

UNIVERSITÀ DEGLI STUDI DI URBINO CARLO BO

Corso di Laurea in Informatica Applicata

Anno Accademico 2024/2025

**Costruzione di un database relazionale per l'analisi dei
servizi sanitari territoriali delle ASL**

Relazione per il Progetto del Corso di Basi di Dati

Milena Balducci, matricola 321791

Indice

1	Introduzione	2
2	Progettazione Concettuale	3
2.1	Raccolta e Analisi dei Requisiti	3
2.2	Diagramma E-R	4
2.3	Dizionario delle entità	4
2.4	Dizionario delle relazioni	7
3	Progettazione Logica	8
3.1	Analisi delle ridondanze	8
3.2	Traduzione nel modello logico	9
3.3	Normalizzazione	10
3.3.1	Verifica Forma Normale di Boyce-Codd	10
3.3.2	Verifica Terza Forma Normale	11
4	Progettazione Fisica	12
4.1	Scelta del DBMS e Ambiente	12
4.2	Definizione dei Tipi di Dati	12
4.3	Definizione delle Chiavi	12
4.3.1	Chiavi Primarie (PK)	12
4.3.2	Chiavi Esterne (FK)	13
4.4	Indicizzazione Secondaria	13
4.5	Scelta del Motore di Archiviazione	13
4.5.1	Set di Caratteri e Collation	14
4.6	Esempio di Mappatura Tabella: ASL	14
4.7	Utilizzo	15
5	Query di esempio	16
6	Conclusioni	19
7	Fonti dataset	20

1 Introduzione

L'obiettivo primario di questo progetto è la realizzazione di un sistema di database per la memorizzazione e la gestione delle informazioni relative alle **Aziende Sanitarie Locali (ASL)** distribuite sul territorio nazionale. La base di partenza per questa implementazione è costituita da un dataset open data reso pubblicamente disponibile dal Ministero della Salute.

L'idea centrale è di progettare una struttura dati che non solo faciliti la consultazione intuitiva e rapida delle informazioni esistenti, ma che sia anche flessibile e scalabile. Questa flessibilità è cruciale per permettere future espansioni del modello dati, consentendo l'integrazione di nuove tipologie di informazioni non presenti nel dataset originale che potrebbero arricchire ulteriormente l'analisi del panorama sanitario italiano.

Questo approccio apre anche la strada a possibili applicazioni che vanno dalla mappatura dei servizi sanitari alla valutazione della loro distribuzione geografica e alla pianificazione strategica futura.

L'impiego di principi di normalizzazione e l'attenzione alla chiarezza dello schema relazionale sono stati i pilastri di questo progetto, mirando a garantire l'integrità dei dati e a semplificare le operazioni di query complesse. Questo documento illustrerà le fasi di progettazione, implementazione e verifica di tale sistema, evidenziando le scelte architetturali e le metodologie adottate per raggiungere gli obiettivi prefissati.

2 Progettazione Concettuale

2.1 Raccolta e Analisi dei Requisiti

Normalmente, la progettazione concettuale parte dalla definizione di un "problema" che deve essere risolto attraverso l'implementazione di un database. In questo caso però si ha già un dataset di partenza, quindi l'obiettivo è analizzarlo per capire come strutturare la base di dati correttamente a partire da esso.

Il dataset originale presenta i seguenti campi:

anno_riferimento	servizio_trasporto_dialisi
codice_regione	servizio_adi
codice_asl	unità_mobile_rianimazione
regione	ambulanze_emergenza_neonato
denominazione_asl	medici
indirizzo_asl	pedici_indennità_associativa
cap_asl	totale_scelte
comune_asl	pediatri
sigla_provincia_asl	pediatri_indennità_associativa
residenti_età_infantile	totale_scelte_dei_pediatri
residenti_età_adulta	ambulatori_laboratori
residenti_anziani	ambulatori_laboratori_convenzionati
residenti_totale	numero_ricette
cup_tipo.1	importo_ricette_euro
cup_tipo.2	ADI_casi_trattati
dipartimento_prevenzione	ADI_casi_trattati_Anziani
dipartimento_materno_infantile	ADI_casi_trattati_Pazienti_Terminali
dipartimento_salute_mentale	

E' possibile raggruppare questi campi in dei macro gruppi: regione, provincia, ASL, servizi, strutture, personale e prescrizioni.

