

Progetto Base di Dati e Sistemi Informativi

A.A. 2020-2021



Autori

Elena Abbate - 7032147 elena.abbate@stud.unifi.it
Lisa Comacchio - 7027412 lisa.comacchio@stud.unifi.it

Centro Aiuti Volontariato (CAV)

1 - Obiettivi, strumenti e metodologia	2
2 - Raccolta delle specifiche	2
3 - Progettazione concettuale	3
3.0 Breve introduzione	3
3.1 Tecniche utilizzate e descrizione	3
4 - Progettazione logica	5
4.1 - Ristrutturazione dello schema E-R al relazionale	5
4.1.1- Analisi delle ridondanze	5
4.1.2 - Eliminazione delle generalizzazioni	7
4.1.3 - Eliminazione degli attributi composti	7
4.1.4 - Scelta degli identificatori	8
4.2 - Trasformazioni entità/relazioni in tabelle	9
5 - Progettazione fisica	11
5.1 Trigger	11
5.2 View	11
5.3 Procedure	11

1 - Obiettivi, strumenti e metodologia

Il progetto ha come scopo la realizzazione di una base di dati relazionale a tema libero utilizzando il DBMS MySQL (MySQL Server 8.8, MySQL Workbench 6.3, Dbeaver per simulazioni).

La metodologia utilizzata segue i seguenti passi:

- raccolta delle specifiche in linguaggio relazionale
- progettazione concettuale: traduzione delle specifiche in un diagramma Entità-Relazione
- ristrutturazione dello schema E-R: analisi di prestazioni e ridondanze, eliminazione delle generalizzazioni
- progettazione logica: traduzione dal diagramma E-R al modello relazionale
- stesura del codice.

2 - Raccolta delle specifiche

Si è realizzata la base di dati per un **centro esami**, composto da dottori volontari, che si occupano di svolgere esami specifici per pazienti non abbienti.

Ogni **paziente** è identificato con nome, cognome, con un codice univoco (il CF) e il recapito. Può prenotare, in una data specifica, uno singolo **esame** tramite il codice della prenotazione, che verrà effettuato nell'apposito reparto e dal dottore responsabile dello stesso, nella fascia oraria indicata nella prenotazione.

Il **dottore** è identificato da nome, cognome e da un codice univoco personale ed opera in un solo reparto. Può seguire fino ad un massimo di 5 prenotazioni al giorno e dopo aver effettuato un esame, potrà assegnare al paziente un farmaco tramite prescrizione.

Vista la limitatezza dei posti e la grande richiesta, ogni paziente, potrà richiedere un solo un esame per ogni **prenotazione** la quale verrà riconosciuta tramite un codice univoco, la data e la fascia oraria richiesti.

Ogni reparto è identificato da un codice, dal nome e il tipo di esami che si svolgono al suo interno.

Essendo un'attività gestita dallo Stato, l'amministrazione e tutto quello che riguarda l'organizzazione viene gestito da esterni: non dovremo quindi preoccuparci della loro gestione.

3 - Progettazione concettuale

3.0 Breve introduzione

Progettare concettualmente una base di dati significa individuare gli oggetti (o entità) che la costituiscono e le relazioni (operazioni o associazioni) tra un oggetto e l'altro. Si tratta del livello più alto della progettazione di un database, quello più vicino all'uomo e più lontano dalla macchina (hardware).

