

Basi di Dati
Prof. L. Tanca e F. A. Schreiber
APPELLO DEL 07 LUGLIO 2015
Tempo: 2h30m

Si consideri il seguente schema di base di dati che vuole tenere traccia di sinistri segnalati alle assicurazioni.

VEICOLO (Targa, Marca, Modello, CFProprietario, CodAssicurazione)
PROPRIETARIO (CodiceFiscale, Cognome, Nome, DataNascita, CittàResidenza)
ASSICURAZIONE (CodiceAssicurazione, Nome, CittàSede)
SINISTRO (Codice, Data, Indirizzo, Località)
PARTISINISTRO (CodiceSinistro, TargaAuto, CircostanzeSinistro, Responsabilità, Danno)

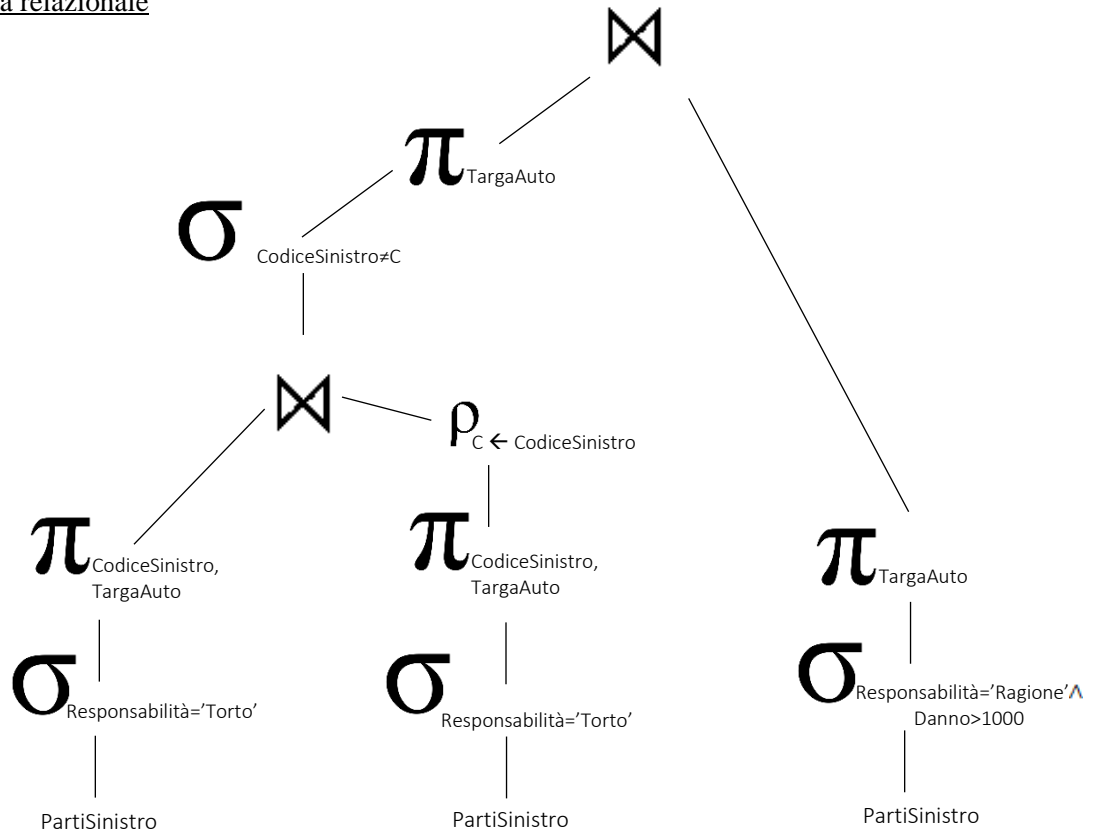
Il campo Responsabilità nella tabella PartiSinistro può assumere i valori “Torto”, “Ragione” o “Concorso”. Il campo Danno riporta invece l’ammontare del danno che verrà liquidato al proprietario dell’auto interamente se la Responsabilità assume il valore “Ragione” o al 50% se la Responsabilità assume il campo “Concorso”.

