 14:38:00','dd/mm/yyyy HH24:mi:ss'),
    'Inalatoria', 'CI61623100');
INSERT INTO FOGLIO_TERAPIA(DATA_ORA, METODO_SOMMINISTRA, COD_CART_INF)
  VALUES(TO_DATE('01/03/2019 09:02:00','dd/mm/yyyy HH24:mi:ss'),
    'Intravascolare', 'CI61623100');
```

```
--FARMACO
INSERT INTO FARMACO(COD_FARMACO, TIPO, NOME_FARMACO, ASSUNZ_GIORNALIERA)
  VALUES('FFF0000001', 'Antimicrobico', 'Bactocin', 2);
INSERT INTO FARMACO(COD_FARMACO, TIPO, NOME_FARMACO, ASSUNZ_GIORNALIERA)
  VALUES('FFF0000002', 'Catartico', 'Plantalax', 1);
INSERT INTO FARMACO(COD_FARMACO, TIPO, NOME_FARMACO, ASSUNZ_GIORNALIERA)
  VALUES('FFF0000003', 'Emetocartartico', 'Lipoxan', 1);
INSERT INTO FARMACO(COD_FARMACO, TIPO, NOME_FARMACO, ASSUNZ_GIORNALIERA)
  VALUES('FFF0000004', 'Oressizzante', 'Betotal', 3);
INSERT INTO FARMACO(COD_FARMACO, TIPO, NOME_FARMACO, ASSUNZ_GIORNALIERA)
  VALUES('FFF0000005', 'Stomachico', 'Enzitasi', 2);
```

```
--CONTIENE
INSERT INTO CONTIENE(DATA_ORA, COD_CART_INF, COD_FARMACO)
  VALUES(TO_DATE('02/01/2019 11:20:00','dd/mm/yyyy
    HH24:mi:ss'),'CI77277340', 'FFF0000001');
INSERT INTO CONTIENE(DATA_ORA, COD_CART_INF, COD_FARMACO)
```

```
VALUES(TO_DATE('26/01/2019 10:30:00','dd/mm/yyyy
HH24:mi:ss'),'CI77277340','FFF0000001');
INSERT INTO CONTIENE(DATA_ORA, COD_CART_INF, COD_FARMACO)
VALUES(TO_DATE('14/02/2019 17:15:00','dd/mm/yyyy
HH24:mi:ss'),'CI77277340','FFF0000002');
INSERT INTO CONTIENE(DATA_ORA, COD_CART_INF, COD_FARMACO)
VALUES(TO_DATE('18/01/2019 14:38:00','dd/mm/yyyy
HH24:mi:ss'),'CI61623100','FFF0000003');
INSERT INTO CONTIENE(DATA_ORA, COD_CART_INF, COD_FARMACO)
VALUES(TO_DATE('01/03/2019 09:02:00','dd/mm/yyyy
HH24:mi:ss'),'CI61623100','FFF0000004');
```

## 2.5 Trigger

I trigger impiegati in questo DB garantiscono un'emulazione relativamente fedele del funzionamento di una Clinica Riabilitativa.

### 1) Per ogni intervento devono esserci massimo 4 infermieri

Il seguente trigger conta per ogni inserimento di assistenza ad intervento di un infermiere quanti infermieri sono già assegnati a tale intervento: se vi sono già 4 infermieri, il trigger si attiva.

```
CREATE OR REPLACE TRIGGER INF2
BEFORE INSERT ON assiste
FOR EACH ROW
DECLARE
CONT_INF NUMBER;
EX1 EXCEPTION;
BEGIN
SELECT COUNT(*) INTO CONT_INF FROM assiste
WHERE ID_INTERVENTO = :NEW.ID_INTERVENTO;
IF(CONT_INF = 4) THEN
RAISE EX1;
END IF;
EXCEPTION
WHEN EX1 THEN
RAISE_APPLICATION_ERROR(-20011,'NUMERO MASSIMO DI INFERMIERI RAGGIUNTO
PER QUESTO INTERVENTO.');
```

### 2) Un medico può partecipare al massimo a 10 interventi in una settimana e di al più di tre tipi diversi

Tale trigger impone il limite massimo di partecipazione al più a 10 interventi a settimana, al massimo di 3 tipi diversi. Sull'inserimento di un medico che effettua

un intervento, vengono contati gli interventi svolti in 7 giorni dallo stesso medico, identificato dal suo codice fiscale: se questi sono già 10 o se la somma di tipi differenti è già 3, viene sollevato il trigger.

```
CREATE OR REPLACE TRIGGER max_10_int_in_una_sett
BEFORE INSERT ON EFFETTUA
FOR EACH ROW
DECLARE
n1    NUMBER(2,0):=0;
n2    NUMBER(2,0):=0;
ecc_1 EXCEPTION;
ecc_2 EXCEPTION;
BEGIN
SELECT COUNT(DISTINCT I.ID_INTERVENTO) INTO n1
FROM INTERVENTO I JOIN EFFETTUA E ON I.ID_INTERVENTO=E.ID_INTERVENTO
WHERE :NEW.CF_MED=E.CF_MED AND I.DATA_PREVISTA >= (SYSDATE-7);
SELECT COUNT(DISTINCT TIPO) INTO n2
FROM (SELECT TIPO
FROM INTERVENTO I JOIN EFFETTUA E ON I.ID_INTERVENTO=E.ID_INTERVENTO
WHERE :NEW.CF_MED=E.CF_MED AND I.DATA_PREVISTA >= (SYSDATE-7));
IF n1 > 10 THEN
RAISE ecc_1;
ELSIF n2 > 3 THEN
RAISE ecc_2;
END IF;
EXCEPTION
WHEN ecc_1 THEN
RAISE_APPLICATION_ERROR(-20097,'Troppi interventi per questo medico in
una settimana');
WHEN ecc_2 THEN
RAISE_APPLICATION_ERROR(-20096,'Troppi interventi diversi per questo
medico in una settimana');
END;
```

### 3) Ogni intervento deve avvenire a distanza di 15 giorni per ogni paziente

Tale trigger viene scatenato quando viene inserito un intervento per un paziente che ha già subito un intervento in meno di 15 giorni. Viene quindi presa la data dell'ultimo intervento (ossia quella col valore massimo) e viene verificato che non sia NULL (cioè se il paziente è già stato precedentemente operato, dunque che esiste una data precedente), se non è NULL, si verifica se sia una data risalente a meno di 15 giorni dalla data dell'intervento che si sta inserendo, in tal caso viene evocata l'eccezione.

```
CREATE OR REPLACE TRIGGER TR_INT
BEFORE INSERT ON INTERVENTO
FOR EACH ROW
DECLARE
DIM DIMISSIONE.ID_DIMISSIONE%TYPE;
```

```

LAST_INT DATE;
C NUMBER;
EX1 EXCEPTION;
EX2 EXCEPTION;
BEGIN
SELECT COUNT(*) INTO C FROM DIMISSIONE D
WHERE D.ID_RICOVERO = :NEW.ID_RICOVERO;
IF(C <> 0) THEN
RAISE EX1;
END IF;
SELECT MAX(DATA_EFFETTIVA) INTO LAST_INT FROM INTERVENTO I
WHERE I.CF_PAZ = :NEW.CF_PAZ;
IF(LAST_INT IS NOT NULL) THEN
IF(:NEW.DATA_PREVISTA - LAST_INT < 15) THEN
RAISE EX2;
END IF;
END IF;
EXCEPTION
WHEN EX1 THEN
RAISE_APPLICATION_ERROR(-20031, 'IL PAZIENTE PER IL RICOVERO INSERITO è
STATO DIMESSO.');
```

#### 4) Un infermiere può partecipare al massimo a 2 interventi in un giorno

Con questo trigger ci si accerta che ad ogni inserimento di assistenza ad un intervento per un infermiere, ogni infermiere possa partecipare al più 2 interventi eseguiti lo stesso giorno.

