
Progetto di Basi di Dati

Università degli studi di Napoli Parthenope

Dipartimento di Scienze e Tecnologie - CdL in Informatica



“Luke, sono tuo padre.”

Banca del Seme

Cristina Buonaugurio - 0124/1372

Giovanni D'Angelo - 0124/1369

Anno Accademico: 2017/2018

Indice

| | |
|------------------------------------------------------------|----|
| Progetto di Basi di Dati | 1 |
| Indice | 2 |
| Generics | 4 |
| Sintesi dei requisiti | 5 |
| Glossario | 7 |
| Modello EE/R | 8 |
| Modello Relazionale | 9 |
| Utenti | 10 |
| Categorie e Permessi | 10 |
| Diagramma degli use-case | 12 |
| Tabelle Descrittore | 13 |
| Tabelle Descrittore - Procedure automaticamente richiamate | 20 |
| Creazione degli Utenti | 22 |
| Volumi | 24 |
| Operazioni DDL | 25 |
| Creazione Tabelle | 25 |
| Creazione Sequenze | 30 |
| Procedure | 31 |
| Funzioni | 48 |
| Viste | 51 |
| Vincoli di Integrità | 54 |
| Vincoli di Integrità statici | 54 |
| Vincoli di Integrità dinamici | 55 |
| Vincoli di Integrità dinamici - Procedure | 55 |
| Vincoli di Integrità dinamici - Trigger | 55 |
| Operazioni DML | 67 |
| Scheduler | 75 |

| | |
|-----------------------------------------------------|----|
| Normalizzazione | 77 |
| Prima forma normale | 77 |
| Seconda forma normale | 77 |
| Terza forma normale e forma normale di Boyce e Codd | 77 |

Generics

- Sintesi dei requisiti
- Glossario
- Modello EE/R
- Modello Relazionale
- Utenti e gestione dei privilegi
- Volumi
- Procedure e Funzioni
- Viste
- Vincoli di Integrità
- Operazioni DML
- Scheduler
- Normalizzazione

Nel corso di questa relazione, saranno analizzati i vari punti elencati sopra. In tal modo, si procederà con una spiegazione dettagliata ed approfondita di tutto ciò che è stato fatto per realizzare questo progetto per l'esame di Basi di Dati.

Saranno descritti, a seguire, i requisiti della base di dati da progettare, gli utenti con i rispettivi ruoli e privilegi (operazioni che possono eseguire con una spiegazione del proprio funzionamento), i vincoli di integrità (statici e dinamici realizzati tramite trigger) e la verifica della forma normale del database.

Per evitare un'eccessiva dispersione delle informazioni, abbiamo deciso di riportare le relative istruzioni SQL nei paragrafi in cui ogni argomento verrà discusso.

Sintesi dei requisiti

Una nota banca del seme necessita di una gestione automatizzata delle principali azioni sanitarie operate dalla suddetta. Dopo un'intervista col proprietario, è stato possibile riconoscerne i requisiti essenziali.

Bisogna tener traccia delle informazioni anagrafiche dei clienti, dei donatori e dei dottori. Al fine di garantire trasparenza, è necessario memorizzare qualsiasi esame del sangue e di seme dei papabili donatori, delle clienti compatibili all'inseminazione, i loro risultati e conservare sia fisicamente che digitalmente ogni fiala prodotta. Gli esami che una persona è tenuta ad eseguire in sede prima di poter essere considerata a tutti gli effetti donatore sono: Spermioграмма, Genetico ed Esame del Sangue per Malattie Veneree. Per le clienti che desiderano fare un'inseminazione nella clinica sono tenute ad eseguire due esami del sangue in sede: quello per le Malattie Veneree e quello per la Fertilità. Anche i soli clienti, i quali desiderano esclusivamente acquistare le fiale, devono registrare le proprie informazioni anagrafiche, ma non possono fare esami. I clienti hanno libero accesso alla lista dei donatori, potendo scegliere il donatore in base a caratteristiche fisiche ben precise: altezza, peso, etnia, età, colore degli occhi, colore dei capelli e gruppo sanguigno. Ogni donatore, d'altro canto, ha il diritto di poter decidere se essere rintracciabile o meno dai clienti e dal figlio procreato dalla propria fiala. Inoltre, per quanto riguarda l'esame del sangue per le Malattie Veneree, questo ha istruzioni ben precise, a differenza degli altri che sono considerati statici nel tempo. Difatti un donatore o un cliente che non si presenta in clinica in un tempo limite di 30 giorni dovrà rifare l'esame del sangue, eliminando il rischio di incidenza di malattie veneree durante il periodo di assenza. Ciò significa che un donatore verificato della Clinica, se per 30 giorni non ha donato altre fiale, dovrà fare un altro esame del sangue prima di poter donare nuovamente. I donatori e le clienti hanno una propria cartella clinica, in cui risiedono tutti gli esami fatti all'interno della clinica, che si deve creare automaticamente al momento della registrazione. La Banca si occupa anche della vendita delle fiale donate e di un servizio di inseminazione interno alla clinica ed entrambe richiedono un pagamento da parte del cliente. Qualsiasi azione, eccetto l'esclusivo acquisto di una fiala senza conseguente inseminazione, dovrà essere preceduta da una prenotazione, che assegnerà un dottore e una sala. Gli orari delle prenotazioni effettuabili verranno automaticamente generati a cadenza annuale: supponendo che ogni esame, o altro evento, necessiti di un arco di tempo massimo pari a 30 minuti, è possibile ogni giorno avere un massimo di 24 prenotazioni disponibili, per ogni sala dell'edificio. Inoltre i dottori sono legalmente obbligati ad assistere o ad eseguire qualsiasi esame, donazione e inseminazione; ragion per cui è di vitale importanza essere a conoscenza dei loro movimenti durante l'orario di lavoro effettuato. I turni dei dottori vengono elaborati in modo pseudo-casuale automaticamente a periodicità settimanale, garantendo che tutti i dottori, dipendenti della clinica, facciano almeno un turno di mattina o di pomeriggio, nei 6 giorni lavorativi. Per quanto riguarda i turni, essi sono principalmente due, con possibilità di mutare in futuro.

Turno lavorativo dei Dottori

| MATTINA | 8:30 - 14:30 |
|------------|---------------|
| POMERIGGIO | 14:30 - 20:30 |

Poiché teniamo conto di ogni singolo movimento, la Banca del Seme mette a disposizione il piano dell'edificio in cui vi risiede la Banca stessa. L'Edificio presenta quindi cinque sale agibili, in due piani, suddivise in ali.

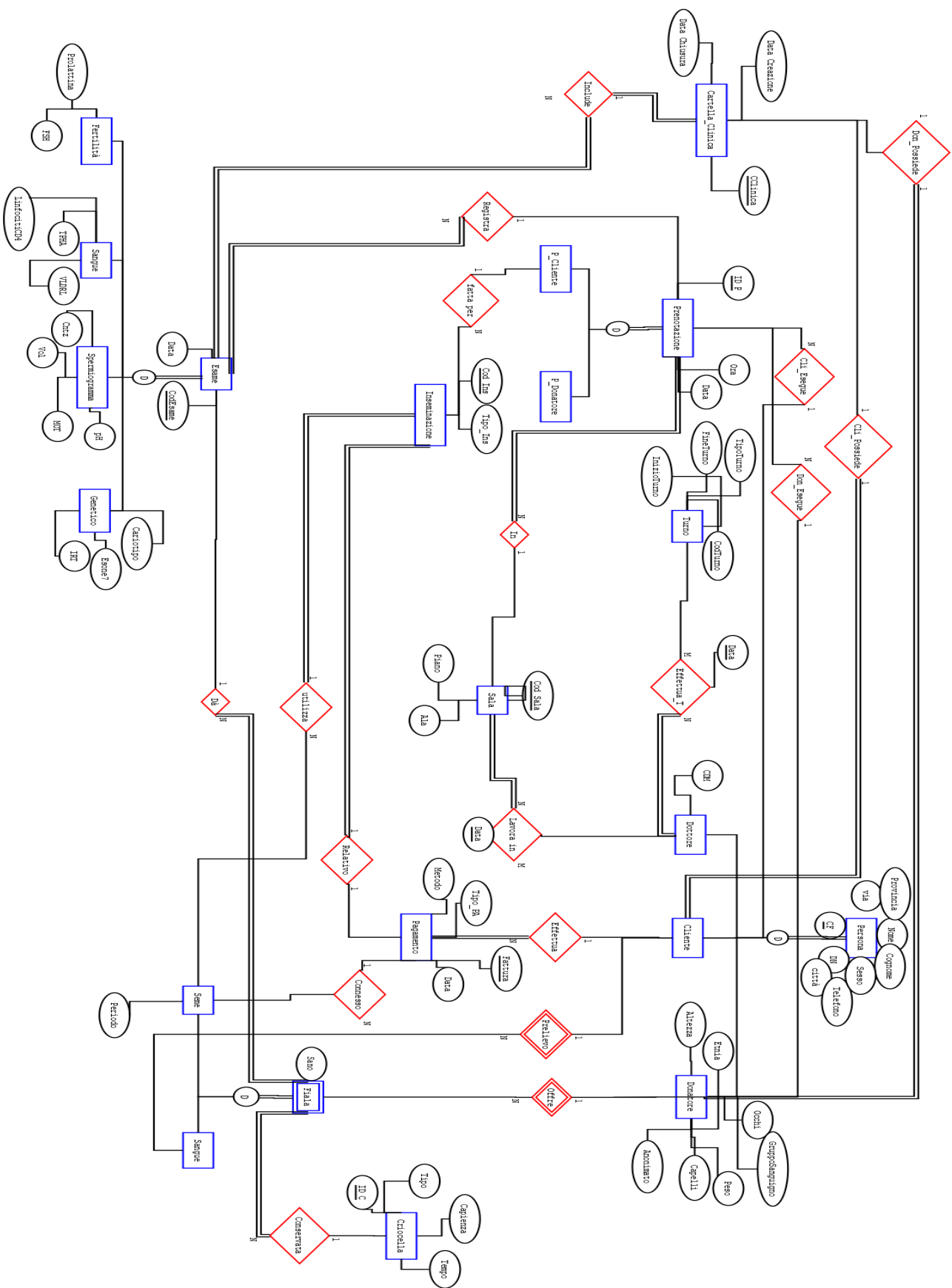
È importante tenere traccia di ogni criocella adibita alla conservazione delle fiale. All'atto dell'intervista sono otto criocelle di cui quattro adibite alla conservazione delle fiale di sangue e quattro per le fiale di seme, con capienza massima che varia da 20 a 80 fiale per ogni criocella. Per il pagamento della fiala o di un servizio di inseminazione, bisogna esclusivamente tenere traccia della modalità di pagamento e dei dati dell'acquirente. È necessario ci siano cinque utenti nel database: il Proprietario committente, il Dottore, la Segreteria, il Cliente e il Donatore. Il Proprietario dovrà essere in grado di aggiungere e/o rimuovere i dottori, aggiungere e/o rimuovere criocelle e modificare i turni lavorativi dei dottori. La Segreteria dovrà gestire le prenotazioni. Il Dottore potrà visualizzare i relativi esami e i loro risultati, i suoi turni e le prenotazioni da dover eseguire. Il Cliente potrà visualizzare i suoi esami e la lista dei donatori. Il Donatore potrà visualizzare i risultati dei suoi esami.

Glossario

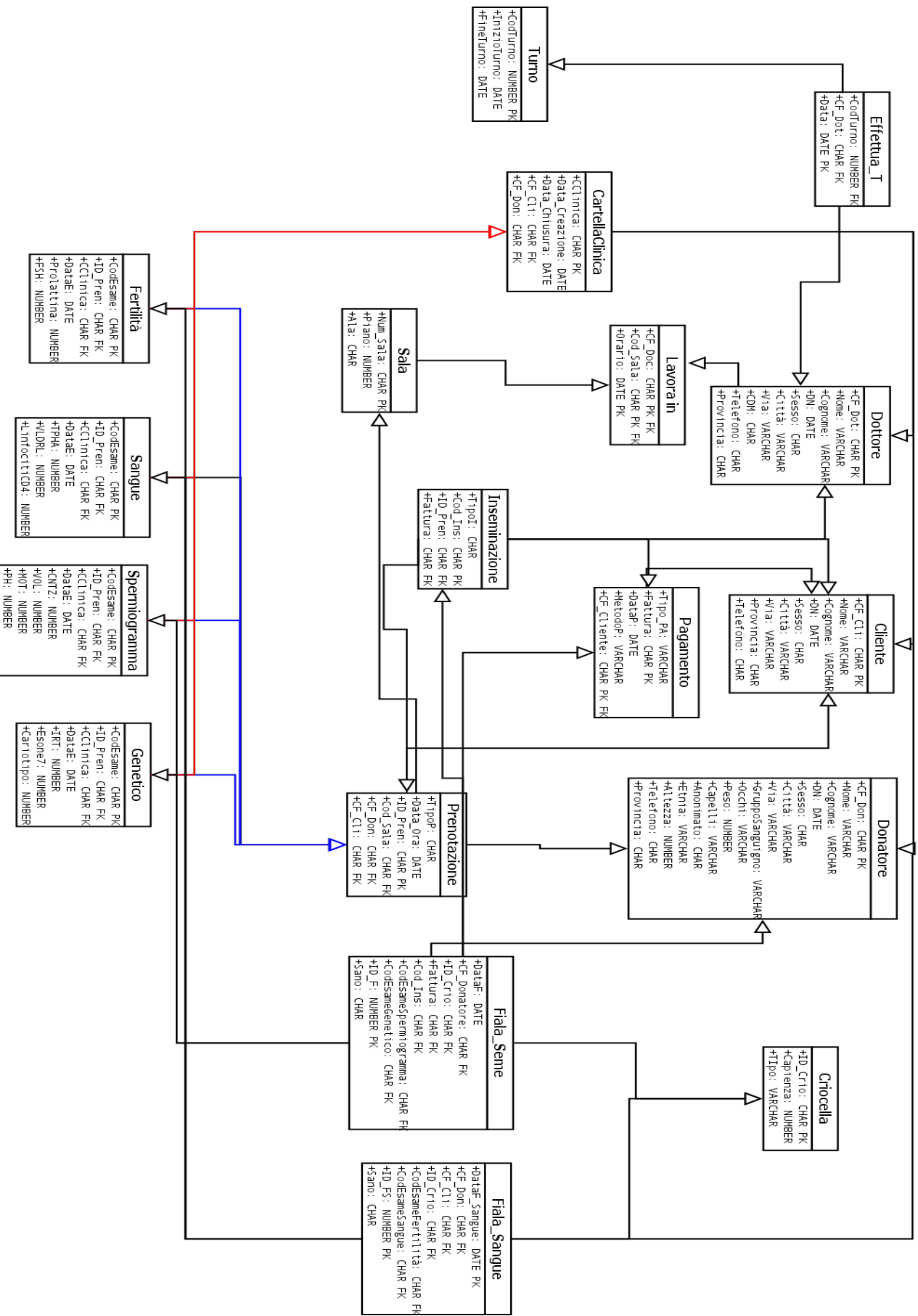
Riportiamo di seguito alcuni termini fuorvianti che saranno utilizzati spesso in seguito.

- **Flag:** un discriminatore utilizzato in varie procedure per riconoscere un evento rispetto ad un altro.
- **Servizio:** utilizzeremo questo termine per fare riferimento agli esami (fertilità, sangue, spermioγραμμα e genetico), all'acquisto della fiala o all'inseminazione.
- **Slot:** si riferisce alla Prenotazione, considerando quindi la fascia oraria in cui un servizio viene svolto.

PROGETTO DI BASI DI DATI - BANCA DEL SEME



Modello Relazionale



Utenti

Ora analizzeremo nel dettaglio i singoli utenti che compongono la base di dati, ognuno con i propri permessi e scopi. Andremo poi a mostrare anche il diagramma degli use-case e le schede descrittore. E concluderemo con la creazione degli utenti in SQL.

Categorie e Permessi

| UTENTE | TIPO | VOLUME | PERMESSI |
|--------------|----------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Admin | Amministratore | 1 | ALL |
| Proprietario | Comune | 1 | EXECUTE INSERISCI_CRIOCELLA INSERT ON DOTTORE UPDATE ON DOTTORE DELETE ON DOTTORE UPDATE ON CRIOCELLA DELETE ON CRIOCELLA |
| Dottore | Comune | 1 | EXECUTE INSERISCI_FIALA_SANGUE EXECUTE INSERISCI_FIALA_SEME EXECUTE INSERT_ES_SANGUE EXECUTE INSERT_ES_SPERM EXECUTE INSERT_ES_GENET EXECUTE INSERT_ES_FERT EXECUTE INSERT_INSEMINAZIONE SELECT ON FIALA_SEME SELECT ON FIALA_SANGUE SELECT ON CRIOCELLA SELECT ON INSEMINAZIONE SELECT ON PREN_GIORNALIERE SELECT ON ESAME_GENETICO SELECT ON ESAME_SANGUE SELECT ON ESAME_SPERMIOGRAMMA SELECT ON ESAME_FERTILITA UPDATE ON FIALA_SEME UPDATE ON FIALA_SANGUE UPDATE ON GENETICO UPDATE ON SPERMIOGRAMMA UPDATE ON SANGUE UPDATE ON FERTILITA DELETE ON FIALA_SEME DELETE ON FIALA_SANGUE DELETE ON GENETICO DELETE ON SPERMIOGRAMMA DELETE ON SANGUE DELETE ON FERTILITA |

| UTENTE | TIPO | VOLUME | PERMESSI |
|------------|--------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Segreteria | Comune | 1 | EXECUTE INSERISCI_PRENOTAZIONE EXECUTE ELIMINA_PRENOTAZIONE EXECUTE INSERISCI_CLIENTE EXECUTE INSERISCI_DONATORE EXECUTE ACQUISTO_FIALA EXECUTE INSERISCI_PAG_INS SELECT ON PRENOTAZIONE_DISPONIBILI SELECT ON DONATORI_DISPONIBILI UPDATE ON PRENOTAZIONE UPDATE ON DONATORE UPDATE ON CLIENTE SELECT ON CARTELLA_CLINICA UPDATE ON CARTELLA_CLINICA |
| Cliente | Comune | 1 | SELECT ON VISITA_DONATORI SELECT ON VISITA_DONATORI_ANONIMI SELECT ON ESAME_SANGUE SELECT ON ESAME_FERTILITA SELECT ON VISUALIZZA_PRENOTAZIONI |
| Donatore | Comune | 1 | SELECT ON ESAME_GENETICO SELECT ON ESAME_SPERMIOGRAMMA SELECT ON ESAME_SANGUE SELECT ON VISUALIZZA_PRENOTAZIONI |

Diagramma degli use-case

Di seguito presentiamo il diagramma use-case, non sono presenti, così come non lo erano nella tabella delle categorie e permessi, tutte quelle procedure che vengono richiamate automaticamente da altre procedure invece segnalate e di cui gli utenti non hanno i permessi per poterle richiamare.

