

Nome: _____ Cognome: _____ Matricola: _____

Esercizio 1

Si considerino le seguenti specifiche relative alla realizzazione del sistema informativo per una società di catering.

La società di catering Gourmet ha deciso di sviluppare un nuovo sistema informatico per gestire le informazioni relative ai clienti, agli eventi, ai menu e al personale. Il sistema deve essere in grado di memorizzare e gestire tutte le informazioni necessarie per facilitare l'organizzazione e la gestione delle attività di catering.

La società offre diversi menù, ciascuno identificato da un codice univoco, e caratterizzato da un nome, una breve descrizione, il prezzo per persona, il tipo di menu (onnivoro, vegetariano o vegano) e l'elenco dei piatti che lo compongono. Per ciascun piatto si memorizzano un codice identificativo, il nome e l'elenco di ingredienti utilizzati per la sua preparazione.

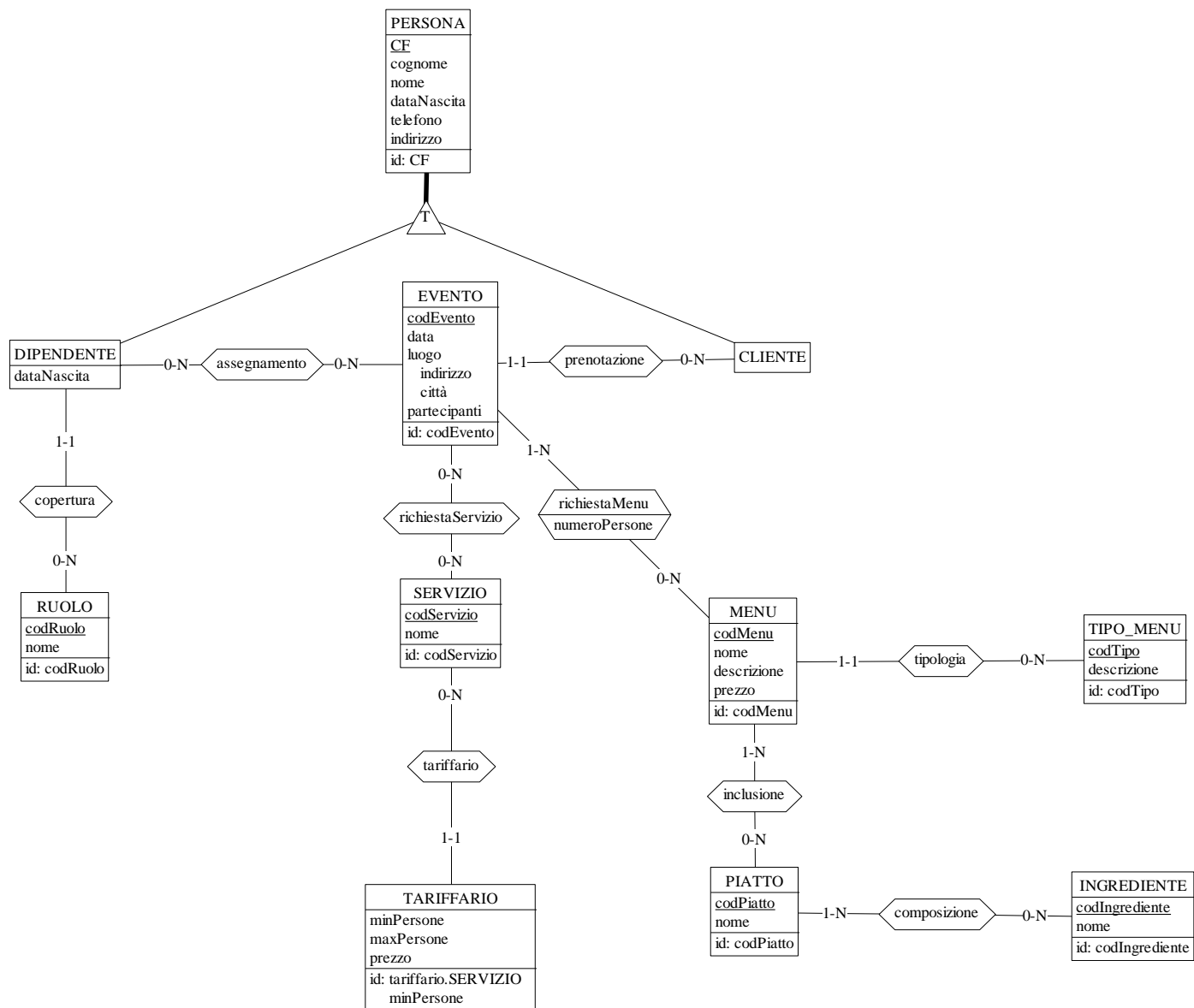
Oltre alla preparazione del cibo, la società di catering offre dei servizi aggiuntivi (es. allestimento, pulizia finale) il cui costo dipende dal numero di persone presenti all'evento (Es. da 0 a 50 persone, da 50 a 100, ecc...).