```

CREATE OR REPLACE TRIGGER max_2_interventi
BEFORE INSERT ON ASSISTE
FOR EACH ROW
DECLARE
number_1 NUMBER(2,0);
eccezione_1 EXCEPTION;
BEGIN
SELECT COUNT(ID_INTERVENTO) INTO number_1 FROM ASSISTE
WHERE :NEW.CF_INF=CF_INF;
IF number_1 > 2 THEN
RAISE eccezione_1;
END IF;
EXCEPTION
WHEN eccezione_1 THEN
RAISE_APPLICATION_ERROR(-20041, 'Troppi interventi nello stesso giorno
per questo infermierie.');
```



**5) La data di intervento su un paziente non può essere precedente alla data del suo ricovero**

Tale trigger garantisce la correttezza della data d'inserimento di un intervento in modo tale da non creare paradossi, quindi che la data di intervento sia successiva alla data di ricovero del paziente.

```
CREATE OR REPLACE TRIGGER DATAINT
BEFORE INSERT ON INTERVENTO
FOR EACH ROW
DECLARE
TIMEPARADOX EXCEPTION;
D DATE;
BEGIN
SELECT DATA_RICOVERO INTO D FROM RICOVERO R
WHERE R.ID_RICOVERO = :NEW.ID_RICOVERO; IF(:NEW.DATA_PREVISTA < D) THEN
RAISE TIMEPARADOX;
END IF;
EXCEPTION
WHEN TIMEPARADOX THEN
RAISE_APPLICATION_ERROR(-20051, 'LA DATA DI INTERVENTO PRECEDE LA DATA DI
RICOVERO');
END;
```

**6) La data di dimissione di un paziente non può essere precedente alla data di ricovero**

Tale trigger garantisce la correttezza della data d'inserimento della dimissione di un paziente in modo tale da non creare paradossi, quindi che la data di dimissione sia successiva alla data di ricovero del paziente.

```
CREATE OR REPLACE TRIGGER DATADIMIS
BEFORE INSERT ON DIMISSIONE
FOR EACH ROW
DECLARE
TIMEPARADOX2 EXCEPTION;
D DATE;
BEGIN
SELECT DATA_RICOVERO INTO D FROM RICOVERO WHERE
ID_RICOVERO = :NEW.ID_RICOVERO;
IF(:NEW.DATA_ORA < D) THEN
RAISE TIMEPARADOX2;
END IF;
EXCEPTION
WHEN TIMEPARADOX2 THEN
RAISE_APPLICATION_ERROR(-20061, 'LA DATA DI DIMISSIONE PRECEDE LA DATA DI
RICOVERO');
END;
```

### 7) In una sala operatoria è possibile effettuare massimo 3 interventi in un giorno

Il seguente trigger viene scatenato quando, all'inserimento di un intervento, nello stesso giorno vi sono già 3 interventi nella sala operatoria dell'intervento che si sta inserendo.

```
CREATE OR REPLACE TRIGGER MAXSALAOP
BEFORE INSERT ON INTERVENTO
FOR EACH ROW
DECLARE
TOO_OP EXCEPTION;
COUNT_OP NUMBER(1,0);
BEGIN
SELECT COUNT(*) INTO COUNT_OP FROM INTERVENTO I WHERE
:NEW.DATA_PREVISTA = TRUNC(I.DATA_PREVISTA) AND :NEW.NUM_SALA_OPERA =
I.NUM_SALA_OPERA;
IF COUNT_OP > 3 THEN
RAISE TOO_OP;
END IF;
EXCEPTION
WHEN TOO_OP THEN
RAISE_APPLICATION_ERROR(-20071, 'CI SONO GIA'' PIU'' DI 3 INTERVENTI IN
QUESTA SALA OPERATORIA PER QUESTO GIORNO');
END;
```

### 8) Ogni paziente ha una sola dieta dalla data di ricovero alla data di dimissione

Questo trigger fa sì che venga rispettata la cardinalità dell'associazione tra dieta e ricovero del paziente, tale per ogni ricovero può avere soltanto una dieta.

```
CREATE OR REPLACE TRIGGER TR_DIETA
BEFORE INSERT ON DIETA
FOR EACH ROW
DECLARE
DIM DIMISSIONE.ID_DIMISSIONE%TYPE;
C NUMBER; EX1 EXCEPTION;
EX2 EXCEPTION;
BEGIN
SELECT ID_DIMISSIONE INTO DIM FROM DIMISSIONE
WHERE ID_RICOVERO = :NEW.ID_RICOVERO;
IF(DIM IS NOT NULL) THEN
RAISE EX1;
END IF;
SELECT COUNT(*) INTO C FROM DIETA D
WHERE D.ID_RICOVERO = :NEW.ID_RICOVERO;
IF(C <> 0) THEN
RAISE EX2;
END IF;
EXCEPTION
WHEN EX1 THEN
```

```

RAISE_APPLICATION_ERROR(-20081, 'IL PAZIENTE INSERITO è GIA STATO
    DIMESSO.');
```

WHEN EX2 THEN

```

RAISE_APPLICATION_ERROR(-20082, 'IL PAZIENTE INSERITO HA GIA UNA DIETA
    ASSEGNATA.');
```

END;

### 9) Gli interventi in ogni sala operatoria non si possono accavallare e solo nella sala 14 si fanno interventi superiori alle 6 ore

Questo trigger fa sì che gli interventi possono essere effettuati solo tra le 8 e le 20 e due o più interventi non possono avere la stessa sala alla stessa ora. Inoltre, solo all'interno della sala operatoria 14 è possibile effettuare interventi di durata superiore alle 6 ore.

