

Tesi di Laurea in
Informatica

Progettazione e implementazione
di una base di dati relazionale
per il percorso gravidanza e parto
nell'Ospedale di Udine

Candidato

Andrea Salvador

Relatore

Prof. Angelo Montanari

Correlatori

Prof.ssa Lorenza Driul

Dott. Andrea Brunello

Dott. Nicola Saccomanno

CONTATTI DELL'ISTITUTO

Dipartimento di Scienze Matematiche, Informatiche e Fisiche

Università degli Studi di Udine

Via delle Scienze, 206

33100 Udine — Italia

+39 0432 558400

<https://www.dmif.uniud.it/>

Alle donne di Gaza,
che nella tragedia di oltre 20 mesi di massacri e sofferenze sono costrette a subire la mancanza di cure
mediche e assistenza durante la gravidanza, e specialmente al momento del parto, spesso non potendo
disporre adeguatamente di cibo, medicinali e anestesia.

Ringraziamenti

Nella realizzazione di questo progetto è stato fondamentale il contributo del personale della Clinica Ostetricia - Ginecologia dell'Ospedale di Udine, sia sul lato dell'acquisizione dei requisiti strettamente necessari per progettare la base di dati sia per quanto riguarda la conoscenza del tema della salute in gravidanza e per i legami che ho potuto scoprire tra la medicina e l'informatica. Vorrei ringraziare quindi prima di tutto la professoressa Driul e le dottoresse Armenise, Battello, Rizzante, Xodo e Zermano, sperando di aver contribuito in modo positivo alle necessità del loro reparto con la costruzione della base di dati.

Ringrazio il professor Montanari e i dottori Brunello e Saccomanno per la disponibilità e la motivazione che mi hanno dato nel portare avanti questo progetto. Insieme a loro ringrazio anche la mia famiglia, la mia fidanzata Martina, i compagni del corso di Informatica e della Scuola Superiore, per il sostegno che ho sempre ricevuto da parte loro e per il piacere di aver condiviso questo percorso.

Sommario

In questa tesi esporremo la progettazione e l'implementazione di una base di dati relazionale, costruita secondo i requisiti ricavati direttamente dal personale medico, per informazioni su gravidanze e parti nel reparto di Ginecologia e Ostetricia nell'Ospedale di Udine, con l'obiettivo di integrare un sistema che sia adatto alle esigenze degli utenti e allo stesso tempo privo di inconsistenze nei dati.

Analizzeremo gli strumenti attualmente in uso, in particolare le loro criticità, e le richieste aggiuntive avanzate dai medici. In seguito, costruiremo uno schema concettuale per rappresentare il dominio del sistema. Attraverso una fase di ristrutturazione, definiremo uno schema logico e successivamente implementeremo tabelle e vincoli di integrità attraverso *script* SQL. Infine, porteremo esempi di interrogazioni possibili su tale base di dati.

Indice

1	Introduzione	1
1.1	Obiettivo del progetto	1
1.2	Realizzazione del progetto	2
2	Analisi del dominio e dei requisiti	3
2.1	Stato di fatto	3
2.2	Tecnologie e collezioni di dati in uso	4
2.2.1	Tabella per la visita del primo trimestre	4
2.2.2	Tabella per la visita del secondo trimestre	6
2.2.3	Tabella per il parto	7
2.2.4	Rilevazione del cardiocotografo	8
2.2.5	Gynbase	9
2.2.6	Tabella per gravidanze a rischio	10
2.3	Problemi riscontrati con le soluzioni attuali	11
2.4	Requisiti funzionali del nuovo sistema	11
2.5	Scelta di una base di dati di tipo relazionale	12
2.5.1	Modello relazionale	12
2.5.2	Confronto con altri modelli	13
2.5.3	Applicazione al caso di studio	13
3	Progettazione concettuale	15
3.1	Area concettuale della gravidanza	15
3.2	Area concettuale delle visite	17
3.3	Area concettuale del parto	17
3.4	Area concettuale del neonato	18
4	Progettazione logica	25
4.1	Ristrutturazione dello schema Entità-Relazione	25
4.1.1	Ristrutturazione dell'area concettuale della gravidanza	25
4.1.2	Ristrutturazione dell'area concettuale delle visite	26
4.1.3	Ristrutturazione dell'area concettuale del parto	27
4.1.4	Ristrutturazione dell'area concettuale del neonato	27
4.2	Schema logico relazionale	33
4.2.1	Area della gravidanza	33
4.2.2	Area delle visite	33
4.2.3	Area del parto	34
4.2.4	Area del neonato	35
5	Progettazione fisica	41
5.1	Definizione dei domini	41
5.2	Definizione delle tabelle	44
5.2.1	Area della gravidanza	44
5.2.2	Area delle visite	44

5.2.3	Area del parto	45
5.2.4	Area del neonato	45
5.3	Definizione dei trigger	51
5.3.1	Sintesi dei vincoli da implementare con trigger	51
5.3.2	Area della gravidanza	52
5.3.3	Area delle visite	53
5.3.4	Area del parto	53
5.3.5	Area del neonato	53
6	Funzionalità della base di dati	63
6.1	Esempi di interrogazioni	63
7	Conclusione	67
7.1	Sintesi della progettazione della base di dati	67
7.2	Applicazioni del nuovo sistema	67
7.3	Possibili sviluppi futuri	68

1

Introduzione

Lo sviluppo di una base di dati centralizzata per la Clinica Ostetrica dell'Ospedale di Udine si inserisce in un progetto di studio più ampio, mirato alla raccolta e all'analisi dei dati relativi a gravidanze e parti, al fine di migliorare l'assistenza fornita dal personale medico e la conseguente esperienza delle pazienti. Nell'ambito medico in generale la raccolta dati investe un ruolo cruciale: attraverso l'analisi e l'interpretazione dei risultati statistici è possibile in particolare:

- migliorare la cura del paziente individuando modelli, tendenze e fattori di rischio, consentendo così ai medici di predisporre strumenti di prevenzione adatti [7, 10];
- promuovere la ricerca scientifica stimolando nuovi studi a partire dai risultati, potenzialmente inediti, estrapolati dalla mole di dati trattata dal personale ospedaliero;
- ottimizzare la gestione ospedaliera grazie alle informazioni su efficienza operativa, uso delle risorse e conseguenti risultati.

Più nello specifico, la Clinica Ostetrica presso l'Ospedale "Santa Maria della Misericordia" di Udine è interessata a sostituire i diversi strumenti software attualmente in uso con un sistema centralizzato, personalizzato secondo i requisiti dei medici e che migliori l'accesso ai dati sia per la sintesi dei dati della singola paziente sia per analisi statistiche di più ampia scala.

1.1 Obiettivo del progetto

Come verrà esposto nella Sezione 2.1, gli usuali sistemi di refertazione adottati a livello aziendale risultano inadatti per la visualizzazione sintetica e le analisi sui dati. I referti sono costituiti di solo testo, oltre a mancare di una struttura logica; ciò ha portato alla scelta di registrare i dati in documenti separati, principalmente in forma di tabelle compilate a mano a seguito delle visite.

Questo sistema parallelo, costruito progressivamente secondo le esigenze specifiche dei suoi utenti, è in uso da circa 3 anni. I medici hanno evidenziato delle problematiche nel suo utilizzo, relative principalmente alla velocità nell'accesso alle informazioni e all'insorgenza di ambiguità legate alla compilazione esclusivamente manuale e da parte di persone diverse.

Al fine di fornire uno strumento informatico che sia più funzionale, accessibile e coerente proponiamo lo sviluppo e l'implementazione di una base di dati di tipo relazionale che sia modellata sulle esigenze specifiche del contesto in cui verrà utilizzata.

1.2 Realizzazione del progetto

Il processo comprenderà una prima fase di analisi dei requisiti, volta a individuare le caratteristiche richieste dagli utenti finali e dagli *stakeholder* in generale, seguita da una fase di modellazione a livello concettuale che determina le caratteristiche del dominio di interesse. A partire dal modello concettuale del dominio costruiremo uno schema logico relazionale che permette di definire operazioni di base sui dati, come interrogazioni e inserimenti. L'ultima fase consisterà nell'implementazione con software specifici che permettono l'interazione attraverso il linguaggio SQL. Mostreremo quindi un insieme di analisi di dati ottenibili grazie a interrogazioni sulla base di dati, per evidenziare in particolar modo le azioni rese possibili dal nuovo sistema che non erano possibili o praticabili con gli strumenti precedenti.

Analisi del dominio e dei requisiti

Il fulcro della progettazione di una base di dati è l'analisi dei requisiti, in quanto essi stabiliscono quali informazioni dovranno essere memorizzate e quali vincoli vengono posti su di esse [8]. Questa fase deve essere condotta in modo accurato ed esaustivo perché eventuali errori commessi al momento della definizione dei requisiti richiedono una quantità molto più elevata di tempo e risorse per essere corretti rispetto a quelli introdotti successivamente, ad esempio nell'implementazione. L'acquisizione dei requisiti deve superare il divario di conoscenza del dominio che si ha tra l'utente finale, in questo caso un membro del reparto di Ginecologia e Ostetricia che si interfacerà con il sistema, e il progettista che lo deve realizzare; ciò si attua con interviste dirette ai committenti e con uno studio accurato dei sistemi già in uso, rilevando le esperienze e le eventuali difficoltà avute.

2.1 Stato di fatto

Il personale medico utilizza sistemi informatici specifici, integrati e uniformati a livello di azienda sanitaria o di sistema sanitario regionale, per compilare e conservare referti di visite ed esami vari; ciò permette il funzionamento dei fascicoli sanitari digitalizzati attualmente in uso. I software in questione sono progettati per rispettare normative legali e standard di qualità e privacy, consentendo di accedere ai singoli referti, ma rendono più difficoltosa l'estrazione delle singole informazioni su pazienti o esami. Per i medici della Clinica che ha richiesto la realizzazione di un nuovo sistema informatico, la necessità è di poter inserire o visualizzare i dati in modo immediato, senza che sia necessario leggere l'intero referto, consentendo inoltre di processare i risultati a fini statistici anche con metodi automatizzati.

I diversi momenti nel corso della gravidanza in cui vengono registrati dati relativi alla paziente (come parametri fisiologici o esami svolti) si sintetizzano nei seguenti.

1. Una visita nel primo trimestre di gravidanza, svolta tra le 11° e la 13° settimana, volta principalmente all'esame chiamato *translucenza nucale* e a un accertamento delle condizioni iniziali della gravidanza.
2. Una visita nel secondo trimestre, detta *morfologica*, che punta ad analizzare lo sviluppo del feto, svolta tra la 19° e la 21° settimana.
3. Una o più visite ecografiche, dette anche *biometriche*, svolte nel corso del terzo trimestre.

4. Il travaglio e il parto, con informazioni sull'intero processo inclusi dati sui neonati.

È importante considerare che il personale medico registra dati soltanto per le visite svolte all'interno della Clinica, e accade frequentemente che una o più visite vengano effettuate in altre strutture o che anche il parto possa avvenire presso altri punti nascita.

Oltre agli esiti di queste visite si vorrebbero conservare informazioni su eventuali patologie delle pazienti, potendo contare sulla presenza di un ambulatorio specifico per le gravidanze *a rischio* che monitora anche le terapie seguite.

2.2 Tecnologie e collezioni di dati in uso

Come presentato nella Sezione 2.1 i medici lamentano la mancanza di un software, personalizzato e parallelo ai sistemi di refertazione “ufficiali” attualmente in uso, che favorisca la lettura dei dati ai fini di consultazione e analisi statistica sull'insieme di pazienti. In assenza di uno strumento centralizzato adatto, nel corso degli ultimi anni (principalmente a partire dal 2022-2023) il personale della Clinica ha registrato i dati di interesse attraverso altri strumenti, affiancati quindi ai referti, variabili sia nel livello di strutturazione dei dati sia nelle modalità di interazione.

Le informazioni raccolte dalle visite vengono registrate principalmente in semplici fogli di calcolo tabellari, accessibili dai diversi calcolatori del reparto e compilati manualmente: elencheremo i diversi campi di queste tabelle con il relativo significato e i possibili valori che possono assumere. A questi si aggiunge un software più complesso e strutturato, ma giudicato non adatto alle esigenze del personale medico.

2.2.1 Tabella per la visita del primo trimestre

Nella visita del primo trimestre vengono annotati aspetti di base ma fondamentali della gravidanza, insieme a primi test genetici e di rischio per determinate condizioni patologiche. In questa visita viene svolto l'esame della translucenza nucale, atto a rilevare malformazioni nel feto già negli stadi iniziali della gravidanza [12]. Questa tabella, così come le successive, contiene un record per ogni gravidanza.

Nella tabella la notazione usata adotta frequentemente abbreviazioni e i campi indicati sono stati predisposti direttamente dal personale medico che poi li deve compilare, adattandosi quindi alle convenzioni a cui gli operatori sono abituati. La compilazione manuale dà ovviamente la possibilità di inserire informazioni sbagliate in tabella, siano esse dati verosimili ma non corrispondenti alla realtà oppure dati corretti ma espressi in modo inconsistente.

- **Data NT:** data della visita. La sigla NT sta per “translucenza nucale” (*nuchal translucency*).
- **Cognome, Nome, DdN:** dati personali identificativi della paziente (cognome, nome e data di nascita).
- **Età all'esame:** età della paziente, espressa come numero intero.
- **Gemellare:** indica se la gravidanza è gemellare, espresso con valori 0/1. In questo attributo e nei seguenti, dove non espresso diversamente, 0 indica un valore falso, assente o negativo e 1 indica un valore vero, presente o positivo.

- **EG:** età gestazionale, indicata come settimane più giorni (ad esempio 12+5).
- **Rischio T21, Rischio T18, Rischio T13:** rischio valutato per le trisomie 21, 18 e 13. I fattori di rischio sono indicati usualmente come classi di rischio, che vanno da alto (A), medio (M oppure I, intermedio), basso (B). In alcuni casi il rischio viene indicato in modo più preciso come rapporto 1:X, con X valore intero, e un livello basso corrisponde a un rapporto 1:1000 o inferiore, un livello medio si trova tra 1:250 e 1:1000, un livello alto è 1:250 o superiore. La scelta di indicare il rischio con una lettera, con l'intera parola o come rapporto dipende dall'operatore che compila la tabella.
- **Morfologia fetale:** indica se sono presenti anomalie nella morfologia fetale, con valori 0/1.
- **Anomalie segnalate:** se il campo precedente ha valore 1, indica le anomalie morfologiche riscontrate come descrizione testuale.
- **NT:** esito dell'esame della translucenza nucale, indica lo spessore della plica nucale in millimetri espresso come numero reale. Il risultato può essere accompagnato da una descrizione testuale.
- **Esito genetica:** risultati di test genetici, eseguiti opzionalmente, successivamente alla visita. I risultati sono molto variabili e indicati come descrizione testuale.
- **Premorfo indicata?:** indica se viene consigliata un'ecografia premorfologica, con valori 0/1 ed eventualmente la motivazione nel caso 1.
- **NIPT:** riporta se è stato effettuato il test prenatale non-invasivo (*Noninvasive Prenatal Testing*).
- **esito NIPT:** esito dell'esame NIPT, riportato come descrizione testuale.
- **BS/BSOB:** indice morfologico misurato sulle dimensioni del tronco encefalico (BS) e della distanza dall'osso occipitale (BSOB), indicato come valore reale.
- **CRL:** indice morfologico, corrispondente alla lunghezza del feto in millimetri.
- **PAPP-A, freeBHCG, PLGF:** esami svolti sul sangue della paziente per la rilevazione di proteine e materiale genetico, danno valori reali.
- **UTPI:** indice di pulsatilità dell'arteria uterina, espresso come valore reale.
- **PAM:** pressione arteriosa media materna, misurata come valore intero in mmHg.
- **Esito <34, Esito <37:** misure della predisposizione alla preeclampsia, rispettivamente entro le 34 e le 37 settimane di gravidanza, espresse come rapporto di rischio.
- **Prescrizione ASA:** indica se c'è stata prescrizione di cardioaspirina, con valori 0/1.
- **Spontanea, PMA, IUI, FIVET, ICSI, Ovodonazione:** informazioni su concepimento ed eventuale tipo di procreazione medicalmente assistita, tutti con valori 0/1. I campi **Spontanea** e **PMA** sono mutuamente esclusivi; gli altri campi, da **IUI** a **Ovodonazione**, sono compilati solo nel caso di procreazione medicalmente assistita, con **IUI**, **FIVET** e **ICSI** mutuamente esclusivi.
- **Peso, Altezza, BMI:** parametri fisici della paziente alla visita.

- **Fumo:** indica se la paziente fuma attualmente, con valori 0/1.
- **Diabete pregestazionale:** indica se la paziente soffre di diabete pregestazionale (precedente o non dovuto alla gravidanza).
- **Malattie autoimmuni:** indica, se ne soffre, le malattie autoimmuni della paziente, elencate in forma testuale.
- **Pregressa PE:** indica se ha sofferto di preeclampsia (PE) in gravidanze precedenti.
- **PE, IUGR, Eseguita ASA, EG al parto:** attributi predisposti per la compilazione alla conclusione del percorso di gravidanza, dato che si riferiscono a dati registrati alla fine della gravidanza o al momento del parto. In realtà questi campi normalmente non vengono compilati e le informazioni si possono reperire nella tabella che contiene i dati del parto.
- **Note:** annotazioni aggiuntive.

2.2.2 Tabella per la visita del secondo trimestre

La visita del secondo trimestre è incentrata sull'esecuzione di un'ecografia morfologica, ovvero volta a esaminare lo sviluppo del corpo del feto. Si registrano anche eventuali esami svolti nel periodo precedente alla visita.

- **Data Morfo:** data di svolgimento della visita.
- **Cognome, Nome, DdN:** dati personali identificativi della paziente.
- **EG, Età all'esame:** età gestazionale ed età della paziente al momento della visita.
- **Premorfo:** indica se è stata svolta un'ecografia premorfologia e l'eventuale esito in forma testuale.
- **Gemellare:** indica se la gravidanza è gemellare, con valori 0/1.
- **Morfologia fetale:** indica se sono presenti anomalie morfologiche nel feto, con valori 0/1.
- **Anomalie segnalate:** se il valore del campo precedente è 1, elenca le anomalie riscontrate in forma testuale.
- **Decorso:** indica, in forma testuale, se ci sono stati eventi degni di nota nel periodo precedente alla visita.
- **Esito genetica:** risultati di test genetici, non sempre presenti, descritti in forma testuale.
- **Note:** annotazioni aggiuntive
- **NIPT:** indica se è stato effettuato il test NIPT, con valori 0/1.
- **Esito NIPT:** esito dell'esame NIPT. Come valori si trovano 0 (o equivalentemente: B, BR, basso, basso rischio), insufficiente (materiale mancante per dare un esito chiaro), oppure una descrizione testuale del risultato.

2.2.3 Tabella per il parto

Durante il ricovero ospedaliero per il periodo di travaglio e parto si registrano dati relativi sia al processo di espulsione (comprese tempistiche o complicazioni) sia parametri fisiologici della partoriente e dei nati. I valori delle colonne da APGAR a Sesso neonato sono riferite ai neonati: se il parto è gemellare sono presenti due (o eventualmente più) valori, separati da uno spazio, per ciascuna di queste colonne, mantenendo ovviamente lo stesso ordine.

- **Data:** data del parto.
- **PZ:** cognome e nome della paziente.
- **S.G.:** età (stato) gestazionale, espresso sempre come settimane+giorni.
- **Parità:** stringa di cifre che sintetizzano le gravidanze della paziente che hanno preceduto quella corrente. Si considerano: il numero di figli nati a termine, il numero di figli nati pretermine, il numero di aborti (spontanei così come per interruzione volontaria di gravidanza), il numero totale di figli nati vivi; la stringa risultante è costituita di questi valori, nell'ordine riportato (ad esempio 1021, 0000), e quindi è costituita generalmente di 4 caratteri, ma occasionalmente può averne di più se la paziente ha avuto un numero elevato di gravidanze.
- **Travaglio:** indica il tipo del travaglio. Può avere i seguenti valori: indotto, spontaneo, pilotato, senza travaglio. L'ultimo caso corrisponde a parti cesarei programmati.
- **Motivo induzione:** indica il motivo dell'induzione del travaglio come descrizione testuale.
- **Metodo induzione:** indica il metodo di induzione del travaglio. Solitamente corrisponde a una lista di abbreviazioni che rappresentano farmaci o metodi meccanici di induzione.
- **Parto:** indica il tipo di parto, con i seguenti valori possibili: spontaneo, cesareo, operativo.
- **Motivo parto operativo:** se il tipo di parto è operativo, riporta i motivi dell'attuazione, come descrizione testuale.
- **Se TC:** se il tipo di parto è cesareo, riporta se è stato programmato come tale, con i seguenti valori possibili: programmato, urgente.
- **Motivo TC:** se il tipo di parto è cesareo, riporta i motivi dell'attuazione, come descrizione testuale.
- **Episiotomia:** indica se è stata svolta l'episiotomia, con valori possibili Sì o No.
- **Motivo episiotomia:** se è stata eseguita l'episiotomia, ne riporta il motivo come descrizione testuale.
- **Lacerazioni:** indica il grado di lacerazioni. I valori possibili sono i seguenti: 0 (equivalenti: No, vuoto), 1, 2, 3, 4, AL (altro). Si applica a tutti i parti eccetto quelli cesarei programmati.
- **Tracheloraffia:** indica la presenza di lacerazioni nel collo dell'utero, con valori possibili Sì o No.
- **Perineo integro:** indica se il perineo è rimasto integro, con valori Sì o No.