Regione e provincia raggruppano i rispettivi dati di base, ovvero il codice/sigla e il nome. Le **ASL** raggruppano i dati di base relativi a ciascuna azienda sanitaria locale, ovvero i dati geografici relativi alla sede principale dell'azienda (indirizzo, CAP, comune), la provincia e la regione a cui fa afferenza, la denominazione, il codice.

Per la **popolazione** servita da ciascuna ASL interessano il numero di residenti in età infantile, in età adulta e anziani, e il totale dei residenti.

Per i **servizi** offerti è invece necessario tenere traccia della denominazione di ciascun servizio e della quantità di servizi offerti per ciascuna denominazione in ciascuna ASL.

Per il **personale medico** interessano il numero dei medici per ciascuna categoria, presenti in ciascuna ASL. Da notare che i campi `totale_scelte` e `totale_scelte_pediatri` non rappresentano la somma degli altri campi.

Per le **infrastrutture**, come per i servizi, interessano i nominativi dei vari tipi di servizio e il totale per ciascun nominativo.

Per le **prescrizioni** interessano il numero e l'importo delle ricette emesse da ciascuna ASL.

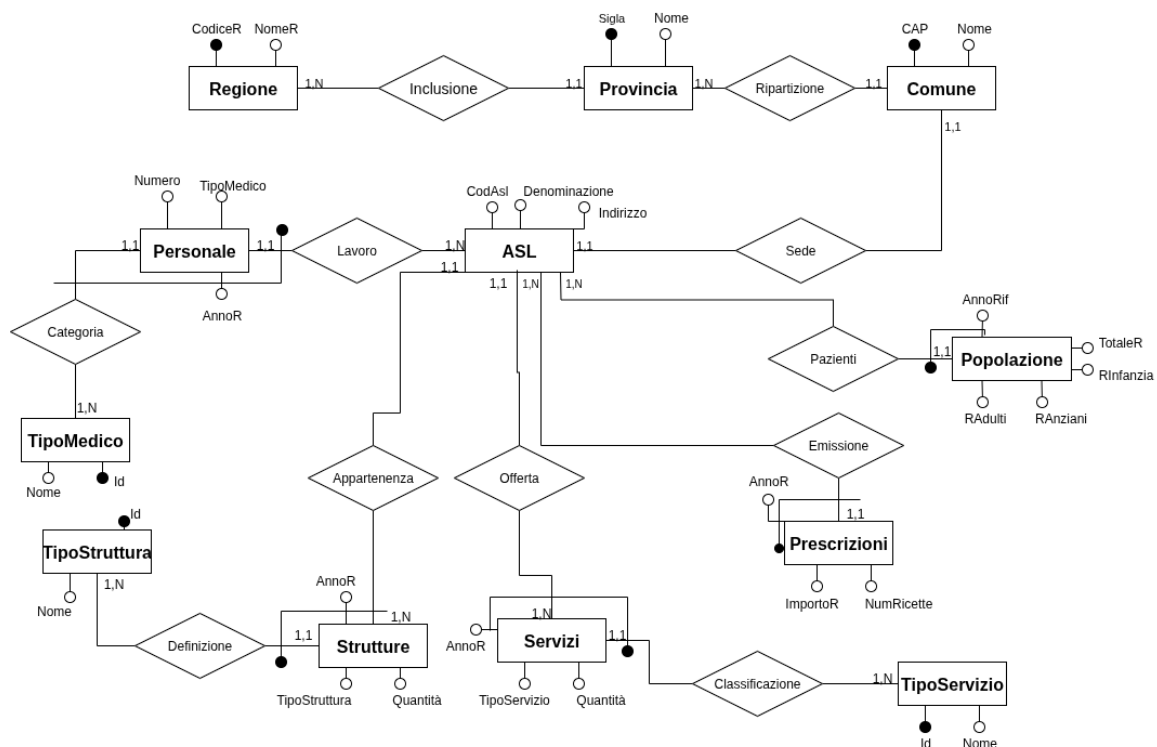
2.2 Diagramma E-R

Dall'analisi dei requisiti effettuata al paragrafo precedente, si individuano facilmente le seguenti entità: ASL, Regione, Provincia, Popolazione, Servizi, Personale, Strutture, Prescrizioni.

Si individua inoltre l'entità Comune, che identifica il comune in cui si trova la sede principale di ciascuna ASL.