```

CREATE OR REPLACE TRIGGER dist_1_ora
BEFORE INSERT ON INTERVENTO
FOR EACH ROW
DECLARE
num_1 NUMBER(1,0) := 0;
ecc_1 EXCEPTION;
ecc_2 EXCEPTION;
ecc_3 EXCEPTION;
BEGIN
IF TO_NUMBER(TO_CHAR(:NEW.DATA_PREVISTA, 'HH24')) < 8 OR
   TO_NUMBER(TO_CHAR(:NEW.DATA_EFFETTIVA, 'HH24')) < 8
   OR TO_NUMBER(TO_CHAR(:NEW.DATA_PREVISTA, 'HH24')) > 20 OR
   TO_NUMBER(TO_CHAR(:NEW.DATA_EFFETTIVA, 'HH24')) > 20 THEN
RAISE ecc_1;
ELSIF :NEW.DURATA > 6 AND :NEW.NUM_SALA_OPERA <> 14 THEN RAISE ecc_2;
END IF; SELECT COUNT(ID_INTERVENTO) INTO num_1 FROM (SELECT * FROM
INTERVENTO WHERE
   TO_CHAR(:NEW.DATA_EFFETTIVA, 'DD/MM/YYYY')=TO_CHAR(DATA_EFFETTIVA,
'DD/MM/YYYY') AND:NEW.NUM_SALA_OPERA=NUM_SALA_OPERA) WHERE
   TO_NUMBER(TO_CHAR(:NEW.DATA_EFFETTIVA, 'HH24')) >
   (TO_NUMBER(TO_CHAR(DATA_EFFETTIVA, 'HH24')) + DURATA); IF num_1 > 0
THEN
RAISE ecc_3;
END IF;
EXCEPTION
WHEN ecc_1 THEN
RAISE_APPLICATION_ERROR(-20091, 'Inserito orario sbagliato');
WHEN ecc_2 THEN
RAISE_APPLICATION_ERROR(-20092, 'Nella sala operatoria numero 14 si
    possono effettuare solo interventi di durata > 6');
WHEN ecc_3 THEN
RAISE_APPLICATION_ERROR(-20093, 'L''intervento da inserire si accavalla
    con un altro intervento');
END;
```

**10 - 11) Un fisioterapista non può effettuare né assistere ad interventi**

Questi trigger impediscono che venga assegnata l'esecuzione o l'assistenza ad un intervento ad un fisioterapista.

```
CREATE OR REPLACE TRIGGER no_eff_fisioterapista
BEFORE INSERT ON EFFETTUA
FOR EACH ROW
DECLARE
num    NUMBER:=0;
ecc1   EXCEPTION;
BEGIN
SELECT COUNT(CF_MED) INTO num FROM MEDICO
WHERE :NEW.CF_MED=CF_MED;
IF num = 1 THEN
RAISE ecc1;
END IF;
EXCEPTION
WHEN ecc1 THEN
RAISE APPLICATION ERROR(-20334, 'NON SI PUO ASSEGNARE UN INTERVENTO AD
    UN FISIOTERAPISTA');
END;
```

```
CREATE OR REPLACE TRIGGER no_int_fisioterapista
BEFORE INSERT ON ASSISTE
FOR EACH ROW
DECLARE
num    NUMBER:=0;
ecc1   EXCEPTION;
BEGIN
SELECT COUNT(CF_FISIO) INTO num FROM FISIOTERAPISTA
WHERE :NEW.CF_INF=CF_FISIO;
IF num = 1 THEN
RAISE ecc1;
END IF;
EXCEPTION
WHEN ecc1 THEN
RAISE APPLICATION ERROR(-10335, 'UN FISIOTERAPISTA NON PUO ASSISTERE UN
    INTERVENTO');
END;
```

**12) Un paziente non può essere tenuto in due reparti contemporaneamente**

Il trigger viene sollevato quando si tenta di assegnare in un certo reparto un paziente che al momento dell'immissione risulta già ricoverato in un altro reparto.

```
CREATE OR REPLACE TRIGGER no_stesso_rep
BEFORE INSERT ON E_TENUTO
FOR EACH ROW
DECLARE
num    NUMBER(2,0):=0;
```

```

cf_pazx CHAR(16);
ecc1 EXCEPTION;
BEGIN
SELECT CF_PAZ INTO cf_pazx FROM
E_TENUTO ET JOIN RICOVERO R ON ET.ID_RICOVERO=R.ID_RICOVERO
WHERE :NEW.ID_RICOVERO=ET.ID_RICOVERO;
SELECT COUNT(R.CF_PAZ) INTO num FROM
E_TENUTO ET JOIN RICOVERO R ON ET.ID_RICOVERO=R.ID_RICOVERO
WHERE :NEW.DATA_ORA=ET.DATA_ORA AND R.CF_PAZ=cf_pazx;
IF num > 1 THEN
RAISE ecc1;
END IF;
EXCEPTION
WHEN ecc1 THEN
RAISE_APPLICATION_ERROR(-20093,'Il paziente non può stare
contemporaneamente in due reparti');
END;

```

### 13) Durata massima degli interventi per i pazienti anziani

Questo trigger impedisce che i pazienti di età superiore a 75 anni vengano sottoposti ad interventi di durata maggiore alle 3 ore.

```

CREATE OR REPLACE TRIGGER durata_int_paz_75
BEFORE INSERT ON INTERVENTO
FOR EACH ROW
DECLARE
datex DATE;
ecc1 EXCEPTION;
BEGIN
SELECT DATA_DI_NASCITA INTO datex FROM PERSONA
WHERE :NEW.CF_PAZ=CF;
IF :NEW.DURATA_PREVISTA > 3 AND ((TO_NUMBER(TO_CHAR(SYSDATE,'yyyy')) -
TO_NUMBER(TO_CHAR(datex,'yyyy')) > 75 THEN
RAISE ecc1;
END IF;
EXCEPTION
WHEN ecc1 THEN
RAISE_APPLICATION_ERROR(-20092,'L''età del paziente è superiore a 75
anni. L''intervento non può durare più di 3 ore.');
```

### 14) Un medico non può assistere un intervento

Un intervento può essere effettuato da più medici, ma la mansione di assistenza è assegnabile ai soli infermieri. Questo trigger impedisce che sia un medico ad essere registrato come assistente per l'intervento.

```

CREATE OR REPLACE TRIGGER no_assiste_int
BEFORE INSERT ON ASSISTE
FOR EACH ROW
DECLARE

```

```

num    NUMBER(2,0):=0;
ecc1   EXCEPTION;
BEGIN
SELECT COUNT(CF_INF) INTO num FROM INFERMIERE
WHERE :NEW.CF_INF=CF_INF;
IF num > 0 THEN
RAISE ecc1;
END IF;
EXCEPTION
WHEN ecc1 THEN
RAISE_APPLICATION_ERROR(-20094,'Si sta tentando di far assistere un
medico ad un intervento');
END;

```

### 15) Un infermiere non può eseguire un intervento

Solo i medici possono eseguire interventi. Questo trigger impedisce che l'esecuzione di un intervento venga assegnata ad un infermiere.

```

CREATE OR REPLACE TRIGGER no_int_infermiere
BEFORE INSERT ON EFFETTUA
FOR EACH ROW
DECLARE
num    NUMBER(2,0):=0;
ecc1   EXCEPTION;
BEGIN
SELECT COUNT(CF_INF) INTO num FROM INFERMIERE
WHERE :NEW.CF_MED=CF_INF;
IF num > 0 THEN
RAISE ecc1;
END IF;
EXCEPTION
WHEN ecc1 THEN
RAISE_APPLICATION_ERROR(-20095,'Si sta tentando di far effettuare un
intervento ad un infermiere');
END;

```

### 16) Riprogrammazione delle visita

Questo trigger si attiva quando viene assegnato un intervento ad un medico, e controlla se nella data dell'intervento il medico ha già una visita programmata: in caso affermativo, dato che gli interventi hanno la priorità sulle visite e un medico non può effettuare visite in un giorno in cui esegue almeno un intervento, tutte le sue visite vengono riprogrammate per il giorno successivo.