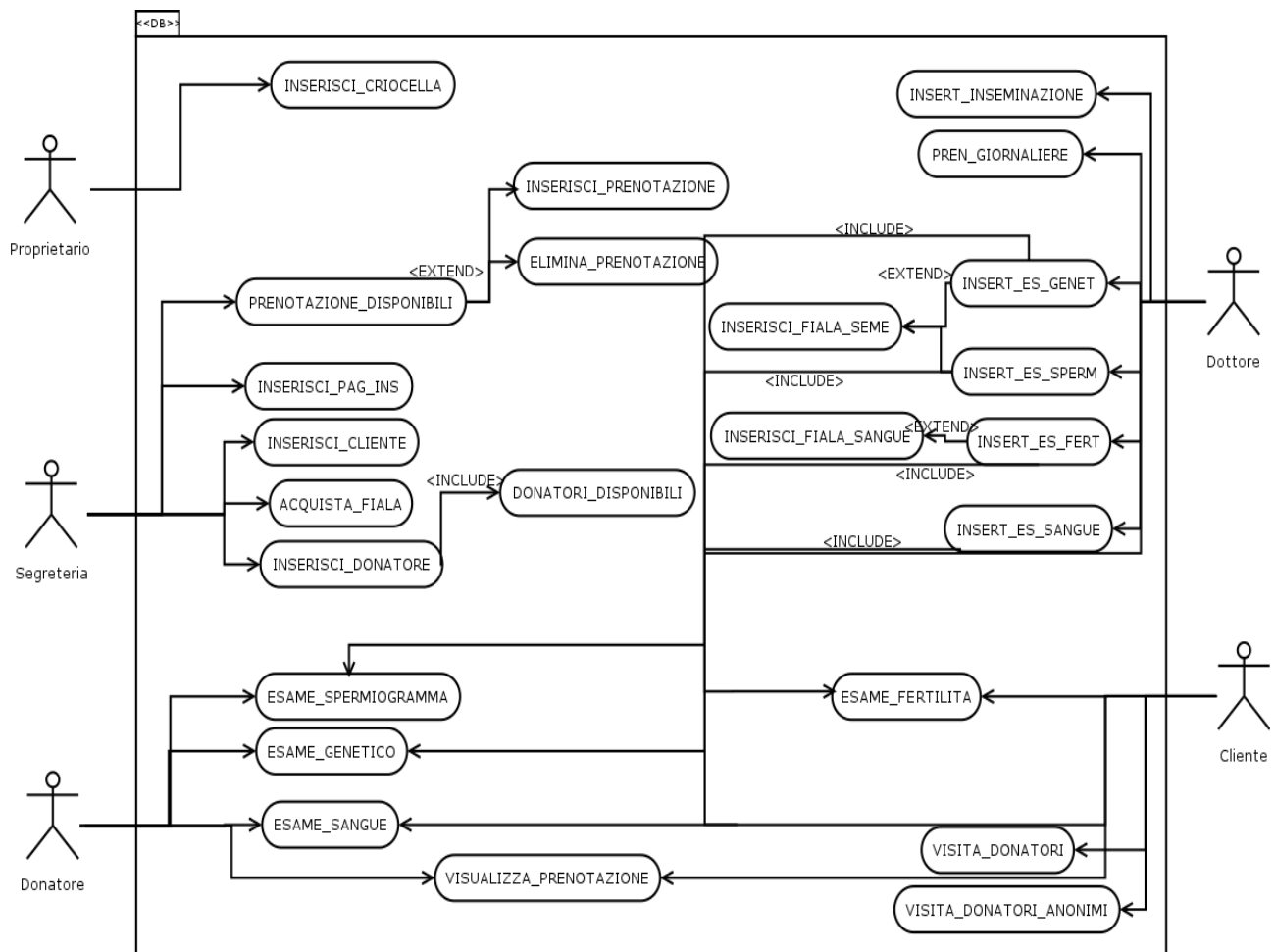


Tabelle Descrittore

Analizzeremo di seguito le procedure principali del database. Si intenderanno non principali le procedure che sostituiscono le operazioni DML: [Inserisci_Donatore](#), [Inserisci_Cliente](#) e [Inserisci_Criocella](#).

| OPERAZIONE | Organizza_Orari_prenotazioni |
|------------|-----------------------------------------------------------------------------------|
| SCOPO | Crea gli slot di prenotazione per un intero anno lavorativo |
| ARGOMENTI | / |
| RISULTATO | Crea gli slot nella tabella Prenotazione |
| ERRORI | / |
| USA | Prenotazione |
| MODIFICA | Prenotazione |
| PRIMA | La tabella di Prenotazione è vuota |
| POI | La tabella di Prenotazione presenta tuple disponibili con ID, Data e Numero sala. |

| OPERAZIONE | Organizza_Turni_Dottori |
|------------|--------------------------------------------------------------------------------|
| SCOPO | Gestisce in modo pseudo_casuale i turni lavorativi dei dottori settimanalmente |
| ARGOMENTI | / |
| RISULTATO | Nuove tuple nella tabella Effettua_T |
| ERRORI | / |
| USA | Effettua_T, Dottore |
| MODIFICA | Effettua_T |
| PRIMA | Non è possibile fare prenotazioni perché non ci sono dottori disponibili |
| POI | È possibile fare prenotazioni |

OPERAZIONE [Inserisci_Prenotazioni](#)

| | |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SCOPO | Permette ad un cliente o ad un donatore di prenotare un servizio |
| ARGOMENTI | Data, Codice Fiscale del Cliente o del Donatore, Tipo di servizio, Numero Sala |
| RISULTATO | Modifica una tupla di prenotazione libera |
| ERRORI | Sala Occupata, Nessun Dottore Disponibile, Codice Fiscale indistinguibile, Il Cliente o il Donatore hanno già una prenotazione in quell'ora, Non è possibile fare l'inseminazione |
| USA | Fiala_Sangue, Effettua_T, Turno, Prenotazione |
| MODIFICA | Prenotazione |
| PRIMA | / |
| POI | Il Cliente o il Donatore ha la prenotazione |

OPERAZIONE [Inserisci_Fiala_Seme](#)

| | |
|------------------|---------------------------------------------------------------------------------------------------------------------------------|
| SCOPO | Inserisce una nuova tupla nella tabella fiala_seme |
| ARGOMENTI | Id della Prenotazione, Codice Esame Spermioγραμμα, Codice Esame Genetico |
| RISULTATO | La fiala seme viene inserita nella tabella e la sua tupla farà riferimento agli esami spermioγραμμα e genetico fatti in clinica |
| ERRORI | / |
| USA | Prenotazione, Tabelle utilizzate dalla funzione utilizzata all'interno della procedura: Criocella, Fiala_Seme, Fiala_Sangue |
| MODIFICA | Fiala_Seme |
| PRIMA | Non c'è la tupla che si riferisce alla fisica fiala di seme presente nella clinica |
| POI | La fiala di seme ha un suo riferimento nel database |

OPERAZIONE [Insert_Es_Sangue](#)

| | |
|------------------|-------------------------------------------------------------------------------------------------------------------------|
| SCOPO | Inserire una nuova tupla nella tabella Sangue con i valori del relativo esame |
| ARGOMENTI | Id della Prenotazione dell'esame del sangue, Valori del sangue: TPHA, VLDRL, LINFOCITICD4 |
| RISULTATO | Inserisce una nuova tupla nella tabella Sangue |
| ERRORI | / |
| USA | Prenotazione, Cliente, Donatore, Cartella_Clinica, Sangue |
| MODIFICA | Sangue |
| PRIMA | Non c'è una tupla dell'esame del sangue fatto dal cliente o dal donatore |
| POI | L'esame del sangue fatto è nel database accessibile al cliente o al donatore e crea una tupla di Fiala_Sangue associata |

OPERAZIONE [Insert_Es_Sperm](#)

| | |
|------------------|-----------------------------------------------------------------------------------------|
| SCOPO | Inserisce una nuova tupla nella tabella Spermiogramma con i valori del relativo esame |
| ARGOMENTI | Id della Prenotazione dell'esame Spermiogramma e i suoi Valori: CNTZ, VOL, MOTILITÀ, PH |
| RISULTATO | Inserisce una nuova tupla nella tabella Spermiogramma |
| ERRORI | / |
| USA | Prenotazione, Donatore, Cartella_Clinica, Spermiogramma |
| MODIFICA | Spermiogramma |
| PRIMA | Non c'è una tupla dell'esame Spermiogramma fatto dal donatore |
| POI | L'esame Spermiogramma è nel database accessibile al donatore |

| | |
|--------------------------------------------------|----------------------------------------------------------------------------------------------------|
| OPERAZIONE Insert_Es_Fert | |
| SCOPO | Inserisce una nuova tupla nella tabella Fertilita con i valori del relativo esame |
| ARGOMENTI | Id della Prenotazione referente all'esame e i suoi valori: PROLATTINA, FSH |
| RISULTATO | Inserisce una nuova tupla nella tabella Fertilita |
| ERRORI | / |
| USA | Prenotazione, Cliente, Cartella_Cliente, Fertilita |
| MODIFICA | Fertilita |
| PRIMA | Non c'è una tupla dell'esame Fertilita fatto dalla Cliente |
| POI | L'esame sulla Fertilità è nel database accessibile dalla Cliente e crea una Fiala_Sangue associata |

| | |
|--------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| OPERAZIONE Inserisci_Fiala_Sangue (1) | |
| SCOPO | Inserisce una nuova tupla nella tabella fiala_sangue |
| ARGOMENTI | Id della Prenotazione, Codice Esame del Sangue, Flag |
| RISULTATO | La fiala sangue viene inserita nella tabella e la sua tupla farà riferimento ad almeno all'ultimo esame del sangue fatto nella clinica |
| ERRORI | / |
| USA | Prenotazione, Fiala_Sangue |
| MODIFICA | Fiala_Sangue |
| PRIMA | Non c'è la tupla che si riferisce alla fisica fiala di sangue presente nella clinica |
| POI | La fiala di sangue ha un suo riferimento nel database |

OPERAZIONE [Insert_Es_Genet](#)

| | |
|------------------|-----------------------------------------------------------------------------------|
| SCOPO | Inserisce una nuova tupla nella tabella Genetico con i valori del relativo esame |
| ARGOMENTI | Id della Prenotazione referente all'esame e i suoi valori: IRT, ESONE7, CARIOTIPO |
| RISULTATO | Inserisce una nuova tupla nella tabella Genetico |
| ERRORI | / |
| USA | Prenotazione, Donatore, Cartella_Clinica, Genetico |
| MODIFICA | Genetico |
| PRIMA | Non c'è una tupla dell'esame Genetico fatto dal donatore |
| POI | L'esame Genetico è nel database accessibile al donatore |

OPERAZIONE [Inserisci_Pag_Ins](#)

| | |
|------------------|------------------------------------------------------------------------------------------------------------|
| SCOPO | Inserisce il pagamento del servizio Inseminazione |
| ARGOMENTI | Metodo di Pagamento, Codice Fiscale della Cliente che chiede il servizio, Data prevista dalla Prenotazione |
| RISULTATO | Inserisce una tupla nella tabella Pagamento |
| ERRORI | La Cliente non è compatibile all'inseminazione |
| USA | Fiala_Sangue, Pagamento |
| MODIFICA | Pagamento |
| PRIMA | L'inseminazione non è stata ancora pagata |
| POI | Il pagamento del servizio è correttamente registrato |

(1) La procedura [Inserisci_Fiala_Sangue](#) andrebbe anche segnalata successivamente per le tabelle descrittore automaticamente richiamate, ma riteniamo che, potendo l'utente aggiungere più di una fiala sangue relativo allo stesso esame, sia necessario presentarla in questa sezione.

OPERAZIONE [Insert_Inseminazione](#)

| | |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SCOPO | Inserisce una tupla nella tabella Inseminazione |
| ARGOMENTI | Numero di Fiale utilizzate, Codice Fiscale del donatore, Tipo di inseminazione, Codice Fattura riferita al pagamento del servizio, Id della Prenotazione referente |
| RISULTATO | Inserimento di una tupla nella tabella Inseminazione |
| ERRORI | Il numero di fiale richiesto è maggiore a quelle disponibili dello specifico donatore |
| USA | Fiala_Seme, Inseminazione |
| MODIFICA | Inseminazione |
| PRIMA | L'inseminazione non può essere effettuata |
| POI | L'inseminazione può essere effettuata ed è regolarmente registrata |

OPERAZIONE [Elimina_Prenotazione](#)

| | |
|------------------|-----------------------------------------------------------------------------------------|
| SCOPO | Svuota una tupla di Prenotazione lasciando la possibilità di essere modificata di nuovo |
| ARGOMENTI | Id della Prenotazione da “eliminare” |
| RISULTATO | La tupla è di nuovo disponibile |
| ERRORI | La tupla è già vuota |
| USA | Prenotazione |
| MODIFICA | Prenotazione |
| PRIMA | La tupla di prenotazione ha già un servizio associato |
| POI | La tupla è di nuovo disponibile |

| | |
|--------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| OPERAZIONE Acquisto_Fiala | |
| SCOPO | Registra l'acquisto di una fiala seme e il suo relativo pagamento nella tabella Pagamento |
| ARGOMENTI | Id della fiala da acquistare, Codice Fiscale del donatore associato alla fiala, Metodo di Pagamento |
| RISULTATO | Inserisce una tupla nella tabella Pagamento |
| ERRORI | La fiala non è disponibile all'acquisto, la fiala non esiste, la fiala è stata già venduta |
| USA | Fiala_Seme, Pagamento |
| MODIFICA | Fiala_Seme, Pagamento |
| PRIMA | La fiala seme desiderata non è ufficialmente acquistata |
| POI | L'acquisto della fiala è correttamente registrato |

Tabelle Descrittore - Procedure automaticamente richiamate

Di seguito avremo le tabelle descrittore per quelle procedure che non vengono direttamente chiamate dall'utente ma richiamate da un'altra procedura. Per un maggior dettaglio, ci avvaliamo della licenza di aggiungere un altro campo alla Tabella: "Chiamato", in cui sarà nominata la procedura interessata.

| OPERAZIONE Apri_Cartella | |
|------------------------------------------|------------------------------------------------------------------------------------------------------------|
| SCOPO | Crea una nuova tupla nella tabella <i>Cartella_Clinica</i> |
| ARGOMENTI | Codice fiscale del Cliente o del Donatore |
| RISULTATO | Il Cliente o il Donatore ha una sua cartella clinica correttamente registrata |
| ERRORI | Il Cliente o il Donatore di cui si vuole creare la cartella non è registrato |
| USA | Cliente, Donatore, <i>Cartella_Clinica</i> |
| MODIFICA | <i>Cartella_Clinica</i> |
| PRIMA | Il Cliente o il Donatore non ha una cartella clinica per registrare i servizi della clinica |
| POI | Il Cliente o il Donatore ha una sua cartella clinica a cui faranno riferimento ogni servizio della Clinica |
| CHIAMATO | Inserisci_Donatore , Inserisci_Cliente |

OPERAZIONE [Assegna_Dot](#)

| | |
|------------------|-------------------------------------------------------------------------------------|
| SCOPO | Automaticamente assegna un dottore libero ad un servizio prenotato |
| ARGOMENTI | Numero della sala in cui si terrà il servizio, data della Prenotazione dello stesso |
| RISULTATO | Il dottore viene assegnato ad un servizio |
| ERRORI | Non ci sono dottori che possono fare quel servizio |
| USA | Effettua_T, Turno, Lavora_In |
| MODIFICA | Lavora_In |
| PRIMA | Il servizio ha bisogno di un dottore |
| POI | C'è un dottore che può assistere |
| CHIAMATO | Inserisci_Prenotazioni |

OPERAZIONE [De_Assegna_Dot](#)

| | |
|------------------|--------------------------------------------------------------------|
| SCOPO | Elimina una tupla nella tabella Lavora_In |
| ARGOMENTI | Data della Prenotazione che è stata “eliminata”, numero della sala |
| RISULTATO | Il dottore non è più occupato con una prenotazione |
| ERRORI | / |
| USA | Lavora_In |
| MODIFICA | Lavora_In |
| PRIMA | Il dottore è occupato con un servizio prenotato |
| POI | Il dottore è libero |
| CHIAMATO | Elimina_Prenotazione |

Creazione degli Utenti

Nella seguente sezione riportiamo sia le istruzioni della creazione degli utenti che popolano la base di dati sia quelle utili a garantire i giusti privilegi a ciascuno di essi.

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

```
-- CREAZIONE DEL PROPRIETARIO E DEI SUOI PERMESSI  
CREATE USER PROPRIETARIO IDENTIFIED BY PROP;  
GRANT CREATE SESSION TO PROPRIETARIO;  
GRANT EXECUTE ON INSERISCI_CRIOCELLA TO PROPRIETARIO;  
GRANT INSERT ON DOTTORE TO PROPRIETARIO;  
GRANT UPDATE ON DOTTORE TO PROPRIETARIO;  
GRANT DELETE ON DOTTORE TO PROPRIETARIO;  
GRANT UPDATE ON CRIOCELLA TO PROPRIETARIO;  
GRANT DELETE ON CRIOCELLA TO PROPRIETARIO;
```

```
-- CREAZIONE DEL DOTTORE E DEI SUOI PERMESSI  
CREATE USER DOTTORE IDENTIFIED BY DOT;  
GRANT CREATE SESSION TO DOTTORE;  
GRANT EXECUTE ON INSERISCI_FIALA_SANGUE TO DOTTORE;  
GRANT EXECUTE ON INSERISCI_FIALA_SEME TO DOTTORE;  
GRANT EXECUTE ON INSERT_ES_SANGUE TO DOTTORE;  
GRANT EXECUTE ON INSERT_ES_SPERM TO DOTTORE;  
GRANT EXECUTE ON INSERT_ES_GENET TO DOTTORE;  
GRANT EXECUTE ON INSERT_ES_FERT TO DOTTORE;  
GRANT EXECUTE ON INSERT_INSEMINAZIONE TO DOTTORE;  
GRANT SELECT ON FIALA_SEME TO DOTTORE;  
GRANT SELECT ON FIALA_SANGUE TO DOTTORE;  
GRANT SELECT ON CRIOCELLA TO DOTTORE;  
GRANT SELECT ON INSEMINAZIONE TO DOTTORE;  
GRANT SELECT ON PREN_GIORNALIERE TO DOTTORE;  
GRANT SELECT ON ESAME_GENETICO TO DOTTORE;  
GRANT SELECT ON ESAME_SANGUE TO DOTTORE;  
GRANT SELECT ON ESAME_SPERMIOGRAMMA TO DOTTORE;  
GRANT SELECT ON ESAME_FERTILITA TO DOTTORE;  
GRANT UPDATE ON FIALA_SEME TO DOTTORE;  
GRANT UPDATE ON FIALA_SANGUE TO DOTTORE;  
GRANT UPDATE ON GENETICO TO DOTTORE;  
GRANT UPDATE ON SPERMIOGRAMMA TO DOTTORE;  
GRANT UPDATE ON SANGUE TO DOTTORE;  
GRANT UPDATE ON FERTILITA TO DOTTORE;  
GRANT DELETE ON FIALA_SEME TO DOTTORE;  
GRANT DELETE ON FIALA_SANGUE TO DOTTORE;  
GRANT DELETE ON GENETICO TO DOTTORE;
```

```

GRANT DELETE ON SPERMIOGRAMMA TO DOTTORE;
GRANT DELETE ON SANGUE TO DOTTORE;
GRANT DELETE ON FERTILITA TO DOTTORE;

-- CREAZIONE DELLA SEGRETERIA E DEI SUOI PERMESSI
CREATE USER SEGRETERIA IDENTIFIED BY SEG;
GRANT CREATE SESSION TO SEGRETERIA;
GRANT EXECUTE ON INSERISCI_PRENOTAZIONE TO SEGRETERIA;
GRANT EXECUTE ON ELIMINA_PRENOTAZIONE TO SEGRETERIA;
GRANT EXECUTE ON INSERISCI_CLIENTE TO SEGRETERIA;
GRANT EXECUTE ON INSERISCI_DONATORE TO SEGRETERIA;
GRANT EXECUTE ON ACQUISTO_FIALA TO SEGRETERIA;
GRANT EXECUTE ON INSERISCI_PAG_INS TO SEGRETERIA;
GRANT SELECT ON PRENOTAZIONI_DISPONIBILI TO SEGRETERIA;
GRANT SELECT ON DONATORI_DISPONIBILI TO SEGRETERIA;
GRANT UPDATE ON PRENOTAZIONE TO SEGRETERIA;
GRANT UPDATE ON DONATORE TO SEGRETERIA;
GRANT UPDATE ON CLIENTE TO SEGRETERIA;
GRANT SELECT ON CARTELLA_CLINICA TO SEGRETERIA;
GRANT UPDATE ON CARTELLA_CLINICA TO SEGRETERIA;

-- CREAZIONE DEL CLIENTE
CREATE USER CLIENTE IDENTIFIED BY CLI;
GRANT CREATE SESSION TO CLIENTE;
GRANT SELECT ON VISITA_DONATORI TO CLIENTE;
GRANT SELECT ON VISITA_DONATORI_ANONIMI TO CLIENTE;
GRANT SELECT ON ESAME_SANGUE TO CLIENTE;
GRANT SELECT ON ESAME_FERTILITA TO CLIENTE;
GRANT SELECT ON VISUALIZZA_PRENOTAZIONI TO CLIENTE;

-- CREAZIONE DEL DONATORE
CREATE USER DONATORE IDENTIFIED BY DON;
GRANT CREATE SESSION TO DONATORE;
GRANT SELECT ON ESAME_GENETICO TO DONATORE;
GRANT SELECT ON ESAME_SPERMIOGRAMMA TO DONATORE;
GRANT SELECT ON ESAME_SANGUE TO DONATORE;
GRANT SELECT ON VISUALIZZA_PRENOTAZIONI TO DONATORE;

```

Volumi

In questa sezione analizzeremo la tabella dei Volumi. È necessario specificare che i valori riportati in questa tabella sono differenti rispetto a quanto abbiamo effettivamente popolato. In quanto per poter ottenere un risultato coerente alla tabella, avremmo impiegato mesi e mesi. Simuliamo, perciò, due settimane di lavoro tipo, ammettendo che la clinica lavori a medio regime.

| Tabella | Tipo | Volume | Incremento | Periodo |
|------------------|------|--------|------------|-------------|
| Cliente | E | 30 | 10 | Mensile |
| Dottore | E | 10 | 5 | Annuale |
| Donatore | E | 40 | 15 | Mensile |
| Pagamento | E | 30 | 10 | Mensile |
| Fiala_Seme | E | 50 | 20 | Mensile |
| Fiala_Sangue | E | 90 | 40 | Mensile |
| Prenotazione | E | 42.000 | 42.000 | Annuale |
| Sala | E | 5 | 0* | / |
| Cartella_Clinica | E | 70 | 25 | Mensile |
| Effettua_T | A | 60 | 60 | Settimanale |
| Turno | E | 2 | 0* | / |
| Criocella | E | 8 | 1 | Semestrale |
| Fertilita | E | 20 | 15 | Mensile |
| Sangue | E | 60 | 25 | Mensile |
| Spermiogramma | E | 40 | 15 | Mensile |
| Genetico | E | 40 | 15 | Mensile |
| Lavora_In | A | 200 | 100 | Mensile |
| Inseminazione | E | 10 | 5 | Mensile |

* Le tuple di Turno e Sala non vengono incrementate nel database poiché rappresentano situazioni aziendali non modificabili con facilità e che quindi richiederebbero l'assistenza dell'Admin.