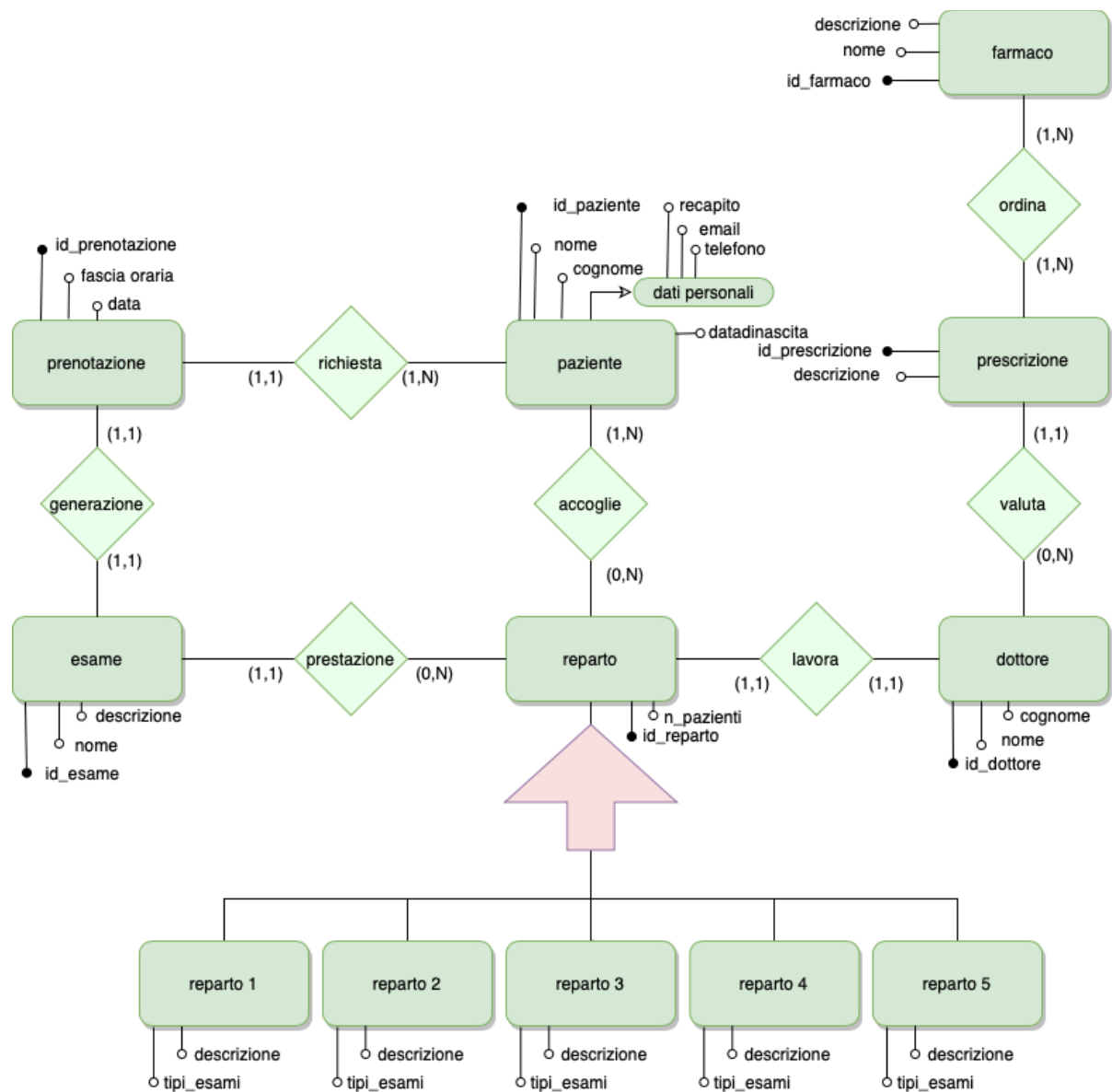
Utilizzeremo quindi un linguaggio indipendente dal sistema.

3.1 Tecniche utilizzate e descrizione

Avendo avuto un'idea chiara fin da subito sul tipo di prestazioni e di servizio che avrebbe dovuto soddisfare il progetto, abbiamo deciso di fare la traduzione delle specifiche in schema ER attraverso la strategia **TOP-DOWN**, considerando le specifiche in modo GLOBALE ed in seguito abbiamo prodotto uno schema iniziale completo ma con POCHI CONCETTI e ASTRATTI.

