



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO DI MATEMATICA

LAUREA TRIENNALE IN INFORMATICA

PROGETTO DI BASI DI DATI

Basi di Dati di un Sistema Informativo per la Gestione di una Catena di Ospedali

Massimiliano de Leoni, Francesco Vinci

Indice

1	Abstract	2
2	Analisi dei Requisiti	2
3	Progettazione Concettuale	4
4	Progettazione Logica	5
4.1	Analisi delle Ridondanze	5
4.2	Eliminazioni delle Generalizzazioni	8
4.3	Schema Relazionale	10
5	Implementazione in PostgreSQL e Definizione delle Query	11
5.1	Definizione delle Query	11
5.2	Creazione degli indici	14
6	Applicazione Software	14

1 Abstract

Questo lavoro sviluppa una base di dati progettata per gestire in modo strutturato e coerente le informazioni relative agli ospedali, ai loro reparti, ai pazienti ricoverati e agli eventi di parto nei reparti di ostetricia. L'obiettivo principale è fornire un sistema che supporti le attività di monitoraggio e ottimizzazione delle risorse sanitarie, garantendo un accesso organizzato ai dati e facilitando la loro analisi.

Nel contesto della sanità pubblica e privata, la base di dati proposta distingue tra ospedali pubblici e privati, gestendo attributi specifici quali affiliazioni universitarie, gruppi di appartenenza e dati relativi ai dirigenti. I reparti ospedalieri, con particolare attenzione a quelli di ostetricia, sono descritti in termini di capacità, superficie e grado di occupazione. Il sistema tiene traccia dei ricoveri dei pazienti e registra eventi di parto con informazioni rilevanti per l'analisi clinica e organizzativa.

Questa base di dati è progettata per garantire un'archiviazione efficiente e strutturata delle informazioni sanitarie, migliorando il recupero e l'analisi dei dati. L'organizzazione sistematica delle informazioni contribuisce a un utilizzo più efficace delle risorse ospedaliere, ottimizzando la gestione delle strutture sanitarie e supportando i processi decisionali.

2 Analisi dei Requisiti

Questa sezione riassume i requisiti a cui deve sottostare la base di dati.

Ospedali. Ogni ospedale è identificato da un codice univoco e contiene le seguenti informazioni:

- **Codice univoco** che distingue in modo univoco ogni struttura sanitaria.
- **Numero di dipendenti** (se noto).
- **La regione** geografica in cui si trova.

Gli ospedali si suddividono in: **pubblici** e **privati**.

Ospedali Pubblici. Oltre alle informazioni generali, per gli ospedali pubblici vengono registrati ulteriori dati:

- **L'anno di fondazione.**
- **L'eventuale università** affiliata con le seguenti informazioni:
 - **La partita IVA.**

- Il **nome** dell'università.
- L'**indirizzo** del rettorato.
- Il **rettore** attuale con la **data di inizio mandato**.

Ospedali Privati. Per gli ospedali privati, oltre alle informazioni di base, si registra:

- Il **gruppo** di appartenenza.
- I nomi delle **persone dirigenti** attualmente in carica con le seguenti informazioni:
 - La **data di inizio mandato**.
 - Il **ruolo**.

Inoltre, una persona può dirigere più ospedali privati solo se questi appartengono allo stesso gruppo.

Reparti Ospedalieri. I reparti costituiscono le unità operative di ciascun ospedale e sono caratterizzati dai seguenti attributi:

- Un **codice univoco** che identifica il reparto all'interno della struttura ospedaliera di appartenenza.
- La **dimensione** in metri quadrati del reparto.
- Il **numero di pazienti** attualmente ricoverati.

I **reparti di ostetricia**, oltre alle caratteristiche sopra elencate, registrano anche il **numero di posti letto** disponibili e gli **eventi di parto**.

Persone ricoverate. I pazienti ospedalizzati sono identificati attraverso:

- **Codice Fiscale.**
- **Nome.**
- **Cognome.**

Ogni episodio di **ricovero** associa una persona a un reparto specifico per un determinato periodo e include:

- **Data del ricovero.**
- **Posto letto occupato** all'interno del reparto.

Una persona può essere ricoverata più volte nello stesso reparto, ma in momenti distinti, e non può essere ricoverata contemporaneamente in più reparti.

