Università degli Studi "Tor Vergata" Corso di Laurea in Informatica
Corso di Basi di Dati e di Conoscenza
Esame del 15 luglio 2024 - Soluzioni

Domanda 1 (30% della valutazione complessiva)

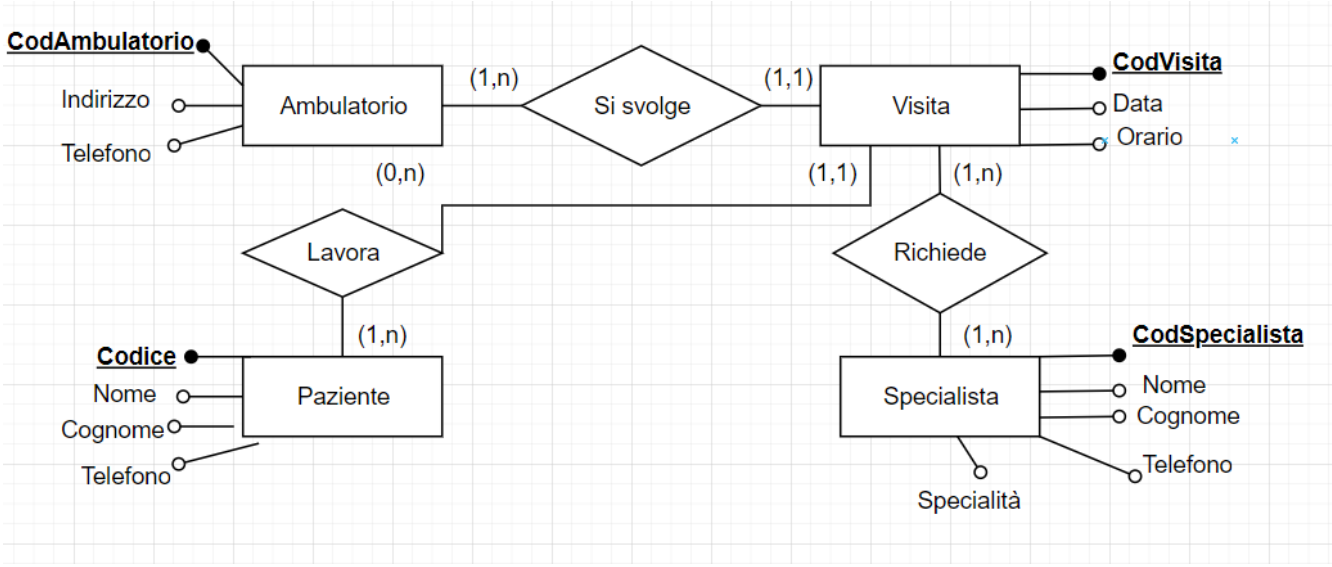
Mostrare uno schema concettuale ER che rappresenti un database per la gestione di **visite mediche** da svolgersi in diversi **ambulatori medic**i.

- Ogni **visita** è svolta in uno ed un solo **ambulatorio** avente un **codice univoco** e caratterizzato da **indirizzo** e n**umero di telefono**
- In un **ambulatorio** possono essere svolte una o più **visite**.
- Ogni **visita** è svolta in una precisa **data** ed orario oltre ad avere un **codice univoc**o che la caratterizza.
- Ogni **visita** può richiedere la prestazione d'opera di uno o più **specialisti**.
- Ogni **specialista** può svolgere più visite
- Ogni **specialista** è identificato da un **codice univoco** e la **specialità**, nome, cognome e recapito telefonico.
- Ogni **visita** è associata ad un **paziente**.
- Un **paziente** può sostenere una o più visite, ciascuna programmata in orari differenti.
- Un **paziente** è caratterizzato dal Codice fiscale, nome, cognome, recapito telefonico.

Determinare:

1. Mostrare lo schema concettuale ER che rappresenti il database.
2. Derivare lo schema relazionale dallo schema concettuale.

