## **DBMS – Master AI&DS 24/25**

Professori: Studenti:

Mario Alviano Andrea Tiloca

Giorgio Terracina Francesco Russo

Marco Maratea Jessica Montesano

## **Capitolo 1**

## **Introduzione**

Il presente lavoro descrive la progettazione di un sistema informativo relazionale per una **clinica privata** specializzata in servizi diagnostici. L’obiettivo principale è quello di supportare in maniera integrata **le attività cliniche e amministrative**, con particolare riferimento alla **gestione delle prenotazioni di esami** da parte dei pazienti. Il sistema è stato pensato per garantire l’efficienza operativa e migliorare l’esperienza del paziente, integrando funzionalità di **registrazione, pagamento, refertazione e feedback**.

## **Capitolo 2**

## **Modellazione concettuale: panoramica dei processi**

I processi principali modellati nel database sono:

* **Registrazione dei pazienti** con dati anagrafici e identificativi univoci (Codice Fiscale).
* **Prenotazione di uno o più esami**, con possibilità di gestione dello stato e orario.
* **Erogazione degli esami**, internamente o tramite laboratori convenzionati.
* **Assegnazione dei medici** agli esami, con registrazione della specializzazione.
* **Gestione dei referti**, che associano l’esito clinico a ogni esame svolto.
* **Gestione dei pagamenti**, con tracciamento del metodo, importo e stato.
* **Inserimento di recensioni** da parte dei pazienti sui servizi ricevuti.

Questa panoramica dei processi costituisce la base per la costruzione dello **schema E/R**.

**Pazienti**

I pazienti sono soggetti che accedono ai servizi della clinica previa registrazione. Di ciascun paziente vengono memorizzati i dati anagrafici, i contatti e il codice fiscale. I pazienti possono effettuare prenotazioni per uno o più esami, ricevere referti, saldare i pagamenti corrispondenti ed eventualmente rilasciare recensioni sui servizi ricevuti.

**Medici**

I medici operano all’interno della struttura come dipendenti oppure in qualità di consulenti esterni. Per ogni medico sono registrati dati identificativi, la specializzazione e il tipo di rapporto con la clinica. I medici sono responsabili dell'esecuzione degli esami e della redazione dei referti.

**Esami**

Gli esami erogati dalla clinica si suddividono in due macro-categorie: **esami di laboratorio** ed **esami di imaging**. Ogni esame è caratterizzato da un tipo specifico, una categoria, un costo, un livello di urgenza e una modalità di erogazione. In particolare, la modalità può essere "in sede" (esame eseguito internamente alla clinica) oppure "esterna" (erogato da un laboratorio convenzionato). Ogni esame è eseguito da un singolo medico.

**Laboratori convenzionati**

Gli esami non effettuabili direttamente presso la clinica possono essere affidati a laboratori diagnostici esterni. Ciascun laboratorio è identificato da un codice, una denominazione, un indirizzo, e può essere o meno accreditato presso il Servizio Sanitario Nazionale (SSN). I laboratori esterni sono coinvolti solo negli esami con modalità "esterna".

**Prenotazioni**

I pazienti effettuano prenotazioni per uno o più esami. Ogni prenotazione è registrata con un identificativo, la data e l’orario richiesto, e uno stato che ne indica la validità (confermata, annullata, ecc.). Per modellare correttamente la relazione molti-a-molti tra prenotazioni ed esami, è prevista un'entità intermedia.

**Referti**

A seguito dell'esecuzione di un esame, viene prodotto un referto diagnostico. Il referto è associato a un unico esame e redatto dal medico responsabile. Contiene l’esito clinico, eventuali note descrittive, la data di emissione e, se previsto, un allegato digitale (come un file PDF o un’immagine DICOM).

**Pagamenti**

Ogni prenotazione può essere associata a uno o più pagamenti, in funzione del numero e del tipo di esami prenotati. I pagamenti sono caratterizzati da un importo, una data, un metodo (es. carta, contanti, bonifico) e uno stato (ad esempio: pagato, in sospeso, rimborsato).

**Recensioni**

Al termine del processo diagnostico, il paziente può esprimere una valutazione del servizio ricevuto mediante una recensione associata al referto. La recensione include un voto numerico, un commento opzionale e la data di inserimento. Ogni recensione è quindi indirettamente riferita sia all’esame che al medico che lo ha eseguito.