- È stata individuata l'entità **paziente** collegato mediante la relazione **richiesta** (*uno a molti*) con l'entità **prenotazione** e mediante la relazione **accoglie** (*uno a molti*) con l'entità **reparto** con l'attributo nome, cognome, dati personali (composto da telefono, indirizzo, e-mail, ecc.).
- È stata aggiunta l'entità **prenotazione** collegata mediante la relazione **richiesta** (*uno a uno*) con l'entità **paziente** e mediante la relazione **generazione** (*uno a uno*) con l'entità **esame**. L'entità prenotazione, sarà identificata da codice, data e fascia oraria.
- È stata aggiunta l'entità **esame** collegato mediante la relazione **generazione** (*uno a uno*) con l'entità **prenotazione**, mediante la relazione **prestazione** (*uno a uno*) con l'entità **reparto**. L'entità esame è identificato dagli attributi nome e descrizione.
- È stato aggiunta l'entità **reparto** collegato mediante la relazione **accoglie** (*zero a molti*) con l'entità **paziente** mediante la relazione **prestazione** (*zero a molti*) con l'entità **esame** e mediante la relazione **lavora** (*uno a uno*) con l'entità **dottore**. L'entità reparto è identificato dal codice e da numero paziente. Si crea inoltre una generalizzazione composta da cinque reparti (INDICARE I NOMI!) differenziati dagli attributi: tipi di esami e descrizioni.
- È stato aggiunta l'entità **dottore** collegato mediante la relazione **lavora** (*uno a uno*) con l'entità **reparto** e mediante la relazione **valuta** (*zero a molti*) con l'entità **prescrizione**. l'entità dottore, sarà identificata dal campo codice, nome e cognome.

- È stata aggiunta l'entità **prescrizione** collegato mediante la relazione **valuta** (uno a uno) con l'entità **dottore** e mediante la relazione **ordina** (uno a molti) con l'entità **farmaco**, identificata dal campo codice e descrizione.
- È stata aggiunta l'entità **farmaco** collegato mediante la relazione **ordina** (uno a molti) con l'entità **prescrizione** identificata dal campo descrizione e nome



4 - Progettazione logica

4.1 - Ristrutturazione dello schema E-R al relazionale

4.1.1- Analisi delle ridondanze

Per confrontare tra loro diverse alternative di traduzione, bisogna conoscere, almeno in maniera approssimativa, il cosiddetto “carico di lavoro” del DB, ovvero:

- I “volumi” dei dati in gioco
- Le principali operazioni che dovrà supportare

Gli indicatori di costo che si utilizzano considerano quindi due aspetti sono:

- spazio: numero di istanze previste (tabella dei volumi)
- tempo: numero di istanze (di entità e associazioni) visitate durante un’operazione (tabella degli accessi)

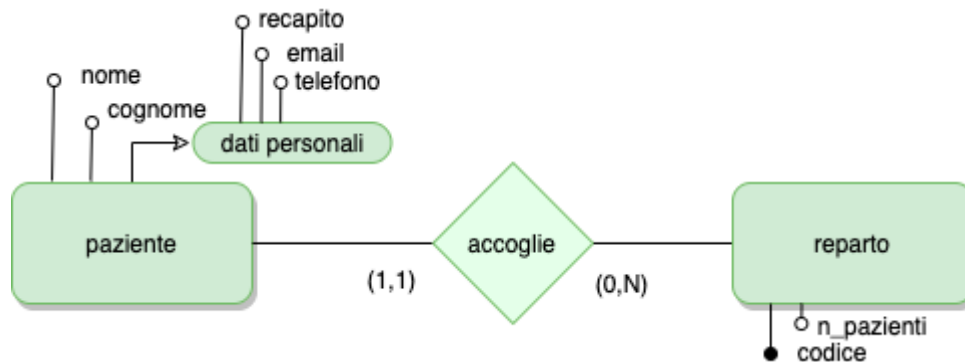
In base a quanto sopra descritto, si costruisce la tabella dei volumi che specifica il numero stimato di istanze per ogni entità (E) e relazione (R) dello schema:

