

Basi di Dati 1
 Prof. L. Tanca e F. A. Schreiber
APPELLO DEL 9 SETTEMBRE 2015
Tempo: 2h30m

Si consideri il seguente schema di base di dati, che vuole memorizzare informazioni relative a un sito di affitto case per privati.

IMMOBILE (Codice, Superficie, Indirizzo, Città, Nazione, Tipologia, NumPostiLetto, NumBagni, PrezzoGiorno, IDProprietario)
 PERSONA (ID, Cognome, Nome, Indirizzo, Città, Nazione)
 PRENOTAZIONE (ID, CodiceImmobile, IDOspite, DataInizio, Numerogiorni)
 PAGAMENTO (IDPrenotazione, Data, CostoServiziAggiuntivi, TotalePagamento)

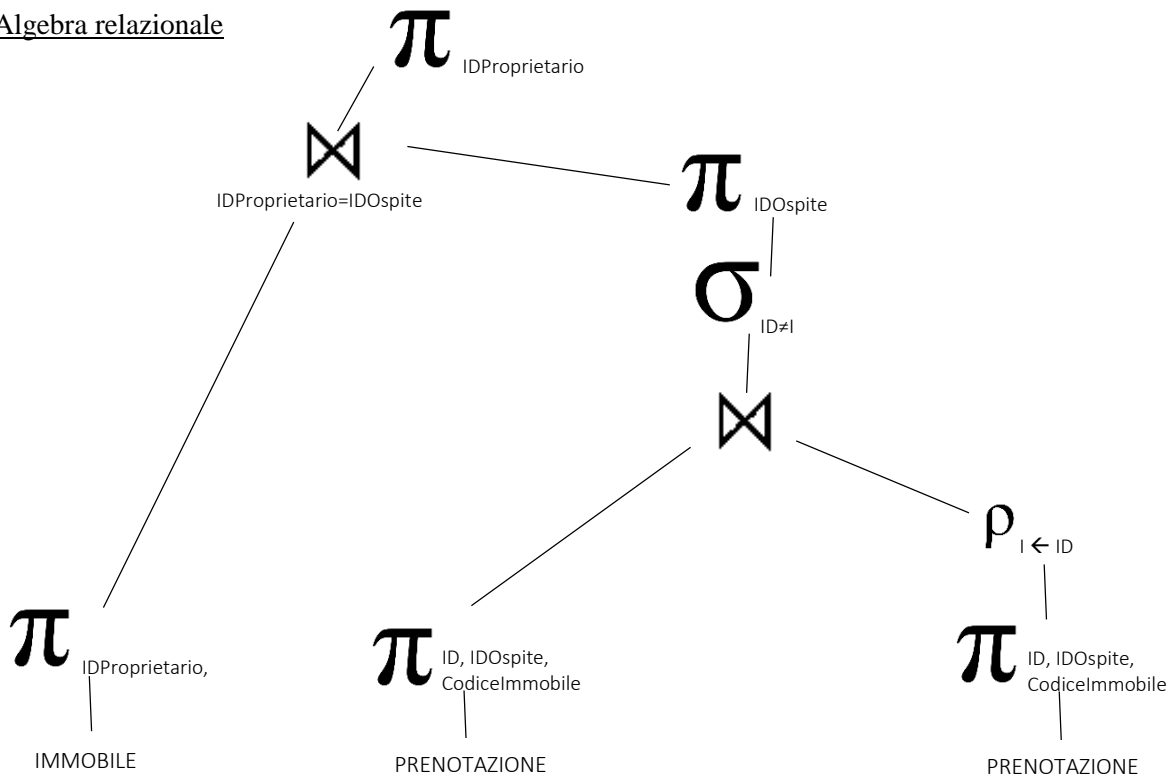
Si noti che l'attributo "Tipologia" della tabella IMMOBILE caratterizza il tipo di immobile e può assumere valori quali "Appartamento", "Casa indipendente", "Stanza".