## **Capitolo 3**

## **Schema Entità/Relazioni (E/R)**

Lo schema E/R rappresenta i principali attori e le relazioni del sistema:

### **Entità principali**

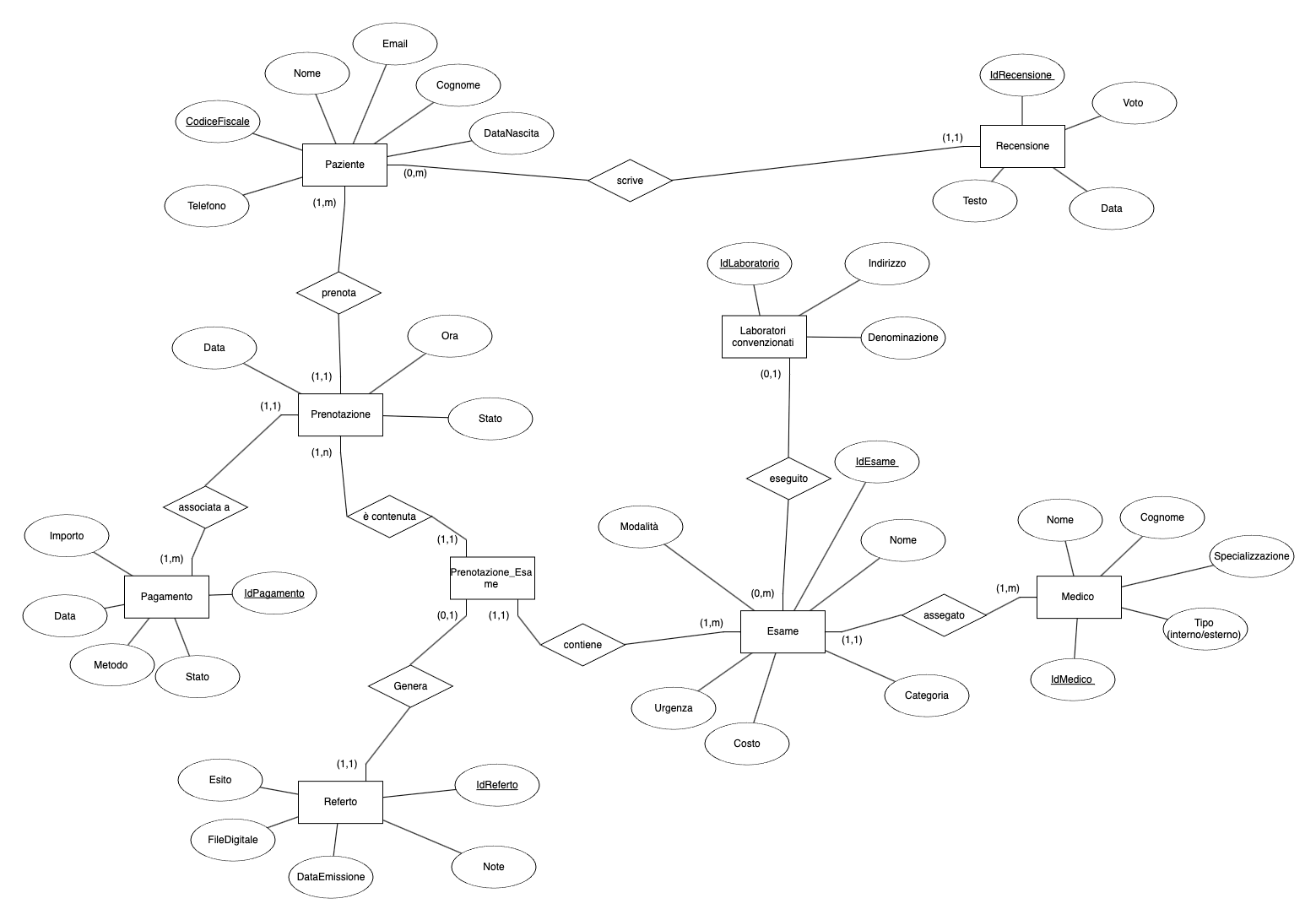
* **Paziente**: contiene i dati anagrafici.
* **Medico**: include nome, cognome, specializzazione e tipo (interno/esterno).
* **Esame**: con caratteristiche tecniche (nome, costo, urgenza, modalità).
* **Laboratorio convenzionato**: per gli esami esterni.
* **Prenotazione**: gestisce l’associazione dei pazienti con uno o più esami.
* **Pagamento**: associato alla prenotazione.
* **Referto**: prodotto in seguito all’esame.
* **Recensione**: espressa dal paziente dopo il ricevimento del referto.