- **Secondamento:** indica la modalità di espulsione della placenta nel processo del parto. I valori possibili sono i seguenti: attivo, strumentale, manuale, scivolamento.
- **Perdite:** indica la perdita ematica sofferta durante il parto, quantificata in mL.
- **Robson:** classificazione del parto con valori della scala di Robson¹.
- **Analgesia:** indica se è stata somministrato un analgesico, con valori Sì o No.
- **Tipo analgesia:** indica il tipo di analgesia somministrata. I valori possibili sono: spinale, epidurale (equivalente: peridurale), spinale e peridurale, calinox.
- **APGAR:** valore di “vitalità” del neonato su una scala di valori interi da 0 a 10. Viene misurato al 1° minuto dalla nascita, al 5° minuto e occasionalmente anche al 10° minuto (ad esempio, 7/7/8).
- **TIN:** indica se il neonato è stato sottoposto a terapia intensiva neonatale, con valori possibili Sì o No.
- **pH, BE:** risultati delle analisi eseguite sul sangue del cordone ombelicale, espressi come valori reali.
- **Sesso neonato:** indica il sesso attribuito al neonato, con valori possibili M o F.

2.2.4 Rilevazione del cardiocotografo

Durante il travaglio viene usato il cardiocotografo, uno strumento che rileva il battito cardiaco del feto (o anche di più feto nel caso di parti gemellari) e l’entità delle contrazioni uterine [2]. Questa apparecchiatura registra e salva i dati in modo da poterli esportare in forma grafica o tabellare. In particolare consideriamo il file risultato dall’esportazione in forma di foglio di calcolo, in cui vengono indicati alcuni dati identificativi della paziente.

- **Nome, Cognome:** informazioni personali della paziente.
- **Num. progressivo:** identificativo progressivo, indicato come numero intero, attribuito dalla macchina alla registrazione.
- **Data e ora inizio tracciato:** timestamp (in formato leggibile) del momento che la macchina assume come “0 secondi”.

I diversi sensori dello strumento sono sincronizzati e rilevano i rispettivi valori 4 volte al secondo. La tabella contiene quindi una riga per ogni rilevazione, ciascuna delle quali ha i seguenti campi.

- **Secondi, Decimi:** indicazione del momento di rilevazione, rispetto al tempo di inizio tracciato indicato all’inizio della tabella. Entrambi i valori sono interi, con **Decimi** [*sic*] che indica i centesimi di secondo (0, 25, 50, 75).

¹Valori possibili nella classificazione di Robson: 1, 2a, 2b, 3, 4a, 4b, 5.1, 5.2, 6, 7, 8, 9, 10.

- **FCF, FCF (2), FCF (3)**: indicano la frequenza cardiaca fetale, come valore intero, rilevata da ciascuno dei tre sensori di cui è dotato lo strumento. Anche se la gravidanza è singola, non è necessariamente il sensore 1 a rilevare il battito del feto. Un valore assente, sia per sensori non utilizzati sia per eventuali errori nella rilevazione, è indicato nella tabella con -1. Nel caso di parti gemellari è comunque possibile stabilire a quale neonato corrisponde ciascun tracciato.
- **TOCO**: misura la contrazione uterina, indicata come valore intero.
- **FCM**: misura la frequenza cardiaca materna, indicata sempre come valore intero.

2.2.5 Gynbase

Tra gli strumenti utilizzati dal personale medico del reparto figura il software *Gynbase*, che si occupa di gestire i dati relativi alla gravidanza e al parto. *Gynbase* è un sistema proprietario che opera su una base di dati relazionale accessibile attraverso un'interfaccia grafica dedicata. Tale programma è predisposto per contenere categorie di dati che non sono direttamente di interesse per la Clinica, per cui non sempre viene compilato quando i dati sono già presenti nelle tabelle esposte precedentemente (Sezione 2.2.1, Sezione 2.2.2 e Sezione 2.2.3). Il contenuto della base di dati, se esportato come tabella in formato CSV, arriva a comprendere circa 150 colonne; analizzeremo quindi gli attributi riportati come di interesse per il personale medico e che non si sovrappongono ad altri già esposti.

- **ID Paziente**: identificativo numerico progressivo attribuito dal sistema alla paziente.
- **ID Gravidanza**: identificativo numerico progressivo attribuito dal sistema alla gravidanza. Questo numero è unico per l'intera base di dati.
- **Cognome, Nome, Nata**: dati personali della paziente.
- **Altezza cm**: altezza della paziente, come valore intero in centimetri.
- **Kg pre gravidanza**: peso della paziente, riportato da lei oppure misurato alla visita del primo trimestre.
- **BMI pre gravidanza**: indice di massa corporea calcolato a partire da peso e altezza all'inizio della gravidanza.
- **Età al concepimento**: indica l'età alla data stimata di concepimento, riportata come numero intero.
- **UM**: riporta la data dell'ultima mestruazione.
- **EPP**: riporta la data prevista del parto.
- **PIH**: indica² se la paziente ha sofferto di ipertensione gestazionale (*pregnancy-induced hypertension*).

²I valori attesi della colonna **PIH**, così come delle seguenti **GDM**, **Tireopatia** e di altre colonne, sarebbero valori booleani del tipo Sì/No o 0/1. In realtà nel file CSV generato dal sistema viene riportato l'identificativo della paziente per significare "vero"/"sì" mentre invece è lasciato vuoto per significare "falso"/"no". Il risultato è che i dati, in questo formato, non sono facilmente leggibili dall'utente.

- **GDM**: indica se la paziente ha sofferto di diabete gestazionale (*gestational diabetes mellitus*).
- **Tireopatia**: indica se la paziente ha sofferto di tiropatia.
- **Data Espulsione**: indica data e ora³ del parto.
- **Data Fase Attiva**: indica data e ora dell’inizio della fase attiva.
- **Data Membrane**: indica data e ora della rottura delle membrane.
- **Data Completa**: indica data e ora del raggiungimento della dilatazione completa.
- **Data Fase Attiva 2**: indica data e ora dell’inizio della fase di espulsione o di spinta.
- **Data Secondamento**: indica data e ora di espulsione della placenta.
- **Complicazioni**: descrizione testuale di eventuali complicazioni avvenute nel processo di travaglio e parto.
- **Indicazioni operativo**: dettagli sullo svolgimento del parto, se operativo.
- **Ind _**: indicazione del motivo o dei motivi dell’induzione del travaglio. Sono presenti diverse colonne, ciascuna rappresentativa di una motivazione (al posto di _): **ipertensione preesistente**, **ipertensione gestazionale**, **proteinuria isolata**, **iugr**, **gdm**, **preeclampsia**, **macrosomia**, **protratta**, **pma**, **oligoamnios**, **poliidramnios**, **altro**, **pprom**, **prom**, **colestasi**, **mef**, **prodromi prolungati**. Per indicare che l’induzione è stata svolta per uno o più determinati motivi la colonna riporta la lettera Y, altrimenti è lasciata vuota.
- **Minuti induzione-inizio travaglio**: tempo trascorso tra l’induzione del travaglio e l’inizio del travaglio, indicato come valore intero.
- **Minuti induzione-parto**: tempo trascorso tra l’induzione del travaglio e l’espulsione, indicato come valore intero.
- **Bishop**: indicatore relativo alla preparazione del collo dell’utero all’induzione del travaglio, riportato come valore intero su una scala da 0 a 16.

2.2.6 Tabella per gravidanze a rischio

Un insieme più ristretto di pazienti viene seguito nel corso della gravidanza da un ambulatorio apposito, detto appunto “gravidanze a rischio”, in presenza di determinate patologie.

- **Nome, Cognome, DdN**: dati identificativi della paziente.
- **Patologie**: diverse colonne che riportano patologie, una per colonna, sofferte dalla paziente e che sono trattate o sorvegliate da questo ambulatorio.
- **Terapie seguite**: descrizione testuale delle terapie seguite. Può non essere indicata una terapia per tutte le patologie.

³Data e ora vengono indicate in formato DD/MM/YYYY HH:MM, oppure con valore “date missing” se assente.

2.3 Problemi riscontrati con le soluzioni attuali

L'insieme di tabelle e fogli di calcolo utilizzato attualmente dai medici del reparto è essenzialmente creato su misura, ovvero sono pensati per contenere le informazioni di loro interesse perché essi stessi ne hanno definito lo schema. Nonostante ciò, questi strumenti sono molto più semplici di una base di dati ben strutturata e questo ne evidenzia gli aspetti negativi.

L'uso di tabelle, rispetto al contenuto testuale dei referti, permette di organizzare almeno visivamente l'insieme di informazioni che si vogliono conservare per ciascuna paziente, ma non vengono imposti vincoli di correttezza o di coerenza dei dati effettivamente inseriti. Le operazioni di accesso ai dati (a partire dai dati identificativi della paziente) sono veloci se si considera una singola visita ma se si vuole analizzare l'intero periodo della gravidanza è necessario leggere contemporaneamente più tabelle; sotto questo aspetto l'interfaccia grafica fornita dai diversi software per visualizzare fogli di calcolo non è ottimale per tali scopi.

Il personale medico riporta anche difficoltà e insoddisfazione nell'uso del software Gynbase che, nonostante appaia meglio strutturato dal punto di vista grafico e concettuale, in realtà presenta aspetti negativi che emergono nel corso degli anni di esperienza. I punti di criticità più importanti sono:

- la mancanza di controlli o avvisi su dati incompatibili (ad esempio sull'ordine cronologico di determinati momenti del parto);
- il fatto che le stesse informazioni possono essere richieste in aree diverse del programma e quindi vengono duplicate, portando a evidenti inconsistenze se non c'è un adeguato controllo da parte di chi si interfaccia con il sistema;
- la poca chiarezza dell'interfaccia, in particolare nella distinzione tra informazioni necessarie e opzionali.

2.4 Requisiti funzionali del nuovo sistema

La richiesta avanzata dal personale medico è di un software integrato e centralizzato che permetta la visualizzazione, in prima analisi, e la manipolazione, in una fase successiva, dei dati relativi a gravidanze e parti, come esposto precedente (Sezione 2.1).

Il sistema dovrà permettere di visualizzare, aggiungere, modificare ed eliminare le informazioni che attualmente sono contenute nei diversi supporti esposti nella Sezione 2.2. In sintesi, i dati riguardano dettagli personali della paziente e di sue eventuali patologie; dettagli della gravidanza, del suo decorso e delle visite svolte; informazioni su travaglio, parto e parametri fisiologici dei neonati; esiti dei vari esami eseguiti nel corso della gravidanza.

Analizzando le tabelle attualmente in uso possiamo dedurre le informazioni di interesse che dovranno sicuramente essere rese accessibili dal sistema che realizziamo. Questo è anche il momento più indicato per valutare l'aggiunta di requisiti che prima potevano non essere stati considerati o non venivano realizzati per la mancanza di supporti adeguati. In particolare, oltre a quanto esposto nella Sezione 2.2, i medici vogliono poter memorizzare informazioni sui seguenti aspetti non trattati prima.

1. Dati relativi ai neonati, in particolare peso, altezza e circonferenza cranica, misurati alla nascita.

2. Visite ecografiche biometriche. Possono essere svolte più visite biometriche nel corso della gravidanza (tendenzialmente 1 o 2) e si raccolgono dati relativi ai seguenti indicatori:
 - BPD (diametro biparietale);
 - CC (circonferenza cranica);
 - AC (circonferenza addominale);
 - FL (lunghezza del femore);
 - EFW (peso fetale stimato);
 - PIAO (valutazione flussimetrica dell'arteria ombelicale);
 - PIACM (valutazione flussimetrica dell'arteria cerebrale media fetale);
 - valutazione del dotto venoso;
 - stato della crescita del feto (crescita regolare, FGR oppure SGA).
3. Esiti di esami vari, eseguiti anche al di fuori delle principali visite elencate nella Sezione 2.1, di una lista eventualmente espandibile su necessità del personale medico. Gli esami danno esiti in forma perlopiù di indici numerici, comprendenti ma non limitati a: HB, TSH, AST/ALT, glicemia, curva da carico di glucosio, sierologie varie, gruppo sanguigno, test di Coombs indiretto.
4. Dettagli su problemi concomitanti, patologie e terapie seguite (anche al di fuori di quelle controllate nell'ambulatorio "gravidanze a rischio") e altri aspetti come interventi, trasfusioni o allergie, da indicare in forma testuale o debolmente strutturata.

Riguardo invece alle informazioni presenti nei sistemi già in uso (Sezione 2.2) vengono fatte alcune precisazioni:

- I parti naturali e operativi possono essere eseguiti o meno con la manovra di Kristeller.
- Un parto gemellare può avere delle complicazioni tali da portare un gemello a nascere naturalmente e l'altro con un parto cesareo oppure operativo. In questi casi si vogliono mantenere informazioni sugli aspetti caratteristici di entrambi i tipi di parto avvenuti.
- Durante il travaglio possono essere somministrati (se farmacologici) o eseguiti (se meccanici) diversi metodi di induzione. Le informazioni su metodo, tempo di somministrazione, motivo, valutazione di Bishop, completamento della procedura, numero di cicli eseguiti e quantità somministrata si vogliono mantenere per ciascuna somministrazione.

2.5 Scelta di una base di dati di tipo relazionale

2.5.1 Modello relazionale

Nel modello relazionale [6] i dati vengono strutturati attraverso relazioni (o equivalentemente, tabelle), ciascuna composta di attributi (colonne) e istanziata come insieme di tuple (righe o record). Uno schema relazionale corretto e ben strutturato mira a prevenire inconsistenze e dati duplicati. Gli strumenti software che operano su basi di dati relazionali (RDBMS) permettono di definire vincoli di integrità

che contribuiscono al mantenimento della coerenza dei dati, all'assenza di ambiguità e all'efficienza delle interazioni con il sistema attraverso controlli automatici. La modellazione del dominio in cui operiamo viene fatta prima a livello concettuale definendo uno schema nel modello “Entità-Relazione” e successivamente traducendolo in uno schema logico nel modello relazionale.

Le basi di dati relazionali si possono interrogare con semplici richieste (*query*) in SQL, un linguaggio dichiarativo basato sulla logica del prim'ordine. Questo linguaggio è supportato da tutti i principali linguaggi di programmazione e ambienti di sviluppo, prestandosi così a integrarsi facilmente in applicazioni complesse e che permettono, ad esempio, l'interazione attraverso un'interfaccia grafica dedicata. In questo modo solo il programmatore ha la necessità di conoscere i dettagli implementativi dello schema mentre l'utente finale, anche se inesperto, può operare sui dati a livello più astratto e vicino al modello concettuale del dominio.

2.5.2 Confronto con altri modelli

Altri modelli di basi di dati possono avere altri vantaggi specifici, ma spesso perdono su alcuni dei punti forti del modello relazionale.

- Il modello gerarchico prevede una struttura ad albero, composto da nodi che possono avere un numero arbitrario di figli ma solo un genitore ciascuno. Questo pone molti limiti sullo schema da realizzare e spesso richiede di duplicare dati.
- Il modello reticolare estende quello gerarchico permettendo un numero arbitrario di genitori, formando quindi un grafo generico. La struttura così ottenuta, eccessivamente libera nella sua topologia, rende più difficoltosa l'interrogazione.
- Il modello a documento, che sta prendendo piede come principale modello di tipo *NoSQL*, prevede che i record siano memorizzati come documenti strutturati ma senza uno schema comune. Questo facilita la memorizzazione di dati con caratteristiche eterogenee ma allo stesso tempo rende più difficoltose le operazioni che legano documenti diversi, come le *join*.

2.5.3 Applicazione al caso di studio

Considerando i requisiti esposti nella Sezione 2.4, proponiamo l'utilizzo di una base di dati basata su un modello relazionale. Le informazioni trattate dal personale medico seguono schemi ben strutturati, come dimostra l'organizzazione tabellare utilizzata finora, e a tali schemi possiamo ricondurre le entità e le relazioni a livello astratto. Riconosciamo infatti entità come la paziente, la gravidanza, la visita, il parto, il neonato, ciascuna con un insieme di attributi. Tra di esse sussistono relazioni che seguono schemi costanti, come la relazione tra gravidanze e visite, tra gravidanze e malattie o tra parti e neonati.

Una base di dati relazionale permette di gestire in modo naturale diversi vincoli di integrità, soprattutto per quanto riguarda gli attributi identificativi (detti “chiavi”) sia il numero di entità coinvolte in determinate relazioni (detto “cardinalità”). Nel caso reale che stiamo trattando è possibile porre vincoli sul numero di visite di ciascun tipo effettuate nel corso della gravidanza, così come sul numero di bambini nati da un parto o di esami svolti in ciascuna visita.

Si vuole che il sistema risultante costituisca, in prima fase, un magazzino dati (*data warehouse*⁴) volto principalmente alla visualizzazione contemporanea e coerente dei dati proveniente dalle diverse fonti attualmente in uso e che possa sostituirle completamente in un secondo momento.

Nel Capitolo 3 si mostra lo schema Entità-Relazione costruito a partire dai requisiti e vengono illustrate le scelte operate a livello concettuale, con tutti i dettagli relativi alle entità presenti, alle relazioni tra esse e agli attributi.

⁴L'accezione di *data warehouse* [8], concetto distinto da *database*, evidenzia come il sistema integri dati provenienti da più fonti in un unico schema e sia orientato alla visualizzazione più che alle operazioni di modifica.

3

Progettazione concettuale

Il modello Entità-Relazione rappresenta il dominio a livello astratto attraverso entità, corrispondenti a elementi o concetti con caratteristiche proprie, e relazioni, ovvero legami che intercorrono tra entità. Le caratteristiche di un'entità o di una relazione sono rappresentate come attributi della stessa; ciascuna entità può avere uno o più attributi che compongono chiavi, ovvero che identificano l'istanza di entità. Attributi e relazioni possono avere cardinalità e partecipazioni diverse e queste vengono indicate nello schema. Tra le diverse relazioni che si possono modellare si mettono in evidenza le relazioni di generalizzazione-specializzazione, corrispondenti a relazioni del tipo "...è un ...", e le strutture di tipo categoriale.

La figura 3.1 mostra lo schema della base di dati nel modello Entità Relazione in una versione semplificata per evidenziare le entità e le relazioni presenti. I quattro contorni tratteggiati identificano quattro aree del diagramma (gravidanza, visite, parto, neonato) che verranno illustrate dettagliatamente con gli attributi presenti. L'analisi dei requisiti e dei supporti esistenti (Sezione 2.2) evidenzia come elemento centrale le gravidanze: solo poche informazioni sono legate alla paziente piuttosto che alla gravidanza e sono essenzialmente quelle che la identificano personalmente.

In generale la presenza di attributi opzionali è dovuta non tanto alla mancanza del dato, ovvero che non sia presente a livello concettuale, bensì alla mancata conoscenza di un suo valore, da attribuirsi a un'errata compilazione dei record o a un dettaglio anamnestico non riportato dalla paziente. Nell'uso del valore *NULL* si ritrova questa stessa ambiguità [11]. Si intende lasciare la possibilità di avere dati mancanti per permettere un'integrazione che sia compatibile con i sistemi attualmente in uso, in modo da implementare il sistema come *data warehouse*, oltre a renderlo tollerante rispetto a eventuali omissioni di informazioni.

3.1 Area concettuale della gravidanza

La Figura 3.2 mostra l'ingrandimento dello schema Entità-Relazione per quanto riguarda le entità Paziente, Gravidanza e Malattia.

Paziente

Si introduce un identificativo per la paziente che risulta essere effettivamente univoco, ovvero il codice fiscale, assumendo che tutte le pazienti trattate abbiano un codice fiscale assegnato. Ciò semplifica la rappresentazione dell'identificativo personale, che nello schema logico verrà inserito in quasi tutte le tabelle, rispetto alla tripla composta da nome, cognome e data di nascita, e previene i possibili, seppur rarissimi, casi di corrispondenza di questi dati per persone diverse.

Gravidanza

A questa entità fanno riferimento alcuni attributi che vengono registrati durante la visita del primo trimestre, oltre all'esito della gravidanza e alla data del primo ingresso. Decidere se assegnare tali attributi a “Gravidanza” oppure a “Visita I° trimestre” può essere talvolta arbitrario: preferiamo quindi assegnare alla prima quelle informazioni che si possono considerare come proprie della gravidanza in sé (come la presenza di feti gemellari o il tipo di PMA), mentre alla seconda le informazioni sulla paziente che si riferiscono al giorno o al periodo della visita (come ad esempio peso o altezza).

Identifichiamo due chiavi candidate, entrambe parziali in quanto l'entità è debole verso Paziente:

- la data del primo ingresso, attributo derivato e corrispondente alla data della prima visita effettuata, se ce n'è stata una, altrimenti alla data del parto;
- la parità registrata all'inizio della gravidanza.

L'attributo “Parità” è composto. Nei sistemi attualmente in uso è scritto solitamente in un unico campo, usando quindi la rappresentazione testuale che è calcolata come concatenazione del numero di figli nati a termine, del numero di figli nati pretermine, del numero di aborti (comprendente sia aborti spontanei sia interruzioni volontarie di gravidanza) e del numero totale di figli nati vivi, in questo ordine.

Sempre sull'attributo “Parità” si pone il seguente vincolo.

- Se una paziente ha più di una gravidanza, per ogni coppia di gravidanze successive i valori degli attributi (non derivati) sono non decrescenti e almeno uno di essi è strettamente crescente.

Le relazioni che l'entità Gravidanza intrattiene con le diverse visite e con Parto hanno tutte partecipazione parziale, per cui viene imposto il seguente vincolo.

- Una gravidanza viene registrata solo se viene effettuata almeno una visita oppure se termina con un parto.