Tavola dei volumi (u.m. un giorno)			
concetto	tipo	volume	descrizione
esame	E	10	tipi di esami
prenotazione	E	25	25 = 1 paziente x 5 fasce orarie x 5 reparti
paziente	E	25	al massimo 25 = 1 paziente x 5 fasce orarie x 5 reparti
dottore	E	5	5 = 1 dottore x 5 reparti
reparto	E	5	
prescrizione	E	≈ 12	considerando circa la metà dei pazienti ricoverati
farmaco	E	10	tipo di farmaco
generazione	R	25	va in base al numero delle prenotazioni
richiesta	R	25	va in base al numero delle prenotazioni
accoglie	R	25	vengono ricoverati al massimo 25 pazienti al giorno
lavora	R	5	lavorano un totale di 5 dottori nel centro
prestazione	R	25	va in base al numero di prenotazioni
valuta	R	≈ 12	considerando la metà dei pazienti
ordina	R	≈ 18	considerando il 25% dei pazienti abbiano bisogno di un solo farmaco e 25% dei pazienti abbiano bisogno di 2 farmaci

Per ogni entità e associazione interessata dall’operazione, si riportano il numero di istanze interessate ed il tipo di accesso (L: lettura; S: scrittura).

Il numero delle istanze si ricava dalla tavola dei volumi mediante semplici operazioni algebriche.

Dopo aver analizzato il numero di operazioni necessarie, si effettua l'analisi e si cercano le possibili ridondanze.



L'attributo `n_pazienti` in `reparto` può essere omissso, poiché calcolabile tramite la relazione con `paziente`.

Come arco temporale si considera una settimana lavorativa (5 giorni). Di seguito due tabelle riassuntive:

Tabella dei volumi		
Concetto	Costrutto	Volume
Paziente	E	125
Accoglie	R	125
Reparto	E	5

Tabella degli accessi			
Concetto	Costrutto	Accessi	Tipo
Paziente	E	1	Scrittura
Accoglie	R	1	Scrittura
Reparto	E	1	Lettura
Reparto	E	1	Scrittura

Prendendo quindi come arco temporale una settimana, con il massimo degli ingressi ogni giorno (cioè 5), si avrà un costo totale massimo di $125 \times 3 S + 125 L$.

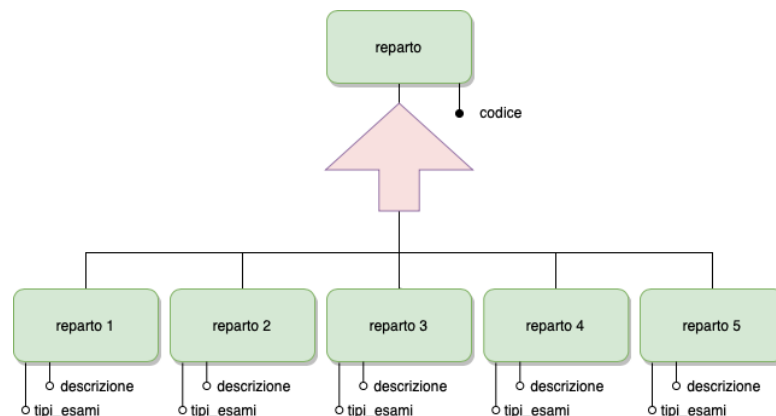
La seguente tabella, prende in considerazione il caso in cui non sia presente l'attributo `n_pazienti`:

Tabella degli accessi			
Concetto	Costrutto	Accessi	Tipo
Paziente	E	1	Scrittura
Accoglie	R	1	Scrittura

Sempre prendendo come arco temporale una settimana, con il massimo degli ingressi consentiti per ogni giorno (cioè 5), si avrà un costo di 125×2 in scrittura .