```

CREATE OR REPLACE TRIGGER INT_VIS
AFTER INSERT ON EFFETTUA
FOR EACH ROW
DECLARE
C      INTEGER;
CF     MEDICO.CF_MED%TYPE;

```

```

DATA_INT DATE;
BEGIN
SELECT DATA_PREVISTA INTO DATA_INT FROM INTERVENTO
WHERE ID_INTERVENTO = :NEW.ID_INTERVENTO;
--Controllo se un medico ha delle visite programmate per il giorno dell
  intervento
SELECT COUNT(*) INTO C FROM VISITA
WHERE CF_MED = :NEW.CF_MED AND to_date(DATA_ORA_PREVISTA, 'dd-mm-yyyy')
  = to_date(DATA_INT, 'dd-mm-yyyy');
DBMS_OUTPUT.PUT_LINE('IL MEDICO IN DATA ' || DATA_INT || ' HA ' || C || '
  VISITE PROGRAMMATE');
--Se ci sono visite prenotate per la data dell intervento, dato che gli
  interventi hanno la priorità,
--tutte le visite del medico programmate per quel giorno vengono
  riprogrammate per il giorno dopo
IF(C != 0) THEN
FOR V IN (SELECT * FROM VISITA
WHERE CF_MED = :NEW.CF_MED AND to_date(DATA_ORA_PREVISTA, 'dd-mm-yyyy')
  = to_date(DATA_INT, 'dd-mm-yyyy'))
LOOP
UPDATE VISITA
SET DATA_ORA_PREVISTA = DATA_ORA_PREVISTA + 1;
END LOOP;
END IF;
END;

```

### 17) ID progressivi

Le entità RICOVERO, INTERVENTO e DIMISSIONE utilizzano degli ID come chiavi primarie artificiali; questi ID vengono generati progressivamente tramite SEQUENCE, dichiarati in fase di costruzione del DB. I tre trigger che seguono assegnano l'ID alla relativa tupla inserita in una delle 3 tabelle sopra citate utilizzando il valore progressivo della relativa sequence.

```

CREATE OR REPLACE TRIGGER ins_ricovero
BEFORE INSERT ON RICOVERO
FOR EACH ROW
BEGIN
SELECT seq_ricovero.nextval INTO :NEW.ID_RICOVERO FROM DUAL;
END;

```

```

CREATE OR REPLACE TRIGGER ins_intervento
BEFORE INSERT ON INTERVENTO
FOR EACH ROW
BEGIN
SELECT seq_intervento.nextval INTO :NEW.ID_INTERVENTO FROM DUAL;
END;

```

```

CREATE OR REPLACE TRIGGER ins_dimissione
BEFORE INSERT ON DIMISSIONE
FOR EACH ROW
BEGIN

```

```
SELECT seq_dimissione.nextval INTO :NEW.ID_DIMISSIONE FROM DUAL;
END;
```

## 2.6 Procedure

Di seguito le procedure progettate per facilitare alcune delle possibili operazioni base effettuabili sul DB di una clinica riabilitativa.

### 1) Visualizzazione delle informazioni di un ricovero

Questa procedura, dato in input l'ID di un ricovero, ritorna tutte le informazioni relative ad interventi, analisi ed esami effettuati sul paziente.

```
CREATE OR REPLACE PROCEDURE VISUALIZZA_PAZIENTE (ID_RIC NUMBER)
IS
  --cartella clinica
  CART_CLIN CHAR(10);
  --intervento
  DATA_INTERVENTO DATE;
  TIPO_INTERVENTO VARCHAR(20);
  DURATA_INTERVENTO NUMBER;
  --analisi
  CLORO_ANALISI NUMBER;
  GLUCOSIO_ANALISI NUMBER;
  POTASSIO_ANALISI NUMBER;
  SODIO_ANALISI NUMBER;
  --esame
  DATA_ESAME DATE;
  TIPO_ESAME2 VARCHAR(30);
  ESITO_ESAME NUMBER;
  BEGIN
    --ottengo il codice della cartella clinica
    SELECT CODICE_CC INTO CART_CLIN FROM CARTELLA_CLINICA
    WHERE ID_RICOVERO = ID_RIC;
    --ottengo i dati da intervento
    SELECT DATA_EFFETTIVA, TIPO, DURATA_EFFETTIVA INTO DATA_INTERVENTO,
      TIPO_INTERVENTO, DURATA_INTERVENTO FROM INTERVENTO I
    WHERE ID_RIC = I.ID_RICOVERO AND DATA_PREVISTA = (SELECT
      MAX(DATA_PREVISTA) FROM INTERVENTO I WHERE I.ID_RICOVERO = ID_RIC);
    DBMS_OUTPUT.PUT_LINE('ULTIMO INTERVENTO EFFETTUATO: ' || DATA_INTERVENTO
      || ' ' || TIPO_INTERVENTO || ' ' || DURATA_INTERVENTO);
    --ottengo i dati da analisi
    SELECT GLUCOSIO, CLORO, POTASSIO, SODIO INTO GLUCOSIO_ANALISI,
      CLORO_ANALISI, POTASSIO_ANALISI, SODIO_ANALISI FROM ANALISI
    WHERE CODICE_CC = CART_CLIN AND DATA_ORA_PRELIEVO = (SELECT
      MAX(DATA_ORA_PRELIEVO) FROM ANALISI A WHERE A.CODICE_CC = CART_CLIN);
    DBMS_OUTPUT.PUT_LINE('ULTIMA ANALISI EFFETTUATA:');
    DBMS_OUTPUT.PUT_LINE('GLUCOSIO: ' || GLUCOSIO_ANALISI);
    DBMS_OUTPUT.PUT_LINE('CLORO: ' || CLORO_ANALISI);
    DBMS_OUTPUT.PUT_LINE('POTASSIO: ' || POTASSIO_ANALISI);
```



```

DBMS_OUTPUT.PUT_LINE('SODIO: ' || SODIO_ANALISI);
--ottengo i dati da esame
SELECT DATA_ORA, TIPO_ESAME, ESITO INTO DATA_ESAME, TIPO_ESAME2,
       ESITO_ESAME FROM ESAMI
WHERE CODICE_CC = CART_CLIN AND DATA_ORA = (SELECT MAX(DATA_ORA) FROM
       ESAMI E WHERE E.CODICE_CC = CART_CLIN);
DBMS_OUTPUT.PUT_LINE('ULTIMO ESAME EFFETTUATO:' || DATA_ESAME || ' ' ||
       TIPO_ESAME2 || ' ' || ESITO_ESAME);
END;

```

## 2) Visualizzazione degli interventi di un paziente

Questa è l'unica procedura su cui un utente paziente possiede permesso di esecuzione. Dato il suo codice fiscali in input, il paziente può vedere l'elenco degli interventi a cui è stato o sarà sottoposto.