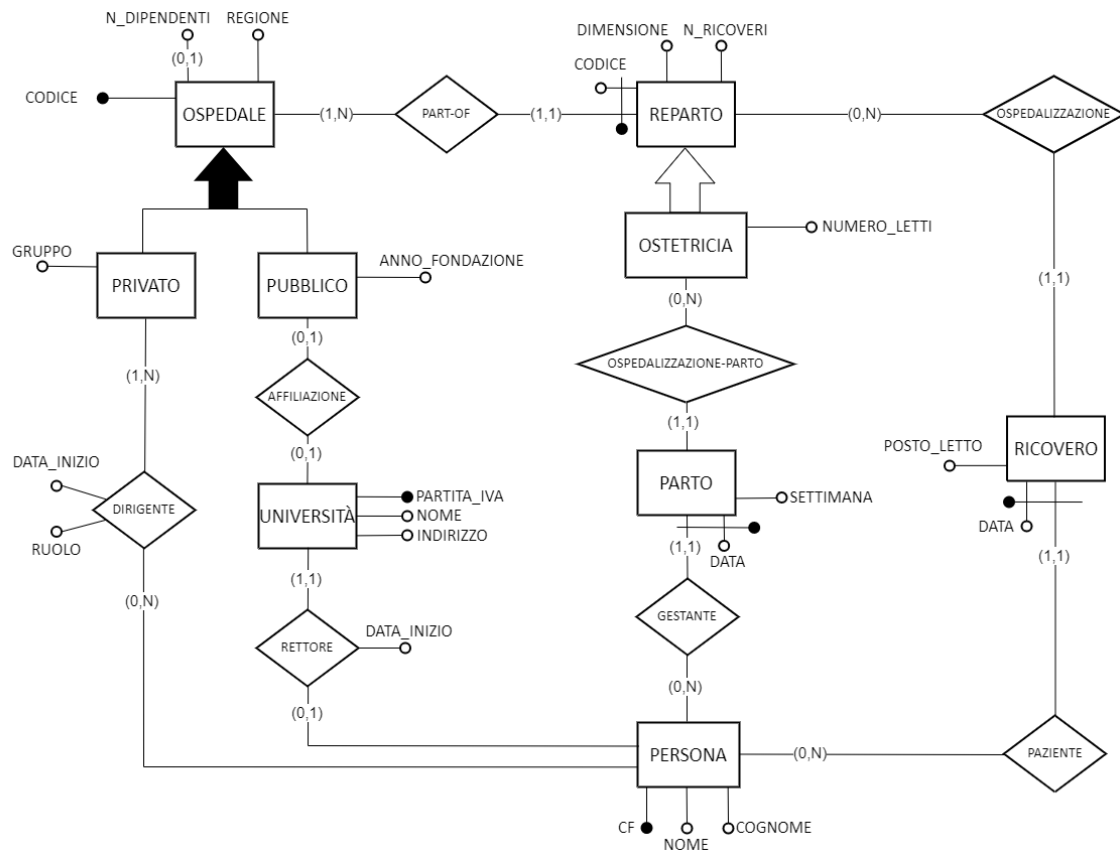


Figura 1: Diagramma E-R della base di dati relativa alla gestione di ospedali.

Eventi di Parto. Gli eventi di parto sono registrati nel sistema con i seguenti dati:

- La **data** del parto.
- La **settimana di gestazione**.
- La **persona gestante**.
- Il **reparto di ostetricia** in cui è avvenuto il parto.

Una stessa gestante può avere più parti nello stesso reparto ma in date diverse.

3 Progettazione Concettuale

Figura 1 mostra il diagramma Entità-Relazione (E-R) che riassume i requisiti descritti in sezione 2.

Esistono esattamente due tipi di ospedali, privati e pubblici. Ogni ospedale privato ha un certo numero di persone come dirigenti, ognuno con un proprio ruolo. Gli ospedali pubblici possono essere parte di un'università, che è guidata da un rettore. Ogni ospedale – sia esso pubblico o privato – è costituito da vari reparti.

L'analisi dei requisiti dà una particolare enfasi ai reparti di ostetricia, il cui concetto è quindi rappresentato come una entità separata, caso particolare del reparto. In particolare, si vuole tenere traccia dei reparti, che avvengono in una certa settimana di gestazione all'interno di un reparto e coinvolgono una persona partorienti.

Ogni ricovero si riferisce ad una persona ed un reparto. Si noti che, essendo ostetricia anche un tipo di reparto, ci possono essere ricoveri anche presso un reparto di ostetricia.