### **Relazioni**

* Un **paziente** può effettuare più **prenotazioni**.
* Ogni **prenotazione** può contenere **più esami**, e ogni **esame** può comparire in più prenotazioni. Questo è gestito tramite la **tabella di join PrenotazioneEsame**, che rappresenta una classica relazione **molti-a-molti**.
* Ogni **esame** è assegnato a un singolo **medico**.
* Se l’esame è esterno, è associato a un **laboratorio convenzionato**.
* A ogni voce nella tabella PrenotazioneEsame può corrispondere un **referto**.
* Ogni **recensione** fa riferimento a un **paziente**, ed è collegata indirettamente all’esame e al medico tramite il referto.

### **Chiavi primarie e chiavi esterne**

* Esempio di **chiavi primarie**: IdPaziente, IdMedico, IdEsame, IdPrenotazione, IdPagamento, IdReferto, IdRecensione.
* Esempio di **foreign key**:
  + IdPaziente in Prenotazione, Recensione
  + IdPrenotazione in Pagamento, PrenotazioneEsame
  + IdEsame in PrenotazioneEsame, Referto
  + IdMedico in Esame
  + IdLaboratorio in Esame (solo per modalità "esterna")



## **Capitolo 4**

## **Schema logico (DDL SQL)**

Lo schema logico è stato implementato in SQL e riflette fedelmente lo schema E/R. Le tabelle sono state create con vincoli di integrità, chiavi primarie, esterne e controlli sui valori ammessi. Di seguito lo Schema logico implementato:

CREATE DATABASE ClinicaPrivata;

USE ClinicaPrivata;

CREATE TABLE Paziente (

IdPaziente INTEGER PRIMARY KEY AUTO\_INCREMENT,

Nome VARCHAR(50) NOT NULL,

Cognome VARCHAR(50) NOT NULL,

Email VARCHAR(100) NOT NULL,

DataNascita DATE NOT NULL,

Telefono VARCHAR(100) NOT NULL,

CodiceFiscale VARCHAR(16) UNIQUE NOT NULL

);

INSERT INTO Paziente (Nome, Cognome, Email, DataNascita, Telefono, CodiceFiscale) VALUES

('Mario', 'Rossi', 'mario.rossi@email.com', '1985-06-12', '3391234567', 'RSSMRA85H12H501Z'),

('Lucia', 'Verdi', 'lucia.verdi@email.com', '1990-11-25', '3487654321', 'VRDLCU90S65C351H'),

('Giovanni', 'Bianchi', 'giovanni.bianchi@email.com', '1978-03-04', '3201122334', 'BNCGVN78C04F205T'),

('Elena', 'Ferrari', 'elena.ferrari@email.com', '2000-07-20', '3479988776', 'FRRLNE00L60C351P'),

('Sara', 'Russo', 'sara.russo@email.com', '1995-10-15', '3335566778', 'RSSSRA95R55H501Y');

Immagine che contiene testo, schermata, Software multimediale, software

Il contenuto generato dall'IA potrebbe non essere corretto.

CREATE TABLE Recensione (

IdRecensione INTEGER PRIMARY KEY AUTO\_INCREMENT,

IdPaziente INTEGER NOT NULL,

Voto INTEGER CHECK(Voto BETWEEN 1 AND 5),

DataRecensione DATE NOT NULL,

TestoRecensione LONGTEXT,

FOREIGN KEY (IdPaziente) REFERENCES Paziente(IdPaziente)

);

INSERT INTO Recensione ( IdPaziente, Voto, DataRecensione, TestoRecensione) VALUES

( 1, 5, '2024-05-20', 'Servizio eccellente, personale molto disponibile.'),

( 2, 3, '2024-05-21', 'Tutto ok, ma ho atteso troppo per il referto.'),

( 3, 4, '2024-05-22', 'Professionali e puntuali, consigliato.'),

( 4, 2, '2024-05-23', 'Struttura buona, ma poca chiarezza alla reception.'),

( 5, 5, '2024-05-24', 'Ottima esperienza, tornerò sicuramente.');

Immagine che contiene testo, schermata, software, Software multimediale

Il contenuto generato dall'IA potrebbe non essere corretto.