```

CREATE OR REPLACE PROCEDURE VISUALIZZA_INT_PAZ(cf_pazx CHAR)
AS
C1 SYS_REFCURSOR;
BEGIN
OPEN C1 FOR
SELECT *
FROM INTERVENTO
WHERE CF_PAZ=cf_pazx;
DBMS_SQL.RETURN_RESULT(C1);
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Questo paziente non ha interventi.');
```

## 3) Visualizzazione di CF, nome, cognome e età dei pazienti in una stanza di un reparto

```

CREATE OR REPLACE PROCEDURE VIS_PAZ_STANZ_REP(NUMERO_S NUMBER, NOM_REP
       VARCHAR)
IS
C1 SYS_REFCURSOR;
BEGIN
OPEN C1 FOR
SELECT CF_PAZ, NOME, COGNOME, (TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')) -
       TO_NUMBER(TO_CHAR(P.DATA_DI_NASCITA, 'YYYY'))) AS ETA
FROM REPARTO R JOIN E_TENUTO ED ON R.NOME_REPARTO=ED.NOME_REPARTO JOIN
       RICOVERO RI ON ED.ID_RICOVERO=RI.ID_RICOVERO JOIN PERSONA P ON
       RI.CF_PAZ=P.CF
WHERE R.NOME_REPARTO=NOM_REP AND N_STANZE=NUMERO_S;
DBMS_SQL.RETURN_RESULT(C1);
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Non ci sono pazienti.');
```

#### 4) Visualizzazione dell'ultimo Foglio Fisioterapia e degli ultimi Esami di un paziente

```

CREATE OR REPLACE PROCEDURE VIS_FOGLIO_FISIO_E_ESAMI_PAZ(cf_pazx CHAR)
AS
C1 SYS_REFCURSOR;
C2 SYS_REFCURSOR;
BEGIN
--Visualizzo l'ultimo Foglio Fisioterapia
OPEN C1 FOR
SELECT
    FF.CF_FISIOTERAPISTA,FF.CODICE_CC,FF.DEAMBULAZIONE,FF.ELETTR_MUSC,FF.RINF_MUSC
FROM FOGLIO_FISIOTERAPIA FF JOIN CARTELLA_CLINICA CC ON
    FF.CODICE_CC=CC.CODICE_CC JOIN RICOVERO R ON
    CC.ID_RICOVERO=R.ID_RICOVERO
WHERE R.CF_PAZ=cf_pazx AND FF.DATA_ORA = (SELECT MAX(FF2.DATA_ORA)
FROM FOGLIO_FISIOTERAPIA FF2 JOIN CARTELLA_CLINICA CC2 ON
    FF2.CODICE_CC=CC2.CODICE_CC JOIN RICOVERO R2 ON
    CC2.ID_RICOVERO=R2.ID_RICOVERO
WHERE R2.CF_PAZ=cf_pazx);
DBMS_SQL.RETURN_RESULT(C1);
--Visualizzo gli ultimi esami
OPEN C2 FOR
SELECT E.CODICE_CC,E.TIPO_ESAME,E.ESITO
FROM RICOVERO R JOIN CARTELLA_CLINICA CC ON CC.ID_RICOVERO=R.ID_RICOVERO
    JOIN ESAMI E ON CC.CODICE_CC=E.CODICE_CC
WHERE R.CF_PAZ=cf_pazx AND E.DATA_ORA = (SELECT MAX(E.DATA_ORA)
FROM RICOVERO R2 JOIN CARTELLA_CLINICA CC2 ON
    R2.ID_RICOVERO=CC2.ID_RICOVERO JOIN ESAMI E2 ON
    E2.CODICE_CC=CC2.CODICE_CC
WHERE R2.CF_PAZ=cf_pazx);
DBMS_SQL.RETURN_RESULT(C2);
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Dati non presenti');
END;

```

#### 5) Visualizzazione degli esami di un paziente

```

CREATE OR REPLACE PROCEDURE VIS_ESAMI_PAZ(cf_pazx CHAR)
AS
C1 SYS_REFCURSOR;
BEGIN
OPEN C1 FOR
SELECT E.CODICE_CC,E.DATA_ORA,E.TIPO_ESAME,E.ESITO
FROM ESAMI E JOIN CARTELLA_CLINICA CC ON E.CODICE_CC=CC.CODICE_CC JOIN
    RICOVERO R ON R.ID_RICOVERO=CC.ID_RICOVERO
WHERE R.CF_PAZ=cf_pazx;
DBMS_SQL.RETURN_RESULT(C1);
EXCEPTION
WHEN NO_DATA_FOUND THEN

```

```
DBMS_OUTPUT.PUT_LINE('Dati non presenti');
END;
```

## 6) Visualizzazione delle analisi di un paziente

```
CREATE OR REPLACE PROCEDURE VIS_ANALISI_PAZ(CF_PAZx CHAR)
AS
C1 SYS_REFCURSOR;
BEGIN
OPEN C1 FOR
SELECT DATA_ORA_PRELIEVO, GLUCOSIO, CLORO, POTASSIO, SODIO
FROM ANALISI A JOIN CARTELLA_CLINICA CC ON A.CODICE_CC=CC.CODICE_CC JOIN
    RICOVERO R ON CC.ID_RICOVERO=R.ID_RICOVERO
WHERE R.CF_PAZ=CF_PAZx;
DBMS_SQL.RETURN_RESULT(C1);
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Questo paziente non ha analisi');
END;
```

## 7) Dimissione di un paziente

Questa procedura viene utilizzata per dimettere un paziente. La dimissione viene accettata solo se un paziente è stato prima sottoposto ad esami per accertamenti.

```
CREATE OR REPLACE PROCEDURE TRY_DIM (id_ric NUMBER, cod_dia NUMBER,
    mod_dim NUMBER)
IS
CLORO_ANALISI      NUMBER;
GLUCOSIO_ANALISI   NUMBER;
POTASSIO_ANALISI    NUMBER;
SODIO_ANALISI      NUMBER;
ID                  NUMBER;
num                 NUMBER:=0;
ecc1                EXCEPTION;
BEGIN
SELECT CLORO, GLUCOSIO, POTASSIO, SODIO INTO CLORO_ANALISI,
    GLUCOSIO_ANALISI, POTASSIO_ANALISI, SODIO_ANALISI
FROM(SELECT CODICE_CC, CLORO, GLUCOSIO, POTASSIO, SODIO
FROM ANALISI
WHERE CODICE_CC = ( SELECT CODICE_CC
FROM CARTELLA_CLINICA
WHERE ID_RICOVERO = id_ric)
ORDER BY DATA_ORA_PRELIEVO)
WHERE ROWNUM=1;
IF CLORO_ANALISI > 60 AND GLUCOSIO_ANALISI > 60 AND POTASSIO_ANALISI >
    60 AND SODIO_ANALISI > 60 THEN
SELECT seq_dimissione.NEXTVAL INTO ID FROM DUAL;
INSERT INTO
    DIMISSIONE(ID_DIMISSIONE, DATA_ORA, CODICE_DIAGNOSI, MOD_DIMISSIONE, ID_RICOVERO)
VALUES (ID, SYSDATE, cod_dia, mod_dim, id_ric);
```

```

END IF;
SELECT COUNT(ID_RICOVERO) INTO num
FROM DIMISSIONE
WHERE ID_RICOVERO=id_ric;
IF num > 1 THEN
RAISE ecc1;
END IF;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RAISE APPLICATION ERROR(-11500, 'Non ci sono esami effettuati.');
```

```

WHEN ecc1 THEN
RAISE APPLICATION ERROR(-11501, 'Un paziente non può avere due
    dimissioni dello stesso ricovero.');
```

```

END;
```

### 8) Programmazione di un intervento

Questa procedura è utilizzata per la programmazione di un intervento da parte di un dato medico in una certa data. Prende in input dei dati che non può calcolare autonomamente, come il tipo di intervento e la durata prevista, e si occupa dei rimanenti aspetti quali l'assegnazione degli infermieri (numero di infermieri determinato dalla durata prevista dell'intervento) o, in casi particolari, l'affiancamento di un ulteriore medico.