Le entità Servizi, Personale e Strutture possono essere modellate in due modi: assegnando loro un attributo per ogni colonna del dataset di partenza che fa parte della categoria rappresentata dall'entità, oppure creando tre entità separate che rappresentino, per ciascuna entità, il tipo di dato rappresentato in ciascuna riga. Si è scelto di optare per la seconda strategia, creando così le entità TipoMedico, TipoStruttura e TipoServizio. Questo perché procedere in questo modo permette, in futuro, di espandere più facilmente le categorie di dati di cui si vuole tenere traccia, in quanto sarebbe sufficiente aggiungere righe nelle tabelle di lookup.

Di seguito è riportato il diagramma E-R:



2.3 Dizionario delle entità

Nome: Regione

Descrizione: Dati identificativi delle regioni italiane

Attributi:

- CodR: intero. Codice univoco che identifica la regione.
- Nome: stringa. Il nome della regione.

Nome: Provincia

Descrizione: Dati identificativi delle province di ciascuna regione

Attributi:

- SiglaP: stringa. Codice univoco che identifica la provincia.
- Nome: stringa. Il nome della provincia.

Nome: Comune

Descrizione: Dati identificativi dei comuni che ospitano le sedi ASL

Attributi:

- CAP: intero. Codice univoco che identifica il comune.
- Nome: stringa. Il nome del comune.

Nome: ASL

Descrizione: Dati identificativi delle ASL

Attributi:

- CodAsl: intero. Il codice dell'ASL. Non è univoco: ASL diverse in regioni diverse possono avere lo stesso codice.
- Denominazione: stringa. Il nome dell'ASL.
- Indirizzo: stringa.

Nome: Popolazione

Descrizione: Dati relativi al numero di residenti nell'area di competenza di ciascuna ASL

Attributi:

- AnnoR: intero. L'anno al quale fanno riferimento i dati.
- RAdulti: intero. Il numero di residenti adulti.
- RInfanzia: intero. Il numero di residenti in età infantile.
- RAnziani: intero. Il numero di residenti anziani.
- TotaleR: intero. Il numero totale dei residenti.

Nome: Prescrizioni

Descrizione: Dati relativi alle spese sostenute dall'ASL per le prescrizioni

Attributi:

- AnnoR: intero. L'anno al quale fanno riferimento i dati.
- ImportoR: decimale. L'importo complessivo delle ricette emesse dall'ASL in un anno specifico.
- NumeroRicette: intero. Il numero delle ricette emesse.

Nome: Servizi

Descrizione: Dati relativi ai servizi offerti dalle ASL

Attributi:

- AnnoR: intero. L'anno al quale fanno riferimento i dati.
- TipoServizio: intero. L'id del servizio offerto.
- Quantità: intero. Il numero di servizi con un determinato id disponibili nell'area di competenza dell'ASL.

Nome: Strutture

Descrizione: Dati relativi alle strutture sanitarie presenti nel territorio delle ASL

Attributi:

- AnnoR: intero. L'anno al quale fanno riferimento i dati.
- TipoStruttura: intero. L'id del tipo di struttura.
- Quantità: intero. Il numero di strutture con un determinato id.

Nome: Personale

Descrizione: Dati relativi ai medici che lavorano nell'ASL

Attributi:

- Numero: intero. Il numero di medici di un determinato tipo.
- TipoMedico: intero. Id della tipologia di medico.
- AnnoR: intero. L'anno al quale fanno riferimento i dati.

Nome: TipoServizio

Descrizione: Tipologie di servizi che le ASL possono offrire

Attributi:

- Id: intero.
- Nome: stringa

Nome: TipoStruttura

Descrizione: Tipologie di strutture che è possibile trovare sul territorio

Attributi:

- Id: intero
- Nome: stringa

Nome: TipoMedico

Descrizione: Tipologie di personale medico

Attributi:

- Id: intero
- Nome: stringa

2.4 Dizionario delle relazioni

Nome	Entità Coinvolte	Descrizione	Cardinalità	Attributi
Inclusione	Regione, Provincia	Indica a quale Regione appartiene ogni Provincia	1:N	-
Ripartizione	Provincia, Comune	Specifica in quale Provincia si trova ciascun Comune	1:N	-
Sede	ASL, Comune	Collega ogni ASL al Comune in cui ha la propria sede principale o operativa	1:N	-
Pazienti	ASL, Popolazione	Quantifica la Popolazione residente di una specifica ASL	1:N	-
Lavoro	ASL, Personale	Quantifica il Personale sanitario delle diverse ASL	1:N	-
Emissione	ASL, Prescrizioni	Quantifica per ogni ASL le Prescrizioni mediche emesse	1:N	-
Categoria	Personale, TipoMedico	Classifica il Personale	1:N	-
Appartenenza	ASL, Strutture	Stabilisce quali Strutture sanitarie sono gestite o afferiscono a una determinata ASL	1:N	-
Offerta	ASL, Servizi	Indica quali Servizi sanitari sono offerti da ciascuna ASL	1:N	-
Definizione	Strutture, TipoStruttura	Classifica le strutture	1:N	-
Classificazione	Servizi, TipoServizio	Classifica i servizi	1:N	-