CREATE TABLE Prenotazione (

IdPrenotazione INTEGER PRIMARY KEY AUTO\_INCREMENT,

IdPaziente INTEGER,

DataPrenotazione DATE NOT NULL,

OraPrenotazione TIME NOT NULL,

StatoPrenotazione VARCHAR(20) NOT NULL CHECK(StatoPrenotazione IN ('Confermato', 'Non confermato')),

FOREIGN KEY (IdPaziente) REFERENCES Paziente(IdPaziente)

);

INSERT INTO Prenotazione ( IdPaziente, DataPrenotazione, OraPrenotazione, StatoPrenotazione) VALUES

( 1, '2024-06-01', '09:00:00', 'Confermato'),

( 2, '2024-06-02', '10:30:00', 'Non confermato'),

( 3, '2024-06-03', '08:45:00', 'Confermato'),

( 4, '2024-06-04', '11:15:00', 'Confermato'),

( 5, '2024-06-05', '14:00:00', 'Non confermato');

Immagine che contiene testo, schermata, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

CREATE TABLE Medico (

IdMedico INTEGER PRIMARY KEY AUTO\_INCREMENT,

Nome VARCHAR(50) NOT NULL,

Cognome VARCHAR(50) NOT NULL,

Specializzazione VARCHAR(100),

Tipo VARCHAR(50)

);

INSERT INTO Medico ( Nome, Cognome, Specializzazione, Tipo) VALUES

('Giulia', 'Rossi', 'Cardiologia', 'Dipendente'),

( 'Marco', 'Bianchi', 'Neurologia', 'Libero Prof.'),

( 'Anna', 'Verdi', 'Pediatria', 'Dipendente'),

( 'Luca', 'Neri', 'Ortopedia', 'Libero Prof.'),

( 'Sara', 'Ferrari', 'Dermatologia', 'Dipendente'),

( 'Paolo', 'Conti', 'Psichiatria', 'Libero Prof.'),

( 'Elena', 'Russo', 'Oncologia', 'Dipendente'),

( 'Giovanni', 'Fontana', 'Endocrinologia', 'Libero Prof.'),

( 'Marta', 'Galli', 'Ginecologia', 'Dipendente'),

( 'Andrea', 'De Luca', 'Chirurgia Generale', 'Dipendente');

Immagine che contiene testo, schermata, software, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

CREATE TABLE Pagamento (

IdPagamento INTEGER PRIMARY KEY AUTO\_INCREMENT,

IdPrenotazione INTEGER,

DataPagamento DATE NOT NULL,

Importo DECIMAL(8,2) NOT NULL,

Metodo VARCHAR(20) NOT NULL CHECK(Metodo IN ('Contanti', 'POS', 'Bonifico')),

StatoPagamento VARCHAR(20) NOT NULL CHECK(StatoPagamento IN ('Confermato', 'Non confermato')),

FOREIGN KEY (IdPrenotazione) REFERENCES Prenotazione(IdPrenotazione)

);

INSERT INTO Pagamento (IdPrenotazione, DataPagamento, Importo, Metodo, StatoPagamento) VALUES

( 1, '2024-06-01', 50, 'Contanti', 'Confermato'),

( 2, '2024-06-02', 75.5, 'POS', 'Confermato'),

( 3, '2024-06-03', 100, 'Bonifico', 'Non confermato'),

( 4, '2024-06-04', 120.75, 'POS', 'Confermato'),

( 5, '2024-06-05', 80, 'Contanti', 'Non confermato');

Immagine che contiene testo, schermata, Carattere, software

Il contenuto generato dall'IA potrebbe non essere corretto.

