



Progetto Basi di Dati

2082849
Perozzo Andrea

2075515
Tutino Giuseppe

1 ABSTRACT

La clinica privata Luzzatti, fondata nel 1990, si distingue tra le più importanti realtà nell'ambito dell'assistenza sanitaria privata, offrendo agli utenti un'alternativa di qualità rispetto all'assistenza pubblica. Nel corso degli anni grazie all'impegno costante di un team medico ed infermieristico altamente specializzato, la clinica si è affermata come uno dei principali punti di riferimento nel settore sanitario. L'organizzazione interna prevede che gli operatori sanitari (medici e infermieri), siano assegnati a specifici reparti ed ognuno di essi sia coordinato da un medico primario. Ogni reparto è suddiviso in specifiche stanze, ognuna delle quali è dedicata alle prestazioni mediche previste in ogni appuntamento, assicurando in tale modo, all'utente, una gestione ottimale delle risorse e dell'organizzazione delle attività cliniche. L'utente che desidera prenotare un appuntamento presso questa clinica potrà scegliere l'ora e il giorno desiderati. Quando la struttura riceverà la richiesta dell'appuntamento, verranno assegnati gli operatori sanitari ed una specifica stanza libera, in quel determinato giorno ed ora. Al termine dell'appuntamento, verrà emessa la fattura da pagare ed il referto medico associato (la validazione del referto potrebbe richiedere alcuni giorni). Gli utenti hanno la possibilità di poter fornire alla clinica le patologie pregresse ed i vaccini effettuati fino a quel momento. In questo modo verranno fornite le migliori cure aumentando la qualità dei trattamenti offerti.

2 ANALISI DEI REQUISITI

2.1 Descrizione

Si vuole creare un database per gestire gli appuntamenti della clinica.

Ad ogni appuntamento possono partecipare medici e infermieri, deve essere presente almeno un medico per appuntamento. L'appuntamento viene svolto in una stanza che non deve avere altri appuntamenti nella stesso giorno ed ora. Per ogni appuntamento serve sapere:

- Data e ora dell'appuntamento

Ogni appuntamento è fissato da un utente, il quale è descritto da:

- Codice Fiscale

- Nome
- Cognome
- Indirizzo E-mail
- Numero di telefono (anche più di uno)
- Data di nascita
- Totale delle fatture da pagare

Il totale delle fatture da pagare è la somma dell'ammontare delle fatture ancora da pagare, di conseguenza il suo valore deve essere maggiore o uguale a zero.

Ogni utente può di fornire alla clinica la lista dei vaccini e delle patologie di cui è in possesso.
Dei vaccini serve sapere:

- Nome del vaccino
- Descrizione breve
- Data di vaccinazione

Delle patologie serve sapere:

- Nome della patologia
- Descrizione breve
- Data di scoperta della patologia

Nella clinica lavorano gli operatori sanitari dei quali occorre sapere le seguenti informazioni:

- Codice identificativo
- Nome
- Cognome
- Data di nascita

Tra gli operatori sanitari sono presenti i medici che sono caratterizzati anche da:

- Specializzazione

Tra gli operatori sanitari sono presenti anche gli infermieri che non hanno bisogno di ulteriori informazioni associate.

La clinica è suddivisa in reparti specializzati, occorre quindi tenere conto di questi reparti che sono caratterizzati da:

- Nome
- Descrizione breve di cosa si occupa

Ogni reparto può avere a capo un medico primario. Ogni reparto ha un certo numero di stanze associate utilizzate per gli appuntamenti che sono identificate da:

- Numero della stanza
- Numero di posti letto
- Piano della clinica in cui si trova

La clinica rilascia dei referti medici alla fine di ogni appuntamento, per ogni referto è utile sapere le seguenti informazioni:

- Esito
- Data di emissione

Infine, viene emessa la fattura per i servizi offerti dalla clinica. La fattura contiene:

- Data di emissione
- Stato attuale della fattura
- Ammontare in euro

2.2 Glossario

- **Referto medico:** un atto scritto nel quale un medico afferma di aver prestato la propria assistenza.

2.3 Operazioni tipiche

- Inserimento di un nuovo appuntamento.
- Inserimento di nuovi referti e fatture.
- Inserimento di un nuovo utente.
- Aggiornamento di attributi di un utente.
- Visualizzazione degli attributi di un medico.