3 Progettazione Logica

3.1 Analisi delle ridondanze

Nello schema proposto, l'attributo **RTotali** nella tabella **Popolazione** rappresenta una potenziale ridondanza, in quanto il suo valore può essere derivato dalla somma di **RAdulti**, **RInfanzia**, e **RAnziani**. La decisione di mantenere o rimuovere questo attributo è stata valutata considerando l'impatto su performance e integrità dei dati.

Sono stati messi a confronto due scenari principali:

Scenario A: **RTotali** rimosso

- **Vantaggi:** Riduce la ridondanza, elimina le anomalie di aggiornamento, occupa meno spazio e semplifica le operazioni di scrittura.
- **Svantaggi:** Ogni query che richiede **RTotali** deve calcolarlo al momento, comportando un leggero overhead della CPU.

Scenario B: **RTotali** mantenuto

- **Vantaggi:** Riduce il costo di lettura per le query che necessitano di **RTotali** e semplifica la scrittura delle query.
- **Svantaggi:** Introduce ridondanza, aumenta lo spazio di archiviazione e la complessità/costo delle operazioni di scrittura (per mantenere la coerenza).

Sono stati presi in considerazione anche i potenziali casi d'uso e il volume previsto dei dati.

1. **Visualizzazione dati popolazione su dashboard (Lettura, Alta Frequenza):** Richiede **RTotali** per riepiloghi rapidi. La criticità è alta.
2. **Aggiornamento annuale dati popolazione (Scrittura, Bassa Frequenza):** Inserimenti/aggiornamenti occasionali che coinvolgono tutti gli attributi. La criticità è media.

Considerando circa 107 ASL, e ipotizzando di conservare dati per 30 anni, la tabella **Popolazione** conterrebbe circa 3.210 righe. Per un database, questo è un volume di dati ridotto. Su tabelle di queste dimensioni, il guadagno di performance derivante dal mantenere **RTotali** precalcolato è minimo. Al contrario, il rischio di inconsistenza dei dati (dove **RTotali** non corrisponde alla somma delle sue parti) è una preoccupazione costante se l'attributo viene mantenuto.

Dato il volume limitato di dati e la bassa frequenza degli aggiornamenti, il costo di ricalcolare **RTotali** in fase di lettura è trascurabile. Mantenerlo introdurrebbe un rischio di inconsistenza senza un significativo beneficio prestazionale. Pertanto, si è deciso di rimuovere l'attributo dalla tabella per garantire una maggiore integrità e semplicità del database.

3.2 Traduzione nel modello logico

Le relazioni presenti tra le entità del modello E-R sono tutte del tipo **uno-a-molti**, e sono state tradotte nel modello logico accorpendo la relazione nell'entità con cardinalità massima 1 e cardinalità minima 1, ovvero includendo la chiave primaria dell'entità con cardinalità uno come chiave esterna nell'entità "molti".

Il risultato di tale mappatura è il seguente schema relazionale:

- Regione(CodR, Nome)
- Provincia(CodR, SiglaP, Nome)
- Comune(SiglaP, CAP, Nome)
- TipoMedico(Id, Nome)
- Personale(CodAsl, CAP, AnnoR, TipoMedico, Numero)
- TipoStruttura(Id, Nome)
- Strutture(CodAsl, CAP, AnnoR, TipoStruttura, Quantità)
- TipoServizio(Id, Nome)
- Servizi(CodAsl, CAP, AnnoR, TipoServizio, Quantità)
- ASL(CodAsl, CAP, NomeASL, Indirizzo)
- Popolazione(CodAsl, CAP, AnnoR, RAdulti, RInfanzia, RAnziani)
- Prescrizioni(CodAsl, CAP, AnnoR, NumeroRicette, ImportoR)

Vincoli di integrità referenziale:

- Provincia.CodR REFERENCES Regione(CodR)
- Comune.SiglaP REFERENCES Provincia(SiglaP)
- ASL.CAP REFERENCES Comune(CAP)
- Personale(CodAsl, CAP) REFERENCES ASL(CodAsl, CAP)
- Personale.TipoMedico REFERENCES TipoMedico(Id)
- Strutture(CodAsl, CAP) REFERENCES ASL(CodAsl, CAP)
- Strutture.TipoStruttura REFERENCES TipoStruttura(Id)
- Servizi(CodAsl, CAP) REFERENCES ASL(CodAsl, CAP)
- Servizi.TipoServizio REFERENCES TipoServizio(Id)
- Popolazione(CodAsl, CAP) REFERENCES ASL(CodAsl, CAP)
- Prescrizioni(CodAsl, CAP) REFERENCES ASL(CodAsl, CAP)

3.3 Normalizzazione

3.3.1 Verifica Forma Normale di Boyce-Codd

Per stabilire se una relazione è in forma normale di Boyce-Codd occorre verificare se ogni attributo o set di attributi da cui dipende funzionalmente un altro attributo è una superchiave.

1. **Regione(CodR, Nome)**
 - PK:** CodR
 - DF:** CodR \rightarrow Nome
 - Risultato:** Già in BCNF.
2. **Provincia(CodR, SiglaP, Nome)**
 - PK:** SiglaP
 - DF:** SiglaP \rightarrow Nome, SiglaP \rightarrow CodR
 - Risultato:** Già in BCNF.
3. **Comune(SiglaP, CAP, Nome)**
 - PK:** CAP
 - DF:** CAP \rightarrow Nome, CAP \rightarrow SiglaP
 - Risultato:** Già in BCNF.
4. **TipoMedico(Id, Nome), TipoStruttura(Id, Nome), TipoServizio(Id, Nome)**
 - PK:** Id
 - DF:** Id \rightarrow Nome
 - Risultato:** Già in BCNF per tutte e tre.
5. **ASL(CodAsl, CAP, NomeASL, Indirizzo)**
 - PK:** (CodAsl, CAP)
 - DF:** (CodAsl, CAP) \rightarrow NomeASL, (CodAsl, CAP) \rightarrow Indirizzo
 - Vincolo Referenziale:** ASL.CAP REFERENCES Comune
 - Analisi:** Tutti gli attributi non-chiave (NomeASL, Indirizzo) dipendono completamente dalla chiave primaria composta (CodAsl, CAP). Non ci sono dipendenze parziali o transitive.
 - Risultato:** Già in BCNF.
6. **Personale(CodAsl, CAP, AnnoR, TipoMedico, Quantità)**
 - PK:** (CodAsl, CAP, AnnoR, TipoMedico)
 - DF:** (CodAsl, CAP, AnnoR, TipoMedico) \rightarrow Quantità
 - Vincolo Referenziale:** (CodAsl, CAP) REFERENCES ASL, TipoMedico REFERENCES TipoMedico.
 - Analisi:** L'attributo non-chiave Numero dipende completamente dalla chiave primaria composta.
 - Risultato:** Già in BCNF.
7. **Strutture(CodAsl, CAP, AnnoR, TipoStruttura, Quantità)**
 - PK:** (CodAsl, CAP, AnnoR, TipoStruttura)
 - DF:** (CodAsl, CAP, AnnoR, TipoStruttura) \rightarrow Quantità
 - Vincolo Referenziale:** (CodAsl, CAP) REFERENCES ASL, TipoStruttura REFERENCES TipoStruttura.

-Analisi: Simile a Personale. L'attributo non-chiave Quantità dipende completamente dalla chiave primaria composta.

-Risultato: Già in BCNF.

8. **Servizi(CodAsl, CAP, AnnoR, TipoServizio, Quantità)**

-PK: (CodAsl, CAP, AnnoR, TipoServizio)

-DF: (CodAsl, CAP, AnnoR, TipoServizio) ->Quantità

-Vincolo Referenziale: (CodAsl, CAP) REFERENCES ASL, TipoServizio REFERENCES TipoServizio.

-Analisi: Simile a Personale e Strutture. L'attributo non-chiave Quantità dipende completamente dalla chiave primaria composta.

-Risultato: Già in BCNF.

9. **Popolazione(CodAsl, CAP, AnnoR, RAdulti, RInfanzia, RAnziani)**

-PK: (CodAsl, CAP, AnnoR)

-DF: (CodAsl, CAP, AnnoR) ->RAdulti, (CodAsl, CAP, AnnoR) ->RInfanzia, (CodAsl, CAP, AnnoR) ->RAnziani

-Analisi: Tutti gli attributi non-chiave dipendono completamente dalla chiave primaria composta.

-Risultato: Già in BCNF.