A. LINGUAGGI FORMALI (6 punti).

Esprimere in algebra relazionale (ottimizzata) e calcolo relazionale la seguente interrogazione:

Trovare le persone che hanno preso in affitto lo stesso immobile almeno due volte in periodi diversi, e inoltre sono a loro volta proprietarie di almeno un immobile. (4 punti)

Algebra relazionale



Calcolo relazionale

$\{t \mid \exists t1 \in \text{Immobile}, \exists t2 \in \text{Prenotazione}, \exists t3 \in \text{Prenotazione} (t[\text{IDProprietario}] = t1[\text{IDProprietario}] \wedge t2[\text{IDospite}] = t3[\text{IDospite}] \wedge t2[\text{CodiceImmobile}] = t3[\text{CodiceImmobile}] \wedge t1[\text{IDProprietario}] = t2[\text{IDospite}] \wedge t2[\text{ID}] \neq t3[\text{ID}])\}$

Esprimere in datalog la seguente interrogazione:

Trovare il codice degli immobili che sono stati affittati sempre a persone residenti in nazioni diverse rispetto a quelle in cui sono situati. (2 punti)

ImmobileStessaNazione(Codice) :- Immobile(Codice, _, _, _, Nazione, _, _, _, _),
Prenotazione(_, Codice, IDospite, _, _),
Persona(IDospite, _, _, _, Nazione)
ImmobileNazioniDiverse(Codice) :- Prenotazione(_, Codice, _, _, _),
¬ImmobileStessaNazione(Codice)
? - ImmobileNazioniDiverse(x)

B. SQL: INTERROGAZIONI (7 punti).

Scrivere in SQL le seguenti interrogazioni:

1. *Trovare il codice e l'indirizzo degli appartamenti di New York che sono stati affittati più di cinque volte ma mai per periodi maggiori di 10 giorni. (3.5 punti)*

```
SELECT I.Codice, I.Indirizzo
FROM Immobile AS I, Prenotazione AS P
WHERE I.Codice=P.CodiceImmobile AND I.Città='New York' AND
      I.Tipologia='Appartamento' AND I.Codice NOT IN (
      SELECT CodiceImmobile
      FROM Prenotazione
      WHERE NumeroGiorni>10
      )
GROUP BY I.Codice, I.Indirizzo
HAVING COUNT(*)>5
```

2. *Trovare, per ogni nazione, il nome e il cognome dei proprietari di stanze che sono residenti in quella nazione e che hanno incassato la somma massima dall'inizio del servizio. (3.5 punti)*

```
CREATE VIEW PropNazIncasso (CodProp, Nome, Cognome, Nazione, Incasso) AS (  
  SELECT Pe.ID, Pe.Nome, Pe.Cognome, Pe.Nazione, SUM(Pa.TotalePagamento)  
  FROM Persona AS Pe, Immobile AS I, Prenotazione AS Pr, Pagamento AS Pa  
  WHERE Pe.ID=I.IDProprietario AND I.Codice=Pr.CodiceImmobile AND  
    Pr.ID=Pa.IDPrenotazione AND I.Tipologia='Stanza'  
  GROUP BY Pe.ID, Pe.Nome, Pe.Cognome, Pe.Nazione  
)
```

```
SELECT P.Nome, P.Cognome, P.Nazione  
FROM PropNazIncasso AS P  
WHERE P.Incasso >= ALL (  
  SELECT P2.Incasso  
  FROM PropNazIncasso AS P2  
  WHERE P2.Nazione=P.Nazione  
)
```