Tabella 1 a pagina 6 riassume tutte le entità e relazioni individuate dalla analisi dei requisiti e rappresentate nel diagramma E-R, con i relativi attributi rilevanti dall'analisi. Per le entità, viene anche fornito l'identificatore, che contiene anche relazioni nei tre casi di *Reparto*, *Parto* e *Ricovero*.

Il presente schema E-R non permette di rappresentare direttamente il seguente vincolo: se una persona pe è dirigente in due ospedali privati pr' e pr'' , allora pr' e pr'' appartengono allo stesso gruppo:

$$(pr', pe) \in \text{Dirigente} \wedge (pr'', pe) \in \text{Dirigente} \Rightarrow (pr'.\text{Gruppo} = pr''.\text{Gruppo})$$

4 Progettazione Logica

In questa sezione viene illustrato il processo di “traduzione” dello schema concettuale in uno schema logico, con l'obiettivo di rappresentare i dati in modo preciso ed efficiente. Il primo passo consiste nell'analizzare le eventuali ridondanze nel modello, al fine di ottimizzare la struttura complessiva. Successivamente, si procede con l'eliminazione delle due generalizzazioni. Infine, viene presentato il diagramma ristrutturato, con una descrizione delle modifiche apportate.

4.1 Analisi delle Ridondanze

L'attributo $N_RICOVERI$ in $REPARTO$, che memorizza il numero di pazienti ricoverati nel reparto presenta una ridondanza. Questo può essere infatti ottenuto contando il numero di ricoveri attivi per quel reparto tramite la relazione $OSPEDALIZZAZIONE$.

Questo attributo viene modificato ogni volta che arriva un nuovo paziente in reparto (circa 100 pazienti nuovi al giorno tra tutti i reparti) e viene visualizzato

Entità	Descrizione	Attributi	Identificatore
Ospedale	Struttura ospedaliera	<i>Codice, N_Dipendenti, Regione</i>	<i>Codice</i>
Privato	Ospedale privato	<i>Gruppo</i>	
Pubblico	Ospedale pubblico	<i>Anno_Fondazione</i>	
Università	Università pubblica	<i>Partita_IVA, Nome, Indirizzo</i>	<i>Partita_IVA</i>
Reparto	Reparto ospedaliero	<i>Codice, Dimensione, N_Ricoveri</i>	<i>Codice, Relazione "Part-Of"</i>
Ostetricia	Reparto di ostetricia	<i>Numero_Letti</i>	
Persona	Paziente	<i>CF, Nome, Cognome</i>	<i>CF</i>
Parto	Evento di parto	<i>Data, Settimana</i>	<i>Data, Relazione "Gestante"</i>
Ricovero	Episodio di ricovero	<i>Data, Posto_Letto</i>	<i>Data, Relazione "Paziente"</i>

(a) Entità

Relazione	Descrizione	Componente	Attributi
Dirigente	Direzione di un ospedale privato	Privato, Persona	<i>Ruolo, Data_Inizio</i>
Affiliazione	Affiliazione tra un'università e un ospedale pubblico	Pubblico, Università	
Rettore	Rettore di un'università	Università, Persona	<i>Data_Inizio</i>
Part-Of	Appartenenza di un reparto a un ospedale	Ospedale, Reparto	
Ospedalizzazione	Ricovero in reparto ospedaliero	Reparto, Ricovero	
Paziente	Persona ricoverata	Persona, Ricovero	
Ospedalizzazione-Parto	Ricovero in reparto di ostetricia	Ostetricia, Parto	
Gestante	Persona gestante	Persona, Parto	

(b) Relazioni

Tabella 1: Dizionario dei dati. Si noti che gli identificatori delle entità Reparto, Parto e Ricovero includono le relazioni con altre entità.

ogni ora del giorno per monitorare il numero di pazienti. Questo si riassume nelle seguenti due operazioni:

- **Operazione 1 (100 al giorno):** memorizza una nuova ospedalizzazione in relativo reparto.
- **Operazione 2 (24 al giorno):** visualizza il numero di ricoveri attuali in un reparto.

Assumendo i seguenti volumi nella base di dati:

Concetto	Costrutto	Volume
REPARTO	E	50
OSPEDALIZZAZIONE	R	20000
RICOVERO	E	20000

la seguente analisi serve per stabilire se sia utile o meno tenere l'attributo ridondante *N_RICOVERI* in REPARTO.