Soluzioni:



- **Ambulatorio** (**cod_ambulatorio**, indirizzo, telefono)
- **Visita** (**cod_visita**, data, orario, cod_ambulatorio [FK], cod_paziente [FK])
- **Specialista** (**cod_specialista**, nome, cognome, telefono, specialità)
- **Paziente** (**cod_fiscale**, nome, cognome, telefono)
- **Visita_Specialista** (**cod_visita [FK]**, **cod_specialista [FK]**) *(Tabella di associazione per la relazione N:N tra visita e specialista)*

Domanda 2 (20% della valutazione complessiva)

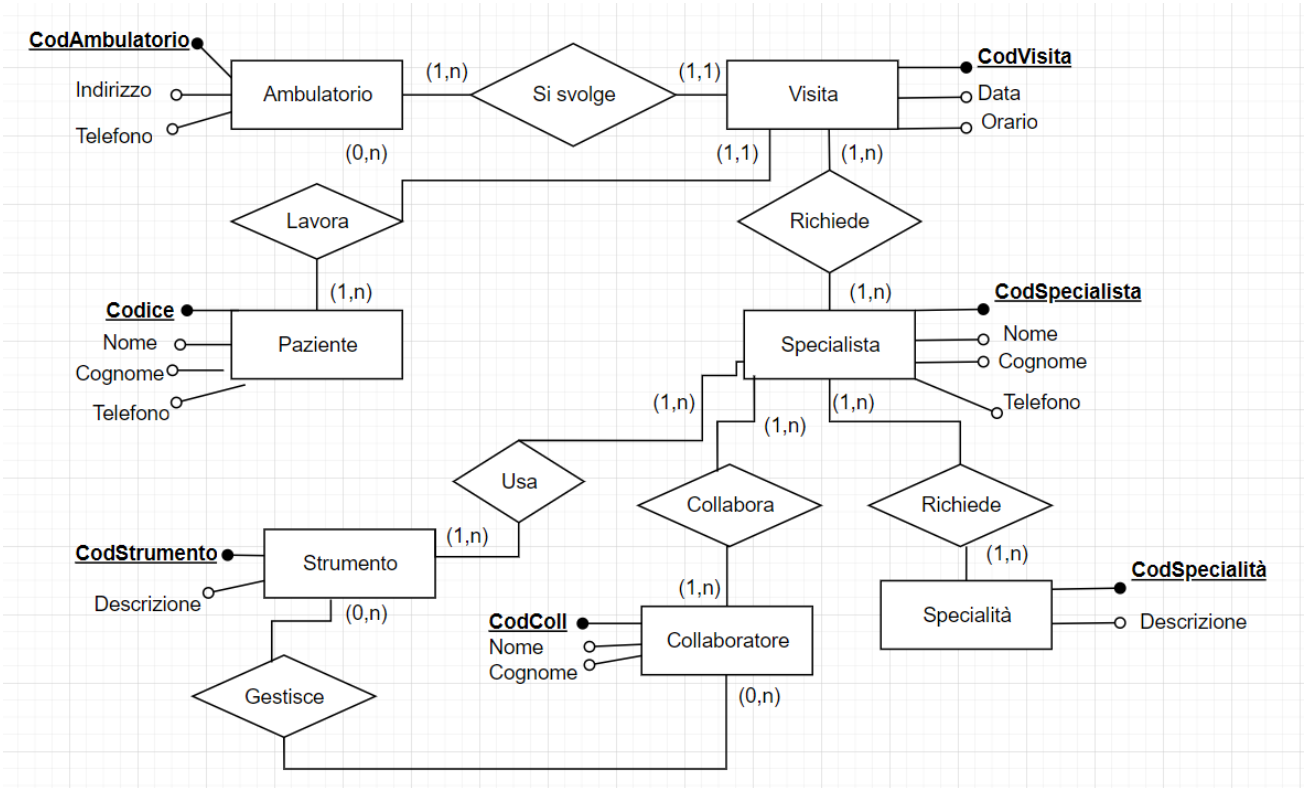
Modificare lo schema in modo tale che:

- Ogni **specialista** può avere almeno una o più **specialità**.
- Ogni **specialista** può avere necessità di un certo insieme di **strumenti** per i quali è presente un codice univoco ed una descrizione.
- Uno **strumento** può essere utilizzato da uno o più **specialisti.**
- Ogni **specialista** può avere bisogno di più **collaboratori** caratterizzati da un codice univoco, nome e cognome.
- Un **collaboratore** collabora con **almeno uno specialista** e può essere responsabile dell'utilizzo di uno o più strumenti.
- Uno **strumento** può essere utilizzato da più collaboratori (ma anche da nessuno)

Determinare:

1. Mostrare lo schema concettuale ER che rappresenti il database.
2. Derivare lo schema relazionale dallo schema concettuale.

Soluzioni:



- **Ambulatorio (cod_ambulatorio,** indirizzo, telefono**)**
- **Visita (cod_visita, d**ata, orario, cod_ambulatorio [FK], cod_paziente [FK]**)**
- **Specialista (cod_specialista,)**
- **Specialità (cod_specialità,** descrizione**)**
- **Specialista_Specialità (cod_specialista [FK], cod_specialità [FK])** *(Tabella di associazione per la relazione N:N)*
- **Paziente (cod_fiscale,** nome, cognome, telefono**)**
- **Visita_Specialista (cod_visita [FK], cod_specialista [FK])** *(Relazione N:N tra visite e specialisti)*
- **Strumento (cod_strumento,** descrizione**)**
- **Specialista_Strumento (cod_specialista [FK], cod_strumento [FK])** *(Relazione N:N tra specialisti e strumenti)*
- **Collaboratore (cod_collaboratore,** nome, cognome**)**
- **Specialista_Collaboratore (cod_specialista [FK], cod_collaboratore [FK])** *(Relazione 1:N tra specialisti e collaboratori)*
- **Collaboratore_Strumento (cod_collaboratore [FK], cod_strumento [FK])** *(Relazione N:N tra collaboratori e strumenti)*

Domanda 3 (30% della valutazione)
In base allo schema relazionale della domanda 2, scrivere le query in SQL che rispondono alle seguenti domande.

a) Stampare l'agenda giornaliera degli specialisti.
b) Per ogni strumento trovare lo specialista che lo ha maggiormente usato
c) Per ogni paziente stampare ile visite effettuate nell'ultimo anno.
d) Scrivere in algebra relazionale la query c).

Soluzioni:

a) Stampare l'agenda giornaliera degli specialisti.

```
SELECT s.cod_specialista, s.nome, s.cognome,
     v.cod_visita, v.data, v.orario, p.nome AS nome_paziente,
     p.cognome AS cognome_paziente
FROM Visita v
JOIN Visita_Specialista vs ON v.cod_visita = vs.cod_visita
JOIN Specialista s ON vs.cod_specialista = s.cod_specialista
JOIN Paziente p ON v.cod_paziente = p.cod_fiscale
WHERE v.data = CURRENT_DATE -- Filtra solo per la data odierna
ORDER BY s.cod_specialista, v.orario;
```

b) Per ogni strumento trovare lo specialista che lo ha maggiormente usato

1) Con clausola WITH

```
WITH UsageCount AS (
    SELECT
        ss.cod_strumento, ss.cod_specialista, COUNT (*) AS utilizzi
    FROM
        Specialista_Strumento ss
```

```
    GROUP BY
        ss.cod_strumento, ss.cod_specialista
)
SELECT
     uc.cod_strumento, s.cod_specialista,
     s.nome, s.cognome, uc.utilizzi
FROM
    UsageCount uc
JOIN Specialista s ON uc.cod_specialista = s.cod_specialista
WHERE
    uc.utilizzi = (
     SELECT MAX (utilizzi)
     FROM UsageCount uc2 WHERE uc2.cod_strumento = uc.cod_strumento);
```