10. **Prescrizioni(CodAsl, CAP, AnnoR, NumeroRicette, ImportoR)**

-PK: (CodAsl, CAP, AnnoR)

-DF: (CodAsl, CAP, AnnoR) ->NumeroRicette, (CodAsl, CAP, AnnoR) ->ImportoR

-Vincolo Referenziale: (CodAsl, CAP) REFERENCES ASL.

-Analisi: Tutti gli attributi non-chiave dipendono completamente dalla chiave primaria composta.

-Risultato: Già in BCNF.

3.3.2 Verifica Terza Forma Normale

Se una relazione è in BCNF (Boyce-Codd Normal Form), allora è automaticamente anche in 3NF. Quindi, per le tabelle che abbiamo già verificato essere in BCNF, possiamo semplicemente confermare che sono anche in 3NF.

4 Progettazione Fisica

4.1 Scelta del DBMS e Ambiente

Per la gestione dei dati, è stato scelto **MySQL 8.0** come Database Management System. La decisione di optare per MySQL è dettata dalla sua natura open source, dall'abbondanza di risorse e documentazione disponibili e, non ultimo, dalla facile disponibilità sulla piattaforma di deployment scelta.

Per quanto riguarda l'ambiente di deployment, il database MySQL è stato configurato per girare all'interno di un container **Docker**. Questo approccio offre numerosi vantaggi in termini di portabilità, isolamento delle dipendenze e facilità di gestione, sia in un contesto di sviluppo locale che su un server dedicato o un servizio cloud.

4.2 Definizione dei Tipi di Dati

Per garantire coerenza e performance, sono state adottate le seguenti linee guida generali nella scelta dei tipi di dati per le tabelle:

Stringhe: si è privilegiato l'utilizzo di **VARCHAR** per i campi testuali di lunghezza variabile. La dimensione massima (**VARCHAR(N)**) è stata specificata per riflettere la lunghezza massima attesa dei dati, bilanciando l'occupazione di spazio e la flessibilità. Per campi con una lunghezza fissa e molto ridotta, come le sigle delle province, si è optato per **CHAR(N)** per un'ottimizzazione leggera dello spazio e una gestione più efficiente delle stringhe di lunghezza costante.

Numeri Interi: il tipo **INT** è stato scelto per la maggior parte dei contatori, degli identificativi numerici (ID) e per campi che rappresentano anni o quantità che rientrano nel range standard degli interi.

Numeri Decimali: per valori monetari, percentuali o qualsiasi quantità che richieda una precisione esatta e non sia soggetta a problemi di arrotondamento floating-point, è stato impiegato il tipo **DECIMAL(P, S)**.

4.3 Definizione delle Chiavi

4.3.1 Chiavi Primarie (PK)

Le chiavi primarie sono state definite per identificare in modo univoco ogni record all'interno di una tabella. A seconda della struttura dell'entità, sono state utilizzate sia chiavi primarie semplici che chiavi composte.

Un aspetto cruciale delle chiavi primarie è che MySQL (e la maggior parte dei DBMS relazionali) crea automaticamente un indice b-tree sulla chiave primaria. Questa indicizzazione automatica è vitale per l'efficienza delle operazioni di ricerca, ordinamento e join, permettendo al database di localizzare rapidamente i record.

4.3.2 Chiavi Esterne (FK)

Le regole di integrità referenziale (ON DELETE e ON UPDATE) sono state definite per governare il comportamento del database quando i dati nella tabella genitore vengono modificati o eliminati, assicurando che le relazioni rimangano coerenti. Le opzioni principali considerate sono state:

- **ON UPDATE CASCADE:** Se un record genitore viene modificato, tutti i record figli correlati vengono automaticamente modificati.
- **ON DELETE NO ACTION:** impedisce l'eliminazione di un record genitore se esistono record figli correlati, garantendo così la massima sicurezza contro eliminazioni non volute. Questa è spesso l'impostazione predefinita e la scelta più sicura.

4.4 Indicizzazione Secondaria

Oltre agli indici creati automaticamente tramite chiavi primarie e vincoli di chiave esterna, sono stati aggiunti indici secondari per ottimizzare le query più comuni, in particolare su colonne frequentemente usate nei filtri e nei join. Ad esempio:

- Strutture(AnnoR), Strutture(TipoStruttura)
- Personale(AnnoR), Personale(TipoMedico)
- Servizi(AnnoR), Servizi(TipoServizio)
- Prescrizioni(AnnoR), Popolazione(AnnoR)
- ASL(NomeASL) per ricerche testuali

Questi indici permettono di migliorare la performance delle interrogazioni, in particolare quelle con filtri per anno di riferimento, tipo di struttura o categoria professionale. La scelta degli indici è stata guidata dall'analisi delle query previste nel progetto.