I clienti della società possono effettuare delle prenotazioni per eventi specifici. Di ogni evento, oltre ai dati anagrafici, indirizzo e recapito del cliente, si memorizzano la data, il luogo (indirizzo completo), il tipo di evento (es. matrimonio, conferenza, evento aziendale), il numero atteso di partecipanti, i menù scelti (indicando per ciascuno il numero di persone) e i servizi aggiuntivi richiesti.

Ad ogni evento la società assegna alcuni membri del proprio personale. Per ciascuno si memorizzano nome, cognome, data di nascita, ruolo (es. chef, cameriere, addetto alla logistica), indirizzo e un recapito telefonico.

*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**.*

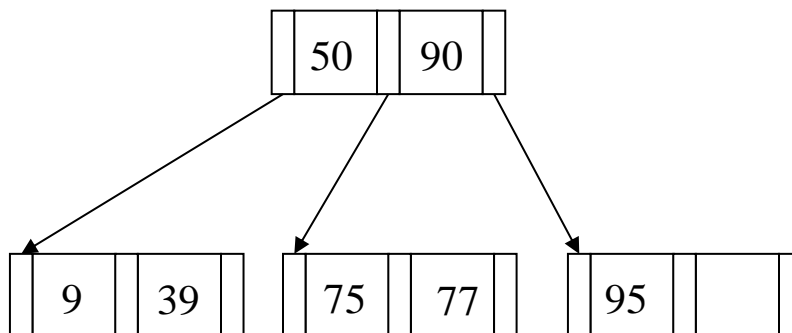
Svolgimento



Nome: _____ Cognome: _____ Matricola: _____

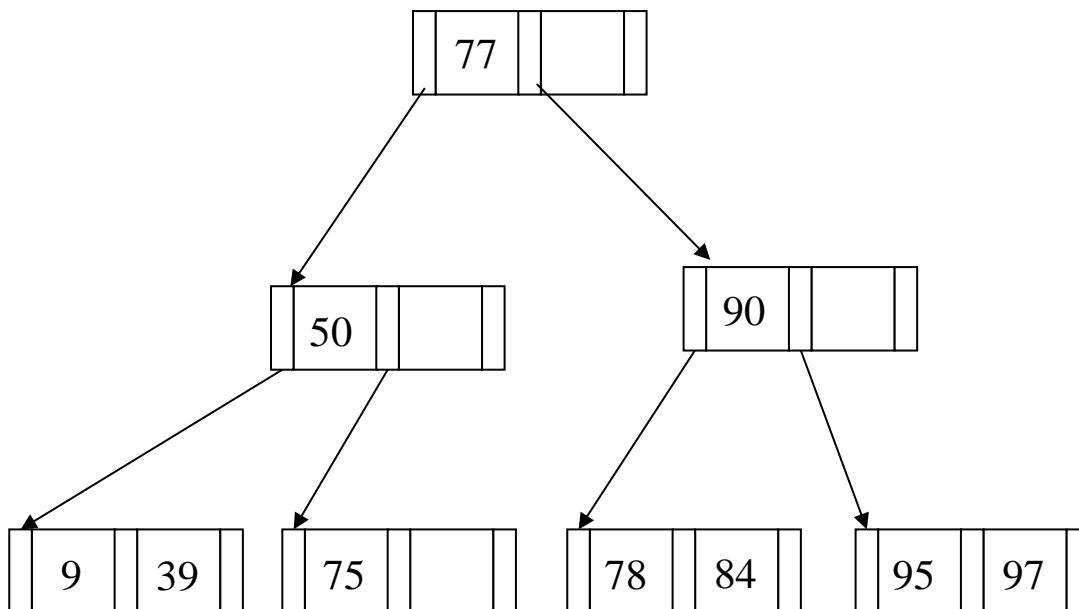
Esercizio 2

- 1) Si descrivano brevemente le strutture B-tree e B+-tree, evidenziandone le principali differenze.
- 2) Si consideri il B-tree sotto riportato (ordine $g=1$), nell'ipotesi di assenza di gestione dell'overflow:



Disegnare la struttura risultante dopo l'inserimento dei seguenti valori di chiave: 97, 78, 84.