Considerando che nella prima opzione abbiamo un totale di 500 operazioni da svolgere mentre nella seconda 250 , è necessario eliminare la ridondanza.

4.1.2 - Eliminazione delle generalizzazioni

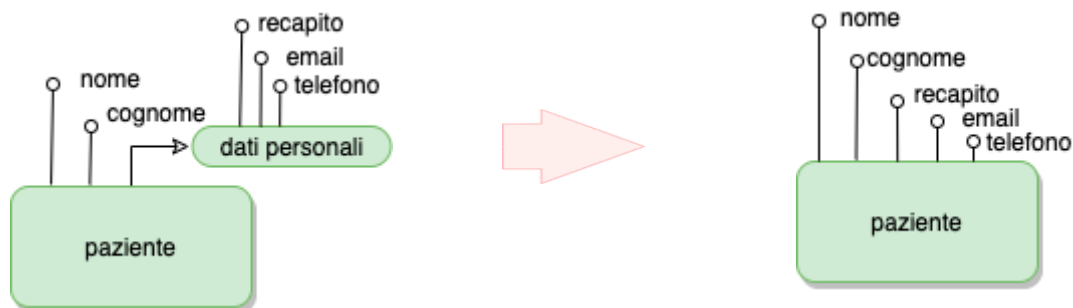


Il modello relazionale non può rappresentare direttamente le generalizzazioni. Nel nostro caso considerando che i tipi di reparto hanno tutti gli stessi attributi e che nessuno di essi dovrà avere relazioni a sè stanti con altre entità, è opportuno inglobare gli attributi dei figli nelle entità padre .

4.1.3 - Eliminazione degli attributi composti

Ai fini della generalizzazione e quindi dell'ottimizzazione del db si reputa opportuno condensare i singoli attributi del campo dati_personali riunendoli all'interno dell'entità padre paziente.

Nello specifico gli attributi recapito, email e telefono vengono direttamente collegati all'entità paziente