3 PROGETTAZIONE CONCETTUALE

3.1 Lista Entità

Il Database è formato dalle seguenti entità. Tutti i campi sono NOT NULL tranne quelli specificati.

- **Operatore Sanitario:** rappresenta un operatore sanitario che lavora nella clinica
 - ID: int

- Nome: *string*
 - Cognome: *string*
 - Nascita: *date*
- **Medico:** rappresenta un medico che lavora nella clinica
 - Specializzazione: *string*
- **Infermiere:** rappresenta un infermiere che lavora nella clinica
- **Reparto:** rappresenta un reparto della clinica
 - Nome: *string*
 - Descrizione: *string*
- **Stanza:** rappresenta una stanza della clinica
 - Numero: *int* (Composta con Reparto)
 - Posti: *int*
 - Piano: *int*
- **Appuntamento:** rappresenta un appuntamento fissato
 - Data: *date&time* (Composta con Utente)
- **Referto:** rappresenta il referto medico rilasciato alla fine dell'appuntamento
 - (Composizione con Appuntamento)
 - Esito: *string*
 - Emissione: *date*
- **Fattura:** rappresenta la fattura relativa all'appuntamento
 - (Composizione con Appuntamento)
 - Stato: *string*
 - Emissione: *date*
 - Ammontare: *float*
- **Utente:** rappresenta l'utente
 - CF: *string*
 - Nome: *string*
 - Cognome: *string*
 - Email: *string*
 - Nascita: *date*
 - Telefono: *char(10)*
 - TotaleDovuto: *float*

- **Patologia:** rappresenta una patologia
 - Nome: *string*
 - Descrizione: *string*
- **Vaccino:** rappresenta un vaccino
 - Nome: *string*
 - Descrizione: *string*

3.2 Lista Relazioni

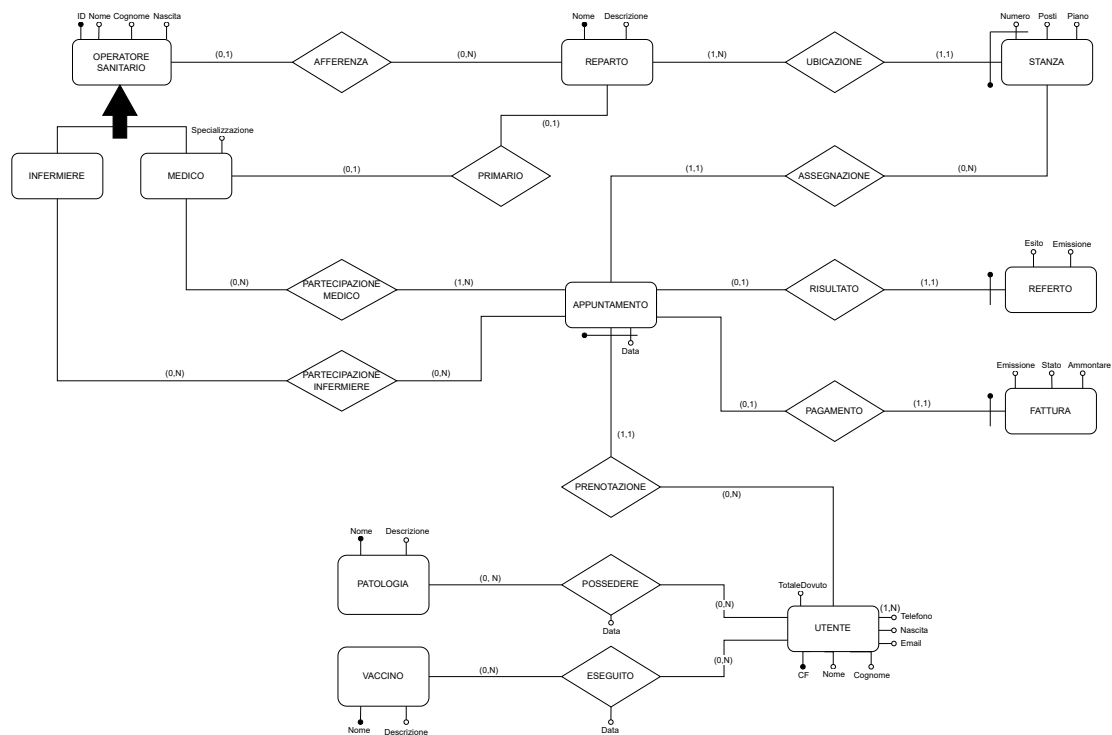
- **Operatore Sanitario-Reperto:** Afferenza (1:N)
 - Un Operatore Sanitario può afferire ad un Reperto
 - Un Reperto ha più afferenti
- **Stanza-Reperto:** Ubicazione (1:N)
 - Un reparto ha più stanze ubicate
 - Una stanza è ubicata in un reparto
- **Medico-Reperto:** Primario (1:1)
 - Un reparto può avere un medico primario
 - Un medico può essere primario di un reparto
- **Appuntamento-Stanza:** Assegnazione (1:N)
 - Un appuntamento viene assegnato ad una stanza
 - Una stanza può essere assegnata a più appuntamenti
- **Infermiere-Appuntamento:** Partecipazione Infermiere (N:M)
 - Un infermiere può partecipare a più appuntamenti
 - Ad ogni appuntamento possono partecipare più infermieri
- **Medico-Appuntamento:** Partecipazione Medico (N:M)
 - Un medico può partecipare a più appuntamenti
 - Ad ogni appuntamento partecipano uno o più medici
- **Appuntamento-Utente:** Prenotazione (1:N)
 - Un utente può prenotare più appuntamenti
 - Un appuntamento viene prenotato da un utente
- **Referto-Appuntamento:** Risultato (1:1)
 - Un referto è il risultato di un appuntamento