A. LINGUAGGI FORMALI (6 punti).

Esprimere in algebra relazionale e calcolo relazionale la seguente query:

Trovare i veicoli coinvolti in almeno due sinistri in cui avevano torto e in un sinistro in cui avevano ragione con un danno maggiore di 1000 euro. (4 punti)

Algebra relazionale



Calcolo relazionale

$\{t \mid (\exists t1 \in PartiSinistro, \exists t2 \in PartiSinistro, \exists t3 \in PartiSinistro$
 $(t[TargaAuto]=t1[TargaAuto] \wedge t1[TargaAuto]=t2[TargaAuto] \wedge$
 $t2[TargaAuto]=t3[TargaAuto] \wedge t1[Responsabilità]='Torto' \wedge$
 $t2[Responsabilità]='Torto' \wedge t3[Responsabilità]='Ragione' \wedge t3[Danno]>1000 \wedge$
 $t1[CodiceSinistro] \neq t2[CodiceSinistro])\}$

Esprimere in datalog la seguente query:

Trovare i sinistri che hanno coinvolto solo macchine della stessa marca. (2 punti)

SinistroMarcheDiverse(CodiceSinistro) :- PartiSinistro(CodiceSinistro, TargaAuto, _, _, _),
 Veicolo(TargaAuto, Marca, _, _, _), PartiSinistro(CodiceSinistro, TargaAuto2, _, _, _),
 Veicolo(TargaAuto2, Marca2, _, _, _), Marca<>Marca2
 SinistroStessaMarca(CodiceSinistro) :- Sinistro(CodiceSinistro, _, _, _),
 ¬SinistroMarcheDiverse(CodiceSinistro)
 ?- SinistroStessaMarca(x)

B. SQL: INTERROGAZIONI (7 punti).

Scrivere in SQL le seguenti interrogazioni:

1. *Trovare i sinistri che hanno coinvolto esattamente tre vetture ma nessuna di marca FIAT. (3.5 punti)*

```
SELECT CodiceSinistro
FROM PartiSinistro
WHERE CodiceSinistro NOT IN (
    SELECT P.CodiceSinistro
    FROM PartiSinistro AS P, Veicolo AS V
    WHERE P.TargaAuto=V.Targa AND V.Marca='Fiat'
)
GROUP BY CodiceSinistro
HAVING COUNT(*)=3
```

2. *Trovare i proprietari di vetture che hanno complessivamente ricevuto più denaro per risarcimenti in concorso di colpa. (3.5 punti)*

```
CREATE VIEW ProprietarioDenaro(Proprietario, Denaro) AS (
    SELECT V.CFProprietario, 0.5*SUM(P.Danno)
    FROM PartiSinistro AS P, Veicolo AS V
    WHERE P.TargaAuto=V.Targa AND P.Responsabilità='Concorso'
    GROUP BY V.CFProprietario
)
```

```
SELECT CodiceFiscale
FROM ProprietarioDenaro
WHERE Denaro = (
    SELECT MAX(Denaro)
    FROM ProprietarioDenaro
)
```

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

1. Specificare in SQL la creazione delle tabelle SINISTRO e PARTISINISTRO, 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 Sinistro (  
    Codice VARCHAR(255) PRIMARY KEY,  
    Data DATE,  
    Indirizzo VARCHAR(255),  
    Località VARCHAR(255)  
)
```

```
CREATE TABLE PartiSinistro (  
    CodiceSinistro VARCHAR(255) REFERENCES Sinistro(Codice) ON UPDATE  
        CASCADE ON DELETE CASCADE,  
    TargaAuto CHAR(7) REFERENCES Veicolo(Targa) ON UPDATE CASCADE  
        ON DELETE NO ACTION,  
    CircostanzeSinistro VARCHAR(500),  
    Responsabilità ENUM('Torto', 'Ragione', 'Concorso'),  
    Danno NUMERIC(10, 2),  
    PRIMARY KEY (CodiceSinistro, TargaAuto)  
)
```

2. Specificare in SQL il vincolo che controlla che per i sinistri in concorso di colpa, sia stata attribuita la responsabilità "Concorso" a tutte le vetture coinvolte nel sinistro (2 punti)