C. DDL: Definizione di Schemi e Vincoli (3 punti)

1. Specificare in SQL la creazione delle tabelle IMMOBILE e PRENOTAZIONE, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo eventuali vincoli di integrità referenziale relativi a tutte le tabelle dello schema. (1 punto)

```
CREATE TABLE Immobile (  
  Codice VARCHAR(255) PRIMARY KEY,  
  Superficie DOUBLE PRECISION,  
  Indirizzo VARCHAR(255),  
  Città VARCHAR(255),  
  Nazione VARCHAR(255),  
  Tipologia ENUM('Appartamento', 'Casa indipendente', 'Stanza'),  
  NumPostiLetto INTEGER,  
  NumBagni INTEGER,  
  PrezzoGiorno NUMERIC(7,2),  
  IDProprietario VARCHAR(255) REFERENCES Persona(ID) ON UPDATE  
    CASCADE ON DELETE NO ACTION  
)  
  
CREATE TABLE Prenotazione (  
  ID VARCHAR(255) PRIMARY KEY,  
  CodiceImmobile VARCHAR(255) REFERENCES Immobile(Codice) ON UPDATE  
    CASCADE ON DELETE CASCADE,  
  IDospite VARCHAR(255) REFERENCES Persona(ID) ON UPDATE CASCADE  
    ON DELETE NO ACTION,  
  DataInizio DATE,  
  NumeroGiorni INTEGER  
)
```

2. Specificare in SQL il vincolo che verifica che nessun pagamento sia inferiore alla cifra dovuta. (2 punti)

```
CREATE ASSERTION VerificaPagamenti CHECK (
  NOT EXISTS (
    SELECT *
    FROM Prenotazione AS Pr, Pagamento AS Pa, Immobile AS I
    WHERE Pr.CodiceImmobile=I.Codice AND Pa.IDPrenotazione=Pr.ID
      AND Pa.TotalePagamento < (I.PrezzoGiorno * Pr.Numerogiorni
      + Pa.CostoServiziAggiuntivi)
  )
)
```

D. TEORIA (4 punti)

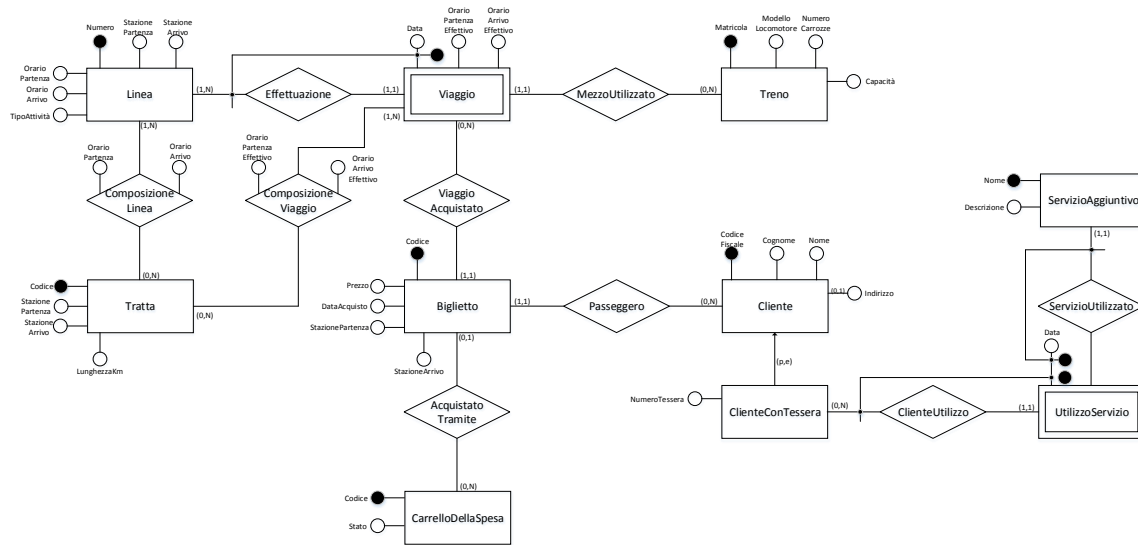
Descrivere le caratteristiche di un sistema di gestione di basi di dati che supporti le transazioni.

E. PROGETTAZIONE DI UNA BASE DI DATI (12 PUNTI)

Progettare la base di dati relativa alla situazione descritta nel seguito. Svolgere il progetto concettuale usando il modello ER, e il progetto logico con il modello relazionale. Specificare le chiavi e gli eventuali vincoli d'integrità.