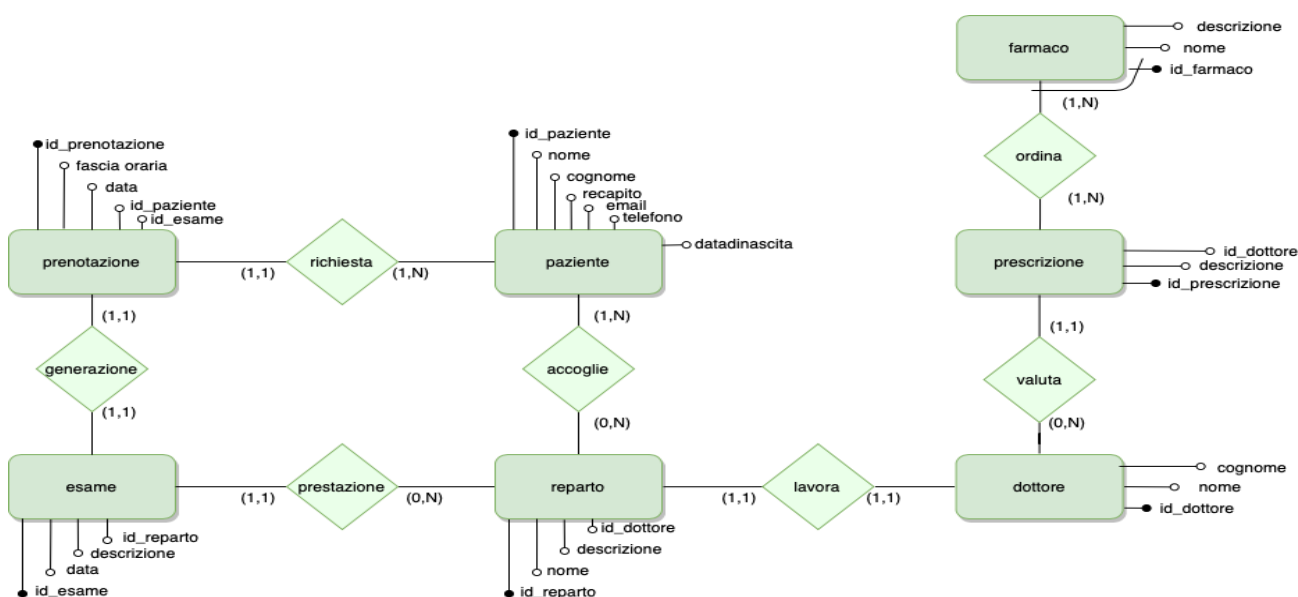
4.1.4 - Scelta degli identificatori

Ogni entità costituisce un sottoinsieme degli attributi, che identificano in maniera univoca ogni occorrenza della stessa entità. Essendo univoche è importante che le stesse:

- non siano nulle
- che siano semplici

Nel caso non sia possibile trovare campi che soddisfano questi requisiti, si creano degli indicatori automatici .

Entità	Campo	Tipo
Reparto	id_reparto	nuovo campo, sequenziale
Dottore	id_dottore	nuovo campo sequenziale
Farmaco	id_farmaco	nuovo campo, sequenziale
Paziente	id_paziente	nuovo campo, corrisponde al codice fiscale
Esame	id_esame	nuovo campo, sequenziale
Prenotazione	id_prenotazione	attributo esistente
Prescrizione	id_prescrizione	attributo esistente



4.2 - Trasformazioni entità/relazioni in tabelle

Per la traduzione dal modello entità-relazione a quello relazionale secondo i pattern standard comunemente utilizzati in questi casi, utilizziamo le seguenti metodologie:

- Le entità sono state tradotte in tabella con campi preesistenti
- le relazioni possono diventare uno dei seguenti casi (in base alle cardinalità fra e/r):

-> uno a uno

Le relationship verranno eliminate inserendo il vincolo d'integrità referenziale all'interno di una delle tue tabelle:

Prenotazione (*id_prenotazione, id_esame fascia_oraria, data, id_paziente*)

Esame (*id_esame, id_reparto, nome, descrizione*)

Esame.id_esame = Prenotazione.id_esame

Reparto (*id_reparto, codice_dottore descrizione, nome*)

Dottore (*codice, nome, cognome*)

Reparto.codice_dottore = Dottore.codice

-> uno a molti

Per non creare ridondanza, le relationship verranno eliminate, inserendo all'interno della tabella con cardinalità max, la chiave primaria dell'altra:

Prenotazione (*id_prenotazione, id_paziente, fascia_oraria, data,*)

Paziente (*id_paziente, nome, cognome, recapito*)

Prenotazione.id_paziente = Paziente.id_paziente

Prescrizione (*id_prescrizione, id_dottore, descrizione*)

Dottore (*codice, nome, cognome*)

Prescrizione.id_dottore = Dottore.codice

Esame (*id_esame, id_reparto, codice_prenotazione, nome, descrizione*)

Reparto (*id_reparto, codice_dottore descrizione, nome*)

Esame.id_reparto = Reparto.id_reparto

-> molti a molti

Si crea una tabella anche per la relazione, che avrà come identificatore il valore di due identificatori di entità:

Farmaco (*id_farmaco, nome, descrizione*)

Prescrizione (*id_prescrizione, id_dottore, descrizione*)

Ordina(id_prescrizione, id_farmaco)

