Compito Basi di Dati. Tempo concesso: 2 ore **11 Gennaio 2023**

Nome: ______________ Cognome: _________________ Matricola: __________

MANCAVA
1H

Esercizio 1

Si considerino le seguenti specifiche per la realizzazione del sistema informativo per un club nautico.

Un club nautico richiede la progettazione di una base di dati la gestione delle informazioni sulle sue imbarcazioni e sui tecnici suoi dipendenti.
Per ciascuna imbarcazione si memorizzano nel sistema il numero di matricola (che le identifica univocamente), il modello ed eventualmente un nome; ciascun modello è caratterizzato da un codice identificativo, la lunghezza, la stazza e il pescaggio.
Dovranno inoltre essere memorizzati nel sistema i dati dei tecnici dipendenti del club nautico. Per ciascun tecnico si memorizzano codice fiscale, nome, cognome, indirizzo, telefono e stipendio. Inoltre, ogni tecnico è abilitato all'intervento su uno o più modelli di imbarcazione.
Il sistema deve infine gestire i test di abilitazione alla navigazione delle imbarcazioni. Ogni tipo di test ha un codice, un nome e un punteggio massimo. Per ogni test effettuato su una determinata imbarcazione è necessario memorizzare le informazioni riguardanti il tecnico che lo ha eseguito, la data in cui è stato effettuato, il tempo impiegato per effettuarlo ed il punteggio assegnato all'imbarcazione. Ai fini della modellazione si consideri che nella stessa data possono essere effettuati test diversi sulla stessa imbarcazione e che lo stesso tipo di test può essere ripetuto su una stessa imbarcazione più volte ma solo in date diverse.

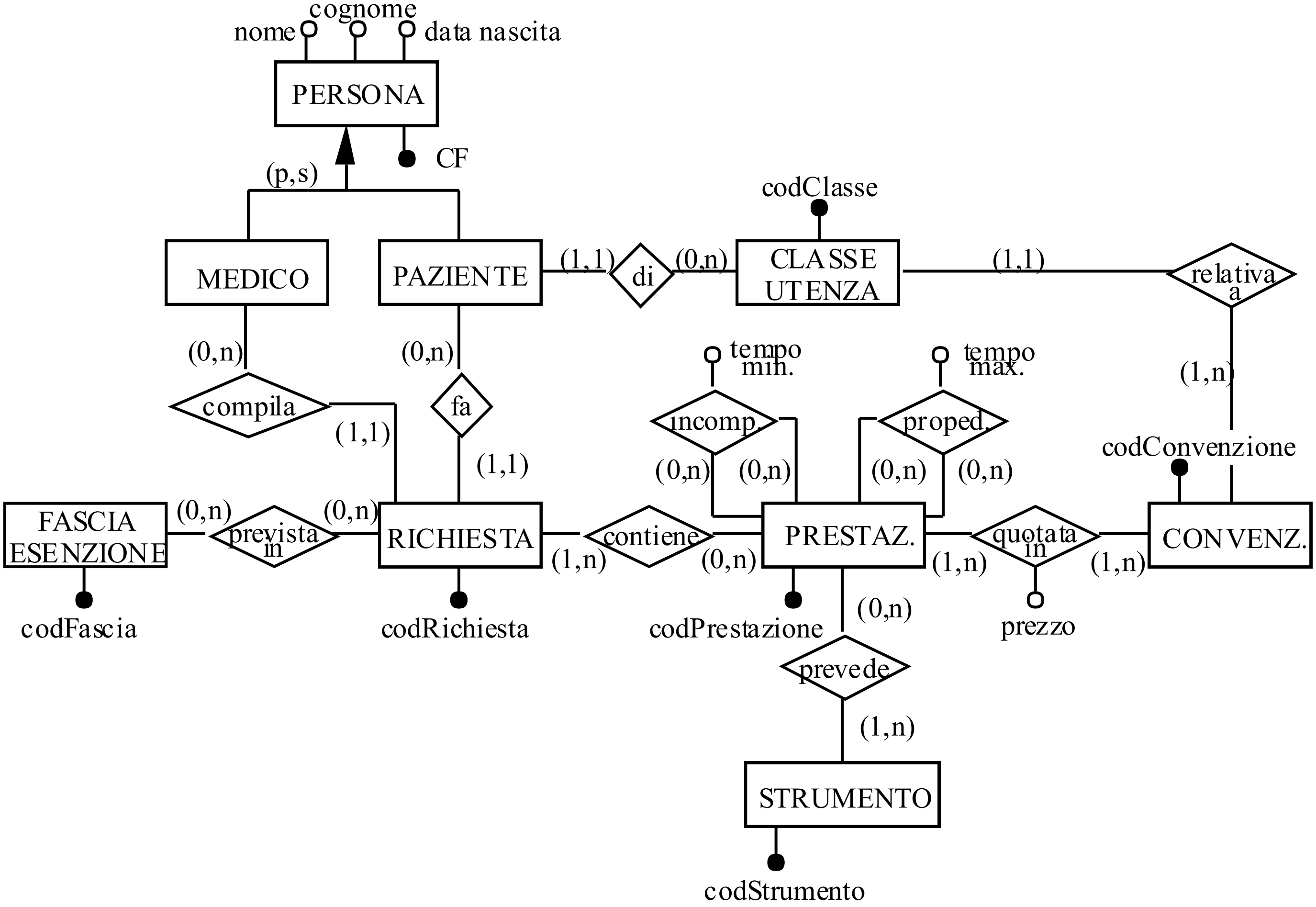
Si definisca il relativo ***schema E/R*** *(nella metodologia proposta a lezione) e si indichino esplicitamente eventuali* ***attributi derivati*** *presenti nelle specifiche, commentandoli adeguatamente. Si evidenzino inoltre eventuali* ***vincoli inespressi****.*

Svolgimento

Nome: _______________ **Cognome:** __________________ **Matricola:** ___________

Esercizio 2

Si effettui la progettazione logica del seguente schema concettuale, producendo uno schema relazionale quanto più fedele possibile allo schema di partenza.