Svolgimento



Esercizio 3

È dato il seguente schema relazionale che gestisce gli appuntamenti di una parrucchiera:

PRENOTAZIONI (IDPrenotazione, DataPrenotazione, OraPrenotazione, IDCliente, NomeCliente, CognomeCliente, IDServizio, DescrizioneServizio, PrezzoServizio, CostoTotale)

Sapendo che:

- Ogni prenotazione è identificata da un codice univoco ed è associata al cliente che l'ha effettuata;
 - Ogni prenotazione può includere più servizi;
 - I prezzi dei servizi sono memorizzati in un tariffario;
 - L'attributo CostoTotale rappresenta il costo totale della prenotazione, ottenuto come somma dei prezzi dei servizi richiesti;
- 1) si evidenzino tutte le *dipendenze funzionali* non banali presenti nello schema, specificandone anche il tipo in caso di dipendenza "problematica";
 - 2) si indichi qual è la forma normale in cui si trova lo schema iniziale;
 - 3) se lo schema non risulta normalizzato, si determini uno *schema in terza forma normale* che risulti equivalente, dal punto di vista informativo, allo schema dato.

Svolgimento

DF:

IDPrenotazione → DataPrenotazione, OraPrenotazione, IDCliente, CostoTotale (DP)

IDCliente → NomeCliente, CognomeCliente (DT)

IDServizio → DescrizioneServizio, PrezzoServizio (DP)

Lo schema è in 1NF.

Schema normalizzato:

CLIENTI (IDCliente, NomeCliente, CognomeCliente)

PRENOTAZIONI (IDPrenotazione, DataPrenotazione, OraPrenotazione, IDCliente: CLIENTI, CostoTotale)

SERVIZI (IDServizio, DescrizioneServizio, PrezzoServizio)

SERVIZI_PRENOTATI (IDPrenotazione: PRENOTAZIONI, IDServizio: SERVIZI)

Nome: _____ **Cognome:** _____ **Matricola:** _____

Esercizio 4

Dato il seguente schema relazionale relativo a uffici collocati in diverse sedi e ai loro dipendenti

DIPENDENTI (CF, Nome, Cognome, CodCategoria: CATEGORIE)

CATEGORIE (CodCategoria, Descrizione, StipendioBase)

SEDI (CodSede, Città, Indirizzo)UFFICI (CodUfficio, Nome, CodSede: SEDI)

AFFERENZE (CF: DIPENDENTI, CodUfficio: UFFICI, DataInizio, DataFine*)

1. Scrivere un'espressione di algebra relazionale che selezioni i dipendenti che hanno lavorato in tutte le sedi (CF, Nome, Cognome).
2. Scrivere una query SQL che visualizzi il numero attuale di dipendenti per ciascuna città in cui l'azienda ha sedi (Città, NumeroDipendenti).
3. Visualizzare tramite query SQL il dipendente che ha lavorato durante la sua carriera nel maggior numero di uffici (CF, Nome, Cognome, NumeroUffici).
4. Scrivere una query SQL che visualizzi i dipendenti che appartengono alla categoria con lo stipendio base più basso (CF, Cognome, Nome).

Svolgimento

1. $\pi_{CF, Nome, Cognome}((\pi_{codSede, CF}(AFFERENZE \bowtie UFFICI) \div \pi_{codSede}(SEDI)) \bowtie DIPENDENTI)$
2. SELECT Città, COUNT(DISTINCT A.CF) AS NumeroDipendenti
FROM AFFERENZE A, UFFICI U, SEDI S
WHERE A.CodUfficio = U.codUfficio
AND U.CodSede = S.CodSede
AND DataFine IS NULL
GROUP BY Città
3. SELECT TOP(1) WITH TIES CF, Nome, Cognome, COUNT(DISTINCT CodUfficio) AS NumeroUffici
FROM DIPENDENTI D, AFFERENZE A
WHERE D.CF = A.CF
GROUP BY CF, Nome, Cognome
ORDER BY COUNT(DISTINCT CodUfficio) DESC
4. SELECT CF, Cognome, Nome
FROM DIPENDENTI D
WHERE CodCategoria IN (SELECT TOP(1) WITH TIES C.CodCategoria
FROM CATEGORIE C
ORDER BY StipendioBase)

SELECT CF, Cognome, Nome

FROM DIPENDENTI D

WHERE CodCategoria IN (SELECT C.CodCategoria

FROM CATEGORIE C

WHERE StipendioBase = (SELECT MIN(C1.StipendioBase)

FROM CATEGORIE C1)

```
)  
SELECT CF, Cognome, Nome  
FROM DIPENDENTI D, CATEGORIE C  
WHERE D.CodCategoria = C.CodCategoria  
AND C.StipendioBase = (SELECT MIN(C1.StipendioBase)  
                        FROM CATEGORIE C1)
```