Malattia

L'entità “Malattia” rappresenta le patologie che possono presentarsi in concomitanza con la gravidanza. Attualmente vengono registrate in modo sistematico solo per le pazienti seguite nell'ambulatorio “gravidanze a rischio” (vedi Sezione 2.2.6) ma è applicabile alla gravidanza di qualsiasi paziente. È importante considerare che le malattie possono variare tra le diverse gravidanze della stessa paziente. Le informazioni riguardo alle eventuali terapie seguite sono memorizzate nell'attributo della relazione che lega gravidanza e malattie.

3.2 Area concettuale delle visite

La Figura 3.3 mostra l'ingrandimento dello schema Entità-Relazione per quanto riguarda le entità relative alle visite e all'entità Esame.

Visite

Dividiamo le possibili visite in quattro entità: Visita I° trimestre, Visita II° trimestre, Visita biometrica, Visita di altro tipo. Gli attributi di queste entità fanno riferimento principalmente alla data e alle informazioni rilevate dalla paziente come età, epoca gestazionale o anomalie fetali. Gli esiti degli esami svolti per una determinata visita non sono modellati come attributi delle visite ma come attributi della relazione che lega Visita e Esame, in modo che gli esami da registrare possano essere modificati nel tempo senza agire sullo schema della base di dati.

I diversi attributi “Età”, presenti in tutte le entità, sono calcolati a partire dalla data di nascita della paziente. L'entità “Visita di altro tipo” raccoglie le istanze di visite che altrimenti non rientrerebbero in nessuna delle tre categorie specifiche ma per le quali si vogliono comunque registrare esiti di esami svolti.

La scelta di organizzare l'entità “Visita” come unione (o categoria) piuttosto che come una generalizzazione è dovuta alla cardinalità delle relazioni che i quattro tipi di visita intrattengono con l'entità Gravidanza. Tutte le visite sono entità deboli rispetto a Gravidanza, ma le visite biometriche o di altro tipo possono essere svolte più volte nel corso di una gravidanza e quindi richiedono un attributo che sia chiave parziale, che invece sarebbe scorretto mettere per le visite del primo e del secondo trimestre le quali sono effettuate una sola volta ciascuna. Se fossero state modellate come specializzazioni di un'unica entità Visita si sarebbero creati numerosi cicli e vincoli aggiuntivi da imporre.

Esame

L'entità Esame rappresenta l'esame in generale, mentre l'esito di uno specifico esame svolto si ritrova nella relazione tra Esame e Visita. Il tipo di dato dell'esito dipende dall'esame, ma l'aspetto implementativo viene trattato a livello di schema logico.

3.3 Area concettuale del parto

La Figura 3.4 mostra l'ingrandimento dello schema Entità-Relazione per quanto riguarda le entità Parto, con le sue specializzazioni, e Travaglio.

Parto

Le diverse specializzazioni dell'entità Parto sono totali e sono possibili le seguenti sovrapposizioni: naturale e operativo, naturale e cesareo urgente, operativo e cesareo urgente. Solo alcune informazioni, come data o informazioni su analgesia e perdite ematiche, sono applicabili ad ogni tipo di parto: gli altri attributi si trovano nelle entità che specializzano Parto, in alcuni casi con sovrapposizioni. Ad esempio, i dati sul travaglio e sulle tempistiche del parto sono applicabili a tutti i tipi di parto eccetto i

parti cesarei programmati, mentre le lacerazioni e il secondamento sono dati presenti nei parti naturali e operativi. La data del parto è coerente con le tempistiche indicate se il parto non è cesareo urgente.

Travaglio

Le informazioni sul travaglio sono applicabili a tutti e soli i parti che non sono cesarei programmati. Se il travaglio è indotto sono presenti anche i dati relativi al metodo e alle tempistiche dell'induzione.

3.4 Area concettuale del neonato

La Figura 3.5 mostra l'ingrandimento dello schema Entità-Relazione per quanto riguarda le entità Neonato e quelle relative alle misurazioni del cardiocotografo.

Neonato

Nell'entità Neonato la chiave parziale è l'ora di nascita. Questo, ugualmente ad altri indicatori dell'ordine di nascita, è necessario per distinguere i gemelli nati in una stessa gravidanza.

Le informazioni relative al neonato sono le seguenti:

- peso, altezza e sesso attribuito;
- indice di Apgar, misurato in diversi momenti dopo la nascita;
- BE: *base excess*, eccesso di basi;
- pH: acidità del sangue;
- CC: circonferenza cranica;
- TIN: terapia intensiva neonatale.

Tracciato e misurazioni

Nello schema la sigla CTG sta per “cardiocotografo”. Un tracciato è identificabile sia con il parto in cui viene registrato sia con il numero progressivo, che viene assegnato unico dal sistema in uso nel reparto.

Ogni misurazione del tracciato riporta i valori FCM (frequenza cardiaca materna) e TOCO (togramma). I valori FCF (frequenza cardiaca fetale) possono essere multipli in ogni misurazione, quindi non sono contenute in un attributo dell'entità Misurazione TCG bensì nella relazione tra essa e Neonato. Si impone quindi il seguente vincolo.

- Per ogni misurazione deve essere presente almeno uno tra: il valore FCM, il valore TOCO e un valore FCF per un neonato associato.

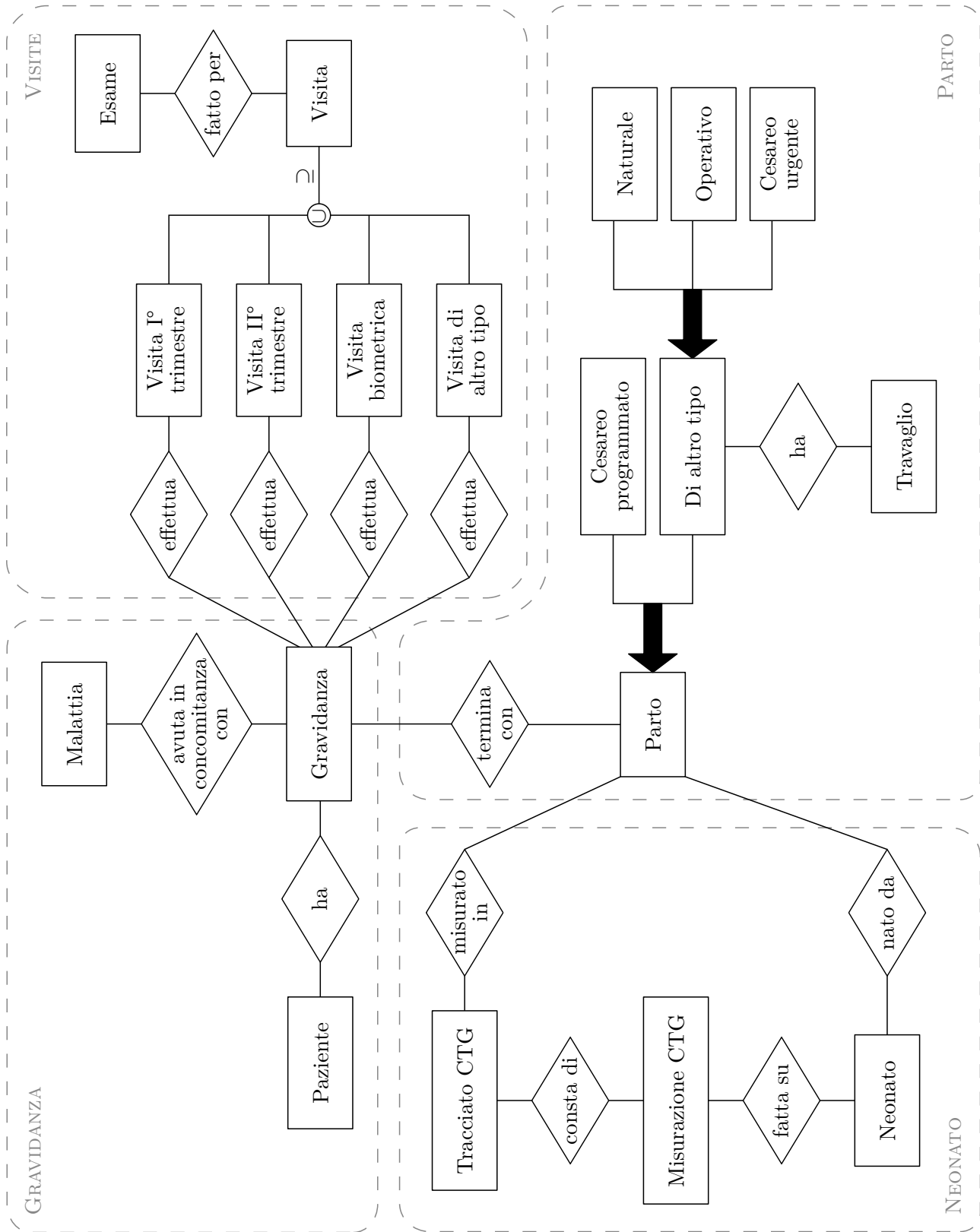


Figura 3.1: Schema Entità-Relazione. Visione d'insieme semplificata, contenente soltanto entità e relazioni del modello. Le aree tratteggiate delimitano i diversi ingrandimenti nei quali vengono specificati gli attributi delle entità e relazioni coinvolte.

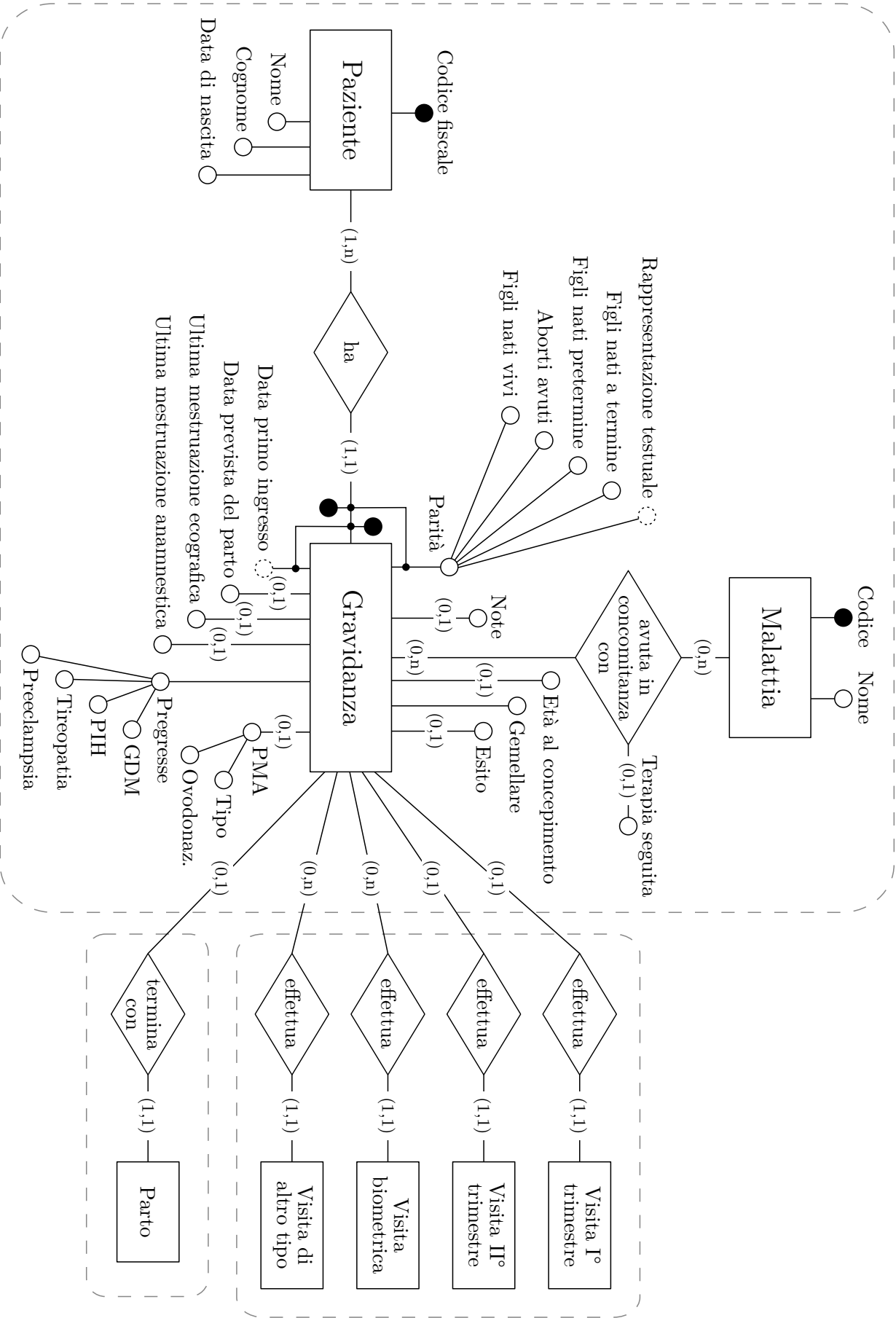


Figura 3.2: Schema Entità-Relazione. Ingrandimento dell'area gravidanza.

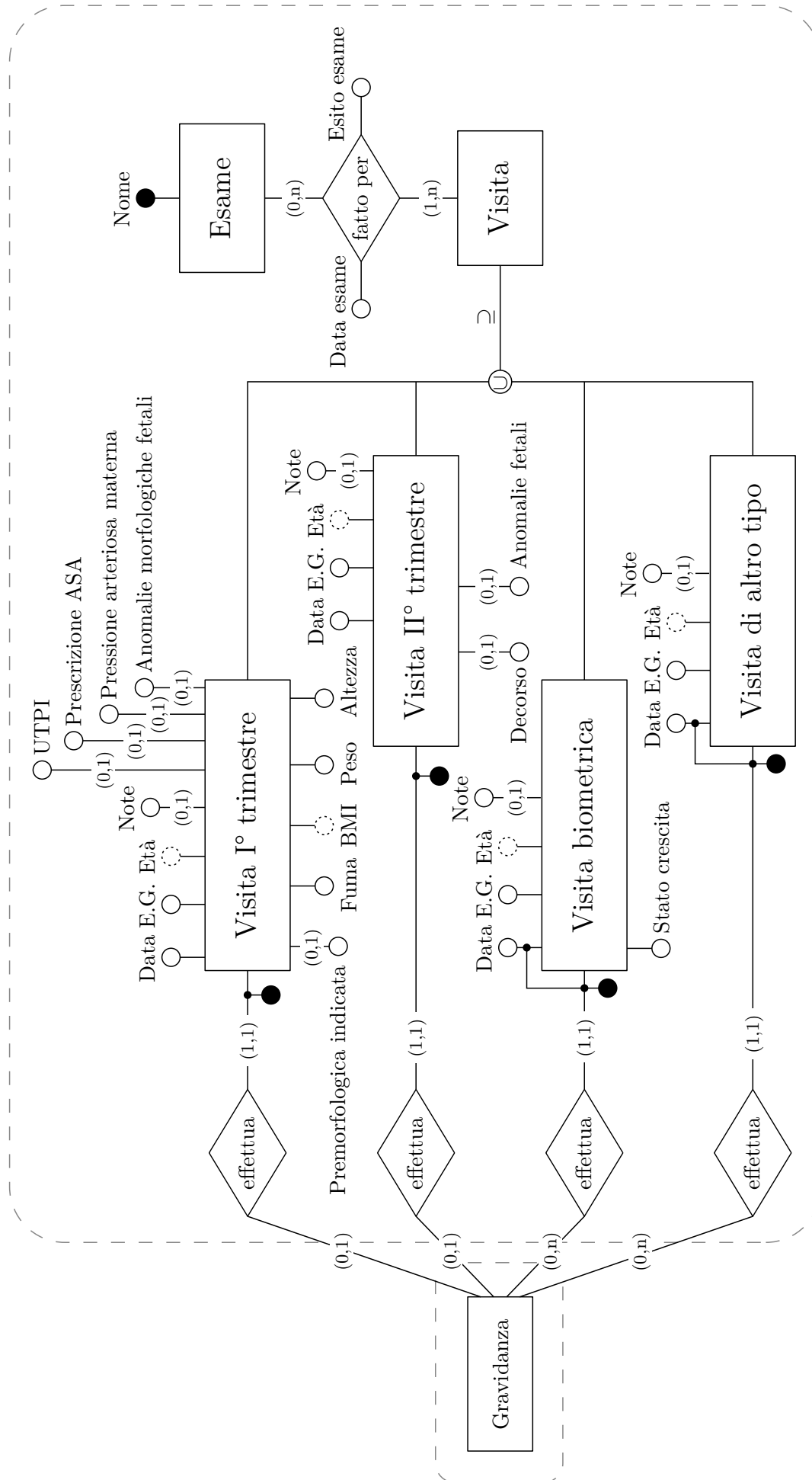


Figura 3.3: Schema Entità-Relazione. Ingrandimento dell'area visite.

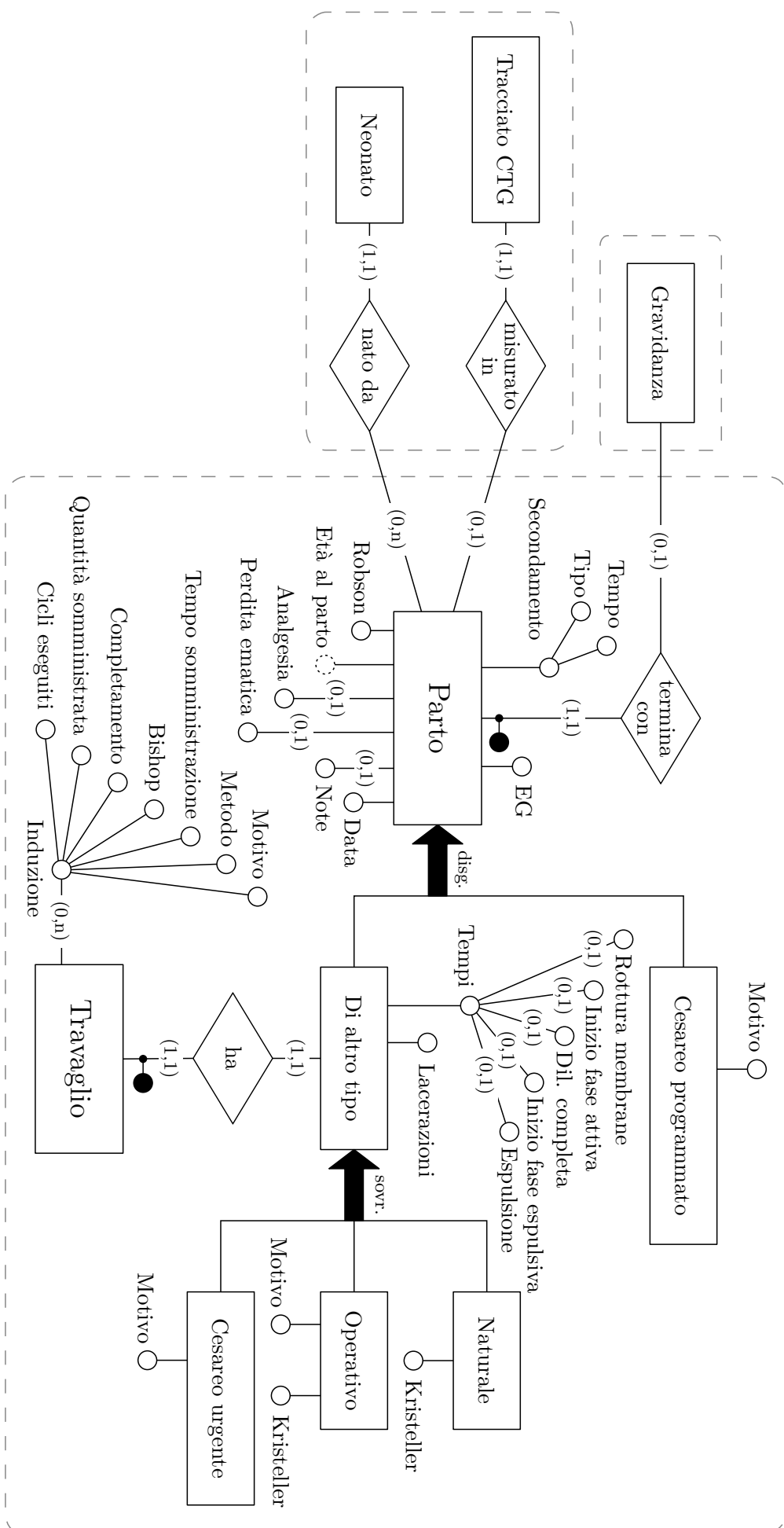


Figura 3.4: Schema Entità-Relazione. Ingrandimento dell'area parto.