CON RIDONDANZA Analizziamo prima il costo totale con ridondanza.

- Operazione 1:

Concetto	Costrutto	Accessi	Tipo	
RICOVERO	E	1	S	$\times 100$
OSPEDALIZZAZIONE	R	1	S	$\times 100$
REPARTO	E	1	L	$\times 100$
REPARTO	E	1	S	$\times 100$

- Operazione 2:

Concetto	Costrutto	Accessi	Tipo	
REPARTO	E	1	L	$\times 24$

Assumendo costo doppio per gli accessi in scrittura:

$$\text{Costo Totale} = 100 \times 3 \times 2 + 100 + 24 = 724$$

SENZA RIDONDANZA Analizziamo il costo totale senza ridondanza.

- Operazione 1:

Concetto	Costrutto	Accessi	Tipo	
RICOVERO	E	1	S	$\times 100$
OSPEDALIZZAZIONE	R	1	S	$\times 100$

- Operazione 2 (con circa $20000/50 = 400$ ospedalizzazioni al giorno):

Concetto	Costrutto	Accessi	Tipo	
REPARTO	E	1	L	$\times 24$
OSPEDALIZZAZIONE	R	400	L	$\times 24$

Assumendo costo doppio per gli accessi in scrittura:

$$\text{Costo Totale} = 100 \times 2 \times 2 + 400 \times 24 + 24 = 10024$$

L'analisi suggerisce quindi di tenere l'attributo ridondante, ottimizzando così il numero di accessi.

4.2 Eliminazioni delle Generalizzazioni

Le generalizzazioni descritte in Sezione 3 vengono eliminate attraverso una ristrutturazione dello schema concettuale, con l'obiettivo di semplificare la successiva implementazione del modello relazionale e ridurre la presenza di valori nulli. Le modifiche vengono applicate come segue:

OSPEDALE La generalizzazione totale di OSPEDALE viene rimossa includendo le relazioni IS-PRIV e IS-PUB, come mostrato in Figura 2. Questa scelta riduce la presenza di valori nulli: se si mantenesse un'unica entità OSPEDALE con tutti gli attributi relativi sia agli ospedali pubblici che a quelli privati, gli ospedali privati avrebbero campi nulli per informazioni come l'università affiliata e il rettore, mentre gli ospedali pubblici avrebbero campi nulli per gruppo privato di appartenenza. Separando le informazioni con relazioni dedicate, si garantisce che ogni entità contenga solo gli attributi rilevanti. In questa soluzione, coerentemente con la metodologia vista a lezione, gli identificatori di *Privato* e *Pubblico* coincidono con le rispettive relazioni con *Ospedale*. Sarebbe stato possibile rimuovere l'entità *Ospedale*, incorporandone gli attributi nelle entità figlie. La soluzione sarebbe stata del tutto legittima, ma avrebbe causato la necessità di duplicare la relazione *Part-Of* in due relazioni, una verso *Privato* ed una verso *Pubblico*.