4.5 Scelta del Motore di Archiviazione

Per questo progetto, il motore di archiviazione **InnoDB** è stato scelto come default per tutte le tabelle. InnoDB è il motore predefinito e generalmente raccomandato per MySQL 8.x, e la sua adozione è motivata da diverse caratteristiche cruciali:

- **Supporto ACID:** InnoDB garantisce le proprietà ACID (Atomicità, Consistenza, Isolamento, Durabilità) per le transazioni. Questo significa che le operazioni sul database sono affidabili e che i dati rimangono integri anche in caso di errori o crash del sistema.
- **Integrità Referenziale:** È l'unico motore di archiviazione ampiamente utilizzato in MySQL che supporta nativamente le chiavi esterne (Foreign Keys), permettendo di definire e far rispettare le relazioni tra le tabelle, come discusso nella sezione precedente.

- **Row-Level Locking:** A differenza di motori che bloccano intere tabelle, InnoDB implementa il blocco a livello di riga (row-level locking). Questo migliora significativamente le prestazioni in ambienti ad alta concorrenza, poiché più utenti possono accedere e modificare righe diverse della stessa tabella contemporaneamente senza bloccarsi a vicenda.
- **Recupero in Caso di Crash:** InnoDB include meccanismi avanzati per il recupero in caso di crash, garantendo che il database possa essere ripristinato a uno stato coerente dopo un'interruzione imprevista.

L'utilizzo di InnoDB è specificato direttamente nella clausola 'CREATE TABLE' di ogni tabella, tramite la clausola 'ENGINE=InnoDB'.

4.5.1 Set di Caratteri e Collation

La corretta configurazione del set di caratteri e della collation è cruciale per la gestione dei dati testuali, assicurando che tutti i caratteri siano visualizzati correttamente e che le operazioni di ordinamento e confronto delle stringhe avvengano come previsto.

Il set di caratteri '**utf8mb4**' è stato scelto come impostazione predefinita per l'intero database e per tutte le tabelle. A differenza del più limitato 'utf8' (che supporta solo i primi 3 byte di un carattere UTF-8, escludendo molti caratteri speciali e emoji), 'utf8mb4' supporta l'intero standard UTF-8 (fino a 4 byte per carattere), garantendo la massima compatibilità e prevenendo problemi di visualizzazione dei caratteri.

La **collation** definisce le regole per l'ordinamento e il confronto delle stringhe all'interno del database. Per questo progetto, la collation raccomandata è '**utf8mb4_unicode_ci**'. '_unicode_ci' indica che l'ordinamento e il confronto avvengono secondo le regole Unicode, garantendo una gestione più accurata e standardizzata delle stringhe in diverse lingue. Il suffisso '_ci' sta per "case-insensitive".

Per le installazioni di MySQL 8.0 e versioni successive, la collation '**utf8mb4_0900_ai_ci**' è spesso la scelta preferita e predefinita. Questa collation è basata sull'algoritmo Unicode Collation Algorithm (UCA) 9.0 e offre un'ulteriore miglioria rispetto a 'utf8mb4_unicode_ci', essendo sia "accent-insensitive" che "case-insensitive", il che può essere vantaggioso per la ricerca e l'ordinamento di dati in lingua italiana.

4.6 Esempio di Mappatura Tabella: ASL

A titolo esemplificativo, di seguito è riportata la sintassi SQL CREATE TABLE per l'entità ASL, che dimostra l'applicazione delle regole di cui sopra e le scelte specifiche dei tipi di dati:

```
CREATE TABLE IF NOT EXISTS ASL (
    CodAsl INT,
    CAP INT,
    NomeASL VARCHAR(255) NOT NULL,
    Indirizzo VARCHAR(255) NOT NULL,
    PRIMARY KEY (CodAsl, CAP),
```

```
FOREIGN KEY (CAP) REFERENCES Comune(CAP)
ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=
utf8mb4_0900_ai_ci;
```

4.7 Utilizzo

Il database può essere utilizzato direttamente da terminale. Per prima cosa è necessario avviare il container Docker con il comando:

```
docker compose up --build
```

Al primo avvio l'operazione potrebbe richiedere del tempo.