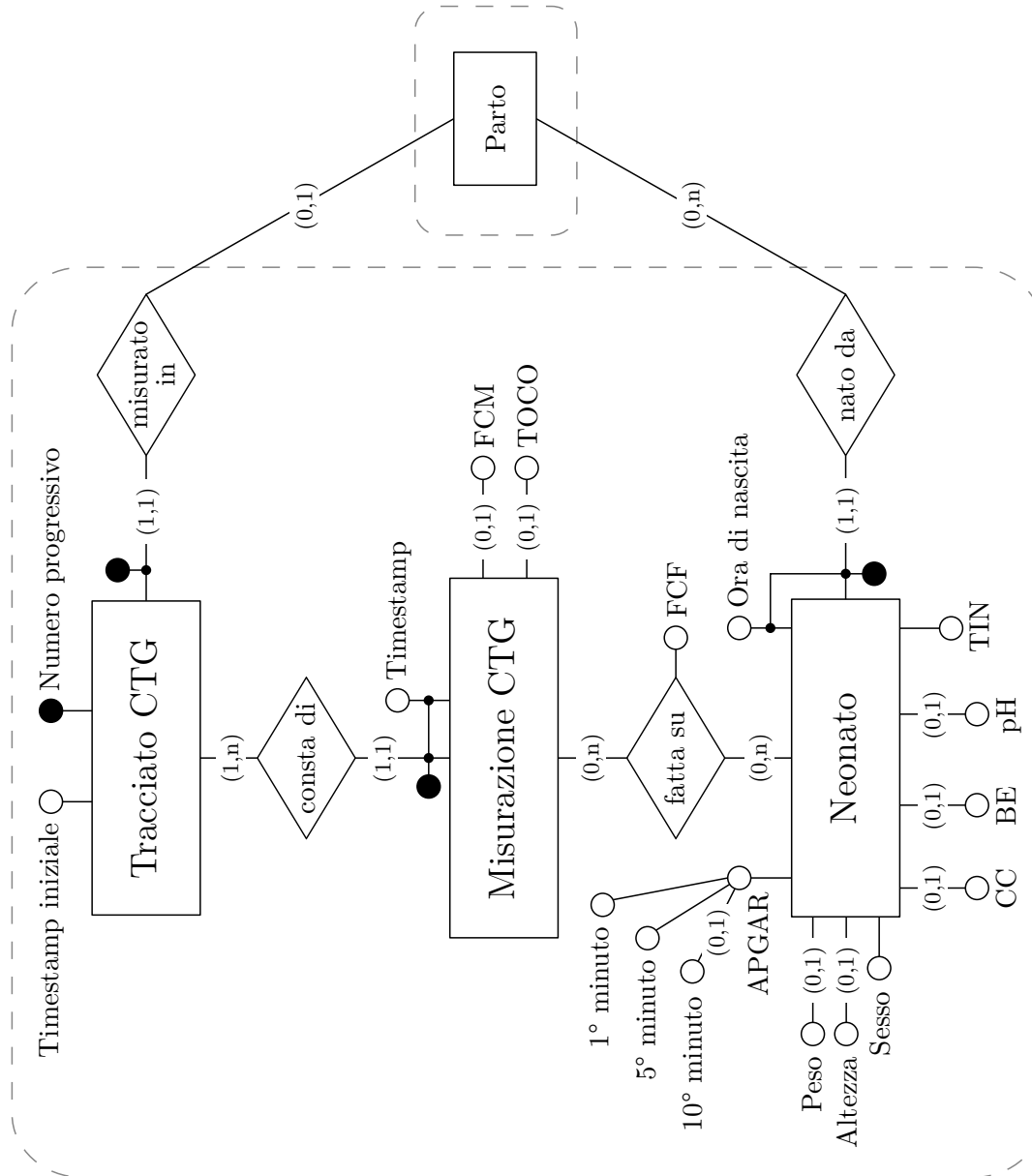


Figura 3.5: Schema Entità-Relazione. Ingrandimento dell'area neonato.

4

Progettazione logica

Lo schema concettuale presentato nel Capitolo 3 rappresenta in modo accurato il dominio a livello astratto ma la traduzione in uno schema logico nel modello relazionale richiede la semplificazione di alcune strutture concettuali. In seguito a questo passaggio di ristrutturazione si definisce uno schema logico, costituito da un insieme di relazioni con eventuali vincoli posti su di esse.

4.1 Ristrutturazione dello schema Entità-Relazione

Il formalismo del modello Entità-Relazione permette di utilizzare strutture piuttosto complesse a livello concettuale, come ad esempio le relazioni di specializzazione o gli attributi multivalore. Per facilitare la traduzione in uno schema logico nel modello relazionale è necessario operare una ristrutturazione o semplificazione dello schema iniziale [1], con l'obiettivo di ottenere soltanto entità senza specializzazioni o categorie, attributi semplici e relazioni binarie. Nella fase di ristrutturazione si valuta se è opportuno mantenere gli attributi ridondanti per motivi di efficienza e si scelgono gli attributi chiave delle entità che hanno più candidati possibili.

Lo schema ristrutturato, mostrato in Figura 4.1, rimane espresso nel modello Entità-Relazione, ma non si può più considerare puramente concettuale perché introduce modifiche dettate non dalle caratteristiche del dominio ma da aspetti implementativi e ottimizzazione delle prestazioni. Dopo la ristrutturazione, la traduzione può essere fatta in modo meccanico perché i costrutti più elementari del modello Entità-Relazione hanno dei corrispettivi diretti nel modello relazionale.

4.1.1 Ristrutturazione dell'area concettuale della gravidanza

La Figura 4.2 mostra l'ingrandimento dello schema Entità-Relazione ristrutturato per quanto riguarda le entità Paziente, Gravidanza e Malattia. Di queste solo Gravidanza subisce delle modifiche rispetto allo schema concettuale.

Gravidanza

Nell'entità Gravidanza semplifichiamo gli attributi composti in attributi semplici, legandoli quindi direttamente alla gravidanza. Viene mantenuto il vincolo posto sulla non decrescenza o crescita stretta dei componenti dell'attributo Parità.

Gravidanza è entità debole ed è identificata dalla relazione con Paziente e nello schema concettuale sono state identificate più possibili chiavi parziali, ovvero la parità e la data di primo ingresso. Considerando che la maggior parte delle entità dell'intero schema sono deboli, direttamente o indirettamente, rispetto alla gravidanza, si otterrà che nello schema logico (vedi Sezione 4.2) in quasi tutte le tabelle sarà presente una chiave esterna che fa riferimento alla gravidanza. Per questo motivo, al fine di alleggerire il numero di colonne presenti in ciascuna tabella e facilitare l'interrogazione, introduciamo un valore numerico identificativo come attributo chiave dell'entità Gravidanza ristrutturata.

La parità e la data del primo ingresso, in quanto chiavi candidate nel modello concettuale, mantengono le proprietà di chiave e quindi viene posto un ulteriore vincolo di unicità¹.

- La quadrupla di attributi che compongono la parità deve essere unica per ogni paziente.
- La data di primo ingresso deve essere unica per ogni paziente.

L'attributo composto relativo alla PMA viene scomposto in due attributi, Tipo e Ovodonazione, su cui si pone un ulteriore vincolo.

- L'attributo Tipo PMA è presente se e solo se è presente l'attributo Ovodonazione PMA.

4.1.2 Ristrutturazione dell'area concettuale delle visite

La Figura 4.3 mostra l'ingrandimento dello schema Entità-Relazione ristrutturato per quanto riguarda le entità relative alle visite e all'entità Esame. Quest'ultima non subisce modifiche nella ristrutturazione.

Visite

Considerando la presenza di molti attributi in comune, le diverse entità coinvolte nella categoria Visita collassano in un'unica entità. Si introduce quindi un attributo ulteriore che indica a quale delle quattro classi di visite appartiene ogni istanza dell'entità Visita.

L'entità Visita rimane debole rispetto a Gravidanza, come erano tutte le diverse visite presenti nello schema concettuale, introducendo come chiave parziale la data per tutte. L'attributo “Anomalie morfologiche fetali” della visita del primo trimestre e l'equivalente “Anomalie fetali” della visita del secondo trimestre confluiscono in un unico attributo. L'attributo BMI, calcolabile facilmente a partire da peso e altezza, è ridondante e si può rimuovere.

Come conseguenza della semplificazione della categoria, la relazione che Visita ha con Esame modifica la partecipazione rendendola parziale, dato che non è richiesto che ad ogni visita vengano effettuati esami. Le diverse relazioni che legavano le entità originali a Gravidanza si riducono a una sola relazione sulla quale si pongono nuovi vincoli per rispettare la cardinalità imposta dalle relazioni originali.

- Ogni gravidanza può effettuare al massimo una sola visita del primo trimestre.
- Ogni gravidanza può effettuare al massimo una sola visita del secondo trimestre.

¹Il vincolo di non avere valore *NULL* è già espresso dall'assenza dell'indicazione (0,1) su tutti gli attributi, quindi nessuno di essi è opzionale.

4.1.3 Ristrutturazione dell'area concettuale del parto

La Figura 4.4 mostra l'ingrandimento dello schema Entità-Relazione ristrutturato per quanto riguarda le entità parto, con le sue specializzazioni, e Travaglio. Entrambe subiscono modifiche rispetto allo schema originale.

Parto

Nello schema concettuale originale l'entità Parto è specializzata in più entità su due livelli. Consideriamo il primo livello di specializzazione, articolata in parti cesarei programmati e parti di altro tipo, che è totale e disgiunta: ristrutturiamo la specializzazione con un'entità Parto, che contiene gli attributi comuni, e le due entità Cesareo programmato e Parto con travaglio, deboli rispetto a Parto, che contengono gli attributi specifici. Per mantenere le proprietà previste dalla specializzazione, ovvero totalità e disgiunzione, è necessario introdurre un vincolo ulteriore.

- Ogni parto deve essere uno e uno solo tra cesareo programmato e parto con travaglio.

Il secondo livello di specializzazione, che distingue parti naturali, operativi e cesarei urgenti, è totale e sovrapposta, con solo pochi attributi che non sono in comune a tutte le sottoentità. Eliminiamo quindi la specializzazione, legando anche gli attributi specifici delle sottoentità a Parto con travaglio. Si impongono i seguenti vincoli.

- L'attributo Motivo è presente se e solo se il parto è operativo o cesareo.
- L'attributo Kristeller è presente se e solo se il parto è naturale o operativo.

Gli attributi composti sia nell'entità Parto (Secondamento) sia nell'entità Parto con travaglio (Tempi) vengono scomposti nei relativi attributi che li compongono.

Travaglio e induzione

L'attributo composto e multivalore Induzione diventa un'entità a sé stante, con l'istante di somministrazione come chiave parziale. L'entità Travaglio rimane senza nessun attributo, quindi la rimuoviamo per legare direttamente Parto con travaglio e Induzione; l'entità Induzione diventa quindi debole rispetto a Parto con travaglio.

4.1.4 Ristrutturazione dell'area concettuale del neonato

La Figura 4.5 mostra l'ingrandimento dello schema Entità-Relazione ristrutturato per quanto riguarda le entità Neonato, Tracciato CTG e Misurazione CTG. Le differenze rispetto allo schema originale sono minime: nell'entità Neonato l'attributo composto APGAR viene diviso nelle sue componenti che vengono collegate direttamente a Neonato, mentre nell'entità Tracciato CTG si sceglie come chiave la relazione identificante con l'entità Parto.

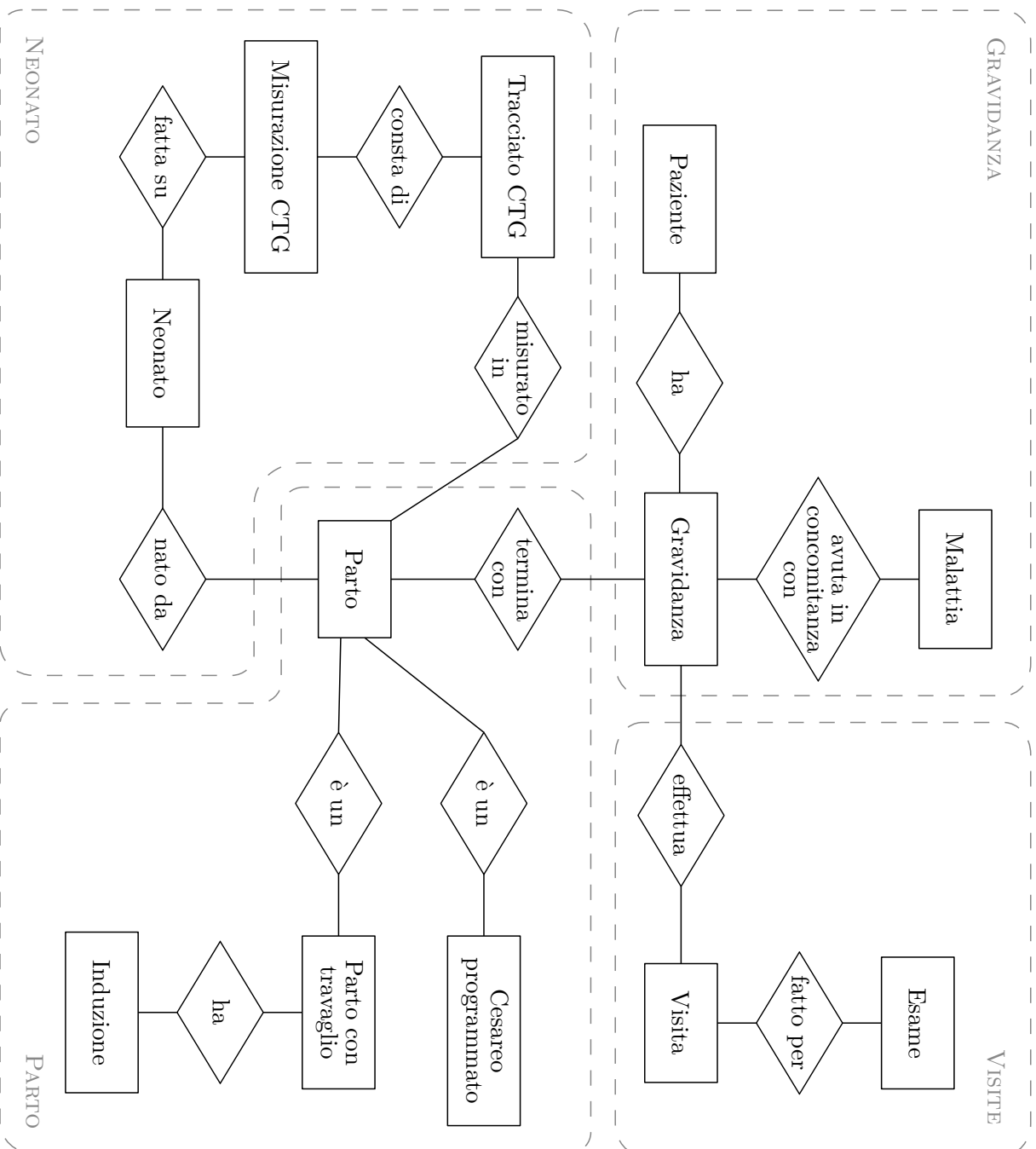


Figura 4.1: Schema Entità-Relazione ristrutturato. Visione d'insieme semplificata, contenente soltanto entità e relazioni del modello. Le aree tratteggiate delimitano gli stessi ingrandimenti dello schema concettuale originale.

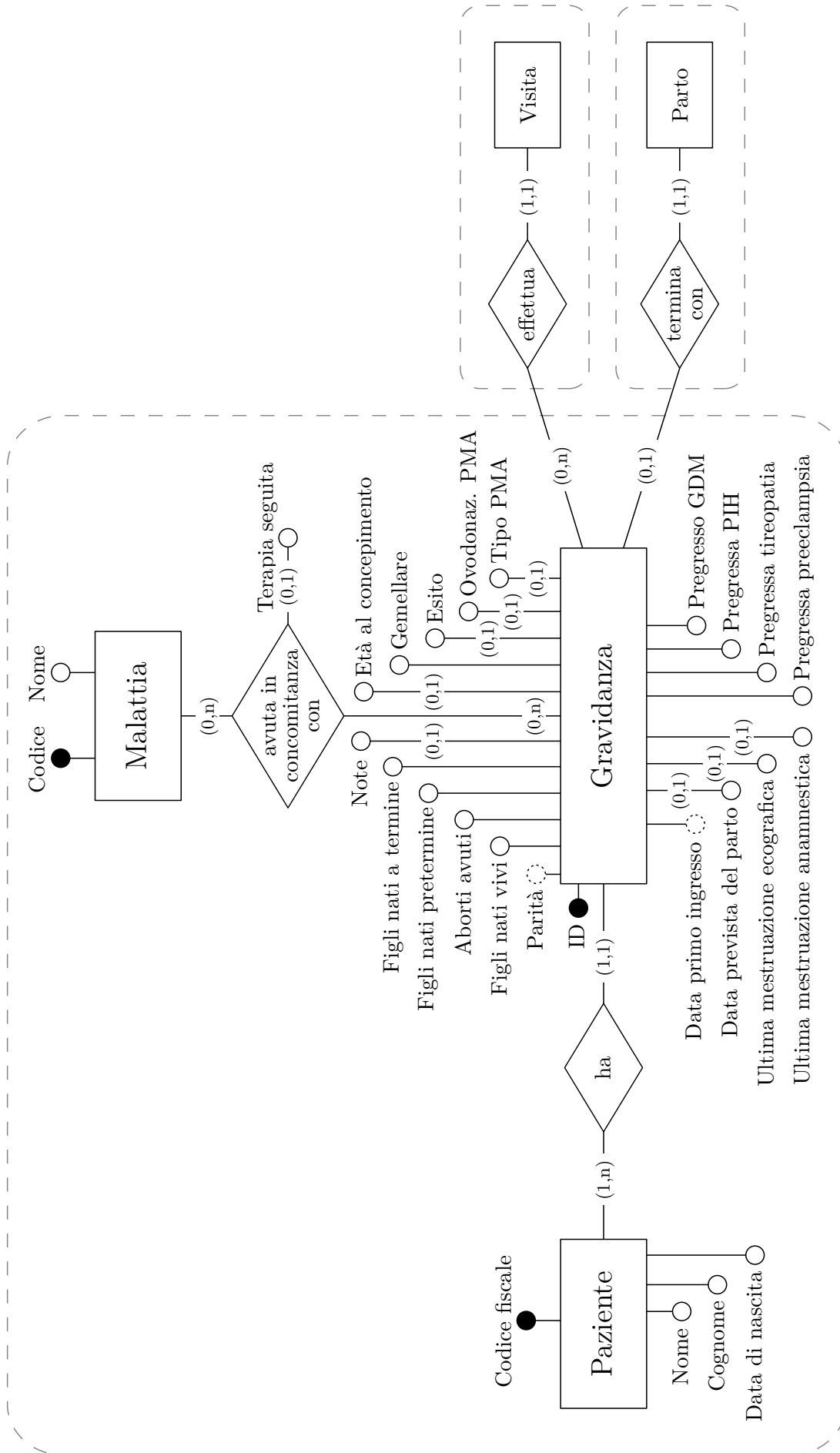


Figura 4.2: Schema Entità-Relazione ristrutturato. Ingrandimento dell'area gravidanza.

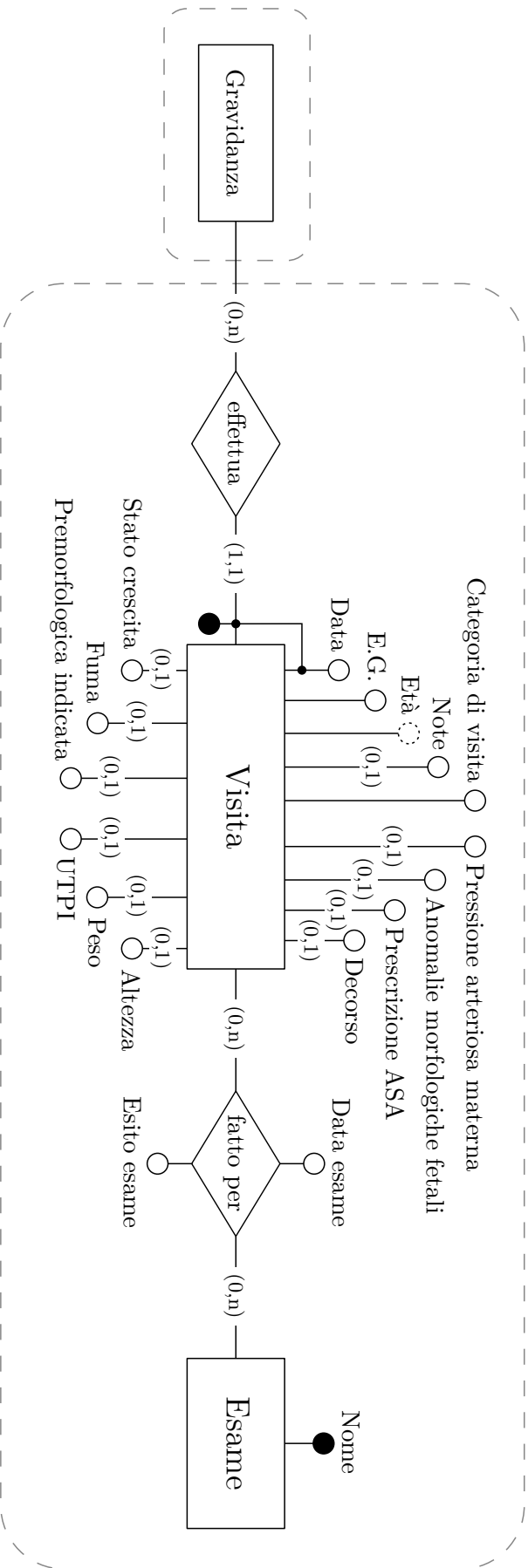


Figura 4.3: Schema Entità-Relazione ristrutturato. Ingrandimento dell'area visite.