CREATE TABLE LaboratoriConvenzionati (

IdLaboratorio INTEGER PRIMARY KEY AUTO\_INCREMENT,

NomeLaboratorio VARCHAR(100) NOT NULL,

Indirizzo VARCHAR(150) NOT NULL,

Citta VARCHAR(50) NOT NULL,

Cap VARCHAR(5) NOT NULL

);

USE ClinicaPrivata;

INSERT INTO LaboratoriConvenzionati (NomeLaboratorio, Indirizzo, Citta, Cap) VALUES

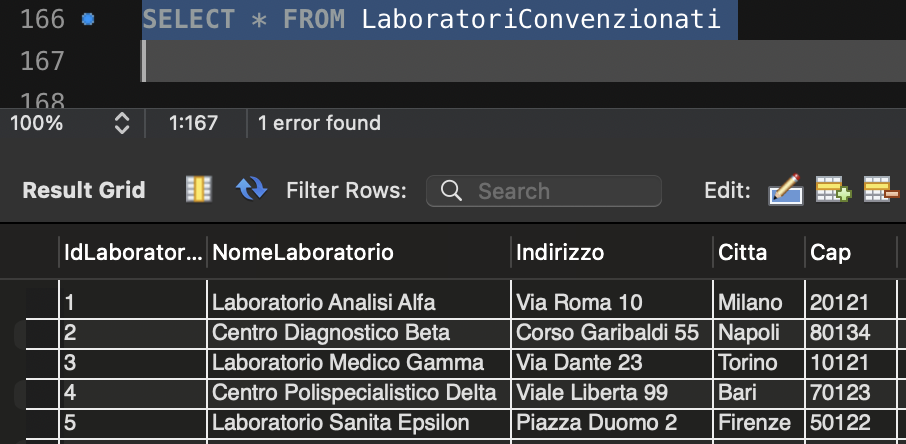
('Laboratorio Analisi Alfa', 'Via Roma 10', 'Milano', '20121'),

('Centro Diagnostico Beta', 'Corso Garibaldi 55', 'Napoli', '80134'),

('Laboratorio Medico Gamma', 'Via Dante 23', 'Torino', '10121'),

('Centro Polispecialistico Delta', 'Viale Liberta 99', 'Bari', '70123'),

('Laboratorio Sanita Epsilon', 'Piazza Duomo 2', 'Firenze', '50122');



CREATE TABLE Esame (

IdEsame INTEGER PRIMARY KEY AUTO\_INCREMENT,

IdMedico INTEGER,

NomeEsame VARCHAR(100) NOT NULL,

Categoria VARCHAR(50) NOT NULL,

Costo DECIMAL(8,2) NOT NULL,

Urgenza VARCHAR(10) NOT NULL,

Modalita VARCHAR(20) NOT NULL CHECK(modalita IN ('Interno', 'Esterno')),

IdLaboratorio INTEGER ,

FOREIGN KEY (IdMedico) REFERENCES Medico(IdMedico),

FOREIGN KEY (IdLaboratorio) REFERENCES LaboratoriConvenzionati(IdLaboratorio)

);

INSERT INTO Esame (IdMedico, NomeEsame, Categoria, Costo, Urgenza, Modalita, IdLaboratorio) VALUES

(1, 'Emocromo', 'Laboratorio', 25, 'No', 'Interno', NULL),

(2, 'Risonanza Magnetica', 'Imaging', 150, 'Si', 'Esterno', 1),

(3, 'Ecografia addome', 'Imaging', 70, 'No', 'Interno', NULL),

(1, 'Tampone COVID', 'Laboratorio', 40, 'Si', 'Interno', NULL),

(2, 'TC torace', 'Imaging', 130, 'Si', 'Esterno', 2);

Immagine che contiene testo, schermata, software, Software multimediale

Il contenuto generato dall'IA potrebbe non essere corretto.