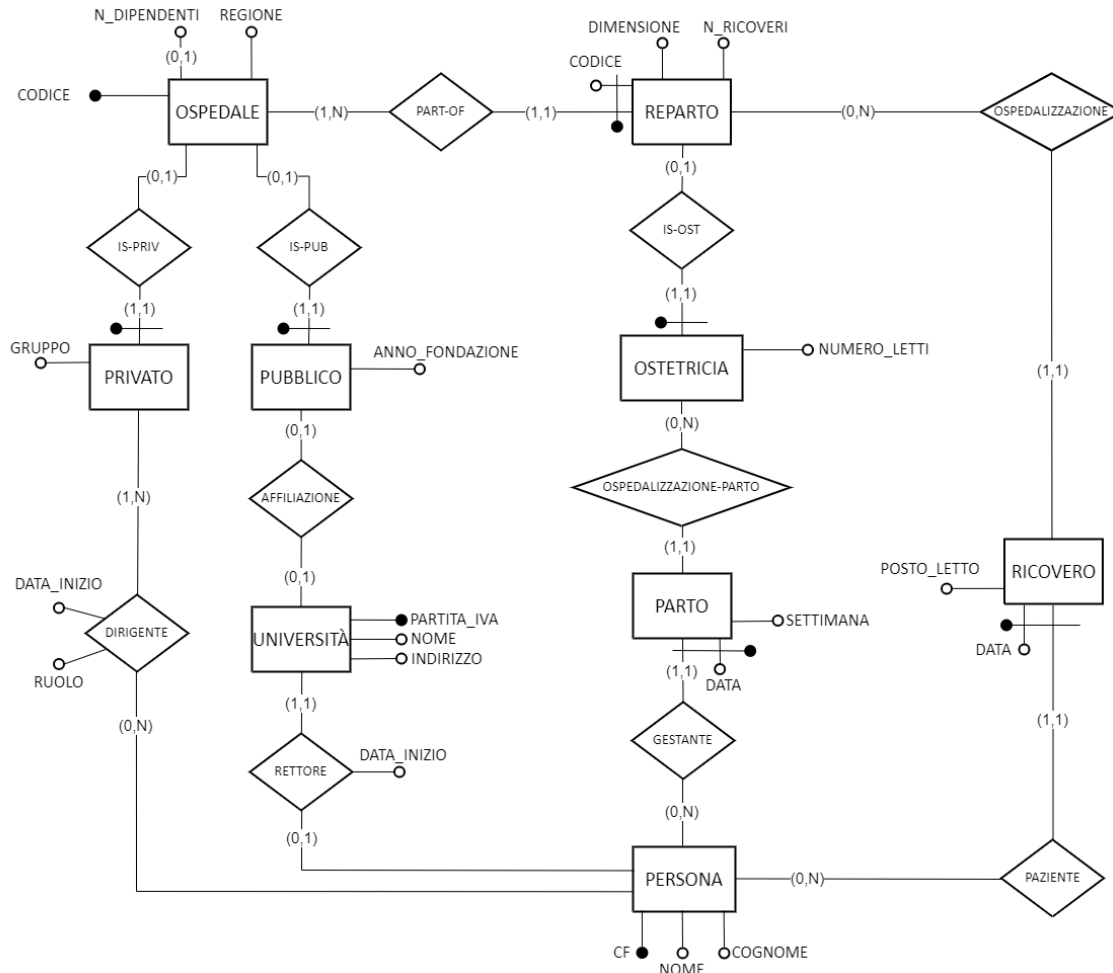


Figura 2: Diagramma E-R ristrutturato.

REPARTO In maniera simile, la generalizzazione parziale REPARTO viene sostituita con la relazione IS-OST (vedi Figura 2). Anche in questo caso, seguendo un ragionamento simile fatto per la generalizzazione precedente, questo cambiamento permette la riduzione di valori nulli. Essendo la generalizzazione parziale (non tutti i reparti sono di ostetricia), la soluzione di eliminare l'entità padre non sarebbe corretta.

Il diagramma E-R ristrutturato, mostrato in Figura 2, riflette queste modifiche rispetto alla versione precedente presentata in Figura 1.

4.3 Schema Relazionale

Lo schema ristrutturato in [Figura 2](#) contiene solamente costrutti mappabili in corrispondenza dello schema relazionale - detto anche schema logico. Lo schema logico è rappresentato a seguire, dove l'asterisco dopo il nome degli attributi indica quelli che ammettono valori nulli.

- **Ospedale**(Codice, N_Dipendenti*, Regione)
- **Privato**(Ospedale, Gruppo)
 - Privato.Ospedale → Ospedale.Codice
- **Pubblico**(Ospedale, Anno_Fondazione, Università_PartitaIVA*)
 - Pubblico.Ospedale → Ospedale.Codice
 - Pubblico.Università_PartitaIVA → Università.Partita IVA
- **Dirigente**(Persona CF, Privato, Ruolo, Data_Inizio)
 - Dirigente.Persona CF → Persona.CF
 - Dirigente.Privato → Privato.Ospedale
- **Università**(Partita IVA, Nome, Indirizzo)
- **Rettore**(Persona CF, Università_PartitaIVA, Data_Inizio)
 - Rettore.Persona CF → Persona.CF
 - Rettore.Università_PartitaIVA → Università.PartitaIVA
- **Persona**(CF, Nome, Cognome)
- **Reparto**(Ospedale, Codice, Dimensione, N_Ricoveri)
 - Reparto.Ospedale → Ospedale.Codice
- **Ostetricia**(Ospedale, Reparto, Numero Letti)
 - Ostetricia.(Ospedale, Reparto) → Reparto.(Ospedale, Codice)
- **Parto**(Persona, Data_Parto, Settimana, Ospedale, Reparto)
 - Parto.Persona → Persona.CF
 - Parto.(Ospedale, Reparto) → Ostetricia.(Ospedale, Reparto)
- **Ricovero**(Persona, Data_Ricovero, Posto Letto, Ospedale, Reparto)
 - Ricovero.Persona → Persona.CF
 - Ricovero.(Ospedale, Reparto) → Reparto.(Ospedale, Codice)