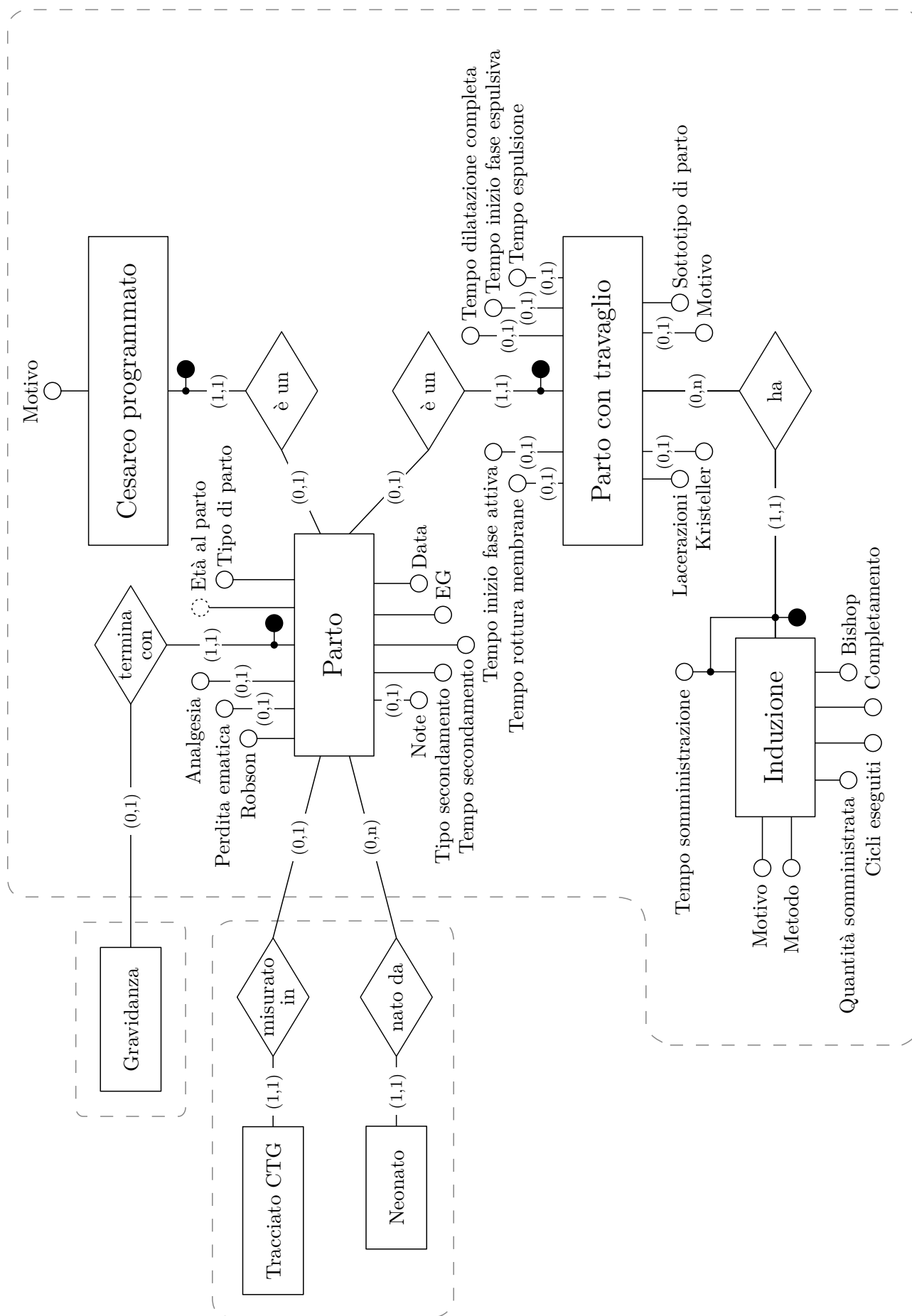


Figura 4.4: Schema Entità-Relazione ristrutturato. Ingrandimento dell'area parto.

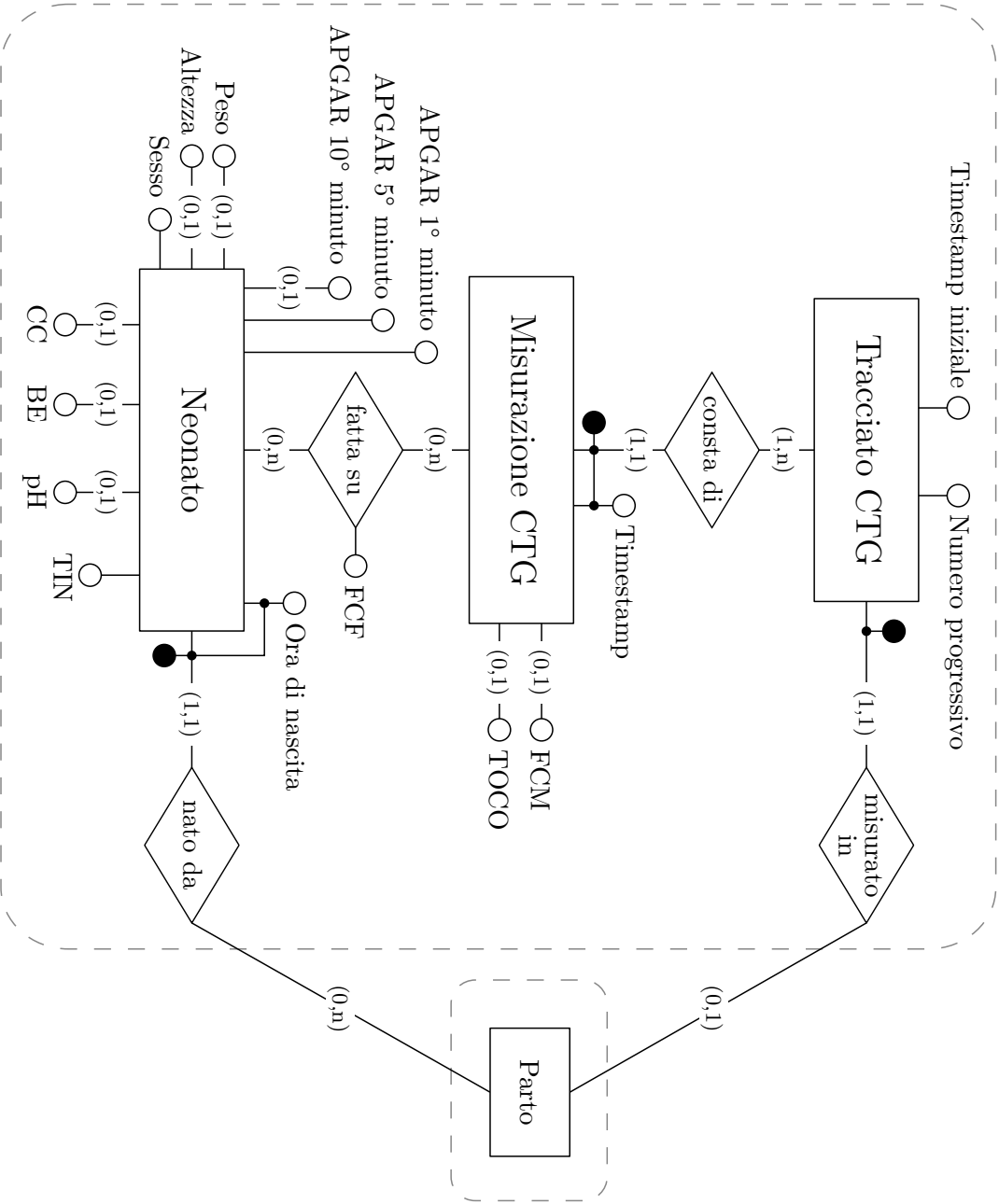


Figura 4.5: Schema Entità-Relazione ristrutturato. Ingrandimento dell'area neonato.

4.2 Schema logico relazionale

Lo schema logico è un insieme di relazioni composte di attributi, corrispondenti concettualmente a tabelle strutturate in colonne; ogni relazione può essere istanziata come insieme di tuple, corrispondenti all'insieme dei record nelle tabelle. Tra diverse tabelle possono sussistere vincoli di chiave esterna, ovvero situazioni in cui ogni istanza di un dato attributo di una relazione deve essere presente come istanza di attributo dell'altra relazione coinvolta.

La traduzione da schema concettuale ristrutturato (Sezione 4.1) a schema logico è molto lineare perché il primo dei due viene elaborato appositamente, a partire dallo schema concettuale originale (Sezione 3), per avvicinarsi alla modellazione tipica degli schemi relazionali. In generale tutte le entità vengono mantenute con i rispettivi attributi, andando a costituire “relazioni”; tra le relazioni concettuali, invece, solo le relazioni molti a molti costituiscono relazioni logiche vere e proprie, mentre le altre fanno confluire i loro attributi nelle tabelle delle entità dal lato in cui hanno molteplicità uno.

La Figura 4.6 mostra uno schema relazionale di massima contenente soltanto i nomi delle relazioni e i vincoli di chiave esterna tra relazioni. Gli schemi relazionali usano la seguente sintassi: gli attributi chiave sono indicati con l'icona della chiave; gli attributi opzionali sono indicati in corsivo; i vincoli di chiave esterna sono rappresentati con la notazione *crow's foot*².

4.2.1 Area della gravidanza

Le entità Paziente, Malattia e Gravidanza costituiscono tabelle a sé stanti. La relazione “ha” viene conservata nell'attributo `paziente_cf` della tabella `gravidanza`, mentre la relazione “avuta in concomitanza con”, che lega gravidanze e malattie, viene memorizzata in una tabella separata. Sintetizziamo di seguito i vincoli precedentemente posti sullo schema concettuale (Sezione 3 e Sezione 4.1) e riformulati come vincoli sullo schema logico.

- La coppia (`paziente_cf`, `data_primo_ingresso`) di `gravidanza` è unica.
- La tupla (`paziente_cf`, `figli_nati_vivi`, `figli_nati_pretermine`, `figli_nati_a_termine`, `aborti_avuti`) di `gravidanza` è unica.
- Se una paziente ha più di una gravidanza, per ogni coppia di gravidanze successive i valori di `figli_nati_vivi`, `aborti_avuti`, `figli_nati_pretermine`, `figli_nati_a_termine` sono non decrescenti e almeno uno tra essi è crescente.
- In `gravidanza` l'attributo `pma_tipo` è presente se e solo se l'attributo `pma_ovodonazione` è presente.

4.2.2 Area delle visite

Le entità Visita e Esame costituiscono tabelle a sé stanti. La relazione molti a molti “fatto per” diventa una tabella separata che lega esami e visite.

²La notazione *crow's foot* indica la cardinalità delle relazioni tra entità (nel modello Entità-Relazione) o tra tabelle (nel modello relazionale). Una linea singola indica cardinalità 1, mentre il simbolo della “zampa di corvo”, che rappresenta idealmente un collegamento con più punti terminali, indica cardinalità N.

Introduciamo un attributo `tipo` alla relazione `esame` così ottenuta che descrive il tipo di dato dei valori di `esito` in `esame_visita`. Questo è utile a livello implementativo perché il tipo di dato dell'esito di un dato esame può variare in base all'esame stesso. L'attributo `categoria_visita` di `visita` ha uno dei seguenti valori: `primo_trimestre`, `secondo_trimestre`, `biometrica`, `altro_tipo`. L'attributo `stato_crescita` di `visita`, se non è `NULL`, ha uno dei seguenti valori: `regolare`, `fgr`, `sga`.

Riassumiamo di seguito i vincoli da porre su tale schema logico.

- Per ogni gravidanza, in `visita` esiste al massimo un solo record che ha valore `primo_trimestre` nel campo `categoria_visita`.
- Per ogni gravidanza, in `visita` esiste al massimo un solo record che ha valore `secondo_trimestre` nel campo `categoria_visita`.
- Per ogni gravidanza deve essere presente almeno un record in `visita` oppure in `parto`.
- Per ogni esame registrato per una data visita, l'esito dell'esame deve essere compatibile con il `tipo_esito` indicato nel record corrispondente nella tabella `esame`.

4.2.3 Area del parto

Le entità Parto, Cesareo programmato, Parto con travaglio e Induzione formano ciascuna una tabella a sé stante mentre tutte le relazioni vengono rimosse e si introduce come attributo chiave (o parte della chiave) l'identificatore della gravidanza in tutte le tabelle.

L'attributo `tipo_parto` di `parto` deve essere uno tra `cesareo_programmato` e `parto_con_travaglio`. Di conseguenza si impone il seguente vincolo.

- Il valore di `tipo_parto` è `cesareo_programmato` se e solo se il parto ha un record corrispondente in `cesareo_programmato`.
- Il valore di `tipo_parto` è `parto_con_travaglio` se e solo se il parto ha un record corrispondente in `parto_con_travaglio`³.

L'attributo `sottotipo_parto` di `parto_con_travaglio` deve essere uno tra `naturale`, `operativo`, `cesareo` (dove “urgente” è sottinteso per semplicità di rappresentazione), `naturale_operativo`, `naturale_cesareo`, `operativo_cesareo`. Si impongono quindi i seguenti vincoli.

- L'attributo `motivo` di `parto_con_travaglio` è assente se e solo se `sottotipo_parto` ha valore `naturale`.
- L'attributo `kristeller` di `parto_con_travaglio` è assente se e solo se `sottotipo_parto` ha valore `cesareo`.

³Questo vincolo è equivalente e soddisfa il seguente: “Ogni Parto è uno e uno solo tra cesareo programmato e parto con travaglio”.

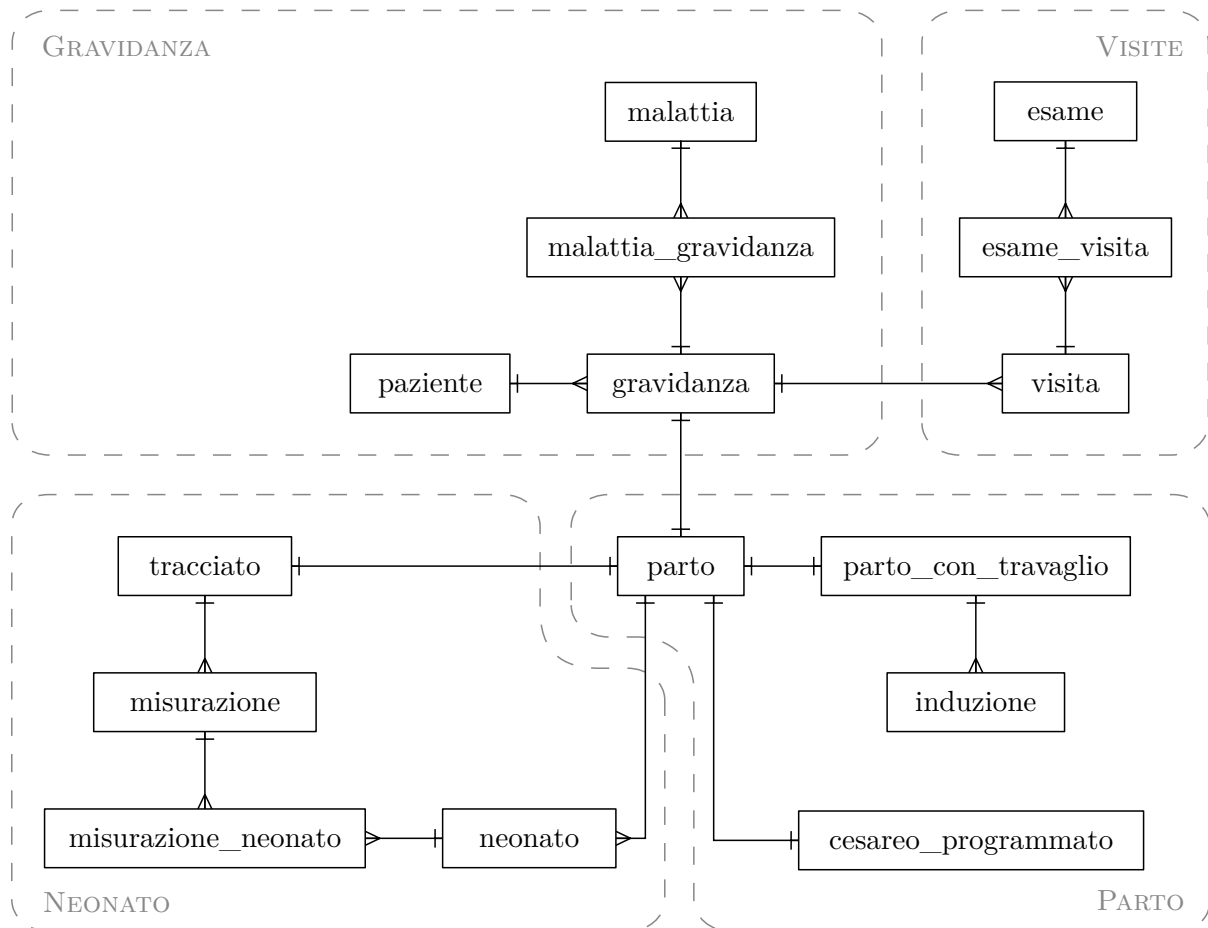


Figura 4.6: Schema relazionale. Visione d'insieme semplificata, contenente soltanto i nomi delle relazioni o tabelle dello schema e le relazioni di chiave esterna.

4.2.4 Area del neonato

Le entità Neonato, Tracciato e Misurazione costituiscono tabelle a sé stanti, mentre solo la relazione “fatta su” diventa una tabella a parte (con il nome di `misurazione_neonato`) e le altre vengono invece accorpate alle altre tabelle.

Il ciclo di relazioni che si forma a livello concettuale, comprendente le entità Parto, Neonato, Misurazione e Tracciato, non crea problemi di integrità perché la relazione di “contatto”, ovvero “fatta su”, coinvolge entità che sono tutte deboli rispetto a Parto; ciò permette di includere questa proprietà imponendo il vincolo di chiave della tabella `misurazione_neonato`.

È necessario imporre un ultimo vincolo.

- Per ogni misurazione deve essere presente almeno uno tra l'attributo `valore_fcm`, l'attributo `valore_toco` e un record nella tabella `misurazione_neonato`.

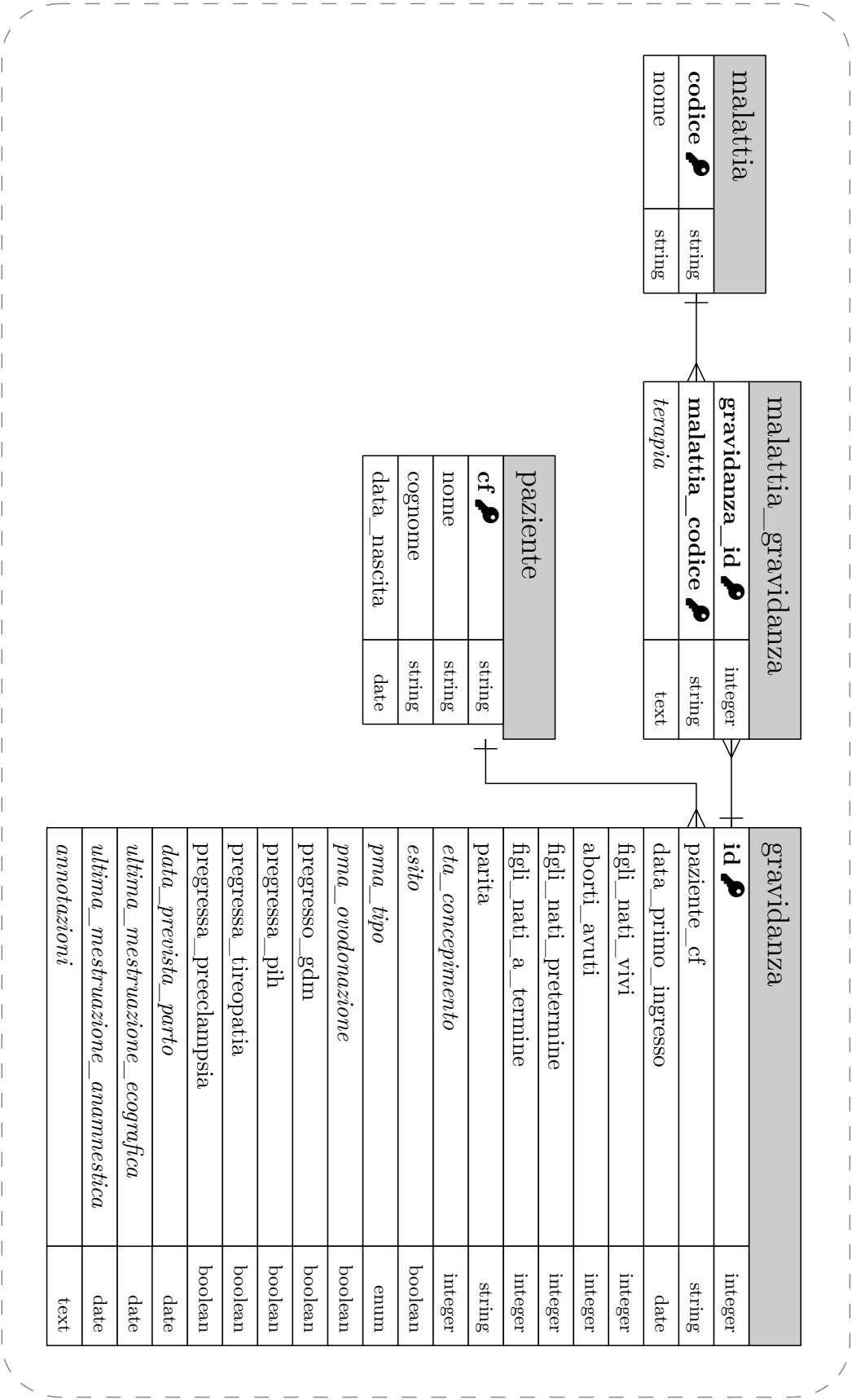


Figura 4.7: Schema relazionale. Ingrandimento dell'area gravidanza.