CREATE TABLE PrenotazioneEsame (

IdPrenotazioneEsame INTEGER PRIMARY KEY AUTO\_INCREMENT,

IdPrenotazione INTEGER,

IdEsame INTEGER,

FOREIGN KEY (IdPrenotazione) REFERENCES Prenotazione(IdPrenotazione),

FOREIGN KEY (IdEsame) REFERENCES Esame(IdEsame)

);

USE ClinicaPrivata;

INSERT INTO PrenotazioneEsame (IdPrenotazione, IdEsame) VALUES

(1, 11),

(1, 12),

(2, 13),

(3, 11),

(4, 14);

Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

CREATE TABLE Referto (

IdReferto INTEGER PRIMARY KEY AUTO\_INCREMENT,

IdPrenotazioneEsame INTEGER UNIQUE,

DataEmissione DATE NOT NULL,

Esito LONGTEXT NOT NULL,

FileDigitale VARCHAR(100) NOT NULL,

FOREIGN KEY (IdPrenotazioneEsame) REFERENCES PrenotazioneEsame(IdPrenotazioneEsame)

);

INSERT INTO Referto (IdPrenotazioneEsame, DataEmissione, Esito, FileDigitale) VALUES

(11, '2024-06-02', 'Valori nella norma', 'referti/001.pdf'),

(12, '2024-06-03', 'Presenza di alterazioni nei tessuti molli', 'referti/002.pdf'),

(13, '2024-06-04', 'Nessuna anomalia riscontrata', 'referti/003.pdf'),

(14, '2024-06-05', 'Positivita al test rapido', 'referti/004.pdf'),

(15, '2024-06-06', 'Versamento pleurico lieve riscontrato', 'referti/005.pdf');

Immagine che contiene testo, schermata, Carattere, software

Il contenuto generato dall'IA potrebbe non essere corretto.

Ogni tabella è accompagnata da vincoli semantici, come:

* CHECK sui campi StatoPrenotazione, Metodo, StatoPagamento.
* UNIQUE sul Codice Fiscale per evitare duplicazioni.
* FOREIGN KEY per assicurare coerenza tra tabelle.

Particolare rilievo assume la **tabella PrenotazioneEsame** che gestisce la relazione molti-a-molti tra prenotazioni ed esami, in quanto una prenotazione può riferire a più esami per il paziente.

## **Capitolo 5**

## **Query sul database ClinicaPrivata**

In quest’ultima parte della relazione, vengono riportate le query sul database:

1. Query di selezione:

La prima Query viene fatta sulla tabella Paziente, in cui si cerca fra i pazienti, quelli nati dopo il 1990-01-01:

-- QUERY DI SELEZIONE

-- Prima query di selezione per i pazienti in tabella nati dopo il 1990

SELECT Nome, Cognome, DataNascita

FROM Paziente AS p

Immagine che contiene testo, schermata, software, Software multimediale

Il contenuto generato dall'IA potrebbe non essere corretto.WHERE p.DataNascita > '1990-01-01';

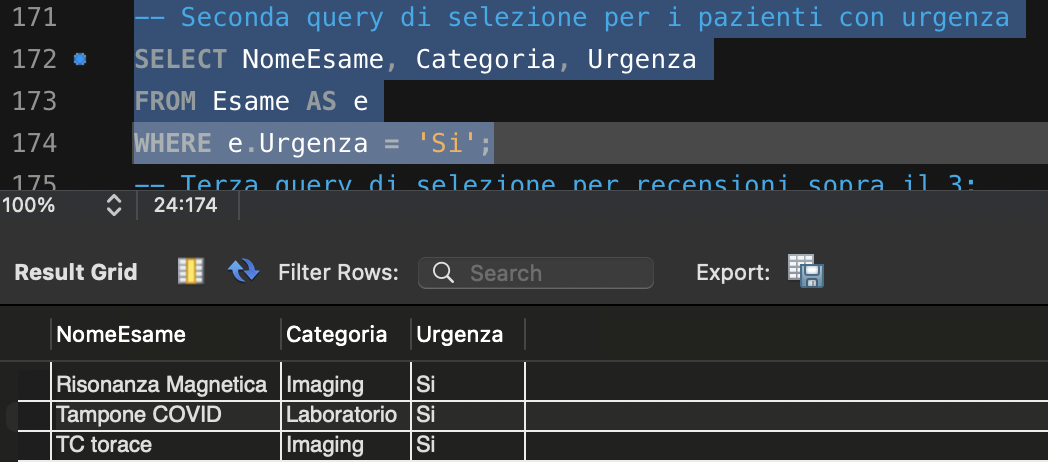
La seconda query viene fatta sulla tabella Esame e vengono selezionati solo quei pazienti che rientrino nella categoria “Urgenze”:

-- Seconda query di selezione per i pazienti con urgenza

SELECT NomeEsame, Categoria, Urgenza

FROM Esame AS e

WHERE e.Urgenza = 'Si';

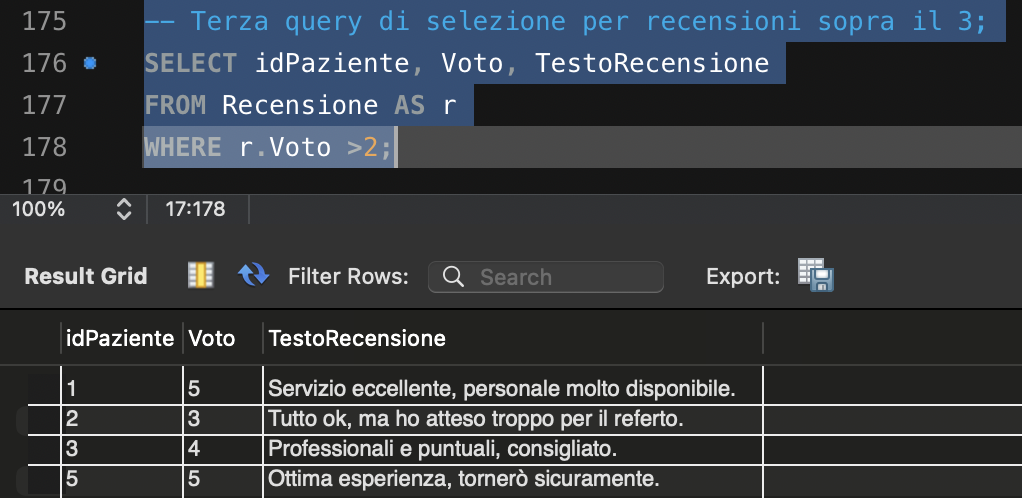


La terza query viene fatta sulla tabella Recensione e vengono selezionate solo le recensioni che abbiano un voto maggiore di 2:

-- Terza query di selezione per recensioni sopra il 3;

SELECT idPaziente, Voto, TestoRecensione

FROM Recensione AS r

WHERE r.Voto >2;

1. Query di join:

Nella prima query di join vengono unite le tabelle Paziente e Prenotazione al fine di trovare individuare il nome dei pazienti e lo stato della loro prenotazione:

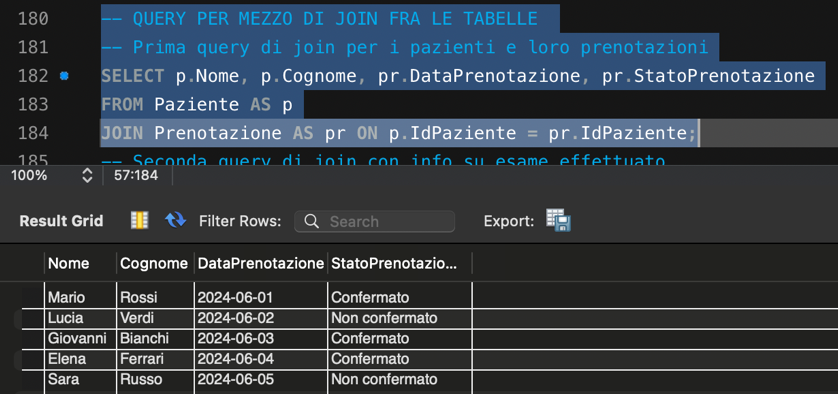
-- QUERY PER MEZZO DI JOIN FRA LE TABELLE

-- Prima query di join per i pazienti e loro prenotazioni

SELECT p.Nome, p.Cognome, pr.DataPrenotazione, pr.StatoPrenotazione

FROM Paziente AS p

JOIN Prenotazione AS pr ON p.IdPaziente = pr.IdPaziente;