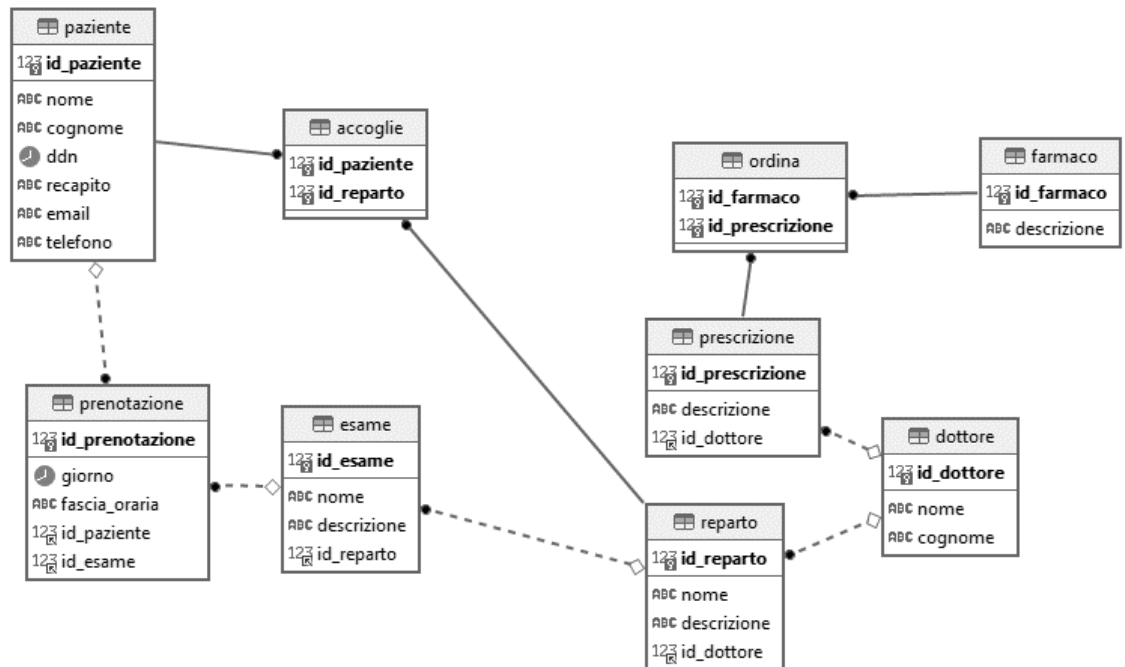
Ordina.id_prescrizione = Prescrizione.id_prescrizione

Ordina.id_farmaco = Farmaco.id_farmaco

Paziente(id_paziente, nome, cognome, recapito)

Reparto(id_reparto, codice_dottore descrizione , nome)

Accoglie (id_paziente, id_reparto)



5 - Progettazione fisica

Di seguito, si riporta una descrizione di massima dei criteri adottati per stendere il codice SQL dello schema logico relazionale individuato nel paragrafo precedente. Per fare questo, esamineremo ad una ad una le categorie di oggetti DB utilizzati, mentre per i dettagli si rimanda direttamente al codice sql nel file allegato.

5.1 Trigger

Il trigger MySQL è un oggetto di database che è associato ad una tabella e si attiva quando un'azione predefinita viene eseguita.

Nel nostro contesto un dottore può effettuare al massimo 5 esami al giorno, quindi le prenotazioni per ogni reparto sono 5. Attraverso l'utilizzo dei trigger sarà possibile inserire solo un set di 5 prenotazioni al giorno:

`trigger_n_prenotazioni_maxGiornaliere`: effettua il controllo sulla prenotazione dei pazienti per un massimo di 5 prenotazioni giornaliere

`trigger_fasceOrarie`: effettua il controllo sulla prenotazione dei pazienti in una determinata fascia oraria in quanto non è possibile avere due pazienti nello stesso momento per un esame

5.2 View

Esattamente come le tabelle, infatti, le Viste sono composte da colonne e righe le quali, però, sono il frutto di una query che è stata memorizzata, all'interno del database, in qualità di oggetto.

5.3 Procedure

Mediante i vincoli di integrità referenziale, gli attributi ed i trigger sono stati rispettati tutti i vincoli indicati in fase di progettazione logica.

Esistono però alcune esigenze di correttezza logica dei dati che non possono essere garantite dai semplici vincoli di integrità, e che però devono essere verificati soprattutto in fase di inserimento dai dati all'interno del DB.