2) Con Subquery

```
SELECT ss.cod_strumento, ss.cod_specialista, s.nome, s.cognome,
       (SELECT COUNT(*)
        FROM Specialista_Strumento ss2
        WHERE ss2.cod_strumento = ss.cod_strumento
          AND ss2.cod_specialista = ss.cod_specialista)
               AS utilizzi
FROM Specialista_Strumento ss
JOIN Specialista s ON ss.cod_specialista = s.cod_specialista
WHERE (ss.cod_strumento,
       (SELECT COUNT(*)
        FROM Specialista_Strumento ss2
        WHERE ss2.cod_strumento = ss.cod_strumento
        GROUP BY ss2.cod_strumento
        ORDER BY COUNT(*) DESC
        LIMIT 1))
      IN
      (SELECT cod_strumento, MAX(utilizzi)
       FROM (SELECT cod_strumento, cod_specialista, COUNT(*)
          AS utilizzi
             FROM Specialista_Strumento
             GROUP BY cod_strumento, cod_specialista)
          AS utilizzo_max
             GROUP BY cod_strumento);
```

c. Per ogni paziente stampare le visite effettuate nell'ultimo anno.

```
SELECT
    p.cod_fiscale, p.nome, p.cognome, v.cod_visita, v.data, v.orario
FROM
    Paziente p
JOIN Visita v ON p.cod_fiscale = v.cod_paziente
```

```
WHERE
    v.data >= CURRENT_DATE - INTERVAL '1 YEAR'
ORDER BY
    p.cod_fiscale, v.data DESC;
```

d) Scrivere in algebra relazionale la query c).

$$\pi_{cod\_fiscale,nome,cognome,cod\_visita,data,orario}\left(\sigma_{data \geq CURRENT\_DATE - INTERVAL'1YEAR'}\left(Paziente \bowtie Visita\right)\right)$$

Domanda 4 (20% della valutazione)
Considerare lil seguente schema relazionale NON in 1 NF, 2 NF, e 3NF.

1. **Libri** (CodLibro, Titolo, Autore, Genere, Editore, IndirizzoEditore, CittàEditore, AnnoPubblicazione, ISBN, NomeSezione, PosizioneScaffale)
2. **Prestiti** (CodPrestito, CodLibro, Titolo, CodUtente, NomeUtente, CognomeUtente, Telefono, DataPrestito, DataScadenza, DataRestituzione)
3. **Utenti** (CodUtente, NomeUtente, CognomeUtente, Indirizzo, Città, CAP, Telefono, E-mail, CodLibroAttualmenteInPrestito, TitoloLibroInPrestito, DataPrestito, DataScadenza),

Determinare:
1. Le violazioni delle forme normali presenti nello schema relazionale iniziale.
2. Uno schema relazionale normalizzato fino alla terza forma normale (3NF).
3. Verificare la BCNF.
4. Un modello Entità-Relazione (ER) che rappresenti in modo chiaro e strutturato la base di dati.

Soluzioni:

1. Analisi delle violazioni delle forme normali

Violazioni della 1NF (Prima Forma Normale)

- **Dati non atomici**:
  - La tabella **Utenti** contiene **CodLibroAttualmenteInPrestito, TitoloLibroInPrestito**, che suggerisce che un utente può avere più libri in prestito, ma sono memorizzati in un solo campo (violazione della atomicità dei dati).
  - In **Libri**, il campo **Autore** può contenere più valori se un libro ha più autori (non è atomico).
- **Valori ripetuti**:
  - **Editore, IndirizzoEditore, CittàEditore** sono ripetuti in più righe se lo stesso editore pubblica più libri.
  - **NomeUtente, CognomeUtente, Telefono** in **Prestiti** ripetono informazioni già presenti in **Utenti**.