- Un appuntamento ha come risultato un referto
- **Fattura-Appuntamento: Pagamento (1:1)**
 - Una fattura attesta il pagamento di un appuntamento
 - Al pagamento di ogni appuntamento viene emessa una fattura
- **Patologia-Utente: Possedere (N:M)**
 - Un utente può possedere più patologie
 - Una patologia può essere posseduta da più utenti
- **Vaccino-Utente: Eseguito (N:M)**
 - Un vaccino può essere stato eseguito su più utenti
 - Un utente può aver eseguito più vaccini

3.3 Lista Generalizzazioni

- **Operatore Sanitario** è una generalizzazione totale di **Infermiere** e **Medico**

3.4 Schema E-R



3.5 Vincoli di integrità non rappresentabili nello schema E-R

- Non possono esistere due **appuntamenti** fissati nella stessa stanza e alla stessa data

4 PROGETTAZIONE LOGICA

4.1 Analisi delle ridondanze

- **Operazione 1 (1 volta alla settimana)** Per ogni utente verificare l'ammontare che deve pagare alla clinica.
- **Operazione 2 (300 volte alla settimana)** Inserimento di una nuova fattura da pagare
- **Operazione 3 (100 volte alla settimana)** Aggiornamento stato di una fattura

Concetto	Tipo	Volume
Utente	Entità	800
Prenotazione	Relazione	2000
Appuntamento	Entità	2000
Pagamento	Relazione	1800
Fattura	Entità	1800

Con ridondanza

- Operazione 1

Concetto	Tipo	Accesso	Tipo
Utente	Entità	800	Lettura

Costo: 800 accessi in lettura

- Operazione 2

Concetto	Tipo	Accesso	Tipo
Fattura	Entità	1	Scrittura
Appuntamento	Entità	1	Lettura
Prenotazione	Relazione	1	Lettura
Utente	Entità	1	Lettura
Utente	Entità	1	Scrittura

Costo: 2x300 accessi in scrittura, 3x300 accessi in lettura

- Operazione 3

Concetto	Tipo	Accesso	Tipo
Fattura	Entità	1	Scrittura
Appuntamento	Entità	1	Lettura
Prenotazione	Relazione	1	Lettura
Utente	Entità	1	Lettura
Utente	Entità	1	Scrittura

Costo: 2x100 accessi in scrittura, 3x100 accessi in lettura

Costo settimanale totale: $800 + 600 \times 2 + 900 + 200 \times 2 + 300 = 3600$

Senza ridondanza

- Operazione 1

Concetto	Tipo	Accesso	Tipo
Utente	Entità	800	Lettura
Prenotazione	Relazione	800	Lettura
Appuntamento	Entità	800	Lettura
Pagamento	Relazione	800	Lettura
Fattura	Entità	800	Lettura