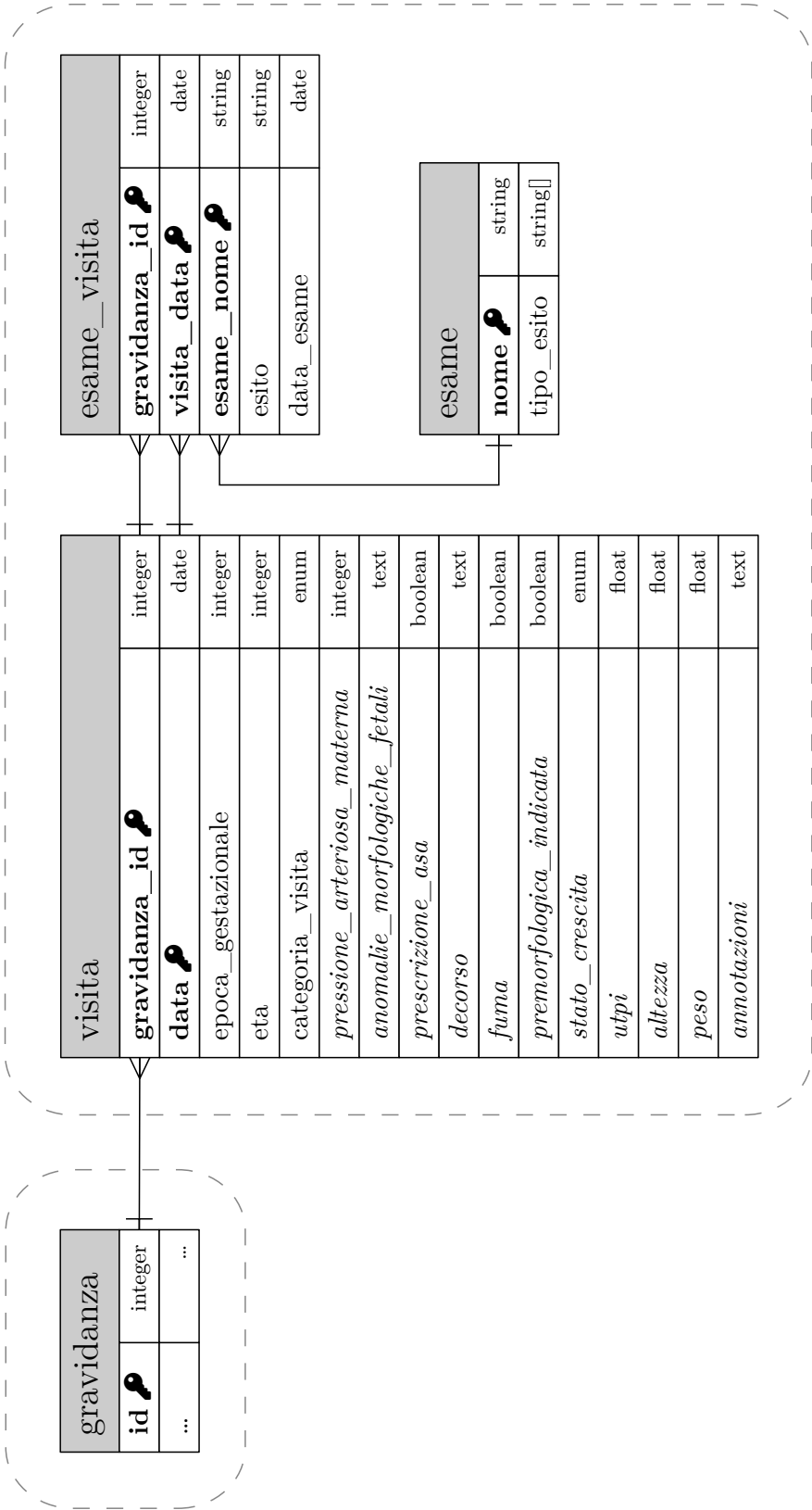


Figura 4.8: Schema relazionale. Ingrandimento dell'area visite.

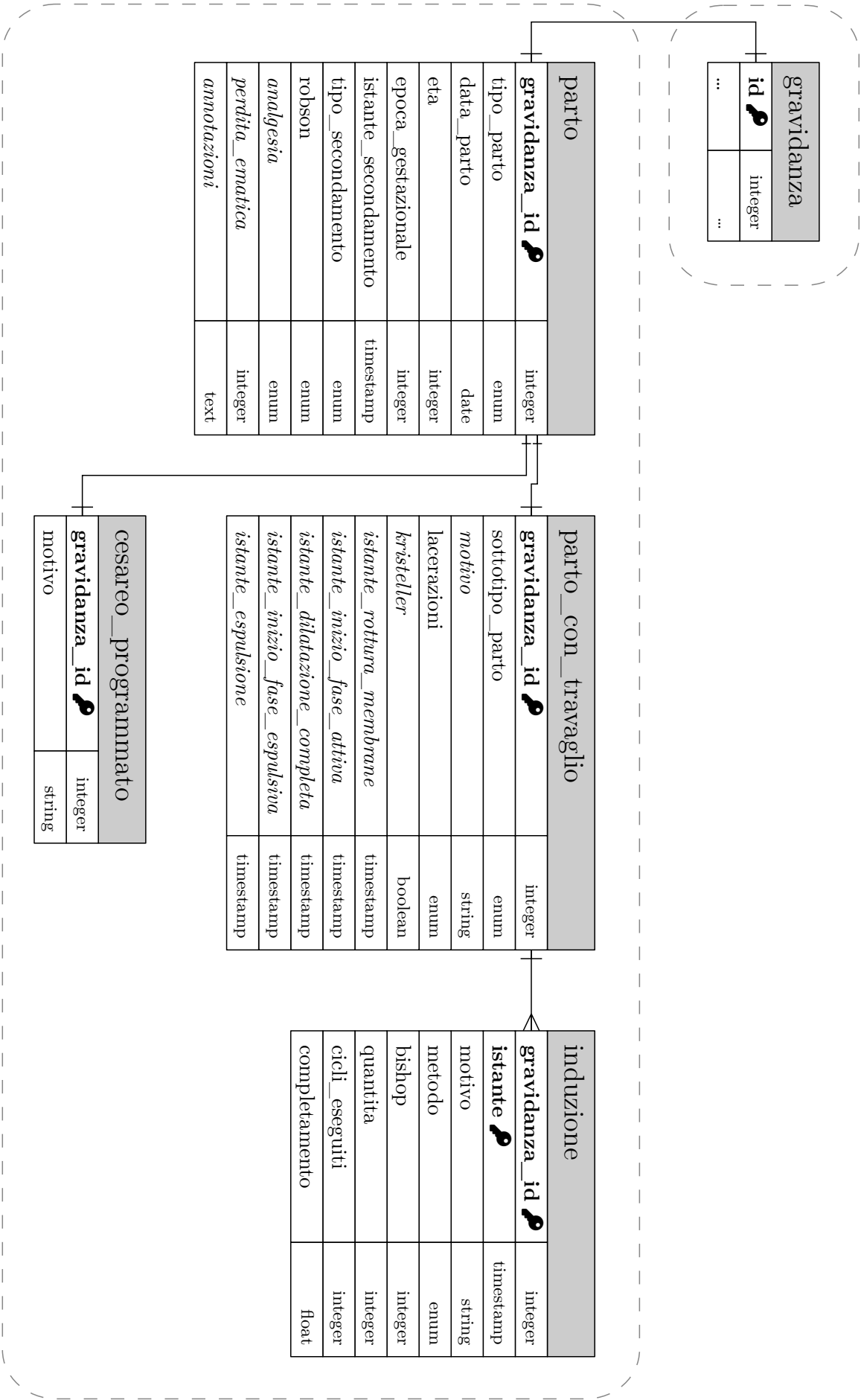


Figura 4.9: Schema relazionale. Ingrandimento dell'area parto.

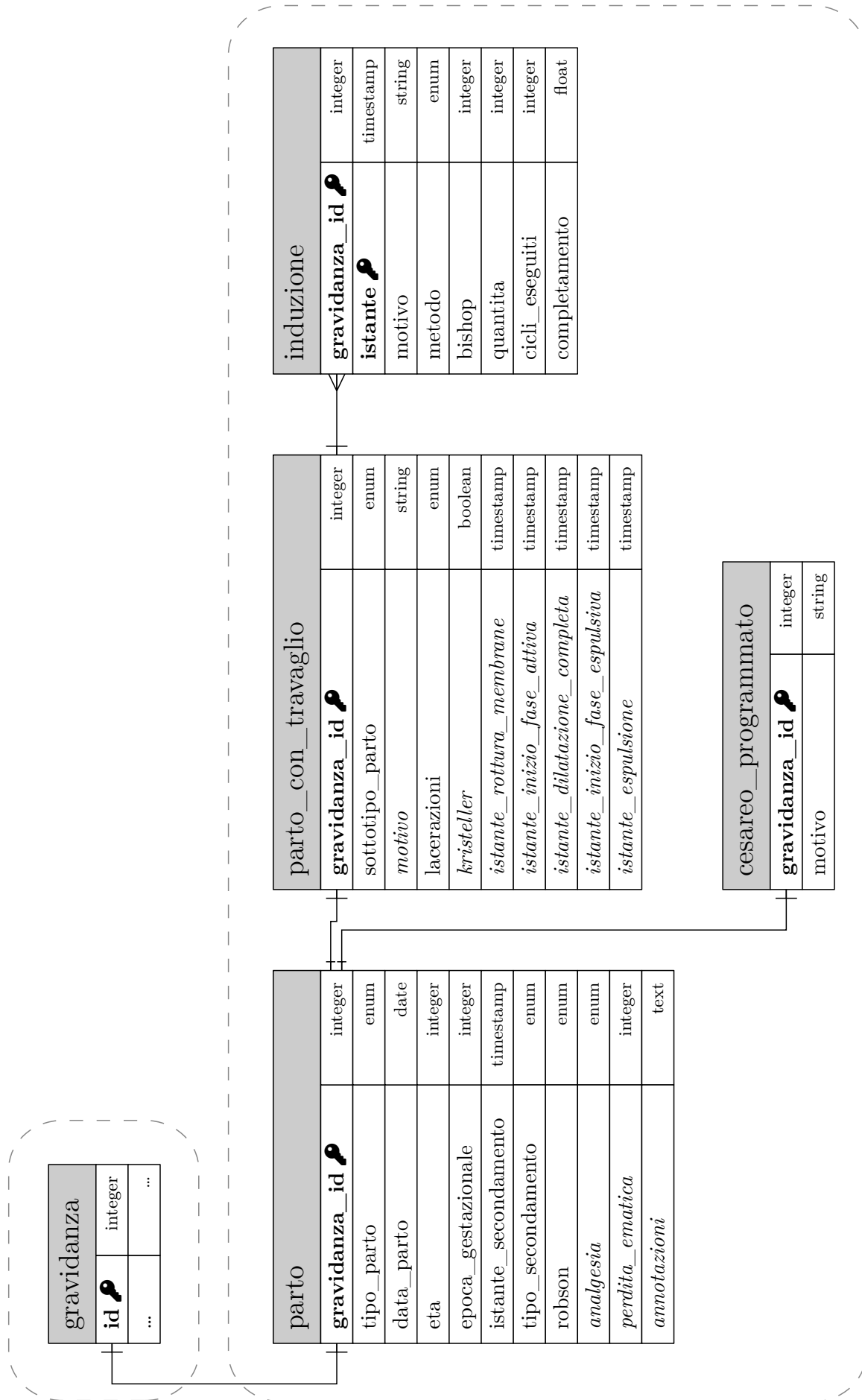


Figura 4.10: Schema relazionale. Ingrandimento dell'area neonato.

Progettazione fisica

La fase di progettazione logica (Sezione 4.2) produce uno schema logico, formalizzato nel modello relazionale, che descrive l'insieme di tabelle e di vincoli su di esse che contengono i dati di interesse. Questo schema è interrogabile, ovvero è possibile formulare delle interrogazioni (anche dette *query* o richieste) sulla base di dati; le interrogazioni si formulano attraverso linguaggi formali specifici per il modello relazionale come algebra relazionale e calcolo relazionale.

Il linguaggio SQL (*Structured Query Language*) è un linguaggio di interrogazione per basi di dati relazionali [11], affermatosi come standard *de facto* (grazie alle numerose implementazioni nei sistemi distribuiti commercialmente) e anche *de iure*¹. SQL è composto di diverse parti: è possibile esprimere sia la definizione delle tabelle (DDL, *Data Definition Language*) sia le operazioni di inserimento, modifica, cancellazione e interrogazione (DML, *Data Manipulation Language*), insieme anche ai vincoli di integrità e ai privilegi degli utenti che possono accedere alla base di dati.

In questo capitolo costruiamo fisicamente la base di dati progettata nei capitoli precedenti attraverso diversi *script* in linguaggio SQL. Il DBMS che utilizziamo è PostgreSQL, uno tra i più diffusi DBMS relazionali.

5.1 Definizione dei domini

Per quanto riguarda i tipi di dato semplici, in PostgreSQL il tipo corrispondente alle stringhe è **varchar** (*variable character*) e il tipo corrispondente ai numeri rappresentati in virgola mobile è **real**.

Alcune tabelle dello schema logico presentano attributi con vincoli sul dominio o con un insieme finito di valori possibili. In SQL è possibile definire domini, utilizzabili poi allo stesso modo di quelli predefiniti del linguaggio, che sintetizzano i vincoli da imporre ed evitano ripetizioni e quindi possibili inconsistenze.

I tipi enumerazione, ovvero che possono assumere un insieme finito e contenuto di valori, si implementano come interi (a cui però serve poi aggiungere una funzione di conversione a livello più alto per tradurre il numero nel valore semantico corrispondente) oppure come stringa (**varchar**) su cui viene imposto il vincolo di appartenenza all'insieme di valori del dominio; per la base di dati che stiamo costruendo scegliamo la seconda implementazione.

¹Il primo standard ANSI e ISO per il linguaggio SQL è del 1986 (SQL-86) [11]. Successivamente sono stati pubblicati diversi aggiornamenti fino alla versione attuale, SQL:2023 (ISO/IEC 9075:2023).

Elenchiamo di seguito i domini definiti per lo schema relazionale della base di dati (Listato 5.1)².

- Nella tabella `paziente` si utilizza come identificatore il codice fiscale italiano, ovvero una stringa alfanumerica di 16 caratteri.
- Nella tabella `gravidanza` l'attributo `pma_tipo` contiene l'informazione relativa al tipo di procreazione medicalmente assistita, che può avere valore `iui`, `fivet` o `icsi`.
- Nella tabella `visita` l'attributo `categoria_visita`, relativo al tipo di visita, può avere valore `primo_trimestre`, `secondo_trimestre`, `biometrica` o `altro_tipo`. Nella stessa tabella, l'attributo `stato_crescita` può avere valore `regolare`, `fgr` o `sga`.
- Per la tabella `parto` è richiesta la dichiarazione di diversi domini. L'attributo `tipo_parto` indica il tipo di parto e può avere valore `cesareo_programmato` o `parto_con_travaglio`. L'attributo `tipo_secondamento` può avere valore `attivo`, `strumentale`, `manuale` o `scivolamento`. L'attributo `robson` può avere valore 1, 2a, 2b, 3, 4a, 4b, 5.1, 5.2, 6, 7, 8, 9 o 10. L'attributo `analgesia` può avere valore `spinale`, `peridurale`, `spinale_peridurale`, `calinox`.
- Nella tabella `parto_con_cesareo` l'attributo `sottotipo_parto` può avere valore uguale a `naturale`, `operativo`, `cesareo`, `naturale_operativo`, `naturale_cesareo`, `operativo_cesareo`, mentre l'attributo `lacerazioni` può avere valore `episiotomia`, `tracheloraffia`, `nessuna`, `grado_1`, `grado_2`, `grado_3`, `grado_4`, `altro`.
- Nella tabella `induzione` l'attributo `metodo_induzione` può avere valore `amnioressi`, `cook`, `cook_misoprostolo`, `cook_ossitocina`, `dilapan`, `dilapan_misoprostolo`, `dilapan_ossitocina`, `misoprostolo`, `propess` o `propidil`.

²Tutti i frammenti di codice mostrati nei listati di questa tesi sono disponibili anche nella *repository* al link <https://github.com/andr3wsm/bach/tree/main/sql-scripts>.

```

-- Dominio codice_fiscale per persona
create domain codice_fiscale as char(16);
-- Dominio pma_tipo_enum per +gravidanza
create domain pma_tipo_enum as varchar
check (value in ('iui','fivet','icsi'));
-- Dominio categoria_visita_enum per visita
create domain categoria_visita_enum as varchar
check (value in ('primo_trimestre','secondo_trimestre','biometrica',
  'altro_tipo'));
-- Dominio stato_crescita_enum per visita
create domain stato_crescita_enum as varchar
check (value in ('regolare','fgr','sga'));
-- Dominio tipo_parto_enum per parto
create domain tipo_parto_enum as varchar
check (value in ('cesareo_programmato','parto_con_travaglio'));
-- Dominio tipo_secondamento_enum per parto
create domain tipo_secondamento_enum as varchar
check (value in ('attivo','strumentale','manuale','scovolamento'));
-- Dominio robson_enum per parto
create domain robson_enum as varchar
check (value in ('1','2a','2b','3','4a','4b','5.1','5.2',
  '6','7','8','9','10'));
-- Dominio analgesia_enum per parto
create domain analgesia_enum as varchar
check (value in ('spinale','peridurale','spinale_peridurale',
  'calinox'));
-- Dominio sottotipo_parto_enum per parto_con_travaglio
create domain sottotipo_parto_enum as varchar
check (value in ('naturale','operativo','cesareo','naturale_operativo',
  'naturale_cesareo','operativo_cesareo'));
-- Dominio lacerazioni_enum per parto_con_travaglio
create domain lacerazioni_enum as varchar
check (value in ('nessuna','episiotomia','tracheloraffia','grado_1',
  'grado_2','grado_3','grado_4','altro'));
-- Dominio metodo_induzione per induzione
create domain metodo_induzione as varchar
check (value in ('amnioressi','cook','cook_misoprostolo','cook_ossitocina',
  'dilapan','dilapan_misoprostolo','dilapan_ossitocina','misoprostolo',
  'propess','propidil'));

```

Listato 5.1: Definizione dei domini.

5.2 Definizione delle tabelle

Definiamo con *script* SQL tutte le tabelle presentate nello schema logico (Sezione 4.2). Insieme agli attributi vengono introdotti anche tutti i vincoli di chiave esterna e alcuni vincoli aggiuntivi che si applicano ad attributi di un singolo record. Rimarranno da definire separatamente i vincoli che coinvolgono dati memorizzati su più tabelle o che verificano condizioni su dati aggregati.

Nei vincoli di chiave esterna (**foreign key**) si indica la tabella e l'attributo o la tupla di attributi a cui si fa riferimento (**references**). Le opzioni **on update cascade** e **on delete cascade** indicano che determinate operazioni alla chiave primaria della tabella puntata dal riferimento vengono riprodotte a cascata sulla tabella dipendente, rispettivamente nei casi di modifiche e di cancellazioni.

5.2.1 Area della gravidanza

La tabella **paziente** (Listato 5.2) utilizza il dominio **codice_fiscale** definito precedentemente.

Nella tabella **gravidanza** (Listato 5.3) è stato introdotto come chiave primaria un attributo **id**, da intendersi come numero progressivo; un'implementazione tipica e di “basso livello” degli identificatori di questo tipo prevede di dichiarare l'attributo come intero e chiave primaria, forzando l'utente a determinare il valore da assegnare a **id** ad ogni inserimento, solitamente calcolando il massimo dei valori **id** presenti nella tabella e aggiungendo 1. In PostgreSQL [13] è possibile ovviare a tale ostacolo definendo il campo **id** come **serial**: non è più necessario indicare né calcolare manualmente il valore **id** da assegnare perché viene determinato all'occorrenza, permettendo alle istruzioni di inserimento di esserne trasparenti. Nell'implementazione, **serial** è *zucchero sintattico* che si traduce nella definizione di una **sequence** di **integer** (a cui vengono dedicati 4 byte) ed esistono le varianti **smallserial** (2 byte, corrispondente a **smallint**) e **bigserial** (8 byte, corrispondente a **bigint**). Quando un valore dichiarato come **serial** viene usato come chiave esterna in altre tabelle deve essere trattato come il tipo **integer** corrispondente.

Nella definizione della tabella **gravidanza** vengono formalizzati i vincoli di unicità da imporre su tuple di attributi, che corrispondono alle chiavi candidate dello schema concettuale, esposti nella Sezione 4.2.1. Aggiungendo ulteriori vincoli a **gravidanza** si possono soddisfare l'integrità referenziale verso la tabella **paziente** e il vincolo relativo agli attributi **pma_tipo** e **pma_ovodonazione**, riportati sempre nella Sezione 4.2.1. Per il vincolo rimanente, relativo alle proprietà di crescita dei valori di parità, si definisce un *trigger* apposito (Sezione 5.3.2).

La tabella **malattia** (Listato 5.4) e la tabella **malattia_gravidanza** (Listato 5.5) seguono le definizioni riportate nello schema logico. Sulla seconda vengono imposti i vincoli di integrità referenziale.

5.2.2 Area delle visite

La tabella **visita** (Listato 5.6) utilizza i domini **categoria_visita_enum** e **stato_crescita_enum** definiti precedentemente; fatta eccezione per il vincolo di chiave esterna i diversi vincoli imposti su questa tabella (Sezione 4.2.2) vengono implementati attraverso *trigger* (Sezione 5.3.3).

Le tabelle **esame** (Listato 5.7) e **esame_visita** (Listato 5.8) sono definite come nello schema logico. Il vincolo imposto sul tipo dell'attributo **esito** della tabella **esame_visita** (Sezione 4.2.2) viene riformulato, considerando l'implementazione in SQL, come segue.

- Per ogni esame registrato per una data visita, si controlla la corrispondenza tra **esito** e il valore di **tipo_esito** della tabella **esame**. Se il valore è `{numeric}`³ il valore di **esito** deve essere la rappresentazione come stringa di un numero. Se il valore è `{string}` il valore può essere una stringa arbitraria. Se il valore è diverso da questi, ovvero è un array (di stringhe) di lunghezza arbitraria, i suoi elementi sono da intendersi come i possibili valori di un tipo enumerazione definito appositamente per quell'esame, quindi il valore di **esito** deve essere uno dei valori possibili di questo tipo.