```

CREATE OR REPLACE PROCEDURE PROGRAMMAZIONE_INTERVENTO
(CF_MED IN VARCHAR, CF_PAZ IN VARCHAR, SALA_OP IN INTEGER,
    TIPO_INTERVENTO IN VARCHAR,
    DURATA IN NUMBER, DATA_PREVISTA IN VARCHAR)
IS
DATA_PREV DATE := to_date(DATA_PREVISTA, 'dd-mm-yyyy HH24:mi:ss');
ID_RIC    RICOVERO.ID_RICOVERO%TYPE;
COUNT_SO INTEGER;
COUNT_MED INTEGER;
C_DIM     INTEGER;
DATA_RIC  DATE;
CF_NEW_MED PERSONA.CF%TYPE;
CF_NEW_INF PERSONA.CF%TYPE;
SPEC      MEDICO.SPECIALIZZAZIONE%TYPE;
ID_INT    INTERVENTO.ID_INTERVENTO%TYPE;
IND       INTEGER;
EX1       EXCEPTION;
EX2       EXCEPTION;
EX3       EXCEPTION;
EX4       EXCEPTION;
EX5       EXCEPTION;
EX6       EXCEPTION;
BEGIN
--Medico e paziente non possono coincidere
IF(CF_PAZ = CF_MED) THEN
RAISE EX1;
END IF;
--La data dell'intervento non può essere precedente a quella del ricovero
```

```

SELECT MAX(DATA_RICOVERO) INTO DATA_RIC FROM RICOVERO R
WHERE R.CF_PAZ = CF_PAZ;
IF(DATA_PREV < DATA_RIC) THEN
RAISE EX2;
END IF;
--In una sala operatoria non possono esserci più di 3 interventi al
giorno
SELECT COUNT(*) INTO COUNT_SO FROM INTERVENTO I
WHERE I.NUM_SALA_OPERA = SALA_OP
AND I.DATA_PREVISTA = DATA_PREV;
IF(COUNT_SO = 3) THEN
RAISE EX3;
END IF;
--Un medico non può essere assegnato a più di 2 interventi al giorno
COUNT_MED := COUNT_INT(CF_MED, DATA_PREV);
IF(COUNT_MED = 2) THEN
RAISE EX4;
END IF;
--Calcolo ID intervento
SELECT seq_intervento.NEXTVAL INTO ID_INT FROM DUAL;
DBMS_OUTPUT.PUT_LINE('ID INTERVENTO: ' || ID_INT);
--Calcolo ID ricovero
SELECT ID_RICOVERO INTO ID_RIC FROM RICOVERO
WHERE DATA_RICOVERO = (SELECT MAX(DATA_RICOVERO) FROM RICOVERO R
WHERE R.CF_PAZ = CF_PAZ);
--Non si può effettuare un intervento su un paziente che è stato dimesso
SELECT COUNT(*) INTO C_DIM FROM DIMISSIONE
WHERE ID_RICOVERO = ID_RIC;
IF(C_DIM <> 0) THEN
RAISE EX5;
END IF;
--Creazione dell'intervento
INSERT INTO INTERVENTO(ID_INTERVENTO, DATA_PREVISTA, DATA_EFFETTIVA,
TIPO, DURATA_PREVISTA, DURATA_EFFETTIVA, NUM_SALA_OPERA, ID_RICOVERO,
CF_PAZ)
VALUES (ID_INT, DATA_PREV, NULL, TIPO_INTERVENTO, DURATA, NULL, SALA_OP,
ID_RIC, CF_PAZ);
--Assegnazione al medico
INSERT INTO EFFETTUA(CF_MED, ID_INTERVENTO) VALUES(CF_MED, ID_INT);
--Se la durata dell'intervento è maggiore a 6 ore si assegna un altro
medico con la stessa specializzazione, se disponibile
IF(DURATA > 6) THEN
SELECT SPECIALIZZAZIONE INTO SPEC FROM MEDICO M
WHERE M.CF_MED = CF_MED;
SELECT CF_MED INTO CF_NEW_MED FROM MEDICO M
WHERE M.SPECIALIZZAZIONE = SPEC AND COUNT_INT(M.CF_MED, DATA_PREV) < 2;
INSERT INTO EFFETTUA(CF_MED, ID_INTERVENTO) VALUES(CF_NEW_MED, ID_INT);
END IF;
--Assegnazione degli infermieri in base alla durata dell'intervento:
--da 1 a 3 ore: 2 infermieri;
--da 4 a 5 ore: 3 infermieri;
--da 6 ore in su: 4 infermieri.
IF(DURATA > 0 AND DURATA < 4) THEN
IND := 2;
ELSIF(DURATA > 3 AND DURATA < 6) THEN
IND := 3;

```

```

ELSE
IND := 4;
END IF;
FOR A IN 1..IND LOOP
CF_NEW_INF := GET_INF(DATA_PREV);
DBMS_OUTPUT.PUT_LINE('CF INFERMIERE: ' || CF_NEW_INF);
INSERT INTO ASSISTE(CF_INF, ID_INTERVENTO) VALUES(CF_NEW_INF, ID_INT);
CF_NEW_INF := '';
END LOOP;
EXCEPTION
WHEN EX1 THEN
RAISE_APPLICATION_ERROR(-20050, 'MEDICO E PAZIENTE NON POSSONO
    COINCIDERE');
WHEN EX2 THEN
RAISE_APPLICATION_ERROR(-20051, 'LA DATA INTERVENTO RISULTA PRECEDENTE
    ALLA DATA DI RICOVERO');
WHEN EX3 THEN
RAISE_APPLICATION_ERROR(-20052, 'LA SALA OPERATORIA HA RAGGIUNTO IL
    LIMITE GIORNALIERO DI INTERVENTI NELLA DATA INSERITA');
WHEN EX4 THEN
RAISE_APPLICATION_ERROR(-20053, 'IL MEDICO HA RAGGIUNTO IL LIMITE
    GIORNALIERO DI INTERVENTI NELLA DATA INSERITA');
WHEN EX5 THEN
RAISE_APPLICATION_ERROR(-20055, 'IL PAZIENTE RISULTA DIMESSO');
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20054, 'NON SONO DISPONIBILI ALTRI MEDICI PER
    EFFETTUARE L INTERVENTO');
END;

```

### 9) Programmazione di un intervento in base ai valori delle analisi

Questa procedura, dato l'ID ricovero di un paziente, recupera i risultati delle sue ultime analisi effettuate e, se i valori ottenuti sono inferiori ad una certa soglia, programma un intervento per il paziente richiamando la procedura 8.