Costo: 5x800 accessi in lettura

- Operazione 2

Concetto	Tipo	Accesso	Tipo
Fattura	Entità	1	Scrittura

Costo: 1x300 accessi in scrittura

- Operazione 3

Concetto	Tipo	Accesso	Tipo
Fattura	Entità	1	Scrittura

Costo: 1x100 accessi in scrittura

Costo settimanale totale: $4000 + 300 \times 2 + 100 \times 2 = 4800$

È evidente che con la ridondanza andiamo a risparmiare 1200 operazioni, quindi conviene tenerla.

4.2 Eliminazione delle generalizzazioni

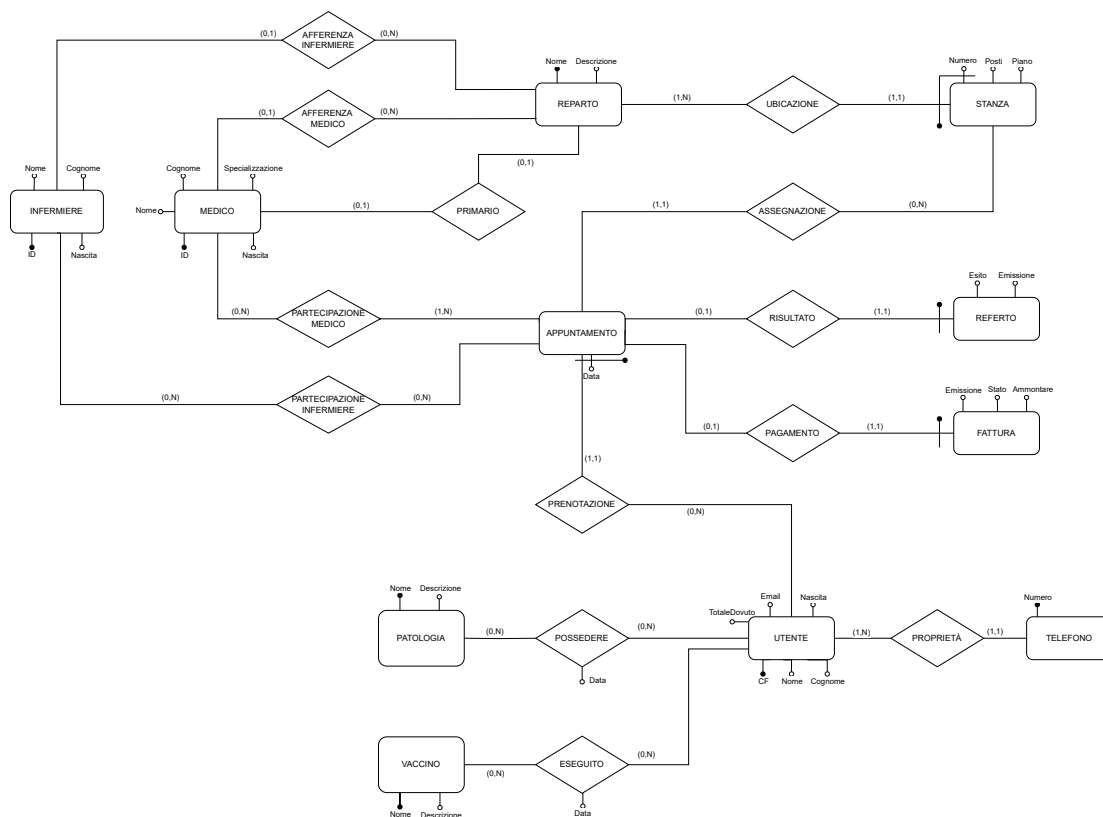
- **Operatore Sanitario:** È una generalizzazione totale, legata da una relazione uno a molti con l'entità **Reparto**. Le entità figlie **Infermiere** e **Medico** sono in relazione molti a molti con l'entità **Appuntamento**. Come obiettivo si è posto di minimizzare il numero di attributi NULL, quindi si è scelto di accorpare il padre nelle entità figlie. Come ulteriore giustificazione, in questo modo si favorisce l'uso delle query che usufruiscono solo di Medici o di Infermieri.