Operazioni DDL

In questa sezione saranno presentati le istruzioni di creazione, aggiungendo gli opportuni commenti quando necessario.

Creazione Tabelle

```
CREATE TABLE CLIENTE (  
CF_CLI CHAR(16) PRIMARY KEY,  
NOME_CLI VARCHAR(15) NOT NULL,  
COGNOME_CLI VARCHAR(30) NOT NULL,  
DN_CLI DATE NOT NULL,  
SESSO_CLI CHAR(1),  
CITTA_CLI VARCHAR(20),  
VIA_CLI VARCHAR(30),  
TELEFONO_CLI CHAR(10) NOT NULL UNIQUE,  
PROV_CLI CHAR(2) NOT NULL,  
CONSTRAINT CHECK_SCL CHECK(SESSO_CLI IN ( 'M' , 'F' ))  
);
```

```
CREATE TABLE DOTTORE (  
CF_DOT CHAR(16) PRIMARY KEY,  
NOME_DOT VARCHAR(15) NOT NULL,  
COGNOME_DOT VARCHAR(30) NOT NULL,  
DN_DOT DATE NOT NULL,  
SESSO_DOT CHAR(1),  
CITTA_DOT VARCHAR(20),  
VIA_DOT VARCHAR(30),  
CDM CHAR(4) NOT NULL UNIQUE,  
TELEFONO_DOT CHAR(10) NOT NULL UNIQUE,  
PROV_DOT CHAR(2) NOT NULL,  
CONSTRAINT CHECK_SDOT CHECK(SESSO_DOT IN ( 'M' , 'F' ))  
);
```

Di seguito presentiamo il codice per la creazione della tabella Donatore, poiché la clinica è orientata alla sola donazione di seme, c'è sembrato superfluo tenere traccia del campo sesso del donatore.

```

CREATE TABLE DONATORE (
CF_DON CHAR(16) PRIMARY KEY,
NOME_DON VARCHAR(15) NOT NULL,
COGNOME_DON VARCHAR(30) NOT NULL,
DN_DON DATE,
CITTA_DON VARCHAR(20),
VIA_DON VARCHAR(30),
G_SANGUIGNO VARCHAR(3) NOT NULL,
OCCHI VARCHAR(10),
PESO NUMBER,
CAPELLI VARCHAR(10),
ANONIMATO CHAR(2) NOT NULL,
ETNIA VARCHAR(15),
ALTEZZA NUMBER,
TELEFONO_DON CHAR(10) NOT NULL UNIQUE,
PROV_DON CHAR(2) NOT NULL,
CONSTRAINT CHECK_COLOREOCCHI CHECK (OCCHI IN
('BLU', 'AZZURRI', 'VERDI', 'MARRONI', 'NERI', 'GRIGI')),
CONSTRAINT CHECK_GS CHECK(G_SANGUIGNO IN
('0-', '0+', 'A-', 'A+', 'B-', 'B+', 'AB-', 'AB+')),
CONSTRAINT CHECK_PESO CHECK((PESO>30) AND (PESO<150)),
CONSTRAINT CHECK_AN CHECK(ANONIMATO IN ('SI', 'NO')),
CONSTRAINT CHECK_HE CHECK((ALTEZZA> 120) AND (ALTEZZA<220))
);

CREATE TABLE PAGAMENTO (
TIPO_PA VARCHAR(20) CHECK(TIPO_PA IN ('INSEMINAZIONE', 'FIALA')),
FATTURA CHAR(10) PRIMARY KEY,
DATAP DATE,
METODOP VARCHAR(10),
CF_CLI CHAR(16) NOT NULL,
CONSTRAINTS EFFETTUA FOREIGN KEY(CF_CLI) REFERENCES
CLIENTE(CF_CLI)
);

CREATE TABLE FIALA_SEME (
DATA_SEME DATE,
ID_FIALA_SE CHAR(4) PRIMARY KEY,
CF_DON CHAR(16) NOT NULL,
ID_CRIO NUMBER,
COD_E_SPERMIOGRAMMA CHAR(8) NOT NULL,
COD_E_GENETICO CHAR (8) NOT NULL,
COD_INS CHAR(8),
FATTURA CHAR(10),
SANO CHAR(2) CHECK (SANO IN ('SI', 'NO')),
CONSTRAINT CONNESSO FOREIGN KEY(FATTURA) REFERENCES
PAGAMENTO(FATTURA)
);

```

```

CREATE TABLE FIALA_SANGUE (
DATA_SANGUE DATE,
ID_FIALA_SA CHAR(4) PRIMARY KEY,
CF_DON CHAR(16),
CF_CLI CHAR(16),
COD_E_FERTILITA CHAR(8),
COD_E_SANGUE CHAR(8),
ID_CRIO_SANGUE NUMBER,
SANO CHAR(2) CHECK (SANO IN ( 'SI', 'NO' ))
);

CREATE TABLE CRIOCELLA(
ID_CRIO NUMBER PRIMARY KEY,
CAPIENZA_MAX NUMBER NOT NULL,
TIPO VARCHAR (10) CHECK(TIPO IN ( 'SANGUE', 'SEME' )),
CONSTRAINT CHECK_CM CHECK(CAPIENZA_MAX IN (20,40,60,80))
);

CREATE TABLE INSEMINAZIONE (
COD_INS CHAR(8) PRIMARY KEY,
TIPO_INS CHAR(3),
FATTURA CHAR(10),
ID_PREN NUMBER NOT NULL,
CONSTRAINT CHECK_TI CHECK(TIPO_INS IN ( 'ICI', 'IUI' )),
CONSTRAINT RELATIVO FOREIGN KEY(FATTURA) REFERENCES
PAGAMENTO(FATTURA)
);

CREATE TABLE CARTELLA_CLINICA (
C_CLINICA CHAR(8) PRIMARY KEY,
DATA_CREAZIONE DATE NOT NULL,
DATA_CHIUSURA DATE,
CF_DON CHAR(16) UNIQUE,
CF_CLI CHAR(16) UNIQUE,
CONSTRAINT POSSIEDE_CLIENTE FOREIGN KEY(CF_CLI) REFERENCES
CLIENTE(CF_CLI),
CONSTRAINT POSSIEDE_DONATORE FOREIGN KEY(CF_DON) REFERENCES
DONATORE(CF_DON)
);

CREATE TABLE GENETICO (
COD_E_GENETICO CHAR (8) PRIMARY KEY,
ID_PREN NUMBER NOT NULL,
C_CLINICA CHAR(8) NOT NULL,
DATA_GEN DATE,
IRT NUMBER NOT NULL,
ESONE7 NUMBER NOT NULL CHECK(ESONE7 IN(1,0)),
CARIOTIPO CHAR(4) NOT NULL,
CONSTRAINT RIF_CC_GEN FOREIGN KEY(C_CLINICA) REFERENCES
CARTELLA_CLINICA(C_CLINICA)

```

);

```
CREATE TABLE SPERMIOGRAMMA (  
  COD_E_SPERMIOGRAMMA CHAR(8) PRIMARY KEY,  
  ID_PREN NUMBER NOT NULL,  
  C_CLINICA CHAR(8) NOT NULL,  
  DATA_E_SP DATE,  
  CNTZ NUMBER NOT NULL,  
  VOL NUMBER NOT NULL,  
  MOT NUMBER NOT NULL CHECK((MOT>0) AND (MOT<100)),  
  PH NUMBER NOT NULL,  
  CONSTRAINT RF_CC_SPE FOREIGN KEY(C_CLINICA) REFERENCES  
  CARTELLA_CLINICA(C_CLINICA)  
);
```

```
CREATE TABLE SANGUE(  
  COD_E_SANGUE CHAR(8) PRIMARY KEY,  
  ID_PREN NUMBER NOT NULL,  
  C_CLINICA CHAR(8) NOT NULL,  
  DATA_E_SA DATE,  
  TPHA NUMBER NOT NULL,  
  VLDRL NUMBER NOT NULL,  
  LINFOCITCD4 NUMBER NOT NULL,  
  CONSTRAINT RF_CC_SAN FOREIGN KEY(C_CLINICA) REFERENCES  
  CARTELLA_CLINICA(C_CLINICA)  
);
```

```
CREATE TABLE FERTILITA(  
  COD_E_FERTILITA CHAR(8) PRIMARY KEY,  
  ID_PREN NUMBER NOT NULL,  
  C_CLINICA CHAR(8) NOT NULL,  
  DATA_E_FE DATE,  
  PROLATTINA NUMBER NOT NULL,  
  FSH NUMBER NOT NULL,  
  CONSTRAINT RF_CC_FER FOREIGN KEY(C_CLINICA) REFERENCES  
  CARTELLA_CLINICA(C_CLINICA)  
);
```

```
CREATE TABLE TURNO(  
  COD_TURNO NUMBER PRIMARY KEY,  
  INIZIO_TURNO DATE NOT NULL CHECK  
  (TO_CHAR(INIZIO_TURNO, 'HH24:MI:SS')>'08:29:00'),  
  FINE_TURNO DATE NOT NULL CHECK  
  (TO_CHAR(FINE_TURNO, 'HH24:MI:SS')<'20:31:00')  
);
```

```
CREATE TABLE SALA(  

```

```

NUM_SALA NUMBER PRIMARY KEY,
ALA CHAR(1) NOT NULL,
PIANO NUMBER NOT NULL
);

```

```

CREATE TABLE PRENOTAZIONE(
TIPO_PREN VARCHAR(20),
DATA_ORA DATE NOT NULL,
ID_PREN NUMBER PRIMARY KEY,
NUM_SALA NUMBER,
CF_DON CHAR(16),
CF_CLI CHAR(16),
CONSTRAINT P_CLIENTE FOREIGN KEY (CF_CLI) REFERENCES
CLIENTE(CF_CLI),
CONSTRAINT P_DONATORE FOREIGN KEY (CF_DON) REFERENCES
DONATORE(CF_DON),
CONSTRAINT SI_ESEGUE_IN FOREIGN KEY (NUM_SALA) REFERENCES
SALA(NUM_SALA),
CONSTRAINT TIPO_PRENOTAZIONE CHECK(TIPO_PREN
IN('DONAZIONE','INSEMINAZIONE','SPERMIOGRAMMA','GENETICO','SANGUE'
,'FERTILITA'))
);

```

```

CREATE TABLE LAVORA_IN(
CF_DOT CHAR(16),
NUM_SALA NUMBER,
DATA_ORA DATE,
CONSTRAINT PK_LAVORA_IN PRIMARY KEY (CF_DOT,NUM_SALA, DATA_ORA),
CONSTRAINT FK_LAVORA_IN_DOT FOREIGN KEY(CF_DOT) REFERENCES
DOTTORE(CF_DOT),
CONSTRAINT FK_LAVORA_IN_SALA FOREIGN KEY(NUM_SALA) REFERENCES
SALA(NUM_SALA)
);

```

```

CREATE TABLE EFFETTUA_T(
DATA_T DATE,
CF_DOT CHAR(16),
COD_TURNO NUMBER,
CONSTRAINT TURNO_DOT_PRE FOREIGN KEY(CF_DOT) REFERENCES
DOTTORE(CF_DOT),
CONSTRAINT TURNO_COD_PRE FOREIGN KEY(COD_TURNO) REFERENCES
TURNO(COD_TURNO),
CONSTRAINT PK_T_PRE PRIMARY KEY (CF_DOT,COD_TURNO, DATA_T)
);

```

```

ALTER TABLE FIALA_SEME ADD CONSTRAINT CONSERVATA FOREIGN
KEY(ID_CRIO) REFERENCES CRIOCELLA(ID_CRIO);

```

```

ALTER TABLE FIALA_SEME ADD CONSTRAINT DA_SEME FOREIGN
KEY(COD_E_SPERMIOGRAMMA) REFERENCES
SPERMIOGRAMMA(COD_E_SPERMIOGRAMMA);

ALTER TABLE FIALA_SEME ADD CONSTRAINT DA_GENETICO FOREIGN
KEY(COD_E_GENETICO) REFERENCES GENETICO(COD_E_GENETICO);

ALTER TABLE FIALA_SEME ADD CONSTRAINT UILIZZA FOREIGN KEY
(COD_INS) REFERENCES INSEMINAZIONE(COD_INS);

ALTER TABLE FIALA_SANGUE ADD CONSTRAINT DA_FERTILITA FOREIGN
KEY(COD_E_FERTILITA) REFERENCES FERTILITA(COD_E_FERTILITA);

ALTER TABLE FIALA_SANGUE ADD CONSTRAINT DA_SANGUE FOREIGN
KEY(COD_E_SANGUE) REFERENCES SANGUE(COD_E_SANGUE);

ALTER TABLE GENETICO ADD CONSTRAINT P_GENETICO FOREIGN
KEY(ID_PREN) REFERENCES PRENOTAZIONE(ID_PREN);

ALTER TABLE SANGUE ADD CONSTRAINT P_SANGUE FOREIGN KEY(ID_PREN)
REFERENCES PRENOTAZIONE(ID_PREN);

ALTER TABLE SPERMIOGRAMMA ADD CONSTRAINT P_SPERMIOGRAMMA FOREIGN
KEY(ID_PREN) REFERENCES PRENOTAZIONE(ID_PREN);

ALTER TABLE FERTILITA ADD CONSTRAINT P_FERTILITA FOREIGN
KEY(ID_PREN) REFERENCES PRENOTAZIONE(ID_PREN);

ALTER TABLE FIALA_SANGUE ADD CONSTRAINT CONSERVATO_SANGUE FOREIGN
KEY(ID_CRIO_SANGUE) REFERENCES CRIOCELLA(ID_CRIO);

ALTER TABLE INSEMINAZIONE ADD CONSTRAINT P_INSEMINAZIONE FOREIGN
KEY(ID_PREN) REFERENCES PRENOTAZIONE(ID_PREN);

```

Creazione Sequenze

```

CREATE SEQUENCE ID_FIALA_SA_S;
CREATE SEQUENCE ID_FIALA_SE_S;
CREATE SEQUENCE ID_CRIO_S;
CREATE SEQUENCE ID_PREN_S;
CREATE SEQUENCE TURNO_DOTTORE;
CREATE SEQUENCE ID_CC_S;
CREATE SEQUENCE COD_SANGUE_S;
CREATE SEQUENCE COD_SPERM_S;
CREATE SEQUENCE COD_GEN_S;
CREATE SEQUENCE COD_FERT_S;
CREATE SEQUENCE FATTURA_S;
CREATE SEQUENCE COD_INS_S;

```

Procedure

Supponendo che ogni servizio occupi in media un lasso di tempo di 30 minuti, creiamo gli slot di prenotazione per ogni mezz'ora, tenendo conto delle festività comandate, le domeniche e i giorni di festività della clinica.

```
CREATE OR REPLACE PROCEDURE ORGANIZZA_ORARI_PRENOTAZIONI IS
ANNO_CORRENTE NUMBER;
FLAG NUMBER;
BEGIN
SELECT      TO_NUMBER(TO_CHAR(TRUNC(SYSDATE, 'YEAR'), 'YYYY'))      INTO
ANNO_CORRENTE FROM DUAL;
IF MOD(ANNO_CORRENTE, 4) = 0
THEN FLAG:=1;
ELSE FLAG:=0;
END IF;

FOR J IN 1..5          -- Per ogni sala della Clinica
LOOP
FOR I IN 1..365+FLAG   -- Per tutti i giorni dell'anno
LOOP
IF TO_CHAR(TRUNC(SYSDATE, 'YEAR')+I, 'D') <> '7'
AND TRUNC(SYSDATE, 'YEAR')+I <> TO_DATE('06/01', 'DD/MM')
AND TRUNC(SYSDATE, 'YEAR')+I <> TO_DATE('25/04', 'DD/MM')
AND TRUNC(SYSDATE, 'YEAR')+I <> TO_DATE('01/05', 'DD/MM')
AND TRUNC(SYSDATE, 'YEAR')+I <> TO_DATE('22/05', 'DD/MM')
AND TRUNC(SYSDATE, 'YEAR')+I <> TO_DATE('25/05', 'DD/MM')
AND TRUNC(SYSDATE, 'YEAR')+I <> TO_DATE('02/06', 'DD/MM')
AND TRUNC(SYSDATE, 'YEAR')+I <> TO_DATE('15/08', 'DD/MM')
AND TRUNC(SYSDATE, 'YEAR')+I <> TO_DATE('11/10', 'DD/MM')
AND TRUNC(SYSDATE, 'YEAR')+I <> TO_DATE('01/11', 'DD/MM')
AND TRUNC(SYSDATE, 'YEAR')+I <> TO_DATE('02/11', 'DD/MM')
AND TRUNC(SYSDATE, 'YEAR')+I <> TO_DATE('08/12', 'DD/MM')
AND TRUNC(SYSDATE, 'YEAR')+I NOT BETWEEN TO_DATE('24/12', 'DD/MM')
AND TO_DATE('31/12', 'DD/MM')
THEN -- Finché non è uno di questi giorni inserisci 24 tuple
in Prenotazione
INSERT INTO PRENOTAZIONE(DATA_ORA, NUM_SALA)
SELECT (1/48)*16+TRUNC(SYSDATE, 'YEAR')+I+(1/48)*LEVEL , J
FROM DUAL
CONNECT BY LEVEL <= 24
ORDER BY LEVEL;
END IF;
END LOOP;
END LOOP;
END;
```