Violazioni della 2NF (Seconda Forma Normale)

- **Dipendenze parziali**:
  - In **Libri**, **Editore → IndirizzoEditore, CittàEditore**, quindi l'indirizzo dipende da Editore e non da CodLibro.
  - In **Prestiti**, **Titolo** dipende solo da **CodLibro**, ma non dall'intera chiave primaria.
  - In **Utenti**, il nome e il cognome dell'utente dipendono solo da **CodUtente**, non dall'intera chiave primaria.

Violazioni della 3NF (Terza Forma Normale)

- **Dipendenze transitive**:
  - In **Libri**, **IndirizzoEditore, CittàEditore** dipendono da **Editore**, che a sua volta dipende da **CodLibro**.
  - In **Prestiti**, **NomeUtente, CognomeUtente, Telefono** dipendono da **CodUtente**, che non è chiave primaria.

2. Schema relazionale normalizzato fino alla 3NF

1. **Libri** (**CodLibro** PK, Titolo, CodAutore FK, CodEditore FK, AnnoPubblicazione, ISBN, CodSezione FK, PosizioneScaffale)
2. **Autori** (**CodAutore** PK, Nome, Cognome)
3. **Editori** (**CodEditore** PK, NomeEditore, IndirizzoEditore, CittàEditore)
4. **Sezioni** (**CodSezione** PK, NomeSezione)
5. **Utenti** (**CodUtente** PK, Nome, Cognome, Indirizzo, Città, CAP, Telefono, Email)
6. **Prestiti** (**CodPrestito** PK, CodLibro FK, CodUtente FK, DataPrestito, DataScadenza, DataRestituzione)

Risultati della normalizzazione:

- **Eliminata la ripetizione delle informazioni sugli editori** spostandole in una tabella separata.
- **Eliminata la ridondanza nei prestiti** rimuovendo dati anagrafici duplicati.
- **Gestione degli autori separata** per consentire più autori per libro.

3. Verifica della BCNF (Forma Normale di Boyce-Codd)

La BCNF richiede che **ogni dipendenza funzionale sia determinata da una superchiave**.

- **Libri**: CodLibro → Titolo, CodAutore, CodEditore, AnnoPubblicazione, ISBN, CodSezione, PosizioneScaffale
- **Autori**: CodAutore → Nome, Cognome
- **Editori**: CodEditore → NomeEditore, IndirizzoEditore, CittàEditore
- **Sezioni**: CodSezione → NomeSezione
- **Utenti**: CodUtente → Nome, Cognome, Indirizzo, Città, CAP, Telefono, Email
- **Prestiti**: CodPrestito → CodLibro, CodUtente, DataPrestito, DataScadenza, DataRestituzione

Lo schema è in BCNF, poiché non ci sono dipendenze funzionali in cui un attributo non chiave dipende da un attributo non superchiave.

4. Modello ER (Entità-Relazione)

Entità principali:

- **Libro** (CodLibro, Titolo, AnnoPubblicazione, ISBN, PosizioneScaffale)
- **Autore** (CodAutore, Nome, Cognome)
- **Editore** (CodEditore, NomeEditore, Indirizzo, Città)
- **Sezione** (CodSezione, NomeSezione)
- **Utente** (CodUtente, Nome, Cognome, Indirizzo, Telefono, Email)
- **Prestito** (CodPrestito, DataPrestito, DataScadenza, DataRestituzione)

Relazioni:

- **Libro ha un Autore** (1:N → un autore può scrivere più libri, un libro ha un solo autore).
- **Libro è pubblicato da un Editore** (N:1 → un editore pubblica più libri, un libro ha un solo editore).
- **Libro appartiene a una Sezione** (N:1 → una sezione contiene più libri, un libro è in una sola sezione).
- **Utente effettua Prestiti** (1:N → un utente può prendere più libri in prestito, un prestito è legato a un solo utente).
- **Prestito riguarda un Libro** (N:1 → un libro può essere prestato più volte, un prestito riguarda un solo libro).