4.3 Partizionamento/Accorpamento di Entità e Relationship

L'entità **Utente** possiede l'attributo **Telefono** il quale può essere trasformato in un'entità che si collega ad **Utente** tramite la relazione **Proprietà**, questo perché un utente può essere in possesso di più numeri di telefono. Quindi si ha la seguente relazione:

- **Telefono-Utente:** Proprietà (1:N)
 - Un utente può essere proprietario di più numeri di telefono
 - Un numero di telefono è di proprietà di un solo utente

4.4 Diagramma ER ristrutturato



4.5 Descrizione schema relazionale e vincoli di integrità referenziale

Tutti gli attributi sono **non null** a meno che non sia presente il simbolo *. Gli attributi sottolineati costituiscono la chiave primaria della relazione.

- **Utente** (CF, Nome, Cognome, Email, Nascita, TotaleDovuto)

- **Telefono** (Numero, Utente)
 - Telefono.Utente → Utente.CF
- **Patologia** (Nome, Descrizione)
- **Possesso** (Patologia, Utente, Data)
 - Possesso.Patologia → Patologia.Nome
 - Possesso.Utente → Utente.CF
- **Vaccino** (Nome, Descrizione)
- **Eseguito** (Vaccino, Utente, Data)
 - Eseguito.Vaccino → Vaccino.Nome
 - Eseguito.Utente → Utente.CF
- **Reparto** (Nome, Descrizione, Primario*)
 - Reparto.Primario → Medico.ID
- **Medico** (ID, Nome, Cognome, Nascita, Specializzazione, Reparto*)
 - Medico.Reparto → Reparto.Nome
- **Infermiere** (ID, Nome, Cognome, Nascita, Reparto*)
 - Infermiere.Reparto → Reparto.Nome
- **Stanza** (Reparto, Numero, Posti, Piano)
 - Stanza.Reparto → Reparto.Nome
- **Appuntamento** (Utente, Data, Reparto, Stanza)
 - Appuntamento.Utente → Utente.CF
 - Appuntamento.(Reparto, Stanza) → Stanza.(Reparto, Numero)
- **PartecipazioneMedico** (Medico, Utente, Data)
 - PartecipazioneMedico.Medico → Medico.ID
 - PartecipazioneMedico.(Utente, Data) → Appuntamento.(Utente, Data)
- **PartecipazioneInfermiere** (Infermiere, Utente, Data)
 - PartecipazioneInfermiere.Infermiere → Infermiere.ID
 - PartecipazioneInfermiere.(Utente, Data) → Appuntamento.(Utente, Data)
- **Referto** (Utente, Data, Esito, Emissione)
 - Referto.(Utente, Data) → Appuntamento.(Utente, Data)
- **Fattura** (Utente, Data, Emissione, Stato, Ammontare)
 - Fattura.(Utente, Data) → Appuntamento.(Utente, Data)

5 QUERY E INDICI

5.1 Query

- Per ogni medico primario dare il numero di appuntamenti in cui ha partecipato

```

CREATE VIEW medico_primario AS
SELECT id
FROM medico, reparto
WHERE medico.id = reparto.primario

SELECT medico.id, nome, cognome, n_appuntamenti
FROM medico, (SELECT medico_primario.id, COUNT(*) n_appuntamenti
               FROM partecipazionemedico pm, medico_primario
               WHERE pm.medico = medico_primario.id
               GROUP BY medico_primario.id) med_p
WHERE medico.id = med_p.id

```

id	nome	cognome	n_appuntamenti
1	Orland	Capelin	2
2	Johnnie	Ryal	2
3	Crysta	Giovannoni	2
4	Dasi	Blundan	1
(4 rows)			

- Numero di stanze in ogni reparto che possiede un primario

<pre>SELECT nome, COUNT(*) n_stanze FROM reparto, stanza WHERE reparto.nome = stanza.reparto AND reparto.primario IS NOT NULL GROUP BY nome</pre>	<table><tr><th>nome</th><th>n_stanze</th></tr><tr><td>Oncologia</td><td>2</td></tr><tr><td>Pediatria</td><td>2</td></tr><tr><td>Neurologia</td><td>2</td></tr><tr><td>Cardiologia</td><td>4</td></tr><tr><td colspan="2">(4 rows)</td></tr></table>	nome	n_stanze	Oncologia	2	Pediatria	2	Neurologia	2	Cardiologia	4	(4 rows)	
nome	n_stanze												
Oncologia	2												
Pediatria	2												
Neurologia	2												
Cardiologia	4												
(4 rows)													