La seguente procedura crea i turni di lavoro dei dottori in modo pseudo_casuale, garantendo che ogni dottore lavori ogni settimana o di mattina o di pomeriggio.

```
CREATE OR REPLACE PROCEDURE ORGANIZZA_TURNI_DOTTORI IS
ID_GIORNO_CORRENTE NUMBER;
OFFSET NUMBER;
COD_TURNO_F NUMBER;
NUMERO_DOT NUMBER;
NUMERO NUMBER;
BEGIN
SELECT TO_NUMBER(TO_CHAR(SYSDATE, 'D')) INTO ID_GIORNO_CORRENTE
FROM DUAL;

IF ID_GIORNO_CORRENTE = 7
THEN
    OFFSET :=1;
    ID_GIORNO_CORRENTE:=1;
ELSE OFFSET := 0;
END IF;

SELECT COUNT(*) INTO NUMERO
FROM DOTTORE;

IF (MOD(NUMERO, 2) = 0)      -- Questo calcolo ci permette di far
    -- lavorare tutti i dottori, nessuno escluso, nel caso in cui il
    -- numero
    -- di dottori presente nella clinica
    sia dispari
THEN
    NUMERO_DOT:= NUMERO/2;
ELSE
    NUMERO_DOT:=(NUMERO/2) + 1;
END IF;

FOR I IN ID_GIORNO_CORRENTE..6 LOOP      -- Per ogni giorno della
    settimana
        INSERT INTO EFFETTUA_T (CF_DOT, COD_TURNO, DATA_T) --
        Inserisci i dottori randomicamente
        SELECT          CF_DOT,          1,          TRUNC(SYSDATE+I-
        ID_GIORNO_CORRENTE+OFFSET)
        FROM DOTTORE
        WHERE ROWNUM<=NUMERO_DOT
        ORDER BY DBMS_RANDOM.VALUE;
    END LOOP; --FINE FOR PER I TURNI DI MATTINA
```

```

--Il seguente FOR assegna i turni per il pomeriggio.
FOR I IN ID_GIORNO_CORRENTE..6 LOOP
    INSERT INTO EFFETTUA_T (CF_DOT, COD_TURNO, DATA_T)
        SELECT          CF_DOT,          2,          TRUNC(SYSDATE+I-
ID_GIORNO_CORRENTE+OFFSET)
        FROM DOTTORE
        WHERE ROWNUM<=NUMERO_DOT AND CF_DOT NOT IN (
            SELECT T.CF_DOT FROM EFFETTUA_T T WHERE
TRUNC(DATA_T)=TRUNC(SYSDATE+I-ID_GIORNO_CORRENTE+OFFSET))
        ORDER BY DBMS_RANDOM.VALUE;
    END LOOP;
END;

```

```

-- Procedure automaticamente richiamata e assegna automaticamente
il dottore per il servizio appena prenotato

```

```

CREATE OR REPLACE PROCEDURE ASSEGNA_DOT (NUMERO_SALA NUMBER,
DATA_P DATE ) IS
TURNO_DA_ASS NUMBER;
ORARIO_NON_DISP EXCEPTION;

CF_FLAG CHAR(16);

BEGIN
TURNO_DA_ASS :=0;
IF TO_CHAR(DATA_P, 'HH24:MI')>=TO_CHAR(TO_DATE('8:30', 'HH24:MI'),
'HH24:MI') AND TO_CHAR(DATA_P, 'HH24:MI') <
TO_CHAR(TO_DATE('14:30', 'HH24:MI'), 'HH24:MI')
THEN
TURNO_DA_ASS :=1;
ELSE IF TO_CHAR(DATA_P, 'HH24:MI')>=TO_CHAR(TO_DATE('14:30',
'HH24:MI'), 'HH24:MI') AND TO_CHAR(DATA_P, 'HH24:MI') <
TO_CHAR(TO_DATE('20:30', 'HH24:MI'), 'HH24:MI')
THEN
TURNO_DA_ASS :=2;
END IF;
END IF;

IF TURNO_DA_ASS = 0
THEN
RAISE ORARIO_NON_DISP;
END IF;

SELECT CF_DOT INTO CF_FLAG
FROM (SELECT * FROM EFFETTUA_T E JOIN TURNO T ON
E.COD_TURNO=T.COD_TURNO

```

```

WHERE T.COD_TURNO = TURNO_DA_ASS AND TO_CHAR(DATA_P, 'DD/MM/YYYY')=TO_CHAR(E.DATA_T, 'DD/MM/YYYY') AND E.CF_DOT NOT IN -- Il
dottore non deve essere occupato
(SELECT CF_DOT FROM LAVORA_IN WHERE TO_CHAR(DATA_ORA, 'DD/MM/YYYY
HH24:MI')=TO_CHAR(DATA_P, 'DD/MM/YYYY HH24:MI'))
ORDER BY DBMS_RANDOM.VALUE)
WHERE ROWNUM<=1;

INSERT INTO LAVORA_IN(CF_DOT, NUM_SALA, DATA_ORA)
VALUES (CF_FLAG, NUMERO_SALA, DATA_P);

EXCEPTION

WHEN ORARIO_NON_DISP
THEN RAISE_APPLICATION_ERROR(-20102, 'ORA SELEZIONATA NON VALIDA');

END;

```

La seguente procedura inserisce una prenotazione nella tabella Prenotazione, richiamando la procedura precedentemente descritta [assegna_dot](#). Prende come dati di input il codice fiscale, la data e l'ora desiderata, il tipo e il numero di sala. Prima di fare l'inserimento controlla che ci sia effettivamente la sala disponibile richiesta, che ci sia un dottore libero per poter essere associato al servizio. In più si assicura che il cliente o il donatore non siano già occupati in un'altra sala, nello stesso orario. Si assicura, inoltre, che il codice fiscale inserito appartenga ad un cliente o a un donatore registrato. E in più, nel caso in cui il tipo della prenotazione sia inseminazione, si assicura che la cliente non abbia malattie infettive o problemi di fertilità, nel caso, la prenotazione non va a buon fine.

```

CREATE OR REPLACE PROCEDURE INSERISCI_PRENOTAZIONE (DATA_INS DATE,
CF_INS CHAR, TIPO_INS VARCHAR, SALA_INS NUMBER) IS
NUM_DOTTORI NUMBER;
NUM_PREN NUMBER;
PRENOTAZIONE_DA_MODIFICARE NUMBER;
NUM_DOTTORI_OCCUPATI NUMBER;
SALE_DISPONIBILI NUMBER;
FLAG NUMBER;
FLAG_2 NUMBER;
NUMERO_FIALE_SANGUE NUMBER;
DISCRIMINATORE NUMBER;
NUM_PREN_STESSA_PERSONA NUMBER;
INIZIO_TURNO_VAR DATE;

SALA_OCCUPATA EXCEPTION;
NO_DOTTORI_DISPONIBILI EXCEPTION;
CF_INESISTENTE EXCEPTION;
ONNIPRESENZA EXCEPTION;

```

```

INSEMINAZIONE_NON_DISPONIBILE EXCEPTION;

BEGIN
PRENOTAZIONE_DA_MODIFICARE := 0;
NUM_DOTTORI:=0;
NUM_PREN:=0;
SALE_DISPONIBILI:=0;
FLAG:=0;
FLAG_2 :=0;
--PRENDO L'INIZIO TURNO POMERIDIANO, CHE CORRISPONDE ALLE 14:30
SELECT INIZIO_TURNO INTO INIZIO_TURNO_VAR FROM TURNO WHERE
COD_TURNO='2';

IF (TIPO_INS = 'INSEMINAZIONE' )
THEN SELECT COUNT(*) INTO NUMERO_FIALE_SANGUE FROM FIALA_SANGUE
WHERE CF_CLI = CF_INS AND SANO = 'SI';
IF NUMERO_FIALE_SANGUE <1
THEN RAISE INSEMINAZIONE_NON_DISPONIBILE;
END IF;
END IF;

IF (TO_CHAR(DATA_INS, 'HH24:MI') < TO_CHAR(INIZIO_TURNO_VAR,
'HH24:MI'))
THEN
SELECT COUNT(*) INTO NUM_DOTTORI
FROM EFFETTUA_T E JOIN TURNO T ON E.COD_TURNO=T.COD_TURNO
WHERE TO_CHAR(E.DATA_T, 'DD/MM/YYYY')=TO_CHAR(DATA_INS, 'DD/MM/
YYYY') AND T.COD_TURNO='1';

SELECT COUNT(*) INTO NUM_DOTTORI_OCCUPATI FROM LAVORA_IN WHERE
TO_CHAR(DATA_INS, 'DD/MM/YYYY HH24:MI') = TO_CHAR(DATA_ORA, 'DD/
MM/YYYY HH24:MI');

NUM_DOTTORI := NUM_DOTTORI - NUM_DOTTORI_OCCUPATI;

ELSE

SELECT COUNT(*) INTO NUM_DOTTORI
FROM EFFETTUA_T E JOIN TURNO T ON E.COD_TURNO=T.COD_TURNO
WHERE TO_CHAR(E.DATA_T, 'DD/MM/YYYY')=TO_CHAR(DATA_INS, 'DD/MM/
YYYY') AND T.COD_TURNO='2';

SELECT COUNT(*) INTO NUM_DOTTORI_OCCUPATI FROM LAVORA_IN WHERE
TO_CHAR(DATA_INS, 'DD/MM/YYYY HH24:MI') = TO_CHAR(DATA_ORA, 'DD/
MM/YYYY HH24:MI');

NUM_DOTTORI := NUM_DOTTORI - NUM_DOTTORI_OCCUPATI;

```

```

END IF;

SELECT COUNT(*) INTO NUM_PREN
FROM PRENOTAZIONE
WHERE TO_CHAR(DATA_ORA, 'DD/MM/YYYY HH24:MI')=TO_CHAR(DATA_INS, 'DD/MM/YYYY HH24:MI') AND CF_CLI IS
NOT NULL AND CF_DON IS NOT NULL;

SELECT COUNT(*) INTO NUM_PREN_STESSA_PERSONA
FROM PRENOTAZIONE
WHERE TO_CHAR(DATA_ORA, 'DD/MM/YYYY HH24:MI')=TO_CHAR(DATA_INS, 'DD/MM/YYYY HH24:MI') AND (CF_CLI =
CF_INS OR CF_DON = CF_INS);

IF (NUM_PREN_STESSA_PERSONA > 0)
THEN RAISE ONNIPRESENZA;
END IF;

IF (NUM_DOTTORI <= NUM_PREN)
THEN RAISE NO_DOTTORI_DISPONIBILI;
END IF;

SELECT COUNT(*) INTO DISCRIMINATORE
FROM PRENOTAZIONE
WHERE TO_CHAR(DATA_ORA, 'DD/MM/YYYY HH24:MI')=TO_CHAR(DATA_INS, 'DD/MM/YYYY HH24:MI') AND NUM_SALA =
SALA_INS AND (CF_CLI IS NOT NULL OR CF_DON IS NOT NULL);

IF (DISCRIMINATORE > 0)
THEN RAISE SALA_OCCUPATA;
END IF;

SELECT ID_PREN INTO PRENOTAZIONE_DA_MODIFICARE FROM PRENOTAZIONE
WHERE TO_CHAR(DATA_ORA, 'DD/MM/YYYY HH24:MI')=TO_CHAR(DATA_INS, 'DD/MM/YYYY HH24:MI') AND NUM_SALA =
SALA_INS;

SELECT COUNT(*) INTO FLAG FROM DONATORE WHERE CF_DON = CF_INS;
IF FLAG >0
THEN
UPDATE PRENOTAZIONE
SET TIPO_PREN = TIPO_INS, CF_DON = CF_INS
WHERE ID_PREN = PRENOTAZIONE_DA_MODIFICARE;
END IF;

```

```

SELECT COUNT(*) INTO FLAG_2 FROM CLIENTE WHERE CF_CLI = CF_INS;
IF FLAG_2 >0
THEN
UPDATE PRENOTAZIONE
SET TIPO_PREN = TIPO_INS, CF_CLI = CF_INS
WHERE ID_PREN = PRENOTAZIONE_DA_MODIFICARE;
END IF;

IF FLAG = 0 AND FLAG_2 = 0
THEN RAISE CF_INESISTENTE;
END IF;

ASSEGNA_DOT(SALA_INS, DATA_INS);

EXCEPTION
WHEN NO_DOTTORI_DISPONIBILI
THEN RAISE_APPLICATION_ERROR(-20027, 'NON CI SONO DOTTORI
DISPONIBILI PER PRENOTARE');

WHEN SALA_OCCUPATA
THEN RAISE_APPLICATION_ERROR(-20037, 'LA SALA E GIA OCCUPATA');

WHEN CF_INESISTENTE
THEN RAISE_APPLICATION_ERROR(-20029, 'NON SO A CHI APPARTENGA
QUESTO CODICE FISCALE');

WHEN ONNIPRESENZA
THEN RAISE_APPLICATION_ERROR(-20049, 'IL CLIENTE E GIA IN UN ALTRA
SALA');

WHEN INSEMINAZIONE_NON_DISPONIBILE
THEN RAISE_APPLICATION_ERROR(-20109, 'INSEMINAZIONE NON
DISPONIBILE. FIALA DEL SANGUE RELATIVA NON TROVATA OPPURE
INFETTA');

END;

```

La procedura seguente svuota la tupla di prenotazione precedentemente riempita con i dati del cliente o del donatore e del tipo di prenotazione. Rendendo a tutti gli effetti quella ora e quella sala nuovamente disponibile per essere prenotata. Richiama all'interno la procedura [de_assegna_dot](#) che vedremo successivamente.

```

CREATE OR REPLACE PROCEDURE ELIMINA_PRENOTAZIONE (ID NUMBER ) IS
FLAG NUMBER;
NO_PREN EXCEPTION;

```

```

DATAS DATE;
NUMSALA NUMBER;
BEGIN
SELECT COUNT(*) INTO FLAG
FROM PRENOTAZIONE
WHERE ID_PREN=ID;

IF FLAG>0
THEN
UPDATE PRENOTAZIONE
SET CF_CLI=NULL, CF_DON=NULL, TIPO_PREN=NULL
WHERE ID_PREN=ID;

SELECT DATA_ORA, NUM_SALA INTO DATAS, NUMSALA
FROM PRENOTAZIONE
WHERE ID_PREN=ID;

DE_ASSEGNA_DOT(DATAS, NUMSALA);

ELSE
RAISE NO_PREN;
END IF;
EXCEPTION
WHEN NO_PREN
THEN RAISE_APPLICATION_ERROR(-20034, 'LA TUPLA E VUOTA');
END;

/

CREATE OR REPLACE PROCEDURE DE_ASSEGNA_DOT (DATAS DATE, NUMSALA
NUMBER) IS
BEGIN
DELETE LAVORA_IN WHERE DATA_ORA = DATAS AND NUM_SALA = NUMSALA;
END;

```

La seguente procedura inserisce una tupla nella tabella Fiala_Sangue, prendendo da input l'id prenotazione di un'esame associato, che può essere sia quello del sangue che quello della fertilità, il codice dell'esame fatto e un discriminatore Flag che ci dice se è un esame del sangue o della fertilità. Richiama una funzione [get_id_crio](#) che vedremo in seguito.

```

CREATE OR REPLACE PROCEDURE INSERISCI_FIALA_SANGUE (ID_PRENO
NUMBER, COD_ESAME CHAR, FLAG NUMBER) IS
ID_FIALA CHAR(4);
CF_DA_INSERIRE CHAR(16);
DATA_DA_INSERIRE DATE;
ID_CRIO_DA_INS NUMBER;

```

```

FLAG_COUNT NUMBER;
BEGIN
FLAG_COUNT:=0;
SELECT COUNT(*) INTO FLAG_COUNT FROM PRENOTAZIONE WHERE CF_CLI IS
NOT NULL AND ID_PREN=ID_PRENO;

IF (FLAG_COUNT>0)
THEN
    SELECT CF_CLI, DATA_ORA INTO CF_DA_INSERTIRE, DATA_DA_INSERTIRE
    FROM PRENOTAZIONE
    WHERE ID_PREN=ID_PRENO;
ELSE
    SELECT CF_DON, DATA_ORA INTO CF_DA_INSERTIRE, DATA_DA_INSERTIRE
    FROM PRENOTAZIONE
    WHERE ID_PREN=ID_PRENO;
END IF;

ID_FIALA:= 'F' || TO_CHAR(ID_FIALA_SE_S.NEXTVAL);

SELECT GET_ID_CRIO(1) INTO ID_CRIO_DA_INS FROM DUAL;

IF (FLAG = 2 AND FLAG_COUNT=0) -- FLAG=2 VUOL DIRE CHE E' UN ESAME
DEL SANGUE CLASSICO ED E' UN DONATORE
THEN
    INSERT INTO FIALA_SANGUE (DATA_SANGUE, ID_FIALA_SA, CF_DON,
ID_CRIO_SANGUE, COD_E_SANGUE )
    SELECT      DATA_DA_INSERTIRE,      ID_FIALA,      CF_DA_INSERTIRE,
ID_CRIO_DA_INS, COD_ESAME FROM DUAL;

ELSE IF (FLAG =2 AND FLAG_COUNT>0) -- VUOL DIRE CHE E' UN CLIENTE
THEN
    INSERT INTO FIALA_SANGUE (DATA_SANGUE, ID_FIALA_SA, CF_DON,
ID_CRIO_SANGUE, COD_E_SANGUE )
    SELECT      DATA_DA_INSERTIRE,      ID_FIALA,      CF_DA_INSERTIRE,
ID_CRIO_DA_INS, COD_ESAME FROM DUAL;

ELSE

    INSERT INTO FIALA_SANGUE (DATA_SANGUE, ID_FIALA_SA, CF_CLI,
ID_CRIO_SANGUE, COD_E_FERTILITA)
    SELECT      DATA_DA_INSERTIRE,      ID_FIALA,      CF_DA_INSERTIRE,
ID_CRIO_DA_INS, COD_ESAME FROM DUAL;
END IF;
END IF;
END;

```

La seguente procedura inserisce l'esame del sangue nella tabella Sangue.

```

CREATE OR REPLACE PROCEDURE INSERT_ES_SANGUE (ID_INS  NUMBER,
TPHA_INS  NUMBER, VLDRL_INS  NUMBER, LINF_INS  NUMBER) IS
DATA_PREN  DATE;
CODICE  CHAR(8);

FLAG  NUMBER;

COD_INS  CHAR(8);

BEGIN
SELECT DATA_ORA INTO DATA_PREN
FROM PRENOTAZIONE
WHERE ID_PREN=ID_INS;

-- POICHE SIA IL CLIENTE CHE IL DONATORE DEVONO FARE L'ESAME DEL
-- SANGUE, RIUSCIRE AD OTTENERE IL CODICE DELLA CARTELLA CLINICA
-- E PIU' COMPLESSO. INFATTI DOBBIAMO TENERE CONTO DI COME E'
-- STRUTTURATA LA TABELLA PRENOTAZIONE, CHE HA SIA IL CODICE FISCALE
-- DEL CLIENTE CHE DEL
-- DONATORE. QUINDI DOBBIAMO PRIMA FILTRARE LE INFORMAZIONI PER
-- OTTENERE IL GIUSTO CODICE.

FLAG:= 0;

SELECT COUNT(*) INTO FLAG
FROM PRENOTAZIONE P JOIN CLIENTE C ON C.CF_CLI=P.CF_CLI
WHERE ID_PREN=ID_INS;

IF (FLAG>0)
THEN

    SELECT C_CLINICA INTO CODICE
    FROM (PRENOTAZIONE P JOIN CLIENTE C ON C.CF_CLI=P.CF_CLI) JOIN
    CARTELLA_CLINICA CC ON CC.CF_CLI=C.CF_CLI
    WHERE ID_PREN=ID_INS;

ELSE

    SELECT COUNT(*) INTO FLAG
    FROM PRENOTAZIONE P JOIN DONATORE D ON D.CF_DON=P.CF_DON
    WHERE ID_PREN=ID_INS;

    IF (FLAG>0)
    THEN
        SELECT C_CLINICA INTO CODICE -- CERCO LA CARTELLA CLINICA
        ASSOCIATA DEL DONATORE
        FROM (PRENOTAZIONE P JOIN DONATORE D ON D.CF_DON=P.CF_DON)
        JOIN CARTELLA_CLINICA CC ON CC.CF_DON=D.CF_DON
        WHERE ID_PREN=ID_INS;
    END IF;