5.2.3 Area del parto

Nella tabella **parto** (Listato 5.9) utilizziamo molti dei domini di tipo enumerazione definiti nella Sezione 5.1. Insieme ad essa definiamo le tabelle **cesareo_programmato** (Listato 5.10) e **parto_con_travaglio** (Listato 5.11) che contengono i dati delle entità specializzate, come da schema ristrutturato. Quest'ultima tabella deve soddisfare i vincoli sugli attributi **motivo** e **kristeller** indicati nella Sezione 4.2.3. Definiamo inoltre la tabella **induzione** (Listato 5.12).

I due vincoli rimanenti della Sezione 4.2.3, che impongono che ogni record di **parto** deve averne uno e uno solo corrispondente in una delle due tabelle **cesareo_programmato** e **parto_con_travaglio** vengono implementati con *trigger* (Sezione 5.3.4).

5.2.4 Area del neonato

Le tabelle **neonato** (Listato 5.13), **tracciato** (Listato 5.14), **misurazione** (Listato 5.15) e **misurazione_neonato** (Listato 5.16) seguono le definizioni previste dallo schema logico. Rimane da implementare un solo vincolo, l'unico della Sezione 4.2.4, che verrà fatto con *trigger* (Sezione 5.3.5).

³La notazione `{numeric}` indica un array composto del solo elemento `numeric`.

```
create table paziente (
  cf codice_fiscale,
  primary key (cf),
  nome varchar not null,
  cognome varchar not null,
  data_nascita date not null
);
```

Listato 5.2: Definizione della tabella **paziente**.

```

create table gravidanza (
  id serial,
  primary key (id),
  paziente_cf codice_fiscale not null,
  data_primo_ingresso date not null,
  unique (paziente_cf, data_primo_ingresso),
  figli_nati_vivi integer not null,
  aborti_avuti integer not null,
  figli_nati_pretermine integer not null,
  figli_nati_a_termine integer not null,
  unique (paziente_cf, figli_nati_vivi, aborti_avuti, figli_nati_pretermine,
    figli_nati_a_termine),
  parita varchar not null,
  eta_concepimento integer,
  esito boolean,
  pma_tipo pma_tipo_enum,
  pma_ovodonazione boolean,
  pregresso_gdm boolean not null,
  pregressa_pih boolean not null,
  pregressa_tireopatia boolean not null,
  pregressa_preeclampsia boolean not null,
  data_prevista_parto date,
  ultima_mestruazione_ecografica date,
  ultima_mestruazione_anamnestica date,
  annotazioni text,
  -- Vincolo di chiave esterna verso paziente
  foreign key (paziente_cf)
    references paziente (cf)
    on update cascade on delete cascade,
  -- Vincolo sugli attributi relativi alla PMA
  check ((pma_tipo is null and pma_ovodonazione is null)
    or (pma_tipo is not null and pma_ovodonazione is not null))
);

```

Listato 5.3: Definizione della tabella gravidanza.

```

create table malattia (
  codice varchar,
  primary key (codice),
  nome varchar not null
);

```

Listato 5.4: Definizione della tabella malattia.


```

create table malattia_gravidanza (
    gravidanza_id integer,
    malattia_codice varchar,
    primary key (gravidanza_id, malattia_codice),
    terapia text,
    -- Vincolo di chiave esterna verso gravidanza
    foreign key (gravidanza_id)
        references gravidanza (id)
        on update cascade on delete cascade,
    -- Vincolo di chiave esterna verso malattia
    foreign key (malattia_codice)
        references malattia (codice)
        on update cascade on delete cascade
);

```

Listato 5.5: Definizione della tabella `malattia_gravidanza`.

```

create table visita (
    gravidanza_id integer,
    data date,
    primary key (gravidanza_id, data),
    epoca_gestazionale integer not null,
    eta integer not null,
    categoria_visita categoria_visita_enum not null,
    pressione_arteriosa_materna integer,
    anomalie_morfologiche_fetali text,
    prescrizione_asa boolean,
    decorso text,
    fuma boolean,
    premorfologica_indicata boolean,
    stato_crescita stato_crescita_enum,
    utpi real,
    altezza real,
    peso real,
    annotazioni text,
    -- Vincolo di chiave esterna verso gravidanza
    foreign key (gravidanza_id)
        references gravidanza (id)
        on update cascade on delete cascade
);

```

Listato 5.6: Definizione della tabella `visita`.

```

create table esame (
    nome varchar,
    primary key (nome),
    tipo varchar[] not null
);

```

Listato 5.7: Definizione della tabella `esame`.

```

create table esame_visita (
  gravidanza_id integer,
  visita_data date,
  esame_nome varchar,
  primary key (gravidanza_id, visita_data, esame_nome),
  esito varchar not null,
  data_esame date not null,
  -- Vincolo di chiave esterna verso visita
  foreign key (gravidanza_id, visita_data)
    references visita (gravidanza_id, data)
    on update cascade on delete cascade,
  -- Vincolo di chiave esterna verso esame
  foreign key (esame_nome)
    references esame (nome)
    on update cascade on delete cascade
);

```

Listato 5.8: Definizione della tabella `esame_visita`.

```

create table parto (
  gravidanza_id integer,
  primary key (gravidanza_id),
  tipo_parto tipo_parto_enum not null,
  data_parto date not null,
  eta integer not null,
  epoca_gestazionale integer not null,
  istante_secondamento timestamp not null,
  tipo_secondamento tipo_secondamento_enum not null,
  robson robson_enum not null,
  analgesia analgesia_enum,
  perdita_ematica integer,
  annotazioni text,
  -- Vincolo di chiave esterna verso gravidanza
  foreign key (gravidanza_id)
    references gravidanza (id)
    on update cascade on delete cascade
);

```

Listato 5.9: Definizione della tabella `parto`.

```

create table cesareo_programmato (
  gravidanza_id integer,
  primary key (gravidanza_id),
  motivo varchar not null,
  -- Vincolo di chiave esterna verso parto
  foreign key (gravidanza_id)
    references parto (gravidanza_id)
    on update cascade on delete cascade
);

```

Listato 5.10: Definizione della tabella `cesareo_programmato`.

```

create table parto_con_travaglio (
  gravidanza_id integer,
  primary key (gravidanza_id),
  sottotipo_parto sottotipo_parto_enum not null,
  motivo varchar,
  lacerazioni lacerazioni_enum not null,
  kristeller boolean,
  istante_rottura_membrane timestamp,
  istante_inizio_fase_attiva timestamp,
  istante_dilatazione_completa timestamp,
  istante_inizio_fase_espulsiva timestamp,
  istante_espulsione timestamp,
  -- Vincolo di chiave esterna verso parto
  foreign key (gravidanza_id)
    references parto (gravidanza_id)
    on update cascade on delete cascade,
  -- Vincolo sull'attributo motivo
  check ((sottotipo_parto = 'naturale' and motivo is null)
    or (sottotipo_parto <> 'naturale' and motivo is not null)),
  -- Vincolo sull'attributo kristeller
  check ((sottotipo_parto = 'cesareo' and kristeller is null)
    or (sottotipo_parto <> 'cesareo' and kristeller is not null))
);

```

Listato 5.11: Definizione della tabella parto_con_travaglio.

```

create table induzione (
  gravidanza_id integer,
  istante timestamp,
  primary key (gravidanza_id, istante),
  motivo varchar not null,
  metodo metodo_induzione not null,
  bishop integer not null,
  quantita integer not null,
  cicli_eseguiti integer not null,
  completamento real check (completamento >= 0 and completamento <= 1),
  -- Vincolo di chiave esterna verso parto_con_travaglio
  foreign key (gravidanza_id)
    references parto_con_travaglio (gravidanza_id)
    on update cascade on delete cascade
);

```

Listato 5.12: Definizione della tabella induzione.

```

create table neonato (
  gravidanza_id integer,
  istante_nascita timestamp,
  primary key (gravidanza_id, istante_nascita),
  peso real,
  altezza real,
  sesso char not null check (sesso in ('M','F')),
  circonferenza_cranica real,
  be real,
  ph real,
  tin boolean not null,
  apgar_min_1 integer not null,
  apgar_min_5 integer not null,
  apgar_min_10 integer,
  -- Vincolo di chiave esterna verso parto
  foreign key (gravidanza_id)
    references parto (gravidanza_id)
    on update cascade on delete cascade
);

```

Listato 5.13: Definizione della tabella `neonato`.

```

create table tracciato (
  gravidanza_id integer,
  primary key (gravidanza_id),
  numero_progressivo varchar not null,
  istante_inizio timestamp not null,
  -- Vincolo di chiave esterna verso parto
  foreign key (gravidanza_id)
    references parto (gravidanza_id)
    on update cascade on delete cascade
);

```

Listato 5.14: Definizione della tabella `tracciato`.

```

create table misurazione (
  gravidanza_id integer,
  istante timestamp,
  primary key (gravidanza_id, istante),
  valore_fcm integer,
  valore_toco integer,
  -- Vincolo di chiave esterna verso tracciato
  foreign key (gravidanza_id)
    references tracciato (gravidanza_id)
    on update cascade on delete cascade
);

```

Listato 5.15: Definizione della tabella `misurazione`.

```

create table misurazione_neonato (
  misurazione_istante timestamp,
  gravidanza_id integer,
  neonato_istante_nascita timestamp,
  primary key (misurazione_istante, gravidanza_id, neonato_istante_nascita),
  -- Vincolo di chiave esterna verso misurazione
  foreign key (gravidanza_id, misurazione_istante)
    references misurazione (gravidanza_id, istante)
    on update cascade on delete cascade,
  -- Vincolo di chiave esterna verso neonato
  foreign key (gravidanza_id, neonato_istante_nascita)
    references neonato (gravidanza_id, istante_nascita)
    on update cascade on delete cascade
);

```

Listato 5.16: Definizione della tabella `misurazione_neonato`.

5.3 Definizione dei trigger

I *trigger* sono istruzioni che il DBMS esegue come effetto collaterale di modifiche alla base di dati [11]. Essi vengono attivati da eventi, eventualmente con condizioni che ne limitano l'avvio, e chiamano procedure che possono operare sui dati presenti nella tabella prima dell'esecuzione oppure dopo l'esecuzione del comando che ha scatenato l'evento.

Il funzionamento della base di dati è di tipo transazionale, ovvero permette di svolgere un'insieme di operazioni distinte come se fossero atomiche: ciò permette di eseguire interrogazioni, anche con *trigger* avviati come effetto collaterale, con la garanzia che alla loro conclusione le modifiche saranno permanenti (*commit*) e le invarianti dei vincoli saranno rispettate; nel caso di errori avvenuti all'interno dell'operazione atomica o in seguito a istruzioni specifiche, si effettua un *rollback* che annulla tutte le modifiche parziali e riporta il sistema nello stato precedente, in cui le invarianti valevano ancora.

Per permettere un corretto funzionamento dei *trigger*, ovvero per consentire alle operazioni interne alla transazione di poter violare temporaneamente i vincoli e di rimandare la verifica della loro validità solo alla fine della transazione, è necessario dichiararli utilizzando le parole chiave `deferrable initially deferred` e le transazioni dovranno includere il comando `set constraints all deferred`.

5.3.1 Sintesi dei vincoli da implementare con trigger

Elenchiamo di seguito i vincoli esplicitati insieme allo schema logico (Sezione 4.2) che non sono soddisfacenti con le sole definizioni delle tabelle, ovvero che richiedono dei *trigger* appositi per essere verificati.

1. Se una paziente ha più di una gravidanza, per ogni coppia di gravidanze successive i valori di `figli_nati_vivi`, `aborti_avuti`, `figli_nati_pretermine` e `figli_nati_a_termine` sono tutti non decrescenti e almeno uno tra essi è crescente (Sezione 4.2.1).
2. Per ogni gravidanza, in `visita` esiste al massimo un solo record che ha `primo_trimestre` nel campo `categoria_visita` (Sezione 4.2.2).

3. Per ogni gravidanza, in `visita` esiste al massimo un solo record che ha `secondo_trimestre` nel campo `categoria_visita` (Sezione 4.2.2).
4. Per ogni gravidanza deve essere presente almeno un record in `visita` oppure in `parto` (Sezione 4.2.2).
5. Per ogni esame registrato per una data visita, si controlla la corrispondenza tra `esito` e il valore di `tipo_esito` della tabella `esame`. Se il valore è `{numeric}` il valore di `esito` deve essere la rappresentazione come stringa di un numero. Se il valore è `{string}` il valore può essere una stringa arbitraria. Se il valore è diverso da questi, ovvero è un array (di stringhe) di lunghezza arbitraria, i suoi elementi sono da intendersi come i possibili valori di un tipo enumerazione definito appositamente per quell'esame, quindi il valore di `esito` deve essere uno dei valori possibili di questo tipo (Sezione 5.2.2)
6. Il valore di `tipo_parto` è `cesareo_programmato` se e solo se il parto ha un record corrispondente in `cesareo_programmato` (Sezione 4.2.3).
7. Il valore di `tipo_parto` è `parto_con_travaglio` se e solo se il parto ha un record corrispondente in `parto_con_travaglio` (Sezione 4.2.3).
8. Per ogni misurazione deve essere presente almeno uno tra l'attributo `valore_fcm`, l'attributo `valore_toco` e un record nella tabella `misurazione_neonato` (Sezione 4.2.4).

A questi aggiungiamo i diversi attributi derivati, ovvero calcolabili o dipendenti da altre informazioni già presenti nella base di dati e che quindi devono essere gestiti automaticamente in caso di inserimenti o modifiche per evitare inconsistenze.

9. Nella tabella `gravidanza` sono derivati gli attributi `data_primo_ingresso` e `parita`.
10. Nella tabella `visita` è derivato l'attributo `eta`.
11. Nella tabella `parto` è derivato l'attributo `eta`.

5.3.2 Area della gravidanza

Il *trigger* `gravidanza_ordinamento` (Listato 5.17) verifica il vincolo 1, ovvero che la tabella `gravidanza`, considerando volta per volta i record relativi a ciascuna paziente, se viene ordinata secondo l'attributo `data_primo_ingresso` allora essa è ordinata anche secondo ciascuno degli attributi che compongono la parità⁴. Ciò viene svolto da un ciclo `for`, costruito disponibile

Il *trigger* `gravidanza_derivati` (Listato 5.18) soddisfa il vincolo 9, ovvero definisce come si calcolano gli attributi derivati della tabella `gravidanza` per ogni riga inserita o aggiornata. La parità è calcolata come concatenazione⁵ dei diversi valori indicati relativi alle gravidanze passate, mentre la data di primo ingresso è la minima nell'insieme delle date delle visite svolte con la data, se presente, del parto.

⁴Il vincolo di unicità posto sulla tupla composta da `paziente_cf` e i quattro attributi componenti della parità, insieme al vincolo di ordinamento (ovvero non decrescenza) per ciascun attributo numerico, implica che per ogni record almeno un attributo numerico è diverso da record precedente e di conseguenza si ottiene la crescita stretta per tale attributo.

⁵L'operatore di concatenazione (`||` in SQL) è applicabile solo ad argomenti di tipo testuale, quindi per applicarlo a valori numerici è richiesto un *casting* esplicito.

Il *trigger* `gravidanza_vincoli` (Listato 5.19) soddisfa i vincoli 2, 3 e 4, ovvero controlla che non ci siano più di una visita del primo trimestre e più di una del secondo trimestre per ogni gravidanza e che ogni gravidanza abbia almeno una visita o un parto associato. In generale per causare il *rollback* di una transazione è sufficiente eseguire l'istruzione `return null`. Per i *constraint trigger* non è possibile perché, essendo di tipo *after*, l'istruzione `return null` non ha nessun effetto concreto, quindi per forzare il *rollback* si deve usare il comando `raise exception`.

5.3.3 Area delle visite

La gestione della corrispondenza tra l'esito degli esami e il tipo previsto nella tabella `esame` è gestito dal *trigger* `esame_esito` (Listato 5.20), che legge i valori presenti in `tipo_esito` e verifica la corrispondenza con il valore `esito` di `esame_visita` e tratta i casi dei valori speciali `{numeric}` e `{string}`.

La cancellazione di visite può rendere falsa l'invariante del vincolo 4 che impone la presenza di almeno una visita o di un parto per ogni gravidanza. Per ovviare a questo problema definiamo il *trigger* `visita_vincoli` (Listato 5.21). L'aggiornamento dell'età, imposto dal vincolo 10, è implementato dal *trigger* `visita_eta` (Listato 5.22).

5.3.4 Area del parto

Anche nella tabella `parto` la cancellazione di record può violare il vincolo 4, quindi si definisce un *trigger* `parto_vincoli` (Listato 5.21) che usa la stessa funzione richiamata dal *trigger* `visita_vincoli`.

Il *trigger* `parto_tipo` soddisfa i vincoli 6 e 7, ovvero verifica che a seconda del tipo indicato nell'attributo `tipo_parto` ci sia un record⁶ nella tabella corrispondente. Una violazione dello stesso vincolo può derivare anche dalla cancellazione di record nelle due tabelle delle sottoentità, quindi è necessario aggiungere due *trigger* `cesareo_programmato_vincoli` e `parto_con_travaglio_vincoli` (Listato 5.24) che fanno riferimento a un'unica funzione di verifica.

L'aggiornamento dell'età, imposto dal vincolo 11, è implementato dal *trigger* `parto_eta` (Listato 5.25) in modo analogo a `visita_eta`.

5.3.5 Area del neonato

Il vincolo numero 8, che impone la presenza di almeno un valore per ogni misurazione di un tracciato, è implementato con il *trigger* `misurazione_vincoli` (Listato 5.26) che esegue il controllo ad ogni inserimento in `misurazione`. Il vincolo può essere violato anche da cancellazioni nella tabella `misurazione_neonato`, quindi è necessario aggiungere un altro *trigger* (Listato 5.27), che è una minima variazione del precedente.

⁶I vincoli di chiave primaria nelle tabelle `cesareo_programmato` e `parto_con_travaglio` garantiscono l'unicità in ciascuna tabella, quindi è sufficiente verificare la presenza in una e l'assenza nell'altra tabella.

```

-- Definizione della funzione di aggiornamento
create or replace function gravidanzaza_controlla_ordinamento()
returns trigger language plpgsql as $$
declare
    rec record;
    is_init boolean := false; -- L'if non si valuta al primo giro
    pre_rec record;
begin
    for rec in
        -- Selezione dei valori di parità per le gravidanze della stessa
        paziente
        (select figli_nati_vivi as fnv, aborti_avuti as aa,
         figli_nati_pretermine as fnp, figli_nati_a_termine as fna,
         data_primo_ingresso
        from gravidanzaza
        where paziente_cf = new.paziente_cf
        order by data_primo_ingresso asc)
    loop
        -- Verifica dell'ordinamento
        if is_init
            and (rec.fnv < pre_rec.fnv
                or rec.aa < pre_rec.aa
                or rec.fnp < pre_rec.fnp
                or rec.fna < pre_rec.fna)
        then
            raise exception 'Almeno un attributo di parità non è ordinato';
        end if;
        is_init := true;
        pre_rec := rec;
    end loop;
end;
$$;

-- Definizione del trigger
create constraint trigger gravidanzaza_ordinamento
after insert or update on gravidanzaza
deferrable initially deferred
for each row
execute procedure gravidanzaza_controlla_ordinamento();

```

Listato 5.17: Definizione del *trigger* gravidanzaza_ordinamento.


```

-- Definizione della funzione di aggiornamento
create or replace function gravidanzaza_aggiorna_derivati()
returns trigger language plpgsql as $$
begin
    -- Calcolo dell'attributo parità
    new.parita := cast(new.figli_nati_a_termine as text)
                || cast(new.figli_nati_pretermine as text)
                || cast(new.aborti_avuti as text)
                || cast(new.figli_nati_vivi as text);
    -- Calcolo dell'attributo data_primo_ingresso
    new.data_primo_ingresso :=
        (select min(data) from
            (select data from visita
              where gravidanzaza_id = new.id
             union
             select data_parto from parto
              where gravidanzaza_id = new.id));
    return new;
end;
$$;
-- Definizione del trigger
create constraint trigger gravidanzaza_derivati
after insert or update on gravidanzaza
deferrable initially deferred
for each row
execute procedure gravidanzaza_aggiorna_derivati();

```

Listato 5.18: Definizione del *trigger* gravidanzaza_derivati.

```

-- Definizione della funzione di aggiornamento
create or replace function gravidanzaza_controlla_vincoli()
returns trigger language plpgsql as $$
begin
    -- Presenza di una visita o di un parto
    if
        not exists (select gravidanzaza_id from visita
                     where gravidanzaza_id = new.id
                     union
                     select gravidanzaza_id from parto
                     where gravidanzaza_id = new.id)
    then
        raise exception 'La gravidanzaza non ha visite né parto associato';
    end if;
    -- Presenza di non più di una visita del primo trimestre
    if
        ((select count(*) from visita
          where gravidanzaza_id = new.id
          and categoria_visita = 'primo_trimestre') > 1)
    then
        raise exception 'Più di una visita del primo trimestre';
    end if;
    -- Presenza di non più di una visita del secondo trimestre
    if
        ((select count(*) from visita
          where gravidanzaza_id = new.id
          and categoria_visita = 'secondo_trimestre') > 1)
    then
        raise exception 'Più di una visita del secondo trimestre';
    end if;
    return new;
end;
$$;

-- Definizione del trigger
create constraint trigger gravidanzaza_vincoli
after insert or update on gravidanzaza
deferrable initially deferred
for each row
execute procedure gravidanzaza_controlla_vincoli();