```

CREATE OR REPLACE PROCEDURE PROG_INT_ANALISI (cf_med CHAR, id_ric
    NUMBER, data_prev DATE, tipo VARCHAR, durata_prev NUMBER, num_sala
    NUMBER)
IS
CLORO_ANALISI    NUMBER;
GLUCOSIO_ANALISI NUMBER;
POTASSIO_ANALISI NUMBER;
SODIO_ANALISI    NUMBER;
ID               CHAR(10);
CF_PAZ2         CHAR(16);
BEGIN
SELECT CLORO, GLUCOSIO, POTASSIO, SODIO INTO CLORO_ANALISI,
    GLUCOSIO_ANALISI, POTASSIO_ANALISI, SODIO_ANALISI
FROM(SELECT CODICE_CC, CLORO, GLUCOSIO, POTASSIO, SODIO
FROM ANALISI
WHERE CODICE_CC = ( SELECT CODICE_CC
FROM CARTELLA_CLINICA
WHERE ID_RICOVERO = id_ric)
ORDER BY DATA_ORA_PRELIEVO)

```

```

WHERE ROWNUM=1;
IF CLORO_ANALISI < 40 AND GLUCOSIO_ANALISI < 40 AND POTASSIO_ANALISI <
  40 AND SODIO_ANALISI < 40 THEN
SELECT CF_PAZ INTO CF_PAZ2 FROM RICOVERO
WHERE RICOVERO.ID_RICOVERO = id_ric;
PROGRAMMAZIONE_INTERVENTO(cf_med, CF_PAZ2, num_sala, tipo, durata_prev,
  data_prev);
END IF;

EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Non ci sono esami effettuati.');
```

### 10) Prenotazione visita

Questa procedura permette la prenotazione automatica di una visita nella prima data disponibile.

```

CREATE OR REPLACE PROCEDURE PRENOTA_VISITA
(CF_PAZ IN VARCHAR, TIPO IN VARCHAR)
IS
DATA  DATE;
C     NUMBER;
CF_MED PERSONA.CF%TYPE;
BEGIN
--Ricerca della prima data disponibile (limite di 10 visite al giorno)
BEGIN
SELECT MAX(DATA_ORA_PREVISTA) INTO DATA FROM VISITA
WHERE to_date(DATA_ORA_PREVISTA, 'dd-mm-yyyy') = (
SELECT MIN(DATA_P) FROM (
SELECT to_date(DATA_ORA_PREVISTA, 'dd-mm-yyyy') AS DATA_P, COUNT(*) AS C
FROM VISITA
GROUP BY to_date(DATA_ORA_PREVISTA, 'dd-mm-yyyy'))
WHERE C < 10);
EXCEPTION
WHEN NO_DATA_FOUND THEN
DATA := NULL;
END;
CASE
WHEN DATA IS NULL THEN
SELECT COUNT(*) INTO C FROM VISITA;
IF(C = 0) THEN
--Se non ci sono registrate la visita viene prenotata a un ora dall
  orario corrente
DATA := to_date(SYSDATE, 'dd-mm-yyyy HH24:mi:ss') + 1/24;
ELSE
--Se non ci sono date disponibili si prenota per il giorno successivo
  all ultima data registrata
SELECT MAX(DATA_ORA_PREVISTA) INTO DATA FROM VISITA;
DATA := DATA + 1;
END IF;
ELSE
```

```

--Se si trova una data disponibile, la visita viene prenotata per l ora
  successiva all ultima visita registrata per quella giornata
DATA := DATA + 1/24;
END CASE;
DBMS_OUTPUT.PUT_LINE('PRIMA DATA DISPONIBILE PER LA VISITA: ' ||
  to_char(DATA, 'dd-mm-yyyy HH24:mi:ss'));
--Assegnazione medico
--Il medico viene scelto casualmente tra i medici che nella data della
  visita non sono impegnati in interventi.
--Nel caso di una visita cardiologica o neurologica si scelgono solo
  medici con le relative specializzazioni,
--altrimenti si sceglie un medico qualsiasi.
IF(TIPO = 'Cardiologica') THEN
SELECT CF INTO CF_MED FROM (
SELECT CF, SPECIALIZZAZIONE, DATA_PREVISTA FROM
(SELECT M.CF_MED AS CF, SPECIALIZZAZIONE, ID_INTERVENTO AS IDINT FROM
MEDICO M LEFT JOIN EFFETTUA E ON M.CF_MED = E.CF_MED)
LEFT JOIN INTERVENTO I ON IDINT = I.ID_INTERVENTO
WHERE SPECIALIZZAZIONE = 'Cardiologia' AND DATA_PREVISTA IS NULL OR
SPECIALIZZAZIONE = 'Cardiologia' AND DATA_PREVISTA <> DATA
ORDER BY DBMS_RANDOM.VALUE)
WHERE ROWNUM = 1;
ELSIF(TIPO = 'Neurologica') THEN
SELECT CF INTO CF_MED FROM (
SELECT CF, SPECIALIZZAZIONE, DATA_PREVISTA FROM
(SELECT M.CF_MED AS CF, SPECIALIZZAZIONE, ID_INTERVENTO AS IDINT FROM
MEDICO M LEFT JOIN EFFETTUA E ON M.CF_MED = E.CF_MED)
LEFT JOIN INTERVENTO I ON IDINT = I.ID_INTERVENTO
WHERE SPECIALIZZAZIONE = 'Neurologia' AND DATA_PREVISTA IS NULL OR
SPECIALIZZAZIONE = 'Neurologia' AND DATA_PREVISTA <> DATA
ORDER BY DBMS_RANDOM.VALUE)
WHERE ROWNUM = 1;
ELSE
SELECT CF INTO CF_MED FROM (
SELECT CF, SPECIALIZZAZIONE, DATA_PREVISTA FROM
(SELECT M.CF_MED AS CF, SPECIALIZZAZIONE, ID_INTERVENTO AS IDINT FROM
MEDICO M LEFT JOIN EFFETTUA E ON M.CF_MED = E.CF_MED)
LEFT JOIN INTERVENTO I ON IDINT = I.ID_INTERVENTO
WHERE DATA_PREVISTA IS NULL OR DATA_PREVISTA <> DATA
ORDER BY DBMS_RANDOM.VALUE)
WHERE ROWNUM = 1;
END IF;
INSERT INTO VISITA(DATA_ORA_PREVISTA, DATA_ORA_EFFETTIVA, TIPO_VISITA,
  CF_PAZ, CF_MED, ID_RICOVERO)
VALUES(DATA, NULL, TIPO, CF_PAZ, CF_MED, NULL);
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('AL MOMENTO NON CI SONO MEDICI DISPONIBILI PER
  EFFETTUARE VISITE.');
```

### 11) Visualizzazione interventi previsti

La procedura, dato in input il codice fiscale di un medico, restituisce gli interventi



da fare del medico e il nome del reparto del relativo paziente.

```
CREATE OR REPLACE PROCEDURE INTERVENTI_DA_FARE(CF_MEDx CHAR)
AS
C1      SYS_REFCURSOR;
BEGIN
OPEN C1 FOR
SELECT I.ID_INTERVENTO,I.CF_PAZ,ET.NOME_REPARTO
FROM INTERVENTO I JOIN EFFETTUA E ON I.ID_INTERVENTO=E.ID_INTERVENTO
      JOIN E.TENUTO ET ON I.ID_RICOVERO=ET.ID_RICOVERO
WHERE E.CF_MED=CF_MEDx AND (I.DATA_EFFETTIVA IS NULL);
DBMS_SQL.RETURN_RESULT(C1);
END;
```

## 12) Conclusione di un intervento

Aggiorna la tupla relativa ad un intervento inserendo valori quali data e durata effettive.