Si vuole progettare una base di dati a supporto delle attività di una compagnia ferroviaria. Occorre prima di tutto rappresentare le linee. Ogni linea è identificata da un numero, e caratterizzata da una stazione di partenza, una stazione di arrivo, un orario di partenza e un orario di arrivo; alcune linee sono attive tutti i giorni, altre solo nei giorni feriali, e altre ancora solo nei giorni festivi. Ogni linea, inoltre, è suddivisa in tratte che connettono stazioni successive sulla linea. Ogni tratta ha un codice identificativo, una stazione di partenza, una stazione di arrivo e una lunghezza in chilometri. Ogni linea è associata a orari di partenza e arrivo per ogni tratta. Ogni linea effettua un viaggio in tutte le date in cui ciò è previsto. Il viaggio è caratterizzato da un orario di partenza effettivo e un orario di arrivo effettivo; tali orari possono differire da quelli della linea corrispondente in caso di ritardi. Ogni viaggio è associato a orari di partenza e arrivo effettivi anche per ciascuna delle tratte che lo compongono. Infine, il viaggio è effettuato da un treno, identificato da un numero di matricola e caratterizzato dal modello del locomotore, dal numero di carrozze e dalla capacità complessiva. Per un viaggio vengono venduti i relativi biglietti. Un biglietto è identificato da un codice, e caratterizzato dal prezzo, dalla data di acquisto, dalle stazioni di partenza e arrivo, e dal cliente che lo ha comprato. Il biglietto può essere acquistato online, e in questo caso è associato a un carrello della spesa. Il carrello della spesa è identificato da un codice e caratterizzato da uno stato (completato o abbandonato). Se il carrello è stato completato è associato all'acquisto di uno o più biglietti, mentre se è stato abbandonato non è associato all'acquisto di alcun biglietto. I clienti sono identificati dal proprio codice fiscale, e caratterizzati da nome, cognome e indirizzo (se disponibile). Alcuni clienti possiedono una tessera fedeltà – caratterizzata da un numero di tessera –, che consente loro di utilizzare servizi aggiuntivi quali bar e wi-fi gratuiti. Ogni servizio aggiuntivo è identificato dal nome e caratterizzato da una descrizione. La compagnia è interessata a registrare quali dei clienti con tessera hanno usufruito di quali servizi in quali date.

Schema concettuale



Schema logico

Linea (Numero, StazionePartenza, StazioneArrivo, OrarioPartenza, OrarioArrivo, TipoAttività)

Tratta (Codice, StazionePartenza, StazioneArrivo, LunghezzaKm)

ComposizioneLinea (NumeroLinea, CodiceTratta, OrarioPartenza, OrarioArrivo)

Treno (Matricola, ModelloLocomotore, NumeroCarrozze, Capacità)

Viaggio (NumeroLinea, Data, OrarioPartenzaEffettivo, OrarioArrivoEffettivo, MatricolaTreno)

ComposizioneViaggio (NumeroLinea, Data, CodiceTratta, OrarioPartenzaEffettivo, OrarioArrivoEffettivo)

Cliente (CodiceFiscale, Cognome, Nome, Indirizzo*, NumeroTessera*)

CarrelloDellaSpesa (Codice, Stato)

Biglietto (Codice, Prezzo, DataAcquisto, StazionePartenza, StazioneArrivo, CodiceFiscaleCliente, CodiceCarrelloDellaSpesa*)

ServizioAggiuntivo (Nome, Descrizione)

UtilizzoServizio (CodiceFiscaleCliente, NomeServizio, Data)

Chiavi esterne

ComposizioneLinea.NumeroLinea → Linea.Numero

ComposizioneLinea.CodiceTratta → Tratta.Codice

Viaggio.NumeroLinea → Linea.Numero

Viaggio.MatricolaTreno → Treno.Matricola

ComposizioneViaggio.(NumeroLinea, Data) → Viaggio.(NumeroLinea, Data)

Biglietto.CodiceFiscaleCliente → Cliente.CodiceFiscale

Biglietto.CodiceCarrelloDellaSpesa → CarrelloDellaSpesa.Codice

UtilizzoServizio.CodiceFiscaleCliente → Cliente.CodiceFiscale

UtilizzoServizio.NomeServizio → ServizioAggiuntivo.Nome