Svolgimento

Nome: _______________ Cognome: __________________ Matricola: __________

Esercizio 3

- Nell’ambito della normalizzazione di schemi relazionali si definiscano formalmente la prima, la seconda e la terza forma normale (1NF, 2NF e 3NF).

- Si consideri il seguente schema relazionale:

  APPARTAMENTI (codAppartamento, indirizzo, mq, numeroCamere, referente, e-mail, prezzoGiornaliero, numeroPersone, dataInizio, dataFine)
  PRENOTAZIONI (codCliente, cognome, nome, telefono, dataArrivo, dataPartenza, dataPrenotazione, numeroPersone, codAppartamento)

  Sapendo che:
  - Ogni appartamento è identificato da un codice ed è caratterizzato da un indirizzo, la superficie in mq, il numero di camere presenti, il nominativo del referente e il suo indirizzo email;
  - Il prezzo degli appartamenti varia a seconda del periodo dell’anno e del numero di persone che alloggiano nell’appartamento;
  - Ogni cliente può prenotare, in una stessa data (dataPrenotazione), appartamenti diversi, specificando per ciascun appartamento il periodo di soggiorno e il numero di persone che vi alloggeranno;
  - I clienti che effettuano prenotazioni sono memorizzati nel sistema, registrando per ciascuno un codice identificativo, il cognome, il nome e un recapito telefonico;

  1) elencare le dipendenze funzionali non banali presenti nello schema;
  2) indicare in quale forma normale si trova lo schema dato, motivando la risposta; 1NF perché ci sono dipendenze funzionali parziali
  3) se lo schema non è in 3NF, trovarne una decomposizione in schemi in 3NF che preservi le dipendenze e risulti senza perdita.

Svolgimento

codAppartamento -> indirizzo, mq, numeroCamere, referente, e-mail (P)
codCliente -> cognome, nome, telefono (P)
codCliente, dataPrenotazione, codAppartamento -> dataArrivo, dataPartenza, numeroPersone (Dipendenza da Preservare)
codAppartamento, numeroPersone, dataInizio -> prezzoGiornaliero (Dipendenza da Preservare)

APPARTAMENTI(codAppartamento, indirizzo, mq, numeroCamere, referente, email)
PREZZI(codAppartamento, numeroPersone, dataInizio, dataFine, prezzoGiornaliero)
CLIENTI(codCliente, cognome, nome, telefono)
PRENOTAZIONI(codCliente, dataPrenotazione, codAppartamento, dataArrivo, dataPartenza, numeroPersone)

Nome: _______________ **Cognome:** ___________________ **Matricola:** ___________

Esercizio 4

Dato il seguente schema relazionale:

IMPIEGATI (codImpiegato, CF, cognome, nome, indirizzo, telefono, dataAssunzione)
STORICOMANSIONI (codImpiegato, dataInizio, dataFine*, codMansione: MANSIONI)
TURNISETTIMANALI (codImpiegato: IMPIEGATI, giorno, oraInizio, oraFine)
MANSIONI (codMansione, nomeMansione, descrizione)

1. Scrivere un’espressione di algebra relazionale che selezioni gli impiegati che non lavorano il martedì (codImpiegato, cognome, nome).
2. Scrivere una query SQL che visualizzi gli impiegati che, nel corso della propria attività lavorativa, hanno svolto almeno due diverse mansioni (codImpiegato, cognome, nome, numeroMansioni).
3. Scrivere una query SQL che selezioni le mansioni che attualmente non sono svolte da nessun impiegato (codMansione, nomeMansione).
4. Scrivere una query SQL che visualizzi gli impiegati che, nel corso della propria attività lavorativa, hanno svolto tutte le mansioni (codImpiegato, cognome, nome, dataAssunzione).

Svolgimento

1)

```
pi codImpiegato (IMPIEGATI) - (pi codImpiegato(sigma giorno='Martedí'(IMPIEGATI natural join TURNISETTIMANALI)))
```

2)

```
SELECT I.codImpiegato, I.cognome, I.nome, COUNT(DISTINCT S.codMansione) AS NumeroMansioni
FROM IMPIEGATI I JOIN STORICOMANSIONI S ON (I.codImpiegato=S.codImpiegato)
GROUP BY I.codImpiegato, I.cognome, I.nome
HAVING NumeroMansioni >= 2
```

3)

```
SELECT codMansione, nomeMansione
FROM MANSIONI
WHERE codMansione NOT IN (SELECT codMansione
FROM STORICOMANSIONI
WHERE datafine IS NULL)
```

4)

```
SELECT I.codImpiegato, cognome, nome, dataAssunzione
FROM IMPIEGATI I, STORICOMANSIONI S
WHERE I.codImpiegato = S.codImpiegato
GROUP BY I.codImpiegato, cognome, nome, dataAssunzione
HAVING COUNT(DISTINCT codMansione) = (SELECT COUNT(*) FROM MANSIONI)
```