```

Listato 5.19: Definizione del *trigger* gravidanzaza_vincoli.

```

-- Definizione della funzione di aggiornamento
create or replace function esame_controlla_esito()
returns trigger language plpgsql as $$
declare
    tipo_esito varchar[];
    esito_val varchar;
    numerico boolean := true;
begin
    tipo_esito := (select tipo_esito from esame
        where nome = new.esame_nome);
    -- Corrispondenza del tipo dell'esito
    if (tipo_esito[1] == 'numeric')
    then
        -- Caso numerico
        begin
            esito_val := new.esito::numeric;
            exception when others then
                numerico := false;
            end;
            if numerico
            then return new;
            else raise exception 'Non corrisponde al tipo numerico';
            end if;
        else
            if (tipo_esito[1] == 'string')
            then
                -- Caso stringa generica
                return new;
            else
                -- Caso tipo enum
                if (new.esito = any (tipo_esito))
                then return new;
                else raise exception 'Non corrisponde al tipo enumerazione';
                end if;
            end if;
        end if;
        return new;
    end;
    $$;
-- Definizione del trigger
create constraint trigger esame_esito
after insert or update on esame_visita
deferrable initially deferred
for each row
execute procedure esame_controlla_esito();

```

Listato 5.20: Definizione del *trigger* esame_esito.

```

-- Definizione della funzione di aggiornamento
create or replace function visita_parto_controlla_vincoli()
returns trigger language plpgsql as $$
begin
    -- Presenza di una visita o di un parto
    if
        not exists (select gravidanza_id from visita
                     where gravidanza_id = old.id
                     union
                     select gravidanza_id from parto
                     where gravidanza_id = old.id)
    then
        raise exception 'La gravidanza non ha visite né parto associato';
    end if;
end;
$$;
-- Definizione del trigger per la tabella visita
create constraint trigger visita_vincoli
after delete on visita
deferrable initially deferred
for each row
execute procedure gravidanza_controlla_vincoli();
-- Definizione del trigger per la tabella parto
create constraint trigger parto_vincoli
after delete on parto
deferrable initially deferred
for each row
execute procedure gravidanza_controlla_vincoli();

```

Listato 5.21: Definizione dei *trigger* visita_vincoli e esame_vincoli.

```

-- Definizione della funzione di aggiornamento
create or replace function visita_calcola_eta()
returns trigger language plpgsql as $$
declare
    data_nascita date;
begin
    data_nascita :=
        (select data_nascita from persona, gravidanza
         where persona.cf = gravidanza.paziente_cf
         and gravidanza.id = new.gravidanza_id);
    new.eta := extract (year from age(new.data,data_nascita));
    return new;
end;
$$;
-- Definizione del trigger
create constraint trigger visita_eta
after insert or update on visita
deferrable initially deferred
for each row
execute procedure visita_calcola_eta();

```

Listato 5.22: Definizione del *trigger* visita_eta.

```

-- Definizione della funzione di aggiornamento
create or replace function parto_controlla_tipo()
returns trigger language plpgsql as $$
begin
    -- Corrispondenza tra tipo_parto e tabelle
    if (new.tipo_parto = 'cesareo_programmato')
    then
        -- Si verifica che sia in cesareo programmato
        -- E non in parto con travaglio
        if
            exists (select count(*) from cesareo_programmato
                    where gravidanza_id = new.gravidanza_id)
            and not exists (select count(*) from parto_con_cesareo
                           where gravidanza_id = new.gravidanza_id)
        then return new;
        else raise exception 'Non corrisponde a cesareo_programmato';
        end if;
    else
        -- Si verifica che sia in parto con travaglio
        -- E non in cesareo programmato
        if
            exists (select count(*) from parto_con_travaglio
                    where gravidanza_id = new.gravidanza_id)
            and not exists (select count(*) from cesareo_programmato
                           where gravidanza_id = new.gravidanza_id)
        then return new;
        else raise exception 'Non corrisponde a parto_con_travaglio';
        end if;
    end if;
end;
$$;
-- Definizione del trigger
create constraint trigger parto_tipo
after insert or update on parto
deferrable initially deferred
for each row
execute procedure parto_controlla_tipo();

```

Listato 5.23: Definizione del *trigger* parto_tipo.

```

-- Definizione della funzione di aggiornamento
create or replace function parto_controlla_tipo_2()
returns trigger language plpgsql as $$
begin
    -- Corrispondenza tra tipo_parto e tabelle
    if (old.tipo_parto = 'cesareo_programmato')
    then
        -- Si verifica che sia in cesareo programmato
        -- E non in parto con travaglio
        if
            exists (select count(*) from cesareo_programmato
                    where gravidanza_id = old.gravidanza_id)
            and not exists (select count(*) from parto_con_cesareo
                           where gravidanza_id = old.gravidanza_id)
        then return old;
        else raise exception 'Non corrisponde a cesareo_programmato';
        end if;
    else
        -- Si verifica che sia in parto con travaglio
        -- E non in cesareo programmato
        if
            exists (select count(*) from parto_con_travaglio
                    where gravidanza_id = old.gravidanza_id)
            and not exists (select count(*) from cesareo_programmato
                           where gravidanza_id = old.gravidanza_id)
        then return old;
        else raise exception 'Non corrisponde a parto_con_travaglio';
        end if;
    end if;
end;
$$;

-- Definizione del trigger
create constraint trigger cesareo_programmato_vincoli
after delete on cesareo_programmato
deferrable initially deferred
for each row
execute procedure parto_controlla_tipo_2();

-- Definizione del trigger
create constraint trigger parto_con_travaglio_vincoli
after delete on parto_con_travaglio
deferrable initially deferred
for each row
execute procedure parto_controlla_tipo_2();

```

Listato 5.24: Definizione dei *trigger* cesareo_programmato_vincoli e parto_con_travaglio_vincoli.

```

-- Definizione della funzione di aggiornamento
create or replace function parto_calcola_eta()
returns trigger language plpgsql as $$
declare
    data_nascita date;
begin
    data_nascita :=
        (select data_nascita from persona, gravidanzaa
         where persona.cf = gravidanzaa.paziente_cf
         and gravidanzaa.id = new.gravidanza_id);
    new.eta := extract (year from age(new.data,data_nascita));
    return new;
end;
$$;
-- Definizione del trigger
create constraint trigger parto_eta
after insert or update on parto
deferrable initially deferred
for each row
execute procedure parto_calcola_eta();

```

Listato 5.25: Definizione del *trigger* parto_eta.

```

-- Definizione della funzione di aggiornamento
create or replace function misurazione_controlla_vincoli()
returns trigger language plpgsql as $$
begin
    -- Presenza di almeno uno dei valori da misurare
    if
        new.valore_fcm is null
        and new.valore_toco is null
        and not exists (select * from misurazione_neonato
                        where gravidanzaa_id = new.gravidanza_id
                        and misurazione_istante = new.istante)
    then
        raise exception 'Nessun valore misurato per questo timestamp';
    end if;
    return new;
end;
$$;
-- Definizione del trigger
create constraint trigger misurazione_vincoli
after insert or update on misurazione
deferrable initially deferred
for each row
execute procedure misurazione_controlla_vincoli();

```

Listato 5.26: Definizione del *trigger* misurazione_vincoli.

```

-- Definizione della funzione di aggiornamento
create or replace function misurazione_neonato_controlla_vincoli()
returns trigger language plpgsql as $$
begin
    -- Presenza di almeno uno dei valori da misurare
    if
        old.valore_fcm is null
        and old.valore_toco is null
        and not exists (select * from misurazione_neonato
            where gravidanza_id = old.gravidanza_id
            and misurazione_istante = old.istante)
    then
        raise exception 'Nessun valore misurato per questo timestamp';
    end if;
    return old;
end;
$$;
-- Definizione del trigger
create constraint trigger misurazione_neonato_vincoli
after delete on misurazione_neonato
deferrable initially deferred
for each row
execute procedure misurazione_neonato_controlla_vincoli();

```

Listato 5.27: Definizione del *trigger* `misurazione_neonato_vincoli`.

6

Funzionalità della base di dati

Come anticipato nella Sezione 2.5, una base di dati relazionale permette di memorizzare dati in maniera strutturata ed efficiente e allo stesso tempo mantiene i vincoli imposti sui dati che prevengono le inconsistenze. Vogliamo mostrare e valutare le qualità del sistema ottenuto e confrontarlo con la soluzione attualmente in uso; ciò viene fatto principalmente con l'esemplificazione di interrogazioni sulla base di dati.

6.1 Esempi di interrogazioni

Generalmente il personale medico è interessato a inserire, visualizzare e modificare le informazioni relative a singole pazienti e raccolte in singoli momenti, ad esempio l'inserimento in blocco dei dati di un parto o di una visita. Queste operazioni, che risultano immediate nella soluzione attuale che impiega fogli di calcolo strutturati proprio secondo i diversi eventi della gravidanza, rimangono non solo possibili ma soprattutto facilmente esprimibili in SQL. Questo avviene perché, nonostante le ristrutturazioni e gli adattamenti necessari all'implementazione, le tabelle dello schema logico e fisico sono ancora perlopiù corrispondenti a entità e relazioni dello schema concettuale.

Ci concentriamo quindi su un livello diverso di analisi dei dati, ovvero principalmente lo studio di dati aggregati e la selezione di record sulla base di caratteristiche di interesse per gli utilizzatori finali. Portiamo alcuni esempi di interrogazioni richieste dai medici che, grazie alla struttura tabellare e alle funzionalità dei DBMS relazionali, vengono svolte efficientemente e senza che sia richiesta una processazione manuale aggiuntiva. Gli esempi di interrogazioni mostrati di seguito non hanno l'obiettivo di portare a un'effettiva analisi delle informazioni presenti nell'istanza di base di dati, bensì hanno il solo scopo di mostrare come tali informazioni possono essere estratte.

Valori di BMI e lacerazioni

Si vuole ottenere, per ogni gravidanza, il valore del BMI registrato all'inizio della gravidanza e il grado di lacerazioni subito nel parto. Questo risultato è ottenibile con l'interrogazione del Listato 6.1.

```
-- Interrogazione principale
select *
from
  (select gravidanza_id, peso/(altezza*altezza) as bmi
   from visita
   where categoria_visita = 'primo_trimestre'
        and altezza is not null
        and altezza <> 0
        and peso is not null)
natural join
(select gravidanza_id, lacerazioni
 from parto_con_travaglio);
```

Listato 6.1: Esempio di interrogazione. Valori di BMI e lacerazioni.

Correlazione tra taglio cesareo e travaglio indotto

Si vuole ottenere la percentuale di tagli cesarei effettuati nei parti con travaglio indotto confrontata con quella nei parti con travaglio spontaneo. Questo risultato è ottenibile con l'interrogazione del Listato 6.2. Definiamo una vista `parto_con_travaglio_induzione` per rendere più concise le diverse sottointerrogazioni nel corpo dell'interrogazione principale.

Correlazione tra parto operativo ed episiotomia

Si vuole ottenere la percentuale di episiotomie effettuate nei parti operativi rispetto ai parti vaginali. Questo risultato è ottenibile con l'interrogazione del Listato 6.3, che risulta essere molto simile al primo esempio.

Correlazione tra taglio cesareo e indice Robson

Si vuole ottenere il numero di tagli cesarei e di parti vaginali per ciascun valore dell'indice Robson. Questo risultato è ottenibile con l'interrogazione del Listato 6.4; definiamo e usiamo una vista `parto_sottotipo` per rendere più concisa l'interrogazione principale.

Per evitare doppi conteggi, questa interrogazione non considera i parti che non sono stati soltanto vaginali o soltanto cesarei. Ricordiamo che è possibile avere una sovrapposizione nel sottotipo dei parti con travaglio nel caso di parti gemellari in cui i gemelli nascono con procedure diverse.

```

-- Definizione di una vista con il numero di induzioni per parto
create view parto_con_travaglio_induzione as
select *
from parto_con_travaglio_natural join
  (select count(*) as induzioni, gravidanza_id
   from induzione
   group by gravidanza_id);
-- Interrogazione principale
select s1.tc_travaglio_spontaneo as tc_travaglio_spontaneo,
       s2.parti_travaglio_spontaneo as parti_travaglio_spontaneo,
       case when s2.parti_travaglio_spontaneo = 0
         then 0
         else s1.tc_travaglio_spontaneo::numeric/s2.parti_travaglio_spontaneo
       end as rapporto_tc_travaglio_spontaneo,
       s3.tc_travaglio_indotto as tc_travaglio_indotto,
       s4.parti_travaglio_indotto as parti_travaglio_indotto,
       case when s4.parti_travaglio_indotto = 0
         then 0
         else s3.tc_travaglio_indotto::numeric/s4.parti_travaglio_indotto
       end as rapporto_tc_travaglio_indotto
from
-- Numero di tagli cesarei in travaglio spontaneo
(select count(*) as tc_travaglio_spontaneo
 from parto_con_travaglio_induzione
 where induzioni = 0
  and sottotipo_parto like '%cesareo%') s1,
-- Numero di parti totali in travaglio spontaneo
(select count(*) as parti_travaglio_spontaneo
 from parto_con_travaglio_induzione
 where induzioni = 0) s2,
-- Numero di tagli cesarei in travaglio indotto
(select count(*) as tc_travaglio_indotto
 from parto_con_travaglio_induzione
 where induzioni > 0
  and sottotipo_parto like '%cesareo%') s3,
-- Numero di parti totali in travaglio indotto
(select count(*) as parti_travaglio_indotto
 from parto_con_travaglio_induzione
 where induzioni > 0) s4;

```

Listato 6.2: Esempio di interrogazione. Correlazione tra parto cesareo e travaglio indotto.

```

-- Interrogazione principale
select s1.ep_operativi as episiotomie_operativi,
       s2.parti_operativi as parti_operativi,
       case when s2.parti_operativi = 0
         then 0
         else s1.ep_operativi::numeric/s2.parti_operativi
       end as rapporto_episiotomie_operativi,
       s3.ep_vaginali as episiotomie_vaginali,
       s4.parti_vaginali as parti_vaginali,
       case when s4.parti_vaginali = 0
         then 0
         else s3.ep_vaginali::numeric/s4.parti_vaginali
       end as rapporto_episiotomie_vaginali
from
-- Numero di tagli cesarei in travaglio spontaneo
(select count(*) as ep_operativi
from parto_con_travaglio
where lacerazioni = 'episiotomia'
   and sottotipo_parto = 'operativo') s1,
-- Numero di parti totali in travaglio spontaneo
(select count(*) as parti_operativi
from parto_con_travaglio
where sottotipo_parto = 'operativo') s2,
-- Numero di tagli cesarei in travaglio indotto
(select count(*) as ep_vaginali
from parto_con_travaglio
where lacerazioni = 'episiotomia'
   and sottotipo_parto = 'naturale') s3,
-- Numero di parti totali in travaglio indotto
(select count(*) as parti_vaginali
from parto_con_travaglio
where sottotipo_parto = 'naturale') s4;

```

Listato 6.3: Esempio di interrogazione. Correlazione tra parto operativo ed episiotomia.

```

-- Definizione di una vista con il sottotipo di ogni parto
create view parto_sottotipo as
select
  gravidanza_id,
  robson,
  case when tipo_parto = 'cesareo_programmato'
    then 'cesareo'
    else sottotipo_parto
  end as sottotipo_parto
from parto natural left join parto_con_travaglio;
-- Interrogazione principale
select robson, parto_sottotipo as tipo, count(*) as numero_parti
from parto_sottotipo
where sottotipo_parto in ('cesareo', 'naturale')
group by robson, sottotipo_parto
order by robson, sottotipo_parto;

```

Listato 6.4: Esempio di interrogazione. Correlazione tra parto cesareo e indice Robson.

Conclusione

7.1 Sintesi della progettazione della base di dati

Analizziamo velocemente le attività svolte per la costruzione della base di dati secondo i principi punti previsti nella Sezione 1.2.

L'analisi della situazione di fatto, degli strumenti in uso e delle richieste di aggiunte ulteriori (Sezione 2) ha permesso la formalizzazione dei requisiti di un sistema complesso per la gestione del percorso gravidanza e parto. Ciò ci ha permesso di scegliere una base di dati relazionale come strumento adatto alla risoluzione del problema specifico che viene posto.

Nei vari stadi della progettazione, passando da un piano concettuale (Sezione 3) a uno logico (Sezione 4.2) fino alla realizzazione concreta attraverso *script* in linguaggio SQL (Sezione 5) e all'esemplificazione delle funzionalità della base di dati (Sezione 6), abbiamo mostrato il processo di costruzione di una base di dati completa e pronta all'impiego. Queste fasi di progettazione si susseguono a cascata in modo naturale e sono guidate da regole e principi che facilitano il raggiungimento di un risultato coerente, anche se in determinati casi è necessario operare delle scelte che, seppure possano aggiungere complicazioni nella gestione delle ridondanze e dell'integrità dei dati, possono agevolare l'interrogazione e l'uso da parte di utenti anche inesperti.

Ogni fase del processo di realizzazione è corredata di schemi, che seguono i formalismi specifici delle singole fasi di progettazione affrontate, insieme a vincoli aggiuntivi e alla spiegazione delle eventuali scelte operate. La documentazione è necessaria non solo per lo sviluppo iniziale del sistema ma anche per l'utilizzo quotidiano, ad esempio per la formulazione di interrogazioni, e per la manutenzione, nel caso in cui emergano criticità, errori o nuovi requisiti da integrare.

7.2 Applicazioni del nuovo sistema

L'analisi dei dati raccolti a partire da gravidanze e parti è un tema emergente e che attira interesse verso di sé, specialmente insieme alla creazione di modelli di intelligenza artificiale e altri metodi *data-driven* volti a studiare e prevedere l'andamento dei parametri fisiologici e delle anomalie.

Ad esempio nel 2014, con la pubblicazione della prima base di dati *open source* con rilevazioni del cardiocitocografo, Chudáček e altri [5] prevedono che l'uso di questi dati sarà utile per il processamento

e infine l'analisi automatica dei tracciati CTG. Anche altri studi hanno evidenziato l'importanza dell'analisi dei tracciati CTG, come Cahill e altri [3], e in particolare con l'ausilio dei recenti sviluppi nel campo del *deep learning*, come McCoy e altri [9].

Ulteriori ricerche sono state condotte grazie all'applicazione del *deep learning* per ottenere modelli predittivi a partire dai dati raccolti durante la gravidanza, come ad esempio lo studio di Cersonsky e altri [4] sulla prevedibilità della mortalità al parto che annovera, tra le variabili analizzate, molte delle informazioni contenute nella nostra base di dati.

7.3 Possibili sviluppi futuri

L'obiettivo finale della realizzazione della base di dati è raggiungere una piena integrazione del nuovo sistema con l'ambiente preesistente in cui lavora il personale medico. Il sistema così ottenuto potrebbe beneficiare dell'uso di un'insieme di strumenti per facilitare questa integrazione.

- Lo sviluppo di un'interfaccia grafica è fondamentale per l'uso della base di dati perché si possono implementare le interrogazioni in un *back-end* programmato a priori e si rende possibile l'uso anche all'utente finale senza che gli sia richiesta la conoscenza del linguaggio SQL né della struttura tabellare di basso livello.
- Anche per il programmatore può essere utile operare a livello più astratto rispetto alle tabelle descritte nell'implementazione (Sezione 5). Si possono quindi definire un'API per facilitare inserimenti singoli o in blocco oppure delle viste per sintetizzare interrogazioni su dati aggregati.
- Per agevolare il popolamento della base di dati a partire dai sistemi attualmente in uso si possono definire *script* o semplici programmi per la conversione dai questi formati a inserimenti nella nuova base di dati relazionale.

Bibliografia

- [1] P. Atzeni e altri. *Basi di dati*. McGraw-Hill, 2023.
- [2] D. Ayres-de-Campos. Electronic fetal monitoring or cardiotocography, 50 years later: what's in a name? *American Journal of Obstetrics & Gynecology*, 218(6):545–546, 2018.
- [3] A.G. Cahill e altri. A prospective cohort study of fetal heart rate monitoring: deceleration area is predictive of fetal acidemia. *American Journal of Obstetrics & Gynecology*, 218(5), 2018.
- [4] T.E.K. Cersonsky e altri. Identifying risk of stillbirth using machine learning. *American Journal of Obstetrics & Gynecology*, 229(3), 2023.
- [5] V. Chudáček e altri. Open access intrapartum ctg database. *BMC Pregnancy Childbirth*, 14(1), 2014.
- [6] E.F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.
- [7] L. Corey, A. Vezina, e R.B. Gala. Using technology to improve women's health care. *Ochsner Journal*, 20(4):422–425, 2020.
- [8] N. Dedić e C. Stanier. An evaluation of the challenges of multilingualism in data warehouse development In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016)*. A cura di S. Hammoudi e altri, volume 1, pp. 196–206, 2016.
- [9] J.A. McCoy e altri. Intrapartum electronic fetal heart rate monitoring to predict acidemia at birth with the use of deep learning. *American Journal of Obstetrics & Gynecology*, 232(1), 2025.
- [10] J.S. McCullough, S.T. Parente, e R. Town. Health information technology and patient outcomes: the role of information and labor coordination. *The RAND Journal of Economics*, 47(1):207–236, 2016.
- [11] A. Silberschatz, H.F. Korth, e S. Sudarshan. *Database System Concepts*. McGraw-Hill, 2011.
- [12] A.P. Souka e altri. Increased nuchal translucency with normal karyotype. *American Journal of Obstetrics & Gynecology*, 192(4):1005–1021, 2005.
- [13] The PostgreSQL Global Development Group. *PostgreSQL 17.5 Documentation*, 2025.