Una volta avviato il database, è possibile accedervi tramite:

```
mysql -u user -p -h 127.0.0.1 -P 3306 -t servizi_strutture_asl
```

In alternativa, è possibile eseguire direttamente uno script SQL in un'unica soluzione, ad esempio:

```
mysql -u user -p -h 127.0.0.1 -P 3306 -t servizi_strutture_asl
< query/tipo_strutture_asl.sql
```

In entrambi i casi, verrà richiesto di inserire la password.

5 Query di esempio

1. Trovare tutte le ASL della provincia di Bologna (BO)

Questa query unisce le tabelle ASL e Comune per filtrare le ASL in base alla provincia del loro CAP.

```
SELECT
    A.NomeASL ,
    A.Indirizzo ,
    C.Nome AS NomeComune
FROM
    ASL AS A
JOIN
    Comune AS C ON A.CAP = C.CAP
JOIN
    Provincia AS P ON C.SiglaP = P.SiglaP
WHERE
    P.SiglaP = 'BO';
```

2. Calcolare il numero totale di persone per ciascuna ASL nel 2022

Questa query aggrega i dati della tabella Popolazione per un anno specifico, fornendo un riepilogo per ASL.

```
SELECT
    A.NomeASL ,
    SUM(Pop.RAdulti + Pop.RInfanzia + Pop.RAnziani) AS
        TotalePopolazione
FROM
    Popolazione AS Pop
JOIN
    ASL AS A ON Pop.CodAsl = A.CodAsl AND Pop.CAP = A.CAP
WHERE
    Pop.AnnoR = 2022
GROUP BY
    A.NomeASL
ORDER BY
    A.NomeASL;
```

3. Elencare le strutture di tipo "DSM" presenti nelle ASL della regione "Lombardia"

Una query che combina più join per filtrare le strutture in base al loro tipo e alla regione di appartenenza.

```
SELECT
    S.Quantita ,
    TS.Nome AS TipoStruttura ,
    A.NomeASL ,
    C.Nome AS ComuneASL ,
    P.Nome AS ProvinciaASL ,
    R.Nome AS RegioneASL
FROM
    Strutture AS S
```

```

JOIN
    TipoStruttura AS TS ON S.TipoStruttura = TS.Id
JOIN
    ASL AS A ON S.CodAsl = A.CodAsl AND S.CAP = A.CAP
JOIN
    Comune AS C ON A.CAP = C.CAP
JOIN
    Provincia AS P ON C.SiglaP = P.SiglaP
JOIN
    Regione AS R ON P.CodR = R.CodR
WHERE
    TS.Nome = 'Dipartimento di Salute Mentale' AND R.Nome
        LIKE '%lombardia%' AND S.AnnoR = 2022;

```

4. Trovare il numero totale di ricette e l'importo complessivo delle prescrizioni per ogni ASL nel 2022

Questa query aggrega i dati delle prescrizioni per fornire totali annuali per ogni ASL.

```

SELECT
    A.NomeASL ,
    SUM(Pres.NumeroRicette) AS RicetteTotali ,
    SUM(Pres.ImportoR) AS ImportoTotalePrescrizioni
FROM
    Prescrizioni AS Pres
JOIN
    ASL AS A ON Pres.CodAsl = A.CodAsl AND Pres.CAP = A.
        CAP
WHERE
    Pres.AnnoR = 2022
GROUP BY
    A.NomeASL
ORDER BY
    ImportoTotalePrescrizioni DESC;

```

5. Visualizzare i dettagli completi di un comune (provincia e regione di appartenenza) dato il suo CAP

Questa query dimostra come ottenere informazioni correlate attraverso più join a partire da un singolo identificatore.

```

SELECT
    C.Nome AS NomeComune ,
    C.CAP ,
    P.Nome AS NomeProvincia ,
    P.SiglaP ,
    R.Nome AS NomeRegione
FROM
    Comune AS C
JOIN
    Provincia AS P ON C.SiglaP = P.SiglaP
JOIN
    Regione AS R ON P.CodR = R.CodR

```

```
WHERE  
  C.CAP = 40124;
```

6 Conclusioni

Brevi note su come le scelte di progettazione fisica contribuiscono alle prestazioni (es. "L'uso oculato degli indici mira a ottimizzare le query più frequenti, mentre la scelta di InnoDB garantisce la robustezza transazionale.").

Accenni a potenziali futuri miglioramenti o sfide (es. "Per carichi di lavoro molto elevati, si potrebbe considerare lo sharding o la replica in futuro.").

7 Fonti dataset