```
CREATE OR REPLACE PROCEDURE END_INT (ID_INT NUMBER, DATA_EFF DATE, DUR
      NUMBER)
IS
DATA_PREV DATE;
TIME_PARADOX3 EXCEPTION;
BEGIN
SELECT DATA_PREVISTA INTO DATA_PREV FROM INTERVENTO
WHERE ID_INTERVENTO = ID_INT;

IF (DATA_EFF < DATA_PREV) THEN
RAISE TIME_PARADOX3;
END IF;
UPDATE INTERVENTO
SET DATA_EFFETTIVA = DATA_EFF,
DURATA_EFFETTIVA = DUR
WHERE ID_INTERVENTO = ID_INT;

EXCEPTION
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20090,'Questo intervento non esiste');

WHEN TIME_PARADOX3 THEN
RAISE_APPLICATION_ERROR(-20091,'La data effettiva precede la data
      prevista');

END;
```

## 13) Visualizzazione CF\_PAZ e Cognome da un dato Nome

```
CREATE OR REPLACE PROCEDURE CF_PAZ_DA_NOME(NOMEx VARCHAR)
AS
C1 SYS_REFCURSOR;
BEGIN
```

```

OPEN C1 FOR
SELECT CF_PAZ, COGNOME
FROM PAZIENTE P JOIN PERSONA PE ON P.CF_PAZ=PE.CF
WHERE PE.NOME=NOME;
DBMS_SQL.RETURN_RESULT(C1);
END;

```

## 2.7 Funzioni

Di seguito sono riportate le funzioni utilizzabili sul DB.

### 1) Numero di interventi effettuati su un paziente

```

CREATE OR REPLACE FUNCTION NUM_INT_EFF_PAZ(CF_PAZX CHAR) RETURN NUMBER
IS
num    NUMBER(2,0):=0;
ecc_1  EXCEPTION;
BEGIN
SELECT COUNT(ID_INTERVENTO) INTO num
FROM INTERVENTO
WHERE CF_PAZ=CF_PAZX AND (DATA_EFFETTIVA IS NOT NULL);
IF num = 0 THEN
RAISE ecc_1;
END IF;
RETURN num;
EXCEPTION
WHEN ecc_1 THEN
DBMS_OUTPUT.PUT_LINE('Non ci sono interventi');
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20088, 'Dati non trovati');
END;

```

### 2) Assegnazione infermieri

Questa funzione viene usata internamente dalla procedura PROGRAMMAZIONE\_INTERVENTO per l'assegnazione degli infermieri ad un intervento, scegliendoli in base a quelli liberi nella data prevista.

```

CREATE OR REPLACE FUNCTION GET_INF
(DATA_INT IN DATE)
RETURN INFERMIERE.CF_INF%TYPE
IS
INF    CHAR(16);
BEGIN
--Cerca gli infermieri che non sono mai stati assegnati ad un intervento
BEGIN
SELECT MIN(CF_INF) INTO INF FROM (

```

```

SELECT I.CF_INF, A.ID_INTERVENTO FROM
INFERMIERE I LEFT JOIN ASSISTE A ON I.CF_INF = A.CF_INF)
WHERE ID_INTERVENTO IS NULL;
RETURN INF;
EXCEPTION
WHEN NO_DATA_FOUND THEN
INF := NULL;
END;
--Cerca gli infermieri liberi nella data e ora inserite e che non hanno
  assistito
--più di due interventi in quel giorno
CASE
WHEN INF IS NULL THEN
SELECT MIN(CF_INF) INTO INF FROM (
SELECT CF_INF, DATA_PREVISTA, COUNT(*) FROM
ASSISTE A JOIN INTERVENTO I ON A.ID_INTERVENTO = I.ID_INTERVENTO
WHERE I.DATA_PREVISTA <> DATA_INT
GROUP BY CF_INF, DATA_PREVISTA
HAVING COUNT(*) < 3);
RETURN INF;
END CASE;
END;

```

### 3) Conteggio degli interventi assegnati ad un medico in una certa data

Come la funzione precedente, anche questa è utilizzata internamente dalla procedura PROGRAMMAZIONE\_INTERVENTO per effettuare i controlli richiesti per l'assegnazione di un medico ad un intervento.

```

CREATE OR REPLACE FUNCTION COUNT_INT
(CF_MED IN VARCHAR, DATE_INT DATE)
RETURN INTEGER
IS
COUNTER INTEGER;
BEGIN
SELECT COUNT(*) INTO COUNTER FROM
EFFETTUA E JOIN INTERVENTO I ON E.ID_INTERVENTO = I.ID_INTERVENTO
WHERE E.CF_MED = CF_MED AND to_date(I.DATA_PREVISTA, 'dd-mm-yyyy') =
  to_date(DATE_INT, 'dd-mm-yyyy');
RETURN COUNTER;
END;

```

## 2.8 Views

Di seguito sono riportate le viste utilizzabili sul DB.

**1) Visualizzazione di tutti gli interventi da fare**

```
CREATE OR REPLACE VIEW VISUALIZZA_INT_DA_FARE
AS
SELECT *
FROM INTERVENTO
WHERE DATA_EFFETTIVA IS NULL
```

**2) Visualizzazione di CF, NOME, COGNOME, REPARTO E DATA\_RICOVERO dei pazienti per i ricoveri dell'ultima settimana**

```
CREATE OR REPLACE VIEW VIS_RIC_ULTIMA_SETT
AS
SELECT CF,NOME,COGNOME,NOME_REPARTO,DATA_RICOVERO
FROM RICOVERO R JOIN PERSONA P ON R.CF_PAZ=P.CF JOIN E_TENUTO ET ON
R.ID_RICOVERO=ET.ID_RICOVERO
WHERE DATA_RICOVERO >= (SYSDATE-7)
```

**3) Visualizzazione dei fogli fisioterapia di tutti i pazienti non ancora dimessi.**

```
CREATE OR REPLACE VIEW VIS_FOGLI_TERAPIA
AS
SELECT FF.CODICE_CC,FF.CF_FISIOTERAPISTA,FF.DATA_ORA,FF.DEAMBULAZIONE,
FF.ELETTR_MUSC,FF.RINF_MUSC
FROM FOGLIO_FISIOTERAPIA FF JOIN CARTELLA_CLINICA CC ON
FF.CODICE_CC=CC.CODICE_CC JOIN RICOVERO R ON
R.ID_RICOVERO=CC.ID_RICOVERO
WHERE NOT EXISTS(SELECT *
FROM DIMISSIONE D
WHERE R.ID_RICOVERO <> D.ID_RICOVERO)
```

## 2.9 Scheduler

Il DB è dotato di uno scheduler che ogni giorno alle 8:00 controlla se nella tabella relativa alle visite sono presenti delle prenotazioni per visite vecchie di almeno due settimane e, se ne trova, le cancella dal DB.

```
BEGIN
DBMS_SCHEDULER.CREATE_JOB (
JOB_NAME => 'cancella_visite_non_effettuate',
JOB_TYPE => 'PLSQL_BLOCK',
```

```
JOB_ACTION => 'BEGIN
DELETE FROM VISITA WHERE DATA_EFFETTIVA IS NULL AND SYSDATE -
    DATA_PREVISTA > 14;
END;',
START_DATE => SYSTIMESTAMP,
REPEAT_INTERVAL => 'FREQ=DAILY; BYHOUR=8;',
ENABLED => TRUE,
AUTO_DROP => FALSE,
COMMENTS => 'Ogni giorno alle ore 8:00 vengono eliminate dal db le
    visite che non sono state effettuate entro due settimane dalla data
    di prenotazione.');
```