```

```

END IF;

COD_INS:='CSA' || TO_CHAR(COD_SANGUE_S.NEXTVAL);

INSERT INTO SANGUE (COD_E_SANGUE, ID_PREN, C_CLINICA, DATA_E_SA,
TPHA, VLDRL, LINFOCITICD4)
SELECT COD_INS, ID_INS, CODICE, DATA_PREN, TPHA_INS, VLDRL_INS,
LINF_INS FROM DUAL;

INSERISCI_FIALA_SANGUE (ID_INS, COD_INS, 2);

END;

/

CREATE OR REPLACE PROCEDURE INSERT_ES_SPERM (ID_INS NUMBER,
CNTZ_INS NUMBER, VOL_INS NUMBER, MOT_INS NUMBER, PH_INS NUMBER )
IS
DATA_PREN DATE;
CODICE CHAR(8);

COD_INS CHAR(8);

BEGIN

SELECT DATA_ORA INTO DATA_PREN
FROM PRENOTAZIONE
WHERE ID_PREN=ID_INS;

SELECT C_CLINICA INTO CODICE -- CERCO LA CARTELLA CLINICA
ASSOCIATA DEL DONATORE
FROM (PRENOTAZIONE P JOIN DONATORE D ON D.CF_DON=P.CF_DON) JOIN
CARTELLA_CLINICA CC ON CC.CF_DON=D.CF_DON
WHERE ID_PREN=ID_INS;

COD_INS:='CSP' || TO_CHAR(COD_SPERM_S.NEXTVAL);

INSERT INTO SPERMIOGRAMMA(COD_E_SPERMIOGRAMMA, ID_PREN, C_CLINICA,
DATA_E_SP, CNTZ,VOL,MOT,PH)
SELECT COD_INS, ID_INS, CODICE, DATA_PREN, CNTZ_INS, VOL_INS,
MOT_INS, PH_INS FROM DUAL;

END;

```

La seguente procedura inserisce i valori dell'esame Genetico nella tabella associata.

```

CREATE OR REPLACE PROCEDURE INSERT_ES_GENET(ID_INS NUMBER, IRT_INS
NUMBER, ESO_INS NUMBER, CARIO_INS CHAR) IS

```

```

DATA_PREN DATE;
CODICE CHAR(8);

COD_INS CHAR(8);

BEGIN

SELECT DATA_ORA INTO DATA_PREN
FROM PRENOTAZIONE
WHERE ID_PREN=ID_INS;

SELECT C_CLINICA INTO CODICE
FROM (PRENOTAZIONE P JOIN DONATORE D ON D.CF_DON=P.CF_DON) JOIN
CARTELLA_CLINICA CC ON CC.CF_DON=D.CF_DON
WHERE ID_PREN=ID_INS;

COD_INS:='CGE' || TO_CHAR(COD_GEN_S.NEXTVAL);

INSERT INTO GENETICO(COD_E_GENETICO, ID_PREN, C_CLINICA, DATA_GEN,
IRT, ESONE7, CARIOTIPO)
SELECT COD_INS, ID_INS, CODICE, DATA_PREN, IRT_INS, ESO_INS,
CARIO_INS FROM DUAL;

END;

/

```

La seguente procedura inserisce i valori dell'esame fertilità nella tabella associata.

```

CREATE OR REPLACE PROCEDURE INSERT_ES_FERT(ID_INS NUMBER, PROL_INS
NUMBER, FSH_INS NUMBER) IS
DATA_PREN DATE;
CODICE CHAR(8);

COD_INS CHAR(8);

BEGIN
SELECT CC.C_CLINICA, P.DATA_ORA INTO CODICE, DATA_PREN
FROM (PRENOTAZIONE P JOIN CLIENTE C ON C.CF_CLI=P.CF_CLI) JOIN
CARTELLA_CLINICA CC ON CC.CF_CLI=C.CF_CLI
WHERE P.ID_PREN=ID_INS;

COD_INS:='CFE' || TO_CHAR(COD_FERT_S.NEXTVAL);

INSERT INTO FERTILITA(COD_E_FERTILITA, ID_PREN, C_CLINICA,
DATA_E_FE, PROLATTINA, FSH)
SELECT COD_INS, ID_INS, CODICE, DATA_PREN, PROL_INS, FSH_INS FROM
DUAL;

INSERISCI_FIALA_SANGUE (ID_INS, COD_INS, 1);

```

END;

La seguente procedura viene automaticamente richiamata dalla procedura [inserisci_donatore](#) e [inserisci_cliente](#). Risponde alla necessità che ogni cliente e donatore della clinica debbano avere una cartella clinica interna alla banca del seme, anche se non dovessero essere donatori o clienti compatibili. Si assicura inoltre che il codice fiscale sia stato effettivamente inserito nella tabella Cliente o Donatore.

```
CREATE OR REPLACE PROCEDURE APRI_CARTELLA (CF_INS CHAR) IS
ID_CC CHAR(8);
FLAG_DON NUMBER;
FLAG_CLI NUMBER;
SESSO_FLAG CHAR(1);

SCONOSCIUTO EXCEPTION;

BEGIN

FLAG_DON:=0;
FLAG_CLI:=0;

SELECT COUNT(*), MAX(SESSO_CLI) INTO FLAG_CLI, SESSO_FLAG FROM
CLIENTE WHERE CF_CLI=CF_INS;
SELECT COUNT(*) INTO FLAG_DON FROM DONATORE WHERE CF_DON=CF_INS;

IF (FLAG_DON>0)
THEN
ID_CC:= 'CC' || TO_CHAR(ID_CC_S.NEXTVAL);

INSERT INTO CARTELLA_CLINICA(C_CLINICA, DATA_CREAZIONE, CF_DON)
SELECT ID_CC, SYSDATE, CF_INS FROM DUAL;

ELSE IF (FLAG_CLI>0 AND SESSO_FLAG = 'F')
THEN
ID_CC:= 'CC' || TO_CHAR(ID_CC_S.NEXTVAL);

INSERT INTO CARTELLA_CLINICA(C_CLINICA, DATA_CREAZIONE, CF_CLI)
SELECT ID_CC, SYSDATE, CF_INS FROM DUAL;
ELSE IF (FLAG_CLI>0 AND SESSO_FLAG = 'M')
THEN RETURN;
ELSE
RAISE SCONOSCIUTO;
END IF;
END IF;
END IF;

EXCEPTION
```

```

WHEN SCONOSCIUTO
THEN RAISE_APPLICATION_ERROR(-20234, 'QUESTA PERSONA NON E
REGISTRATA');
END;

/

CREATE OR REPLACE PROCEDURE INSERISCI_CLIENTE (CF CHAR, NOME CHAR,
COGNOME CHAR, DATA_N DATE, S CHAR, CITTA CHAR, VIA CHAR, TELEFONO
CHAR, PROVINCIA CHAR) IS
BEGIN

INSERT INTO CLIENTE (CF_CLI, NOME_CLI, COGNOME_CLI, DN_CLI,
SESSO_CLI, CITTA_CLI, VIA_CLI, TELEFONO_CLI, PROV_CLI)
SELECT CF, NOME, COGNOME, DATA_N, S, CITTA, VIA, TELEFONO,
PROVINCIA FROM DUAL;

APRI_CARTELLA(CF);

END;

/

CREATE OR REPLACE PROCEDURE INSERISCI_DONATORE(CF CHAR, NOME CHAR,
COGNOME CHAR, DATA_N DATE, CITTA CHAR, VIA CHAR, G CHAR, OCC CHAR,
PE NUMBER, CAP CHAR, AN CHAR, ETN CHAR, ALT NUMBER, TELEFONO CHAR,
PROVINCIA CHAR) IS
BEGIN
INSERT INTO DONATORE(CF_DON, NOME_DON, COGNOME_DON, DN_DON,
CITTA_DON, VIA_DON, G_SANGUIGNO, OCCHI, PESO, CAPELLI, ANONIMATO,
ETNIA, ALTEZZA, TELEFONO_DON, PROV_DON)
SELECT CF, NOME, COGNOME, DATA_N, CITTA, VIA, G, OCC, PE, CAP, AN,
ETN, ALT, TELEFONO, PROVINCIA FROM DUAL;

APRI_CARTELLA(CF);

END;

/

CREATE OR REPLACE PROCEDURE INSERISCI_FIALA_SEME(ID_PRENO NUMBER,
COD_SPERM CHAR, COD_GENET CHAR) IS
ID_FIALA CHAR(4);
CF_DA_INSERIRE CHAR(16);
DATA_DA_INSERIRE DATE;

ID_CRIO_DA_INS NUMBER;
BEGIN

-- CERCO IL CODICE FISCALE DEL DONATORE E LA DATA DELLA
PRENOTAZIONE

```

```

SELECT CF_DON, DATA_ORA INTO CF_DA_INSERTIRE, DATA_DA_INSERTIRE
FROM PRENOTAZIONE
WHERE ID_PREN=ID_PRENO;

ID_FIALA:= 'F' || TO_CHAR(ID_FIALA_SE_S.NEXTVAL);

SELECT GET_ID_CRIO(2) INTO ID_CRIO_DA_INS FROM DUAL;

INSERT INTO FIALA_SEME(CF_DON, DATA_SEME, ID_FIALA_SE, ID_CRIO,
COD_E_SPERMIOGRAMMA, COD_E_GENETICO)
SELECT CF_DA_INSERTIRE, DATA_DA_INSERTIRE, ID_FIALA, ID_CRIO_DA_INS,
COD_SPERM, COD_GENET FROM DUAL;

END;

/

```

La seguente procedura serve a registrare l'acquisto di una fiala. Si assicura che la fiala del seme sia disponibile e sana.

```

CREATE OR REPLACE PROCEDURE ACQUISTO_FIALA (ID_FIALA CHAR, CF_INS
CHAR, METODO_PAG CHAR) IS
FLAG_ESIST NUMBER;
FLAG_VEND NUMBER;
FLAG_SANO CHAR(2);
COD_FATTURA CHAR(10);
TIPO_PAG VARCHAR(20);

NO_DISPONIBILE EXCEPTION;
NON_ESISTE EXCEPTION;
GIA_VENDUTA EXCEPTION;

BEGIN
TIPO_PAG := 'FIALA';
FLAG_ESIST:=0; FLAG_VEND:=0;

-- VEDIAMO SE LA FIALA ESISTE PER ESSERE ACQUISTATA
SELECT COUNT(*) INTO FLAG_ESIST
FROM FIALA_SEME
WHERE ID_FIALA_SE=ID_FIALA;

IF (FLAG_ESIST<=0)
THEN RAISE NON_ESISTE;
END IF;

-- LA FIALA NON DEVE ESSERE GIA STATA VENDUTA
SELECT COUNT(*) INTO FLAG_VEND
FROM FIALA_SEME
WHERE ID_FIALA_SE=ID_FIALA AND FATTURA IS NOT NULL;

```

```

IF (FLAG_VEND>0)
THEN RAISE GIA_VENDUTA;
END IF;

-- NON SI PUO ACQUISTARE UNA FIALA NON SANA
SELECT SANO INTO FLAG_SANO
FROM FIALA_SEME
WHERE ID_FIALA_SE=ID_FIALA;

IF(FLAG_SANO='NO')
THEN RAISE NO_DISPONIBILE;
END IF;

-- BISOGNA INSERIRE LA TUPLA IN PAGAMENTO
COD_FATTURA:= 'FAT' || TO_CHAR(FATTURA_S.NEXTVAL);

INSERT INTO PAGAMENTO(TIPO_PA, FATTURA, DATAP, METODOP,CF_CLI)
SELECT TIPO_PAG, COD_FATTURA, SYSDATE, METODO_PAG, CF_INS FROM
DUAL;

UPDATE FIALA_SEME SET FATTURA=COD_FATTURA WHERE
ID_FIALA_SE=ID_FIALA;
EXCEPTION
WHEN NON_ESISTE
THEN RAISE_APPLICATION_ERROR(-20335,'NON SI PUO ACQUISTARE UNA
FIALA NON ESISTENTE');

WHEN GIA_VENDUTA
THEN RAISE_APPLICATION_ERROR(-20336,'LA FIALA E STATA GIA
VENDUTA');

WHEN NO_DISPONIBILE
THEN RAISE_APPLICATION_ERROR(-20337,'LA FIALA NON PUO ESSERE
UTILIZZATA NE VENDUTA');

END;

/

CREATE OR REPLACE PROCEDURE INSERISCI_CRIOCELLA(CAPIENZA NUMBER,
TIPO_CRIO CHAR) IS
BEGIN
INSERT INTO CRIOCELLA (ID_CRIO, CAPIENZA_MAX, TIPO)
SELECT ID_CRIO_S.NEXTVAL, CAPIENZA, TIPO_CRIO FROM DUAL;
END;

/

```

La seguente procedura inserisce una tupla nella tabella di Inseminazione, controllando che il numero di fiale richieste sia effettivamente disponibile.

```
CREATE OR REPLACE PROCEDURE INSERT_INSEMINAZIONE (NUM_FIALE
NUMBER, CF_DONATORE_F CHAR, TIPO_INSEMINAZIONE CHAR, FATTURA_INS
CHAR, ID_PREN_INS NUMBER) IS
COD_INSEM CHAR(8);
FIALE_ESISTENTI NUMBER;
ID_FIALA_SEME_AGG CHAR(4);
FIALE_MANCANTI EXCEPTION;
BEGIN

SELECT COUNT(*) INTO FIALE_ESISTENTI
FROM FIALA_SEME
WHERE CF_DON = CF_DONATORE_F AND SANO = 'SI' AND FATTURA IS NULL
AND COD_INS IS NULL;

IF FIALE_ESISTENTI < NUM_FIALE
THEN RAISE FIALE_MANCANTI;
END IF;

COD_INSEM:='INS' || TO_CHAR(COD_INS_S.NEXTVAL);

INSERT INTO INSEMINAZIONE (COD_INS, TIPO_INS, FATTURA, ID_PREN)
SELECT COD_INSEM, TIPO_INSEMINAZIONE, FATTURA_INS, ID_PREN_INS FROM
DUAL;

FOR I IN 1..NUM_FIALE LOOP

SELECT ID_FIALA_SE INTO ID_FIALA_SEME_AGG
FROM FIALA_SEME
WHERE SANO = 'SI' AND FATTURA IS NULL AND COD_INS IS NULL AND
CF_DON=CF_DONATORE_F AND ROWNUM = 1;
UPDATE FIALA_SEME SET FATTURA = FATTURA_INS, COD_INS = COD_INSEM
WHERE ID_FIALA_SE=ID_FIALA_SEME_AGG;

END LOOP;

EXCEPTION

WHEN FIALE_MANCANTI
THEN RAISE_APPLICATION_ERROR(-21490, 'MANCANO LE FIALE PER
EFFETTUARE L INSEMINAZIONE');

END;

/
```

La procedura per l'inserimento di una tupla in Inseminazione controlla che la cliente sia compatibile per l'inseminazione. Nel caso non lo sia, l'inserimento fallisce.

```
CREATE OR REPLACE PROCEDURE INSERISCI_PAG_INS (METODO VARCHAR,  
CF_INS CHAR, DATA_INS DATE) IS --BISOGNA PRIMA PAGARE  
-- DI FARE L'INSEMINAZIONE  
COD_FATTURA CHAR(10);  
COUNT_FLAG NUMBER;  
  
NOT_GRANTED EXCEPTION;  
BEGIN  
  
COUNT_FLAG:=0;  
  
SELECT COUNT(*) INTO COUNT_FLAG  
FROM FIALA_SANGUE  
WHERE CF_CLI=CF_INS AND SANO='NO';  
  
IF (COUNT_FLAG>=1)  
THEN RAISE NOT_GRANTED;  
END IF;  
  
COD_FATTURA:='FAT' || TO_CHAR(FATTURA_S.NEXTVAL);  
  
INSERT INTO PAGAMENTO (TIPO_PA, FATTURA, DATAP, METODOP,CF_CLI)  
SELECT 'INSEMINAZIONE',COD_FATTURA, DATA_INS, METODO, CF_INS FROM  
DUAL;  
  
EXCEPTION  
WHEN NOT_GRANTED  
THEN RAISE_APPLICATION_ERROR(-20267,'LA CLIENTE NON E COMPATIBILE  
ALL INSEMINAZIONE');  
  
END;
```

Funzioni

Nel database si utilizza una sola funzione che, all'inserimento di una fiala sangue o seme, viene richiamata restituendo come valore l'id della criocella libera in cui immagazzinare la nuova fiala.

```
CREATE OR REPLACE FUNCTION GET_ID_CRIO (TIPO NUMBER) RETURN NUMBER  
IS  
ID_DA_RESTITUIRE NUMBER;  
ID_DISPONIBILE NUMBER;  
  
SPAZIO_NON_DISPONIBILE EXCEPTION;  
BEGIN
```

— 1= SANGUE, 2= SEME.

IF (TIPO = 1)

THEN

```
SELECT COUNT(*) INTO ID_DISPONIBILE FROM(
SELECT CRIO.ID_CRIO
FROM CRIOCELLA CRIO
WHERE CRIO.TIPO = 'SANGUE'
GROUP BY CRIO.ID_CRIO
HAVING
(SELECT COUNT(*)
FROM CRIOCELLA C JOIN FIALA_SANGUE F ON C.ID_CRIO =
F.ID_CRIO_SANGUE
WHERE C.ID_CRIO = CRIO.ID_CRIO) < MAX(CRIO.CAPIENZA_MAX)
ORDER BY CRIO.ID_CRIO) WHERE ROWNUM=1;
```

IF (ID_DISPONIBILE = 0)

THEN RAISE SPAZIO_NON_DISPONIBILE;

END IF;

```
SELECT ID_CRIO INTO ID_DA_RESTITUIRE FROM(
SELECT CRIO.ID_CRIO
FROM CRIOCELLA CRIO
WHERE CRIO.TIPO = 'SANGUE'
GROUP BY CRIO.ID_CRIO
HAVING
(SELECT COUNT(*)
FROM CRIOCELLA C JOIN FIALA_SANGUE F ON C.ID_CRIO =
F.ID_CRIO_SANGUE
WHERE C.ID_CRIO = CRIO.ID_CRIO) < MAX(CRIO.CAPIENZA_MAX)
ORDER BY CRIO.ID_CRIO) WHERE ROWNUM=1;
```

RETURN ID_DA_RESTITUIRE;

ELSE

```
SELECT COUNT(*) INTO ID_DISPONIBILE FROM(
SELECT CRIO.ID_CRIO
FROM CRIOCELLA CRIO
WHERE CRIO.TIPO = 'SEME'
GROUP BY CRIO.ID_CRIO
HAVING
(SELECT COUNT(*)
FROM CRIOCELLA C JOIN FIALA_SEME F ON C.ID_CRIO =
F.ID_CRIO
WHERE C.ID_CRIO = CRIO.ID_CRIO) < MAX(CRIO.CAPIENZA_MAX)
ORDER BY CRIO.ID_CRIO) WHERE ROWNUM=1;
```

IF (ID_DISPONIBILE = 0)

```

    THEN RAISE SPAZIO_NON_DISPONIBILE;
END IF;

SELECT ID_CRIO INTO ID_DA_RESTITUIRE FROM(
SELECT CRIO.ID_CRIO
FROM CRIOCELLA CRIO
WHERE CRIO.TIPO = 'SEME'
GROUP BY CRIO.ID_CRIO
HAVING
(SELECT COUNT(*)
FROM CRIOCELLA C JOIN FIALA_SEME F ON C.ID_CRIO =
F.ID_CRIO
WHERE C.ID_CRIO = CRIO.ID_CRIO) < MAX(CRIO.CAPIENZA_MAX)
ORDER BY CRIO.ID_CRIO) WHERE ROWNUM=1;

RETURN ID_DA_RESTITUIRE;

END IF;
EXCEPTION
WHEN SPAZIO_NON_DISPONIBILE
THEN RAISE_APPLICATION_ERROR(-20595,'NON SI TROVA SPAZIO NELLE
CRIOCELLE!');

END;

```

Viste

La creazione delle viste facilitano la progettazione di alcune procedure e/o per rendere possibile alcuni compiti all'applicazione web collegata alla base di dati.