```
CREATE ASSERTION VerificaConcorsoColpa CHECK (  
    NOT EXISTS (  
        SELECT *  
        FROM PartiSinistro  
        WHERE Responsabilità='Concorso' AND CodiceSinistro IN (  
            SELECT CodiceSinistro  
            FROM PartiSinistro  
            WHERE Responsabilità<>'Concorso'  
        )  
    )  
)
```

D. TEORIA (4 punti)

Il ruolo delle dipendenze funzionali e della normalizzazione nella progettazione di basi di dati

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

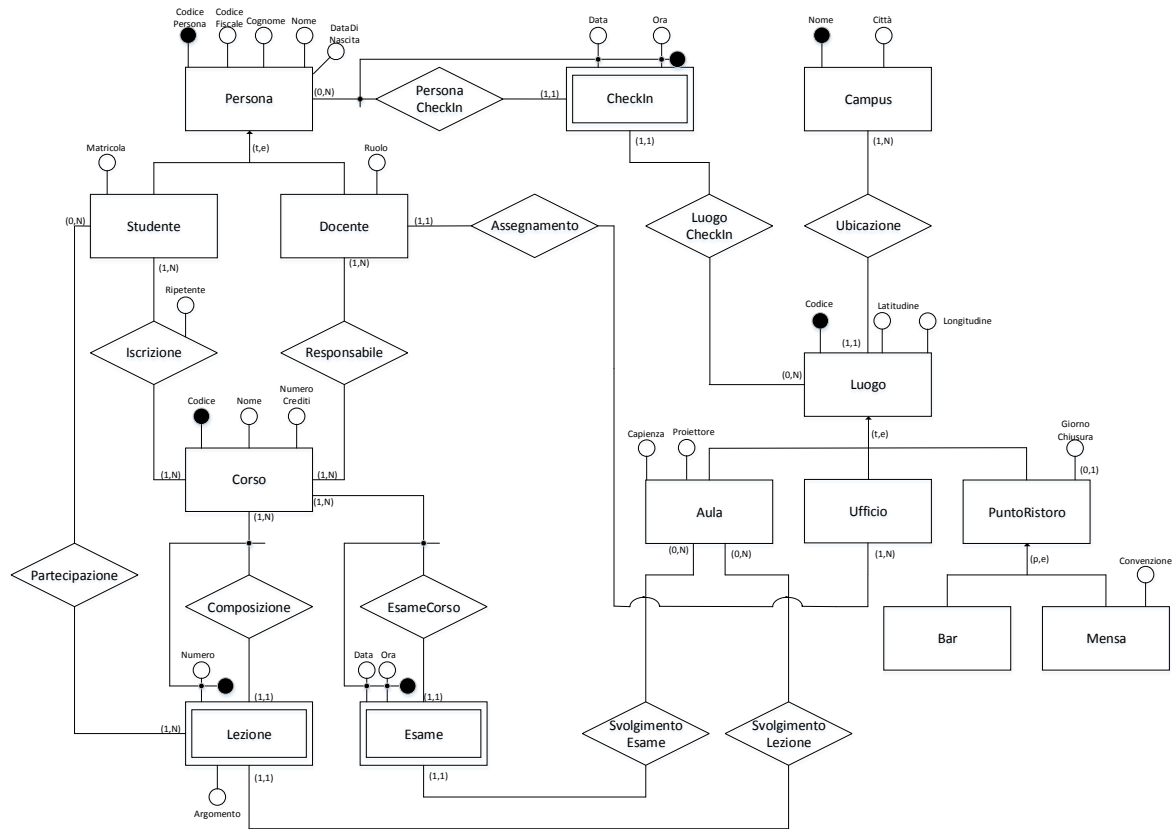
Progettare la base di dati relativa alla situazione descritta nel seguito. Svolgere il progetto concettuale usando il modello ER, e il progetto logico con il modello relazionale. Specificare le chiavi e gli eventuali vincoli d'integrità.

Si vuole progettare la base di dati di supporto a un sistema che affianca studenti e docenti nella vita all'interno dei campus di un'università.

La base di dati deve prima di tutto memorizzare le informazioni sui docenti, sugli studenti e sulle attività didattiche. Ogni persona è identificata da un codice persona, ed è caratterizzata da codice fiscale, nome, cognome e data di nascita; gli studenti sono dotati anche di un numero di matricola univoco, e i docenti hanno un ruolo (ad esempio professore associato) e un ufficio eventualmente in condivisione con altri docenti. Ogni corso è identificato da un codice univoco, e caratterizzato dal nome, dal numero di crediti, dal docente responsabile e dall'elenco degli studenti iscritti; ogni iscritto può essere o meno ripetente per quel corso. Ogni corso è costituito da un insieme di lezioni, ognuna con un numero unico per lo specifico corso; la lezione è caratterizzata dall'argomento e dall'aula in cui si svolge. È necessario registrare la partecipazione degli studenti alle singole lezioni. Si memorizzano infine gli esami dell'anno accademico corrente, ognuno associato a un corso, a una data, a un'ora e all'aula in cui si svolge.