- Per ogni reparto elencare il totale fatturato

<pre>CREATE VIEW ammontare_reparto AS SELECT stanza.reparto AS rep, SUM(fattura.ammontare) AS tot FROM stanza, appuntamento, fattura WHERE appuntamento.reparto = stanza.reparto AND appuntamento.stanza = stanza.numero AND fattura.utente = appuntamento.utente AND fattura.data = appuntamento.data GROUP BY stanza.reparto SELECT * FROM ammontare_reparto UNION SELECT reparto.nome AS rep, CAST(0 AS money) AS tot FROM reparto WHERE reparto.nome NOT IN (SELECT rep FROM ammontare_reparto) ORDER BY tot DESC</pre>	<table><tr><th>rep</th><th>tot</th></tr><tr><td>Ortopedia</td><td>\$360.00</td></tr><tr><td>Neurologia</td><td>\$220.00</td></tr><tr><td>Oncologia</td><td>\$190.00</td></tr><tr><td>Cardiologia</td><td>\$150.00</td></tr><tr><td>Pediatria</td><td>\$0.00</td></tr><tr><td colspan="2">(5 rows)</td></tr></table>	rep	tot	Ortopedia	\$360.00	Neurologia	\$220.00	Oncologia	\$190.00	Cardiologia	\$150.00	Pediatria	\$0.00	(5 rows)	
rep	tot														
Ortopedia	\$360.00														
Neurologia	\$220.00														
Oncologia	\$190.00														
Cardiologia	\$150.00														
Pediatria	\$0.00														
(5 rows)															

- Per ogni vaccino indicare il numero di utenti nati prima del 1990 che l'ha effettuato

```
SELECT vaccino.nome, count(cf)
FROM (vaccino LEFT JOIN eseguito ON vaccino.nome = eseguito.vaccino)
     LEFT JOIN (SELECT cf FROM utente WHERE nascita < '1990-01-01') utente ON utente.cf = eseguito.utente
GROUP BY vaccino.nome
```

nome	count
HPV	0
MPR	0
Hib	0
Influenza stagionale	1
Pneumococco	0
Meningococco	0
Epatite B	1
Rotavirus	1
DTP	0
Polio	0
(10 rows)	

- Tutti gli utenti che soffrono di 'Demenza Alzheimer' con almeno due fatture da pagare

```
SELECT appuntamento.utente
FROM appuntamento, fattura,
     (SELECT utente FROM possesso WHERE patologia = 'Demenza Alzheimer') utente_alzheimer
WHERE appuntamento.data = fattura.data AND appuntamento.utente = fattura.utente
     AND utente_alzheimer.utente = appuntamento.utente AND fattura.stato = 'Da pagare'
GROUP BY appuntamento.utente
HAVING COUNT(fattura.utente) >= 2
```

utente
GRNTNA75M01D612U
RSSMRA85M01H501Z
(2 rows)

5.2 Indice

È stato aggiunto un indice di tipo hash table sull'attributo **stato** della relazione **fattura** in modo da rendere l'ultima query più efficiente. Il tipo dell'indice è hash table perchè nell'ultima query si fa il controllo `fattura.stato = 'Da pagare'`, di conseguenza essendo un'uguaglianza risulta più efficiente.

6 Discussione dell'applicazione software che accede al DB per effettuare le query

Nella funzione main vengono definiti due array di *char**, query e query_description. Query contiene tutte le query e query_description contiene la descrizione testuale di una query, quindi ad esempio il primo elemento di query_description contiene la descrizione testuale della prima query. Il programma accetta un argomento da riga di comando, ovvero un numero che indica il numero della query che si vuole eseguire, se non viene passato l'argomento allora vengono stampate tutte le query. Es: Stampa la terza query: **./query 3**

Quando si collega al db e quando esegue le query controlla sempre che non ci siano errori, in caso contrario il programma chiude la connessione con il db e termina con exit code 1.