Questa vista permette alla Segreteria di vedere le prenotazioni ancora disponibili nella settimana corrente.

```

CREATE OR REPLACE VIEW PRENOTAZIONI_DISPONIBILI AS
SELECT TO_CHAR(DATA_ORA, 'DD/MM/YYYY HH24:MI') AS ORARIO, NUM_SALA
AS SALA FROM PRENOTAZIONE
WHERE CF_CLI IS NULL AND CF_DON IS NULL AND DATA_ORA >= SYSDATE AND
DATA_ORA < (
SELECT NEXT_DAY(TRUNC(SYSDATE, 'DAY'), 'DOMENICA') FROM DUAL);

```

La vista seguente permette alla Segreteria di visualizzare i donatori con ancora fiale associate disponibili per l'inseminazione o per l'acquisto.

```

CREATE OR REPLACE VIEW DONATORI_DISPONIBILI AS
SELECT D.NOME_DON AS NOME, D.COGNOME_DON AS COGNOME
FROM DONATORE D JOIN FIALA_SEME FS ON FS.CF_DON=D.CF_DON
WHERE FS.SANO='SI' AND FS.FATTURA IS NULL AND FS.COD_INS IS NULL;

```

Le due seguenti viste, permettono al cliente di filtrare la lista dei donatori secondo le loro preferenze. Infatti i clienti possono scegliere, rispetto alle caratteristiche fisiche, il seme desiderato. Poiché alcuni donatori scelgono legalmente di non essere rintracciabili, ci sono due viste diverse per lo stesso scopo.

```

--  VISUALIZZARE TUTTI I DONATORI CHE SONO LEGALMENTE
RINTRACCIABILI

```

```

CREATE OR REPLACE VIEW VISITA_DONATORI AS
SELECT NOME_DON AS NOME, COGNOME_DON AS COGNOME, G_SANGUIGNO,
OCCHI, PESO, CAPELLI, ETNIA, ALTEZZA, TELEFONO_DON AS CONTATTO
FROM DONATORE WHERE ANONIMATO='NO';

```

/

```

--  VISUALIZZARE I DONATORI CHE NON SONO LEGALMENTE RINTRACCIABILI

```

```

CREATE OR REPLACE VIEW VISITA_DONATORI_ANONIMI AS
SELECT NOME_DON AS NOME, COGNOME_DON AS COGNOME, G_SANGUIGNO,
OCCHI, PESO, CAPELLI, ETNIA, ALTEZZA FROM DONATORE WHERE
ANONIMATO='SI';

```

La seguente vista è per i dottori. Attraverso un filtro, usato da applicazioni esterne, il dottore può vedere le proprie prenotazioni giornaliere.

```

CREATE OR REPLACE VIEW PREN_GIORNALIERE AS
SELECT          DISTINCT          D.NOME_DOT,          D.COGNOME_DOT,
TO_CHAR(L.DATA_ORA, 'HH24:MI')    AS    ORA,          S.NUM_SALA    FROM
((PRENOTAZIONE P JOIN SALA S ON S.NUM_SALA=P.NUM_SALA) JOIN
LAVORA_IN L ON L.NUM_SALA=S.NUM_SALA) JOIN DOTTORE D ON
L.CF_DOT=D.CF_DOT

```

```
WHERE          TO_DATE(L.DATA_ORA, 'DD/MM/YYYY')=TO_DATE(SYSDATE, 'DD/MM/YYYY');
```

Le seguenti quattro viste sono utilizzate per vedere i risultati degli esami. Grazie all'applicazione web, saranno filtrati rispetto al codice della cartella clinica.

```
-- VISUALIZZARE I RISULTATI DELL'ESAME GENETICO
CREATE OR REPLACE VIEW ESAME_GENETICO AS
SELECT C_CLINICA, IRT, ESONE7, CARIOTIPO FROM GENETICO;
```

```
-- VISUALIZZARE I RISULTATI DELL'ESAME DEL SANGUE
CREATE OR REPLACE VIEW ESAME_SANGUE AS
SELECT C_CLINICA, TPHA, VLDRL, LINFOCITICD4 FROM SANGUE;
```

```
-- VISUALIZZARE I RISULTATI DELL'ESAME SPERMIOGRAMMA
CREATE OR REPLACE VIEW ESAME_SPERMIOGRAMMA AS
SELECT C_CLINICA, CNTZ, VOL, MOT, PH FROM SPERMIOGRAMMA;
```

```
-- VISUALIZZARE I RISULTATI DELL'ESAME FERTILITA
CREATE OR REPLACE VIEW ESAME_FERTILITA AS
SELECT C_CLINICA, PROLATTINA, FSH FROM FERTILITA;
```

L'ultima vista proposta permette di visualizzare le prenotazioni effettuate.

```
CREATE OR REPLACE VIEW VISUALIZZA_PRENOTAZIONI AS
SELECT ID_PREN, DATA_ORA, CF_DON, CF_CLI, NUM_SALA FROM
PRENOTAZIONE WHERE DATA_ORA>=SYSDATE AND (CF_DON IS NOT NULL OR
CF_CLI IS NOT NULL);
```

Vincoli di Integrità

In questa sezione ci occuperemo dei vincoli di integrità, facendo riferimento sia a quelli statici, ovvero quelli già presenti nel codice della creazione delle tabelle, e quelli dinamici, realizzati mediante Trigger in PL/SQL.

Vincoli di Integrità statici

La lista seguente si occuperà dei vincoli di integrità statici, non tenendo conto delle chiavi primarie e chiavi esterne, ponendo l'attenzione sui campi univoci e gli specifici domini per i valori.

- Il sesso dei clienti e del dottore devono assolutamente rispondere a 'M' per maschio e 'F' per femmina.
- Il donatore, avendo molti campi che descrivono caratteristiche fisiche, ha una serie di vincoli. Colore degli occhi: 'blu', 'azzurri', 'verdi', 'marroni', 'neri', 'grigi'. Gruppo Sanguigno: '0-', '0+', 'A-', 'A+', 'B-', 'B+', 'AB-', 'AB+'. Il peso del donatore deve rientrare nel range di '30-150' (è sottintesa la misura in chilogrammi). L'altezza del donatore deve anch'essa rientrare in un range di '120-220' (è sottintesa la misura in centimetri). In più il campo "anonimato" risponde all'esigenza dei donatori di scegliere se essere rintracciabile o meno: può avere valori 'Sì', 'No'.
- Nella tabella Pagamento, il tipo di pagamento può essere soltanto 'inseminazione' e 'fiala', che sono rispettivamente il pagamento per l'inseminazione e per il solo acquisto della fiala.
- Le tabelle Fiala_Sangue e Fiala_Seme hanno il campo "sano" che può avere solo due valori "sì" e "no" che rispondono all'esigenza di dover facilmente riconoscere se un donatore è compatibile alla donazione o se una cliente può fare l'inseminazione.
- La tabella Inseminazione ha un campo chiamato "Tipo" che descrive i due tipi di inseminazione attualmente possibili in medicina, il tipo 'ICI' e 'IUI'.
- La tabella Criocella ha il campo "tipo" perché ogni criocella può conservare solo un tipo di fiale perciò il dominio di quel campo è composto da due valori: "Sangue" e "Seme".
- Nella tabella Genetico il campo "Esone7" ne indica la presenza con valori (booleani) '0' e '1'.
- La motilità nella tabella Spermioγραμμα non può avere valori che eccedono il range '0-100'.
- Nella tabella Cartella_Clinica, sia il campo "cf_don" che "cf_cli", che rispettivamente corrispondono al codice fiscale di donatore e cliente, sono Unique, in quanto ogni donatore o cliente può avere una e una sola cartella clinica a loro associata.
- Il campo "telefono" in Donatore, Cliente e Dottore è anch'esso Unique.
- Il codice dei medici nella tabella Dottore è anch'esso Unique.

Vincoli di Integrità dinamici

Ora analizzeremo i vincoli di integrità dinamici corrispondenti alle regole di business.

- Il donatore non può essere minorenne né superare l'età di 100 anni.
- Un dottore non può avere meno di 25 anni né che ne abbia più di 70, in quanto, in teoria dovrebbe essere in pensione, quindi non può più esercitare.
- Il cliente non può essere minorenne né essere in età di pensione.
- La data di chiusura della cartella clinica non può superare la data di apertura della stessa.
- Per questioni scientifiche non si possono sottoporre ad un'inseminazione: le clienti affette da una malattia infettiva o con un grado di fertilità troppo basso; le clienti che non hanno un esame del sangue per malattie infettive eseguito negli ultimi 30 giorni.
- Non si possono prenotare inseminazioni per clienti e donatori di sesso maschile; un cliente non può donare a meno che non sia registrato come donatore; ai clienti di sesso maschile non è concesso fare nessun esame offerto dalla clinica.
- All'inserimento di una fiala del sangue e del seme bisogna calcolare la salute della fiala, facendo riferimento agli esami a loro associati.
- Bisogna assicurarsi che gli esami (fertilità, sangue, genetico e spermogramma) siano stati effettivamente prenotati.
- Non si può chiudere una cartella clinica che non esiste o è stata già chiusa o a cui sono associati servizi prenotati oltre la data di chiusura.

Vincoli di Integrità dinamici - Procedure

Alcuni dei vincoli di integrità sono stati già presentati nella sezione della creazione delle Procedure a causa dei problemi di mutating table all'interno dei trigger. Di seguito, citeremo soltanto i vincoli e non il codice stesso. Aggiungeremo, però, ad ogni vincolo, la procedura a cui fa riferimento.

- I dottori devono sempre supervisionare un servizio prenotato. Non ci può essere più di una prenotazione per una stessa sala in un determinato orario. Una cliente senza esami del sangue o che abbia un'esame del sangue con risultato sangue infetto non può sottoporsi ad un'inseminazione. Una persona che si sottopone ad un'esame deve essere obbligatoriamente un donatore o una cliente. Procedure referenti: [Inserisci_Prenotazione](#) e [Inserisci_Pag_Ins](#).
- Non si possono acquistare fiale di seme non sane. Procedura referente: [Acquisto_Fiala](#).

Vincoli di Integrità dinamici - Trigger

Di seguito presenteremo il codice in SQL dei vincoli dinamici precedentemente citati.

```

-- Il donatore non dev'essere minorenne né troppo maggiorenne
[DONATORE]
CREATE OR REPLACE TRIGGER ETA_DONATORE
BEFORE INSERT ON DONATORE
FOR EACH ROW
DECLARE
MINORENNE_DON EXCEPTION;
TOO_OLD EXCEPTION;

BEGIN
IF MONTHS_BETWEEN(SYSDATE, :NEW.DN_DON) <= (12*18)
THEN
    RAISE MINORENNE_DON;
END IF;

IF MONTHS_BETWEEN(SYSDATE, :NEW.DN_DON) >= (12*100)
THEN
    RAISE TOO_OLD;
END IF;

EXCEPTION
WHEN MINORENNE_DON THEN RAISE_APPLICATION_ERROR(-20224, 'Il
donatore non deve essere minorenne');
WHEN TOO_OLD THEN RAISE_APPLICATION_ERROR(-20227, 'Il donatore non
puo essere dracula');
END;

-- Il dottore dev'avere almeno 25 anni ma non deve superare gli
anni della pensione DOTTORE]
CREATE OR REPLACE TRIGGER ETA_DOTTORE
BEFORE INSERT ON DOTTORE
FOR EACH ROW
DECLARE
MINORENNE_DOT EXCEPTION;
PENSIONE EXCEPTION;
BEGIN
IF MONTHS_BETWEEN(SYSDATE, :NEW.DN_DOT) <= (12*25)
THEN
    RAISE MINORENNE_DOT;
END IF;
IF MONTHS_BETWEEN(SYSDATE, :NEW.DN_DOT) >= (12*70)
THEN RAISE PENSIONE;
END IF;
EXCEPTION
WHEN MINORENNE_DOT THEN RAISE_APPLICATION_ERROR(-20225, 'Il dottore
non puo essere un bimbo prodigio');
WHEN PENSIONE THEN RAISE_APPLICATION_ERROR(-20229, 'Il dottore ha
la pensione non puo esercitare');
END;

```

```

-- Il cliente non dev'essere minorenne né troppo anziano [CLIENTE]
CREATE OR REPLACE TRIGGER ETA_CLIENTE
BEFORE INSERT ON CLIENTE
FOR EACH ROW
DECLARE
MINORENNE_CLI EXCEPTION;
TOO_OLD EXCEPTION;
BEGIN
IF MONTHS_BETWEEN(SYSDATE, :NEW.DN_CLI) <= (12*18)
THEN
    RAISE MINORENNE_CLI;
END IF;
IF MONTHS_BETWEEN(SYSDATE, :NEW.DN_CLI) >= (12*70)
THEN
    RAISE TOO_OLD;
END IF;
EXCEPTION
WHEN MINORENNE_CLI THEN RAISE_APPLICATION_ERROR(-20226, 'Il cliente
non puo essere minorenne');
WHEN TOO_OLD THEN RAISE_APPLICATION_ERROR(-20230, 'Il cliente e
troppo vecchio');
END;

-- La data di chiusura della cartella clinica non deve essere piu
giovane della data di apertura della stessa [CARTELLA CLINICA]
CREATE OR REPLACE TRIGGER TRIGGER_C_CLINICA
BEFORE INSERT OR UPDATE ON CARTELLA_CLINICA
FOR EACH ROW
DECLARE
PARADOX_CLINIC EXCEPTION;
BEGIN
IF (:NEW.DATA_CHIUSURA < :NEW.DATA_CREAZIONE
OR :NEW.DATA_CHIUSURA < :OLD.DATA_CREAZIONE)
THEN
    RAISE PARADOX_CLINIC;
END IF;
EXCEPTION
WHEN PARADOX_CLINIC THEN RAISE_APPLICATION_ERROR(-20221, 'Non si
puo chiudere la cartella prima di aprirla');
END;

CREATE OR REPLACE TRIGGER CONTROLLO_INSEMINAZIONE
BEFORE INSERT OR UPDATE ON INSEMINAZIONE
FOR EACH ROW
DECLARE
COUNT_FIALE NUMBER;

```

```

CF_FLAG CHAR(16);
DATA_INSEMINAZIONE DATE;
MANCA_FIALE EXCEPTION;
MALATTIA_TROVATA EXCEPTION;

BEGIN

COUNT_FIALE:=0;

SELECT DATA_ORA, CF_CLI INTO DATA_INSEMINAZIONE, CF_FLAG
FROM PRENOTAZIONE
WHERE ID_PREN = :NEW.ID_PREN;

SELECT COUNT(*) INTO COUNT_FIALE --CONTROLLA SE HA AVUTO MALATTIE
INFETTIVE IN PRECEDENZA
FROM FIALA_SANGUE
WHERE CF_CLI=CF_FLAG AND SANO = 'NO';

IF COUNT_FIALE>0
THEN RAISE MALATTIA_TROVATA;
END IF;

SELECT COUNT(*) INTO COUNT_FIALE --CONTROLLA SE ESISTONO DELLE
FIALE SANGUE VALIDE ENTRO TRENTA GIORNI
FROM FIALA_SANGUE
WHERE CF_CLI=CF_FLAG AND TO_DATE(DATA_SANGUE, 'DD/MM/YYYY')
BETWEEN TO_DATE(DATA_INSEMINAZIONE, 'DD/MM/YYYY')-30 AND
TO_DATE(DATA_INSEMINAZIONE, 'DD/MM/YYYY');

IF COUNT_FIALE<=0
THEN RAISE MANCA_FIALE;
END IF;

EXCEPTION

WHEN MALATTIA_TROVATA
THEN RAISE_APPLICATION_ERROR(-2391, 'TROVATA UNA MALATTIA VENEREA,
IMPOSSIBILE ESEGUIRE UNA INSEMINAZIONE');

WHEN MANCA_FIALE
THEN RAISE_APPLICATION_ERROR(-2392, 'NON CI SONO FIALE DI SANGUE
NEGLI ULTIMI 30 GIORNI, IMPOSSIBILE ESEGUIRE UNA INSEMINAZIONE');

END;
CREATE OR REPLACE TRIGGER ID_PRENOTAZIONE
BEFORE INSERT OR UPDATE ON PRENOTAZIONE
FOR EACH ROW
DECLARE
NO_MPREG EXCEPTION;

```

```

BE_DONATORE EXCEPTION;
NOT_GRANTED EXCEPTION;

SESSO_FLAG CHAR(1);
FLAG NUMBER;

BEGIN
IF INSERTING
THEN
SELECT ID_PREN_S.NEXTVAL INTO :NEW.ID_PREN
FROM DUAL;
END IF;
FLAG:= 0;
IF (:NEW.TIPO_PREN='INSEMINAZIONE' OR :NEW.TIPO_PREN='FERTILITA')
THEN

    SELECT COUNT(*) INTO FLAG
    FROM DONATORE
    WHERE :NEW.CF_DON=CF_DON;

    IF FLAG>0
    THEN RAISE NO_MPREG;
    END IF;

    SELECT COUNT(*) INTO FLAG
    FROM CLIENTE
    WHERE :NEW.CF_CLI=CF_CLI AND (SESSO_CLI='M');

    IF FLAG>0
    THEN RAISE NO_MPREG;
    END IF;

END IF;

IF (:NEW.TIPO_PREN='DONAZIONE' AND (:NEW.CF_CLI IS NOT NULL))
THEN RAISE BE_DONATORE;
END IF;

IF(:NEW.CF_CLI IS NOT NULL)
THEN SELECT SESSO_CLI INTO SESSO_FLAG FROM CLIENTE WHERE
CF_CLI=:NEW.CF_CLI;
IF (SESSO_FLAG='M')
THEN RAISE NOT_GRANTED;
END IF;
END IF;

EXCEPTION

WHEN NO_MPREG

```

```
THEN RAISE_APPLICATION_ERROR(-20025, 'LA SCIENZA NON E ARRIVATA
COSI LONTANO');

WHEN BE_DONATORE
THEN RAISE_APPLICATION_ERROR(-20026, 'DEVE PRIMA REGISTRARSI COME
DONATORE');

WHEN NOT_GRANTED
THEN RAISE_APPLICATION_ERROR (-20029, 'AL CLIENTE NON E CONCESSO
FARE QUESTI ESAMI');

END;
```

```
CREATE OR REPLACE TRIGGER SANGUE_SANO
BEFORE INSERT OR UPDATE ON FIALA_SANGUE
FOR EACH ROW
DECLARE
LINF_FLAG NUMBER;
TPHA_FLAG NUMBER;
VLDRL_FLAG NUMBER;

PROL_FLAG NUMBER;
FSH_FLAG NUMBER;

BEGIN

:NEW.SANO:='SI';
IF :NEW.CF_DON IS NOT NULL -- INIZIO
THEN

    SELECT LINFOCITICD4, TPHA, VLDRL INTO LINF_FLAG, TPHA_FLAG,
VLDRL_FLAG
    FROM SANGUE
    WHERE COD_E_SANGUE=:NEW.COD_E_SANGUE;

    IF (LINF_FLAG<500 OR LINF_FLAG>1100 OR TPHA_FLAG<4 OR
VLDRL_FLAG<1280)
    THEN
        :NEW.SANO:='NO';
    END IF;

ELSE IF (:NEW.COD_E_SANGUE IS NOT NULL)
    THEN
        SELECT LINFOCITICD4, TPHA, VLDRL INTO LINF_FLAG, TPHA_FLAG,
VLDRL_FLAG
        FROM SANGUE
        WHERE COD_E_SANGUE=:NEW.COD_E_SANGUE;

    ELSE IF (:NEW.COD_E_FERTILITA IS NOT NULL)
```

```

    THEN
    SELECT PROLATTINA, FSH INTO PROL_FLAG, FSH_FLAG
    FROM FERTILITA
    WHERE COD_E_FERTILITA=:NEW.COD_E_FERTILITA;
    END IF;
    END IF;

    IF ((LINF_FLAG<500 OR LINF_FLAG>1100 OR TPHA_FLAG<4 OR
VLDRL_FLAG<1280) OR (FSH_FLAG<1 OR FSH_FLAG>10 OR PROL_FLAG<2 OR
PROL_FLAG>28))
    THEN
        :NEW.SANO:='NO';
    END IF;

END IF; -- FINE
END;

CREATE OR REPLACE TRIGGER SEME_VALIDO
BEFORE INSERT OR UPDATE ON FIALA_SEME
FOR EACH ROW
DECLARE
DO_PRELIEVO EXCEPTION;
CARIOTIPO_FLAG CHAR(4);
ESONE7_FLAG NUMBER;
IRT_FLAG NUMBER;
CNTZ_FLAG NUMBER;
VOL_FLAG NUMBER;
MOT_FLAG NUMBER;
PH_FLAG NUMBER;
FLAG NUMBER;

BEGIN
IF INSERTING
THEN
:NEW.SANO:='SI';

SELECT CARIOTIPO, IRT, ESONE7 INTO CARIOTIPO_FLAG, IRT_FLAG,
ESONE7_FLAG
FROM GENETICO
WHERE COD_E_GENETICO=:NEW.COD_E_GENETICO;

IF (CARIOTIPO_FLAG<>'46XY' OR ESONE7_FLAG<>1 OR (IRT_FLAG<5 OR
IRT_FLAG>50))
THEN
:NEW.SANO:='NO';
END IF;
--VALORI SPERMIOGRAMMA
SELECT CNTZ, VOL, MOT, PH INTO CNTZ_FLAG, VOL_FLAG, MOT_FLAG,
PH_FLAG