La base di dati memorizza anche informazioni sulla localizzazione dei luoghi all'interno dell'università. Per ogni luogo si conosce il codice identificativo, la latitudine, la longitudine e il campus in cui è ubicato; ogni campus è identificato dal nome e caratterizzato dalla città in cui si trova. Aule e uffici sono luoghi; le aule sono caratterizzate anche dalla capienza e da un valore booleano che indica la presenza di un proiettore. Tra i luoghi vi sono anche i punti di ristoro, caratterizzati da un giorno di chiusura (se esistente). Tra i punti di ristoro vi sono sicuramente bar e mense, ma non si esclude che vi possano essere anche punti di ristoro di altro tipo (ad esempio veri e propri ristoranti). Le mense sono caratterizzate da un valore booleano che indica se esistono convenzioni per gli studenti. Infine, il sistema è in grado di rilevare il check-in delle persone nei luoghi; ogni check-in si verifica in una specifica data e in una specifico orario.

Schema concettuale



Schema logico

Studente(CodicePersona, CodiceFiscale, Cognome, Nome, DataDiNascita, Matricola)

Docente(CodicePersona, CodiceFiscale, Cognome, Nome, Ruolo, Ufficio)

Corso(Codice, Nome, NumeroCrediti, DocenteResponsabile)

Iscrizione(Studente, Corso, Ripetente)

Lezione(Numero, Corso, Argomento, Aula)

Partecipazione(Studente, NumeroLezione, Corso)

Esame(Corso, Data, Ora, Aula)

Luogo(Codice, Latitudine, Longitudine, Campus, Tipo, Capienza*, Proiettore*,
GiornoChiusura*, TipoPuntoRistoro*, Convenzione*)

Campus(Nome, Città)

CheckInStudente(Studente, Data, Ora, Luogo)

CheckInDocente(Docente, Data, Ora, Luogo)

Chiavi esterne

Docente.Ufficio → Luogo.Codice

Corso.DocenteResponsabile → Docente.CodicePersona

Iscrizione.Studente → Studente.CodicePersona

Iscrizione.Corso → Corso.Codice

Lezione.Corso → Corso.Codice
Lezione.Aula → Luogo.Codice
Partecipazione.Studente → Studente.CodicePersona
Partecipazione.(NumeroLezione, Corso) → Lezione.(Numero, Corso)
Esame.Corso → Corso.Codice
Esame.Aula → Luogo.Codice
Luogo.Campus → Campus.Nome
CheckInStudente.Studente → Studente.CodicePersona
CheckInStudente.Luogo → Luogo.Codice
CheckInDocente.Docente → Docente.CodicePersona
CheckInDocente.Luogo → Luogo.Codice