5 Implementazione in PostgreSQL e Definizione delle Query

Il file `ospedali.sql` contiene il codice SQL necessario per la creazione e il popolamento delle tabelle del database. Questo file include inoltre una serie di query per l'estrazione dei dati e un indice creato specificamente per migliorare le prestazioni di una di queste interrogazioni.



5.1 Definizione delle Query

Di seguito vengono presentate e descritte le query con i relativi output generati e viene motivato l'utilizzo dell'indice proposto.

Query 1 Contare il numero di reparti per ciascun ospedale stampandoli in ordine decrescente.

```
01 | SELECT O.Codice AS CodiceOspedale, COUNT(R.Codice) AS
    | NumeroReparti
02 | FROM Ospedale O
03 | JOIN Reparto R ON O.Codice = R.Ospedale
04 | GROUP BY O.Codice
05 | ORDER BY NumeroReparti DESC;
```

Di seguito un'estratto dell'output:

	codiceospedale integer		numeroreparti bigint	
1		75		8
2		4		8
3		2		7
4		11		6
5		9		6
6		5		6
7		14		6
8		7		6
Total rows: 100 of 100		Query complete 00:00:00.251		

Query 2 Trovare tutte le persone, indicandone nome e cognome, che hanno partorito dopo una certa data (ad es. 2024-08-15) e mostrare la regione e la data del parto. Ordinare per data di parto decrescente.

```
01 | SELECT p.Nome, p.Cognome, pa.Data_Parto, o.Regione
02 | FROM Parto pa
```

```

03 | JOIN Persona p ON pa.Persona = p.CF
04 | JOIN Ostetricia os ON pa.Ospedale = os.Ospedale AND pa.
    Reparto = os.Reparto
05 | JOIN Reparto r ON os.Ospedale = r.Ospedale AND os.
    Reparto = r.Codice
06 | JOIN Ospedale o ON r.Ospedale = o.Codice
07 | WHERE pa.Data_Parto > '2024-08-15'
08 | ORDER BY pa.Data_Parto DESC;

```

Estratto dell'output:

	nome character varying (100)	cognome character varying (100)	data_parto date	regione character varying (100)
1	Sofia	Morandi	2025-04-01	Campania
2	Francesca	Ferrara	2025-03-29	Piemonte
3	Michela	Lombardi	2025-03-25	Umbria
4	Lucia	Marinelli	2025-03-24	Sicilia
5	Roberta	Nori	2025-03-20	Lombardia
6	Chiara	Ricci	2025-03-16	Emilia-Romagna
7	Simona	Caputo	2025-03-16	Emilia-Romagna
Total rows: 75 of 75 Query complete 00:00:00.086				

Query 3 Calcolare il numero medio di letti nei reparti di Ostetricia per ciascuna regione, ordinandoli in maniera decrescente.

```

01 | SELECT O.Regione, ROUND(AVG(Os.Numero_Letti),2) AS
    MediaLettiOstetriciaPerRegione
02 | FROM Ospedale O
03 | JOIN Reparto R ON O.Codice = R.Ospedale
04 | JOIN Ostetricia Os ON R.Ospedale = Os.Ospedale AND R.
    Codice = Os.Reparto
05 | GROUP BY O.Regione
06 | ORDER BY MediaLettiOstetriciaPerRegione DESC;

```

Estratto dell'output:

Query 4 Elencare le persone che hanno avuto più di un parto in ordine decrescente indicandone nome e cognome, ordinati anche essi in maniera decrescente, prima per cognome e poi per nome.

```

01 | SELECT P.Nome, P.Cognome, COUNT(Pa.Data_Parto) AS
    NumeroParti
02 | FROM Persona P
03 | JOIN Parto Pa ON P.CF = Pa.Persona
04 | GROUP BY P.CF, P.Nome, P.Cognome
05 | HAVING COUNT(Pa.Data_Parto) > 1