```

```

FROM SPERMIOGRAMMA
WHERE COD_E_SPERMIOGRAMMA=:NEW.COD_E_SPERMIOGRAMMA;

IF (CNTZ_FLAG<15 OR MOT_FLAG<40 OR PH_FLAG<7 OR VOL_FLAG<2)
THEN :NEW.SANO:='NO';
END IF;

--VALORI SUL SANGUE
FLAG:=0;

SELECT COUNT(*) INTO FLAG --CONTROLLA SE HA AVUTO MALATTIE
INFETTIVE IN PRECEDENZA
FROM FIALA_SANGUE
WHERE CF_DON=:NEW.CF_DON AND SANO = 'NO';

IF FLAG>0
THEN :NEW.SANO:='NO';
END IF;

SELECT COUNT(*) INTO FLAG --CONTROLLA SE ESISTONO DELLE FIALE
SANGUE VALIDE ENTRO TRENTA GIORNI
FROM FIALA_SANGUE
WHERE CF_DON=:NEW.CF_DON AND TO_DATE(DATA_SANGUE,'DD/MM/YYYY')
BETWEEN TO_DATE(:NEW.DATA_SEME,'DD/MM/YYYY')-30 AND
TO_DATE(:NEW.DATA_SEME,'DD/MM/YYYY');

IF FLAG<=0
THEN RAISE DO_PRELIEVO;
END IF;

END IF;

EXCEPTION
WHEN DO_PRELIEVO
THEN RAISE_APPLICATION_ERROR(-20227,'BISOGNA FARE L ESAME DEL
SANGUE PRIMA DI DONARE');
END;

CREATE OR REPLACE TRIGGER PREN_SANG
BEFORE INSERT OR UPDATE ON SANGUE
FOR EACH ROW
DECLARE
NO_MATCH EXCEPTION;
TIPO VARCHAR(20);

BEGIN

```

```

SELECT TIPO_PREN INTO TIPO
FROM PRENOTAZIONE
WHERE ID_PREN=:NEW.ID_PREN;

IF (TIPO<>'SANGUE')
THEN
RAISE NO_MATCH;
END IF;

EXCEPTION
WHEN NO_MATCH
THEN RAISE_APPLICATION_ERROR(-20230, 'LA PRENOTAZIONE
CORRISPONDENTE A QUESTO ESAME NON DICHIARA LO STESSO TIPO');

END;

/

SHO ERR;

CREATE OR REPLACE TRIGGER PREN_GENE
BEFORE INSERT OR UPDATE ON GENETICO
FOR EACH ROW
DECLARE
NO_MATCH EXCEPTION;
TIPO VARCHAR(20);

BEGIN

SELECT TIPO_PREN INTO TIPO
FROM PRENOTAZIONE
WHERE ID_PREN=:NEW.ID_PREN;

IF (TIPO<>'GENETICO')
THEN
RAISE NO_MATCH;
END IF;

EXCEPTION
WHEN NO_MATCH
THEN RAISE_APPLICATION_ERROR(-20231, 'LA PRENOTAZIONE
CORRISPONDENTE A QUESTO ESAME NON DICHIARA LO STESSO TIPO');

END;

/

```

SHO ERR;

CREATE OR REPLACE TRIGGER PREN_SPERM
BEFORE INSERT OR UPDATE ON SPERMIOGRAMMA
FOR EACH ROW

DECLARE

NO_MATCH **EXCEPTION**;

TIPO **VARCHAR**(20);

BEGIN

SELECT TIPO_PREN **INTO** TIPO
FROM PRENOTAZIONE
WHERE ID_PREN=:NEW.ID_PREN;

IF(TIPO<>'SPERMIOGRAMMA')

THEN

RAISE NO_MATCH;

END IF;

EXCEPTION

WHEN NO_MATCH

THEN RAISE_APPLICATION_ERROR(-20232,'LA PRENOTAZIONE
CORRISPONDENTE A QUESTO ESAME NON DICHIARA LO STESSO TIPO');

END;

/

SHO ERR;

CREATE OR REPLACE TRIGGER PREN_FERT
BEFORE INSERT OR UPDATE ON FERTILITA
FOR EACH ROW

DECLARE

NO_MATCH **EXCEPTION**;

TIPO **VARCHAR**(20);

BEGIN

SELECT TIPO_PREN **INTO** TIPO
FROM PRENOTAZIONE
WHERE ID_PREN=:NEW.ID_PREN;

IF(TIPO<>'FERTILITA')

THEN

RAISE NO_MATCH;

END IF;

EXCEPTION

```
WHEN NO_MATCH
THEN RAISE_APPLICATION_ERROR(-20233, 'LA PRENOTAZIONE
CORRISPONDENTE A QUESTO ESAME NON DICHIARA LO STESSO TIPO');
```

```
END;
```

```
-- CONTROLLO CHE LA DATA DI CHIUSURA NON SIA MINORE DELLA DATA DI
APERTURA DELLA STESSA
```

```
CREATE OR REPLACE TRIGGER CHIUSURA_CARTELLA
BEFORE UPDATE ON CARTELLA_CLINICA
FOR EACH ROW
DECLARE
COUNTER NUMBER;
```

```
NOT_GRANTED EXCEPTION;
BACK_IN_TIME EXCEPTION;
```

```
BEGIN
```

```
IF (:NEW.DATA_CHIUSURA IS NOT NULL)
```

```
THEN
```

```
    IF (:NEW.DATA_CHIUSURA < :OLD.DATA_CREAZIONE)
```

```
    THEN RAISE BACK_IN_TIME;
```

```
    END IF;
```

```
END IF;
```

```
IF (:OLD.DATA_CHIUSURA IS NOT NULL AND :NEW.DATA_CHIUSURA IS NOT
NULL) -- controlla che la nuova data non sia minore della data di
apertura e che non siano previsti
-- esami dopo la data della nuova chiusura
```

```
THEN
```

```
    IF (:NEW.DATA_CHIUSURA < :OLD.DATA_CREAZIONE)
```

```
    THEN RAISE BACK_IN_TIME;
```

```
    END IF;
```

```
COUNTER:=0;
```

```
IF(:OLD.CF_CLI IS NOT NULL) -- se la cartella clinica
appartiene al cliente
```

```
THEN
```

```
SELECT COUNT(*) INTO COUNTER
```

```
FROM PRENOTAZIONE
```

```
WHERE CF_CLI=:OLD.CF_CLI AND DATA_ORA>:NEW.DATA_CHIUSURA;
```

```
IF (COUNTER>0)
```

```
THEN RAISE NOT_GRANTED;
```

```
END IF;
```

```
ELSE
SELECT COUNT(*) INTO COUNTER
FROM PRENOTAZIONE
WHERE CF_DON=:OLD.CF_DON AND DATA_ORA>:NEW.DATA_CHIUSURA;

    IF (COUNTER>0)
    THEN RAISE NOT_GRANTED;
    END IF;
END IF;
END IF;
EXCEPTION
WHEN BACK_IN_TIME
THEN RAISE_APPLICATION_ERROR(-20289, 'LA CARTELLA CLINICA NON HA
UNA DELOREAN');

WHEN NOT_GRANTED
THEN RAISE_APPLICATION_ERROR(-20290, 'NON SI PUO CHIUDERE LA
CARTELLA PERCHE SONO PREVISTI SERVIZI');

END;
```

Operazioni DML

In questa sezione mostriamo le operazioni di inserimento nel database. Si fa notare che il popolamento è completamente fittizio e risponde ad una gestione tipo in medio regime nell'arco di tempo di due settimane.

```
INSERT INTO TURNO (COD_TURNO, INIZIO_TURNO, FINE_TURNO)
VALUES ('1', TO_DATE('8:30', 'HH24:MI'), TO_DATE('14:30',
'HH24:MI'));
INSERT INTO TURNO (COD_TURNO, INIZIO_TURNO, FINE_TURNO)
VALUES ('2', TO_DATE('14:30', 'HH24:MI'), TO_DATE('20:30',
'HH24:MI'));
```

```
--Per popolare i Donatori ed i Clienti, uso la procedura apposita
--Cio' mi creerà anche le corrispettive cartelle cliniche.
--in totale saranno 10 Donatori e 10 clienti.
```

```
EXECUTE INSERISCI_DONATORE('FRNGVN62C01D811U', 'Gavino',
'Fiorentini', '1-MAR-1962', 'Frosolone', 'Via Firenze', 'A+',
'MARRONI', 63, 'NERI', 'NO', 'CAUCASICO', 175, '3339283765', 'FI')
EXECUTE INSERISCI_DONATORE('PSNNG75S23Z110U', 'Angelo', 'Pisano',
'23-NOV-1975', 'Carcassonne', 'Rue Marie De Mèdicis', 'A+',
'VERDI', 95, 'BIONDI', 'NO', 'CAUCASICO', 176, '0669138477', 'FR')
EXECUTE INSERISCI_DONATORE('NDRVDO75T30H151V', 'Ovidio',
'Endrizzi', '30-DIC-1975', 'Raccuja', 'Via Nuova Del Campo',
'A-', 'AZZURRI', 68, 'NERI', 'NO', 'ASIATICO', 171, '3384911994',
'RA')
EXECUTE INSERISCI_DONATORE('CLBDLM70M19C573R', 'Delmo',
'Calabrese', '19-AGO-1970', 'Cesena', 'Via Foria', '0-',
'MARRONI', 74, 'CASTANI', 'SI', 'CAUCASICO', 189, '3920699543',
'CE')
EXECUTE INSERISCI_DONATORE('MRTLRT94S21E054P', 'Alberto',
'Moretti', '21-NOV-1994', 'Giugliano', 'Via San Vito', 'B+',
'MARRONI', 70, 'NERI', 'NO', 'CAUCASICO', 170, '0817467587', 'NA')
EXECUTE INSERISCI_DONATORE('MARFLL80S38F391U', 'Mario',
'Fratelli', '2-GIU-1980', 'Casavatore', 'Via Campanariello',
'A-', 'MARRONI', 85, 'CASTANI', 'NO', 'MEDIO ORIENTALE', 160,
'0816545690', 'NA')
EXECUTE INSERISCI_DONATORE('SPSFLC72E18A149T', 'Felice',
'Esposito', '18-MAG-1972', 'Alberobello', 'Via Tenente Simonelli',
'B+', 'AZZURRI', 100, 'BIONDI', 'SI', 'CAUCASICO', 180,
'0789132328', 'AB')
EXECUTE INSERISCI_DONATORE('PGLPTL46P07G702M', 'Pantaleone',
'Pugliesi', '07-SET-1946', 'Pisa', 'Via Del Bosco', '0+',
'MARRONI', 90, 'NERI', 'NO', 'AFRICANO', 190, '3234919311', 'FI')
```

```
EXECUTE INSERISCI_DONATORE('DNIMTT96E22A462R', 'Matteo', 'Diana',
'22-MAG-1996', 'Ascoli Piceno', 'Via Partenza', 'B-', 'VERDI', 67,
'BIONDI', 'NO', 'CAUCASICO', 160, '3485900058', 'SA')
EXECUTE INSERISCI_DONATORE('HNDBSS66S16H555Z', 'Basso',
'Hunderhill', '16-NOV-1966', 'Roreto', 'Piazza Guglielmo Pepe',
'0+', 'MARRONI', 80, 'NERI', 'NO', 'ISPANICO', 152, '5556595423',
'AO')
```

--Inserisco 9 Femmine ed 1 Maschio, il quale potrà solo comprare le Fiale.

```
EXECUTE INSERISCI_CLIENTE ('TRNFNC54B53L379X', 'Francesca',
'Trentino', '13-FEB-1954', 'F', 'Trentino', 'Via Callicratide',
'3949193955', 'FI')
EXECUTE INSERISCI_CLIENTE ('GRCMLN42A66B598L', 'Marilena',
'Greece', '26-GEN-1982', 'F', 'Caneva', 'Corso Vittorio Emanuele',
'2524717119', 'PN')
EXECUTE INSERISCI_CLIENTE ('CLBTBT70C52E084E', 'Tabita',
'Calabrese', '12-MAR-1970', 'F', 'Goni', 'Via del Viminale',
'4630758036', 'CA')
EXECUTE INSERISCI_CLIENTE ('BRSLDA81H41G494R', 'Alida', 'Barese',
'1-GIU-1981', 'F', 'Pesconuovo', 'Via Domenico Morelli',
'3425848743', 'PE')
EXECUTE INSERISCI_CLIENTE ('DFNPSN66C57L723B', 'Dafne', 'Pisani',
'17-MAR-1966', 'F', 'Velo', 'Via del Carmine', '3862222143', 'VI')
EXECUTE INSERISCI_CLIENTE ('GMNPRZ81C42F113X', 'Germana',
'Pirozzi', '3-SET-1981', 'F', 'Braies', 'Via Alessandro Farnese',
'3292283276', 'BZ')
EXECUTE INSERISCI_CLIENTE ('LCLRMN88S59G6780', 'Lucilla',
'Romano', '19-NOV-1988', 'F', 'Pino', 'Via Spalato', '3908744662',
'VA')
EXECUTE INSERISCI_CLIENTE ('SLLLRT59E02F839T', 'Alberto', 'Sollo',
'02-MAG-1959', 'M', 'Napoli', 'Via Roma', '0815959091', 'NA')
EXECUTE INSERISCI_CLIENTE ('LSSMCN75A47A783K', 'Alessandra',
'Mancini', '7-GEN-1975', 'F', 'Benevento', 'Viale dei Giardini',
'0881931293', 'BV')
EXECUTE INSERISCI_CLIENTE ('SNCMNO80T42E335P', 'Moana', 'Sinceri',
'2-DIC-1980', 'F', 'Isernia', 'Via Poco', '800900313', 'IS')
```

--Inserisco 7 Dottori

```
INSERT INTO DOTTORE (CF_DOT, NOME_DOT, COGNOME_DOT, DN_DOT,
SESSO_DOT, CITTA_DOT, VIA_DOT, CDM, TELEFONO_DOT, PROV_DOT)
VALUES ('SPSGNR60T25F839S', 'Gennaro', 'Esposito', '25-DIC-1960',
'M', 'Mugnano', 'Via San Pietro', '0135', '3389209200', 'NA');
INSERT INTO DOTTORE (CF_DOT, NOME_DOT, COGNOME_DOT, DN_DOT,
SESSO_DOT, CITTA_DOT, VIA_DOT, CDM, TELEFONO_DOT, PROV_DOT)
VALUES ('FRRRMD63R53I455T', 'Armida', 'Ferrari', '13-OTT-1963',
'F', 'Sassinoro', 'Via Giacinto Gigante', '2336', '3519546049',
'MO');
```

```

INSERT INTO DOTTORE (CF_DOT, NOME_DOT, COGNOME_DOT , DN_DOT,
SESSO_DOT, CITTA_DOT , VIA_DOT , CDM,TELEFONO_DOT, PROV_DOT)
VALUES ( 'BNOPRD87D16C525Z', 'Paride', 'Boni', '16-APR-1987', 'F',
'Cerreto Sannita', 'Via Piccinni', '8991', '3480514421', 'BN');
INSERT INTO DOTTORE (CF_DOT, NOME_DOT, COGNOME_DOT , DN_DOT,
SESSO_DOT, CITTA_DOT , VIA_DOT , CDM,TELEFONO_DOT, PROV_DOT)
VALUES ( 'LSKTNI92H64H757Q', 'Tiina', 'Leskinen', '24-
GEN-1992', 'F', 'SANARICA', 'Via Antonio Beccadelli', '3402',
'3745708542', 'LE');
INSERT INTO DOTTORE (CF_DOT, NOME_DOT, COGNOME_DOT , DN_DOT,
SESSO_DOT, CITTA_DOT , VIA_DOT , CDM,TELEFONO_DOT, PROV_DOT)
VALUES ( 'GRGHSO59E15F839R', 'House', 'Gregory', '15-
MAG-1959', 'M', 'NAPOLI', 'Via Bastone', '1000', '3568984732', 'NA');
INSERT INTO DOTTORE (CF_DOT, NOME_DOT, COGNOME_DOT , DN_DOT,
SESSO_DOT, CITTA_DOT , VIA_DOT , CDM,TELEFONO_DOT, PROV_DOT)
VALUES ( 'JKYHYD87C11B157O', 'Hyde', 'Jekyll', '11-
MAR-1987', 'M', 'MILANO', 'Via San Cosmo', '1886', '3739861286',
'MI');
INSERT INTO DOTTORE (CF_DOT, NOME_DOT, COGNOME_DOT , DN_DOT,
SESSO_DOT, CITTA_DOT , VIA_DOT , CDM,TELEFONO_DOT, PROV_DOT)
VALUES ( 'LNGBLB54M62H526R', 'Belba', 'Longhole', '22-
AGO-1954', 'F', 'RONCADELLO', 'Via Partenope',
'2222', '3574385189', 'FO');

```

-- Aggiungo 5 sale. L'unico che ha il potere di aggiungere altre sale è l'amministratore.

```

INSERT INTO SALA (NUM_SALA, ALA, PIANO)
VALUES (1, 'A', '1');
INSERT INTO SALA (NUM_SALA, ALA, PIANO)
VALUES (2, 'B', '1');
INSERT INTO SALA (NUM_SALA, ALA, PIANO)
VALUES (3, 'A', '2');
INSERT INTO SALA (NUM_SALA, ALA, PIANO)
VALUES (4, 'B', '2');
INSERT INTO SALA (NUM_SALA, ALA, PIANO)
VALUES (5, 'B', '2');

```

-- L'inserimento delle criocelle è esclusivo del proprietario e dell'admin

```

EXECUTE INSERISCI_CRIOCELLA(20, 'SEME')
EXECUTE INSERISCI_CRIOCELLA(40, 'SEME')
EXECUTE INSERISCI_CRIOCELLA(60, 'SEME')
EXECUTE INSERISCI_CRIOCELLA(80, 'SEME')
EXECUTE INSERISCI_CRIOCELLA(20, 'SANGUE')
EXECUTE INSERISCI_CRIOCELLA(40, 'SANGUE')
EXECUTE INSERISCI_CRIOCELLA(60, 'SANGUE')
EXECUTE INSERISCI_CRIOCELLA(80, 'SANGUE')