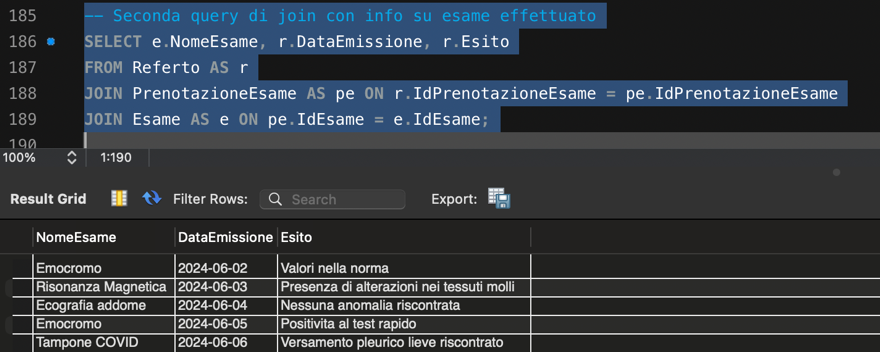
La seconda query di join invece sfrutta l’unione delle due tabelle Referto e PrenotazioneEsame rispetto all’ID della prenotazione e della tabella Esame e PrenotazioneEsame rispetto all’ID dell’esame per individuare il nome degli esami eseguiti dalla clinica e il rispettivo esito a referto:

-- Seconda query di join con info su esame effettuato

SELECT e.NomeEsame, r.DataEmissione, r.Esito

FROM Referto AS r

JOIN PrenotazioneEsame AS pe ON r.IdPrenotazioneEsame = pe.IdPrenotazioneEsame

JOIN Esame AS e ON pe.IdEsame = e.IdEsame;

1. Query con aggregati:

Nella prima query con aggregati si contano nella tabella Esame il numero di esami per categoria d’esame eseguito fra esami di laboratorio ed esami di imaging diagnostico:

-- QUERY CON AGGREGATI

-- Prima query con aggregati per contare il numero di esami per categoria d'esame

SELECT e.Categoria, COUNT(\*) AS NumeroEsami

FROM Esame AS e

GROUP BY e.Categoria;

Immagine che contiene testo, software, Software multimediale, Software per la grafica

Il contenuto generato dall'IA potrebbe non essere corretto.

La seconda query con aggregati calcola la media degli importi dei pagamenti confermati e non della tabella Pagamento:

-- QUERY CON AGGREGATI

-- Prima query con aggregati per contare il numero di esami per categoria d'esame

SELECT e.Categoria, COUNT(\*) AS NumeroEsami

FROM Esame AS e

Immagine che contiene testo, software, Software multimediale, Software per la grafica

Il contenuto generato dall'IA potrebbe non essere corretto.GROUP BY e.Categoria;

1. Query innestate:

Nella prima query innestata si cercano quegli esami nella tabella Esame che avessero un costo superiore alla media fra gli esami presenti in tabella:

-- QUERY INNESTATE

-- Prima query innestata per costi esami superiori alla media

SELECT NomeEsame, Costo

FROM Esame

WHERE Costo > (SELECT AVG(Costo) FROM Esame);

Immagine che contiene testo, schermata, software, Software multimediale

Il contenuto generato dall'IA potrebbe non essere corretto.

La seconda query invece, si basa sulla ricerca dei nomi dei pazienti che presentassero almeno una prenotazione:

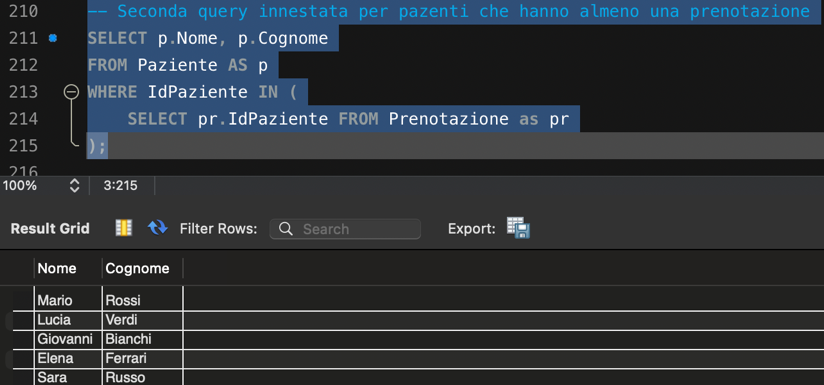
-- Seconda query innestata per pazenti che hanno almeno una prenotazione

SELECT p.Nome, p.Cognome

FROM Paziente AS p

WHERE IdPaziente IN (

SELECT pr.IdPaziente FROM Prenotazione as pr

);