```

	regione character varying (100)	medialettistetriciaiperregione numeric
1	Puglia	19.00
2	Sardegna	18.50
3	Valle d Aosta	17.50
4	Abruzzo	17.00
5	Veneto	16.50
6	Liguria	16.25
7	Lombardia	16.00
8	Friuli Venezia Giulia	15.67
Total rows: 20 of 20		Query complete 00:00:00.089

06 | **ORDER BY** NumeroParti **DESC** , P.Cognome , P.Nome ;

Estratto dell'output:

	nome character varying (100)	cognome character varying (100)	numeroparti bigint
1	Carla	Focaccia	4
2	Michela	Geraci	3
3	Alessia	Montagna	3
4	Fiorenza	Serra	3
5	Simona	Serri	3
6	Stefania	Ametista	2
7	Bianca	Bianchi	2
8	Giovanna	Cammarota	2
Total rows: 24 of 24		Query complete 00:00:00.109	

Query 5 Elencare in ordine decrescente le regioni con la media più alta di ricoveri per reparto, mostrando il numero di reparti, il numero totale di ricoveri e la media per ogni regione.

```

01 | SELECT O.Regione , COUNT(R.Codice) AS Numero_Reparti , SUM
    |      (R.N_Ricoveri) AS Totale_Ricoveri , ROUND(AVG(R.
    |      N_Ricoveri),2) AS Media_Ricoveri_Per_Reparto
02 | FROM Ospedale O JOIN Reparto R ON O.Codice = R.Ospedale
03 | GROUP BY O.Regione
04 | ORDER BY Media_Ricoveri_Per_Reparto DESC ;

```

Estratto dell'output:

	regione character varying (100)	numero_reparti bigint	totale_ricoveri bigint	media_ricoveri_per_reparto numeric
1	Lombardia	22	2960	134.55
2	Veneto	21	2750	130.95
3	Umbria	19	2460	129.47
4	Trentino-Alto Adige	16	2070	129.38
5	Toscana	21	2680	127.62
6	Puglia	21	2670	127.14
7	Calabria	19	2410	126.84
8	Valle d Aosta	17	2140	125.88
Total rows: 20 of 20		Query complete 00:00:00.084		

5.2 Creazione degli indici

Si suppone di voler ottimizzare la query 1, per la quale occorre considerare:

1. Condizione di Join: $O.Codice = R.Ospedale$;
2. Group by sulla colonna $O.Codice$ e aggregazione $COUNT(R.Codice)$
3. Ordinamento su attributo $NumeroReparti$

Per il punto 1, è opportuno creare due indici hash su entrambe le colonne del join:

```
01 | CREATE INDEX idx_reparto ON Reparto USING HASH (Ospedale);
02 | CREATE INDEX idx_ospedale ON Ospedale USING HASH (Codice);
```

Questo permette di fare il join usando i codice hash, che è estremamente efficiente per gli equi-join.

Il punto 2 richiede di raggruppare e contare le tuple con lo stesso valore di $R.Ospedale$ e potrebbe quindi essere ottimizzato tramite il seguente indice B+ Tree:

```
01 | CREATE INDEX idx_reparto_2 ON Reparto (Ospedale);
```

Nella pratica, non è necessario tuttavia creare tale indice perché, come anche altri sistemi, PostgreSQL automaticamente crea un indice B+ Tree per ogni chiave primaria; in particolare, quindi, esiste già un indice B+ Tree su $Ospedale$ e $Codice$, che mantiene un ordinamento delle tuple per lo stesso Ospedale.

6 Applicazione Software

Non viene fornito il codice C dell'applicazione poiché la maggior parte degli studenti in passato hanno fatto plagio del codice fornito, senza produrre una soluzione originale.

Il file `query.c` contiene il codice C necessario per connettersi al database e visualizzare a schermo i risultati delle query presentate nella Sezione 5. Una volta compilato ed eseguito il programma, verrà mostrato all'utente un menu con la seguente lista di interrogazioni disponibili:

1. Numero di reparti per ospedale.
2. Persone che hanno partorito dopo una certa data.
3. Numero medio letti in reparti di ostetricia per regione.
4. Persone con più di un parto.
5. Regioni con media più alta di ricoveri per reparto.

L'utente potrà selezionare la query da eseguire digitando il numero corrispondente. In particolare, la query 2 è stata resa parametrica: al momento della selezione, il programma richiederà l'inserimento della data a partire dalla quale visualizzare i parti registrati.