```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('15/05/2018 17:30', 'DD/MM/YYYY HH24:MI'), 'FRNGVN62C01D811U', 'SPERMIOGRAMMA',1)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('16/05/2018 12:30', 'DD/MM/YYYY HH24:MI'), 'FRNGVN62C01D811U', 'GENETICO',1)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('16/05/2018 12:00', 'DD/MM/YYYY HH24:MI'), 'FRNGVN62C01D811U', 'SANGUE',1)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('16/05/2018 15:30', 'DD/MM/YYYY HH24:MI'), 'FRNGVN62C01D811U', 'DONAZIONE',2)
```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('15/05/2018 11:30', 'DD/MM/YYYY HH24:MI'), 'PSNNGL75S23Z110U', 'SPERMIOGRAMMA',4)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('15/05/2018 12:30', 'DD/MM/YYYY HH24:MI'), 'PSNNGL75S23Z110U', 'GENETICO',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('16/05/2018 10:00', 'DD/MM/YYYY HH24:MI'), 'PSNNGL75S23Z110U', 'SANGUE',1)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('16/05/2018 14:00', 'DD/MM/YYYY HH24:MI'), 'PSNNGL75S23Z110U', 'DONAZIONE',5)
```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('15/05/2018 8:30', 'DD/MM/YYYY HH24:MI'), 'NDRVDO75T30H151V', 'SPERMIOGRAMMA',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('16/05/2018 12:30', 'DD/MM/YYYY HH24:MI'), 'NDRVDO75T30H151V', 'GENETICO',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('16/05/2018 9:00', 'DD/MM/YYYY HH24:MI'), 'NDRVDO75T30H151V', 'SANGUE',5)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('16/05/2018 15:30', 'DD/MM/YYYY HH24:MI'), 'NDRVDO75T30H151V', 'DONAZIONE',4)
```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('15/05/2018 12:00', 'DD/MM/YYYY HH24:MI'), 'CLBDLM70M19C573R', 'SPERMIOGRAMMA',4)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('15/05/2018 13:00', 'DD/MM/YYYY HH24:MI'), 'CLBDLM70M19C573R', 'GENETICO',4)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('17/05/2018 13:30', 'DD/MM/YYYY HH24:MI'), 'CLBDLM70M19C573R', 'SANGUE',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('18/05/2018 14:00', 'DD/MM/YYYY HH24:MI'), 'CLBDLM70M19C573R', 'DONAZIONE',1)
```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('16/05/2018 9:30', 'DD/MM/YYYY HH24:MI'), 'MRTLRT94S21E054P', 'SPERMIOGRAMMA',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('17/05/2018 14:30', 'DD/MM/YYYY HH24:MI'), 'MRTLRT94S21E054P', 'GENETICO',4)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('17/05/2018 14:00', 'DD/MM/YYYY HH24:MI'), 'MRTLRT94S21E054P', 'SANGUE',2)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('17/05/2018 15:30', 'DD/MM/YYYY HH24:MI'), 'MRTLRT94S21E054P', 'DONAZIONE',1)
```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('15/05/2018 19:30', 'DD/MM/YYYY HH24:MI'), 'MARFLL80S38F391U', 'SPERMIOGRAMMA',1)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('15/05/2018 19:00', 'DD/MM/YYYY HH24:MI'), 'MARFLL80S38F391U', 'GENETICO',3)
```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('16/05/2018 20:00', 'DD/MM/YYYY HH24:MI'), 'MARFLL80S38F391U', 'SANGUE',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('17/05/2018 20:00', 'DD/MM/YYYY HH24:MI'), 'MARFLL80S38F391U', 'DONAZIONE',3)
```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('17/05/2018 18:30', 'DD/MM/YYYY HH24:MI'), 'SPSFLC72E18A149T', 'SPERMIOGRAMMA',5)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('18/05/2018 17:30', 'DD/MM/YYYY HH24:MI'), 'SPSFLC72E18A149T', 'GENETICO',4)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('18/05/2018 15:00', 'DD/MM/YYYY HH24:MI'), 'SPSFLC72E18A149T', 'SANGUE',5)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('18/05/2018 16:30', 'DD/MM/YYYY HH24:MI'), 'SPSFLC72E18A149T', 'DONAZIONE',2)
```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('15/05/2018 14:30', 'DD/MM/YYYY HH24:MI'), 'PGLPTL46P07G702M', 'SPERMIOGRAMMA',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('18/05/2018 13:30', 'DD/MM/YYYY HH24:MI'), 'PGLPTL46P07G702M', 'GENETICO',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('18/05/2018 13:00', 'DD/MM/YYYY HH24:MI'), 'PGLPTL46P07G702M', 'SANGUE',4)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('19/05/2018 16:30', 'DD/MM/YYYY HH24:MI'), 'PGLPTL46P07G702M', 'DONAZIONE',5)
```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('15/05/2018 8:30', 'DD/MM/YYYY HH24:MI'), 'DNIMTT96E22A462R', 'SPERMIOGRAMMA',2)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('16/05/2018 19:30', 'DD/MM/YYYY HH24:MI'), 'DNIMTT96E22A462R', 'GENETICO',1)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('17/05/2018 9:00', 'DD/MM/YYYY HH24:MI'), 'DNIMTT96E22A462R', 'SANGUE',1)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('18/05/2018 13:30', 'DD/MM/YYYY HH24:MI'), 'DNIMTT96E22A462R', 'DONAZIONE',5)
```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('18/05/2018 8:30', 'DD/MM/YYYY HH24:MI'), 'HND BSS66S16H555Z', 'SPERMIOGRAMMA',4)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('18/05/2018 10:30', 'DD/MM/YYYY HH24:MI'), 'HND BSS66S16H555Z', 'GENETICO',5)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('19/05/2018 9:00', 'DD/MM/YYYY HH24:MI'), 'HND BSS66S16H555Z', 'SANGUE',2)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('19/05/2018 9:30', 'DD/MM/YYYY HH24:MI'), 'HND BSS66S16H555Z', 'DONAZIONE',5)
```

```
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('23/05/2018 12:30', 'DD/MM/YYYY HH24:MI'), 'NDRVDO75T30H151V', 'DONAZIONE',1)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('23/05/2018 20:00', 'DD/MM/YYYY HH24:MI'), 'MARFLL80S38F391U', 'DONAZIONE',2)
```

--7 Esami Genetici validi, 3 non validi

```
EXECUTE INSERT_ES_GENET (2745, 7, 1, '46XY')
```

```

EXECUTE INSERT_ES_GENET (2759, 10, 1, '46XY')
EXECUTE INSERT_ES_GENET (16881, 55, 1, '46XY')
EXECUTE INSERT_ES_GENET (16894, 49, 1, '46XY')
EXECUTE INSERT_ES_GENET (16905, 35, 1, '46XY')
EXECUTE INSERT_ES_GENET (16955, 31, 1, '46XY')
EXECUTE INSERT_ES_GENET (23962, 20, 1, '46XY')
EXECUTE INSERT_ES_GENET (24013, 30, 0, '46XY')
EXECUTE INSERT_ES_GENET (24043, 55, 1, '45XY')
EXECUTE INSERT_ES_GENET (31109, 40, 1, '46XY')

```

--9 Esami del sangue validi, 1 non valido

```

EXECUTE INSERT_ES_SANGUE (2740, 3, 1200, 600)
EXECUTE INSERT_ES_SANGUE (2744, 5, 1470, 740)
EXECUTE INSERT_ES_SANGUE (2762, 6, 2240, 830)
EXECUTE INSERT_ES_SANGUE (9852, 8, 2180, 1010)
EXECUTE INSERT_ES_SANGUE (9890, 7, 1950, 950)
EXECUTE INSERT_ES_SANGUE (16920, 9, 1840, 960)
EXECUTE INSERT_ES_SANGUE (16931, 5, 2010, 720)
EXECUTE INSERT_ES_SANGUE (24034, 6, 1890, 510)
EXECUTE INSERT_ES_SANGUE (31058, 6, 2100, 690)
EXECUTE INSERT_ES_SANGUE (31118, 7, 1600, 710)

```

--8 Esami del seme validi, 2 non validi

```

EXECUTE INSERT_ES_SPERM (2731, 16, 5, 59, 8)
EXECUTE INSERT_ES_SPERM (2735, 20, 10, 79, 9)
EXECUTE INSERT_ES_SPERM (9793, 14, 3, 99, 8)
EXECUTE INSERT_ES_SPERM (16873, 22, 4, 89, 10)
EXECUTE INSERT_ES_SPERM (16885, 31, 11, 70, 9)
EXECUTE INSERT_ES_SPERM (16899, 17, 6, 46, 10)
EXECUTE INSERT_ES_SPERM (23959, 36, 3, 69, 12)
EXECUTE INSERT_ES_SPERM (23960, 24, 3, 71, 13)
EXECUTE INSERT_ES_SPERM (24025, 21, 7, 94, 6)
EXECUTE INSERT_ES_SPERM (31101, 43, 5, 55, 8)

```

```

EXECUTE INSERISCI_FIALA_SEME(2796, 'CSP8', 'CGE17')
EXECUTE INSERISCI_FIALA_SEME(2775, 'CSP6', 'CGE18')
EXECUTE INSERISCI_FIALA_SEME(9831, 'CSP1', 'CGE11')
EXECUTE INSERISCI_FIALA_SEME(9881, 'CSP10', 'CGE19')
EXECUTE INSERISCI_FIALA_SEME(16944, 'CSP2', 'CGE14')
EXECUTE INSERISCI_FIALA_SEME(23991, 'CSP4', 'CGE15')
EXECUTE INSERISCI_FIALA_SEME(31068, 'CSP7', 'CGE13')
EXECUTE INSERISCI_FIALA_SEME(31115, 'CSP3', 'CGE12')
EXECUTE INSERISCI_FIALA_SEME(31131, 'CSP9', 'CGE20')
EXECUTE INSERISCI_FIALA_SEME(31145, 'CSP5', 'CGE16')
EXECUTE INSERISCI_FIALA_SEME(31145, 'CSP5', 'CGE16')
EXECUTE INSERISCI_FIALA_SEME(23991, 'CSP4', 'CGE15')
EXECUTE INSERISCI_FIALA_SEME(23991, 'CSP4', 'CGE15')
EXECUTE INSERISCI_FIALA_SEME(23991, 'CSP4', 'CGE15')

```

```

EXECUTE INSERISCI_FIALA_SEME(23991 , 'CSP4' , 'CGE15')
EXECUTE INSERISCI_FIALA_SEME(23991 , 'CSP4' , 'CGE15')
EXECUTE INSERISCI_FIALA_SEME(23991 , 'CSP4' , 'CGE15')
EXECUTE INSERISCI_FIALA_SEME(9960 , 'CSP2' , 'CGE14')
EXECUTE INSERISCI_FIALA_SEME(9960 , 'CSP2' , 'CGE14')
EXECUTE INSERISCI_FIALA_SEME(9960 , 'CSP2' , 'CGE14')
EXECUTE INSERISCI_FIALA_SEME(9960 , 'CSP2' , 'CGE14')
EXECUTE INSERISCI_FIALA_SEME(9960 , 'CSP2' , 'CGE14')

```

--Faccio ora le prenotazioni per i clienti. 4 fanno la procedura per l'inseminazione

```

EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('17/05/2018 12:30', 'DD/MM/YYYY HH24:MI'), 'TRNFNC54B53L379X', 'SANGUE',2)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('17/05/2018 19:30', 'DD/MM/YYYY HH24:MI'), 'TRNFNC54B53L379X', 'FERTILITA',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('18/05/2018 14:30', 'DD/MM/YYYY HH24:MI'), 'TRNFNC54B53L379X', 'INSEMINAZIONE',4)

```

```

EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('17/05/2018 8:30', 'DD/MM/YYYY HH24:MI'), 'GRCMLN42A66B598L', 'SANGUE',2)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('19/05/2018 13:30', 'DD/MM/YYYY HH24:MI'), 'GRCMLN42A66B598L', 'FERTILITA',3)

```

```

EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('21/05/2018 11:30', 'DD/MM/YYYY HH24:MI'), 'CLBTBT70C52E084E', 'SANGUE',2)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('23/05/2018 12:30', 'DD/MM/YYYY HH24:MI'), 'CLBTBT70C52E084E', 'FERTILITA',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('24/05/2018 13:30', 'DD/MM/YYYY HH24:MI'), 'CLBTBT70C52E084E', 'INSEMINAZIONE',4)

```

```

EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('23/05/2018 12:00', 'DD/MM/YYYY HH24:MI'), 'BRSLDA81H41G494R', 'SANGUE',1)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('26/05/2018 13:30', 'DD/MM/YYYY HH24:MI'), 'BRSLDA81H41G494R', 'FERTILITA',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('23/05/2018 14:30', 'DD/MM/YYYY HH24:MI'), 'BRSLDA81H41G494R', 'INSEMINAZIONE',4)

```

```

EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('26/05/2018 9:30', 'DD/MM/YYYY HH24:MI'), 'LCLRMN88S59G678O', 'SANGUE',5)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('26/05/2018 16:30', 'DD/MM/YYYY HH24:MI'), 'LCLRMN88S59G678O', 'FERTILITA',3)
EXECUTE INSERISCI_PRENOTAZIONE (TO_DATE('21/05/2018 11:30', 'DD/MM/YYYY HH24:MI'), 'LCLRMN88S59G678O', 'INSEMINAZIONE',4)

```

```

EXECUTE INSERT_ES_SANGUE (9849, 4, 1500, 700)
EXECUTE INSERT_ES_SANGUE (9841, 8, 3000, 950)
EXECUTE INSERT_ES_SANGUE (9919, 5, 2100, 800)

```

```

EXECUTE INSERT_ES_SANGUE (2864, 6, 3100, 850)
EXECUTE INSERT_ES_SANGUE (31227, 4, 2500, 1000)

EXECUTE INSERT_ES_FERT(16943, 4, 3)
EXECUTE INSERT_ES_FERT(16979, 1, 7)
EXECUTE INSERT_ES_FERT(17025, 5, 9)
EXECUTE INSERT_ES_FERT(17075, 6, 8)
EXECUTE INSERT_ES_FERT(17081, 7, 5)

--inserisco i pagamenti per le inseminazioni
EXECUTE INSERISCI_PAG_INS('MASTERCARD', 'TRNFNC54B53L379X',
SYSDATE)
EXECUTE INSERISCI_PAG_INS('VISA', 'CLBTBT70C52E084E', SYSDATE)
EXECUTE INSERISCI_PAG_INS('MASTERCARD', 'BRSLDA81H41G494R',
SYSDATE)
EXECUTE INSERISCI_PAG_INS('CONTANTE', 'LCLRMN88S59G6780', SYSDATE)

--inserisco le inseminazioni

EXECUTE INSERT_INSEMINAZIONE (2, 'PGLPTL46P07G702M', 'IUI',
'FAT1', 24037 )
EXECUTE INSERT_INSEMINAZIONE (1, 'FRNGVN62C01D811U', 'ICI',
'FAT15', 24079 )
EXECUTE INSERT_INSEMINAZIONE (1, 'MARFLL80S38F391U', 'IUI',
'FAT16', 24109 )
EXECUTE INSERT_INSEMINAZIONE (1, 'NDRVDO75T30H151V', 'ICI',
'FAT17', 24131 )
EXECUTE INSERT_INSEMINAZIONE (1, 'MARFLL80S38F391U', 'IUI',
'FAT16', 24109 )
EXECUTE INSERT_INSEMINAZIONE (1, 'NDRVDO75T30H151V', 'ICI',
'FAT17', 24131 )

--fiale acquistate da vari clienti
EXECUTE ACQUISTO_FIALA ('F125', 'SLLLRT59E02F839T', 'VISA')
EXECUTE ACQUISTO_FIALA ('F162', 'SNCMNO80T42E335P', 'CONTANTE')
EXECUTE ACQUISTO_FIALA ('F163', 'GMNPRZ81C42F113X', 'MASTERCARD')
EXECUTE ACQUISTO_FIALA ('F165', 'GMNPRZ81C42F113X', 'MASTERCARD')
EXECUTE ACQUISTO_FIALA ('F167', 'DFNPSN66C57L723B', 'VISA')
EXECUTE ACQUISTO_FIALA ('F168', 'DFNPSN66C57L723B', 'VISA')
EXECUTE ACQUISTO_FIALA ('F160', 'DFNPSN66C57L723B', 'VISA')
EXECUTE ACQUISTO_FIALA ('F161', 'DFNPSN66C57L723B', 'VISA')

```

Scheduler

In questa sezione parleremo dello scheduling di alcune attività implementate tramite job di Oracle DBMS. Procedendo con ordine, saranno analizzate e descritte singolarmente.

Il seguente job si occupa di chiamare la procedura Organizza_Orari_Prenotazioni che ricordiamo utile per la realizzazione degli slot delle Prenotazioni. Supponiamo che la prima esecuzione sia manuale mentre le successive avverranno una volta all'anno, il 1 Gennaio di ogni anno.

```
BEGIN
DBMS_SCHEDULER.CREATE_JOB (
    job_name          => 'Inizializza_Prenotazioni',
    job_type          => 'STORED_PROCEDURE',
    job_action         => 'ORGANIZZA_ORARI_PRENOTAZIONI',
    start_date        => TO_DATE('24/04/2018','DD/MM/YYYY'),
    repeat_interval    => 'FREQ=YEARLY',
    enabled            => TRUE,
    comments           => 'Crea gli slot per gli esami prenotabili in
tutto l''anno.');
```

END;

Il job Pulisci_Prenotazioni permette di ripulire tutti gli slot della tabella Prenotazione che non sono stati utilizzati per registrare un servizio prenotato.

```
BEGIN
DBMS_SCHEDULER.CREATE_JOB (

    job_name          => 'Pulisci_Prenotazioni',
    job_type          => 'PLSQL_BLOCK',
    job_action         => 'BEGIN
                           DELETE FROM PRENOTAZIONE
                           WHERE TRUNC(DATA_ORA)<TRUNC(SYSDATE) AND
CF_CLI IS NULL AND CF_DON IS NULL;

                           COMMIT;
                           END;',
    start_date        => NEXT_DAY(SYSDATE, 'DOMENICA'),
    repeat_interval    => 'FREQ=WEEKLY',
    enabled            => TRUE,

    comments           => 'Elimina le tuple di prenotazione che non
sono state utilizzate.');
```

END;

Quest'ultimo job richiama la procedura Organizza_Turni_Dottori a cadenza settimanale, in particolare ogni domenica, assegnando in modo pseudo-casuale i turni ai dottori.

BEGIN

```
DBMS_SCHEDULER.CREATE_JOB (  
    job_name          => 'Organizza_turni',  
    job_type          => 'STORED_PROCEDURE',  
    job_action         => 'ORGANIZZA_TURNI_DOTTORI',  
    start_date        => NEXT_DAY(SYSDATE, 'DOMENICA'),  
    repeat_interval    => 'FREQ=WEEKLY',  
    enabled            => TRUE,  
    comments           => 'Organizza i turni dei dottori ogni  
settimana di domenica.');
```

END;

Normalizzazione

In questa ultima sezione ci occuperemo ad effettuare la normalità di questa base di dati. Per ogni forma normale, analizzeremo e argomenteremo le motivazioni per cui ne impediscono il raggiungimento o ne favoriscono.

Prima forma normale

Il database, volendo essere precisi, non risponde alla prima forma normale a causa dei campi Data e Ora rappresentati come attributi atomici, quando in realtà non lo sono. Riteniamo, però, di poter sorvolare su questa *anomalia*, volendola chiamare così, di Oracle DBMS, che implementa questo tipo di attributo come atomico. Solo nel caso della tabella Turno ci potrebbero essere dei problemi, in cui è necessario inserire soltanto l'orario. Questo problema è sorvolabile mediante un'opportuna chiamata di `to_date()` per inserire soltanto l'orario.

Seconda forma normale

Sono presenti chiavi primarie multiattributo. Ciononostante sono relative a tabelle di transizione: Lavora_In, Effettua_T. Questa fa sì che nessuna di essa determini parzialmente alcun attributo della relazione, per cui si può affermare che il database è in seconda forma normale.

Terza forma normale e forma normale di Boyce e Codd

Non esiste nessuna dipendenza transitiva tra attributi non-chiave. Inizialmente, era previsto un campo per memorizzare il CAP nei vari indirizzi. Questo attributo, se non opportunamente gestito, avrebbe portato al non rispetto della terza forma normale a causa della dipendenza di questo attributo non-chiave da un altro campo non-chiave, la provincia. Per semplicità, e poiché si trattava di un'informazione per nulla indispensabile, abbiamo deciso di rimuoverlo, così che il DB potesse essere in BNCF.