

**BASI DI DATI 1 – PROFF. S. CERI, G. POZZI, E. QUINTARELLI, L. TANCA –
A.A. 2017/2018**

SECONDO APPELLO – 6 FEBBRAIO 2018

TEMPO A DISPOSIZIONE: 2 h e 30 m

Matricola o Codice Persona:	
Cognome e Nome: (IN STAMPATELLO MAIUSCOLO)	
Firma:	

Si consideri il seguente schema di base di dati nell'ambito del bike sharing:

ISCRITTO (IdIscritto, Nome, Cognome, DataNascita, Email)

VIAGGIO (IdBici, TimestampInizio, TimestampFine, PosizioneInizio, PosizioneFine, IdIscritto, KmPercorsi)

DANNO (IdBici, TimestampInizio, TimestampFine, Descrizione, CostoRiparazione)

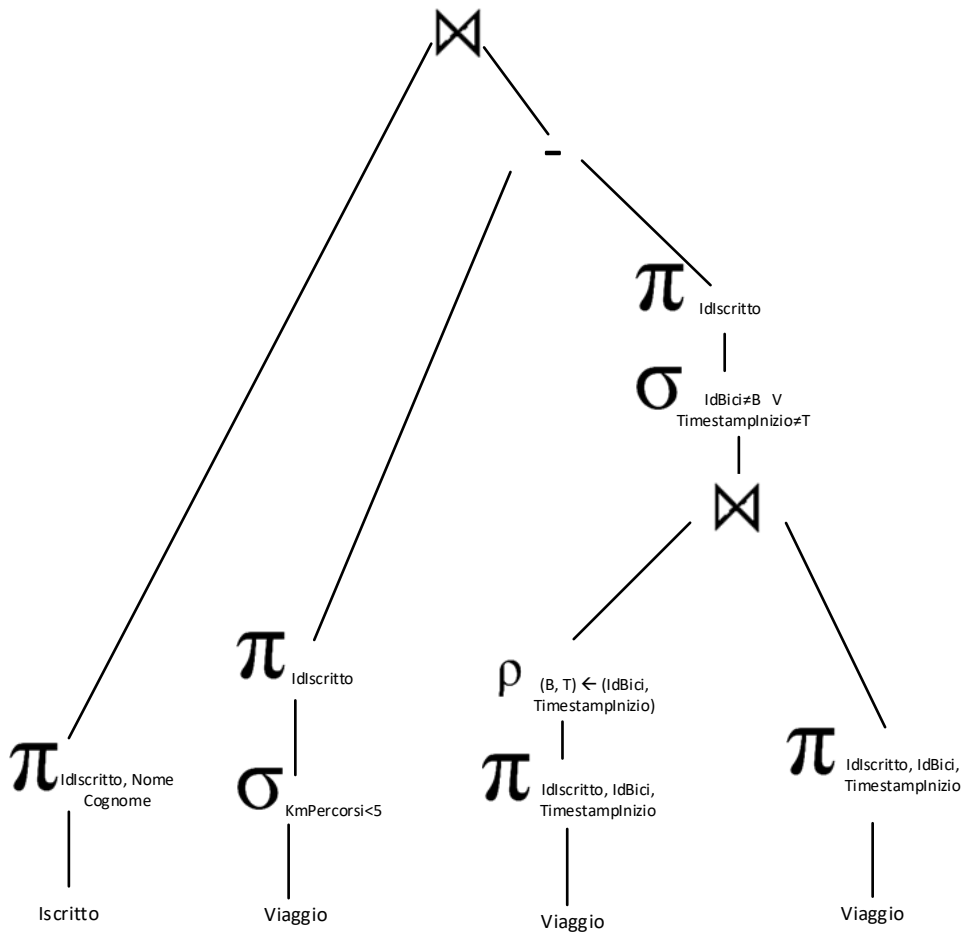
Si noti che VIAGGIO contiene tutti i viaggi fatti con una certa bici, un viaggio inizia nel momento specificato da TimestampInizio e finisce quando specificato da TimestampFine. La relazione DANNO, invece, indica tutti i periodi in cui una certa bici è stata "inutilizzabile" perché danneggiata (indipendentemente dai viaggi che sono stati fatti con tale bici); un periodo di danno inizia nell'istante specificato da TimestampInizio e finisce nell'istante specificato da TimestampFine.

A. Linguaggi formali (6 punti)

Esprimere in algebra relazionale, calcolo relazionale e Datalog la seguente interrogazione:

Trovare id, nome e cognome degli iscritti che hanno fatto un unico viaggio e in tale viaggio hanno percorso meno di 5 chilometri.

Algebra relazionale



Calcolo relazionale

$$\{ t \mid \exists t_{v1} \in \text{VIAGGIO}, \exists t_i \in \text{ISCRITTO} (t[\text{IdIscritto}, \text{Nome}, \text{Cognome}] = t_i[\text{IdIscritto}, \text{Nome}, \text{Cognome}] \wedge t_{v1}[\text{IdIscritto}] = t_i[\text{IdIscritto}] \wedge t_{v1}[\text{KmPercorsi}] < 5 \wedge \neg (\exists t_{v2} \in \text{VIAGGIO} (t_{v2}[\text{IdIscritto}] = t_{v1}[\text{IdIscritto}] \wedge t_{v1}[\text{IdBici}, \text{TimestampInizio}] < t_{v2}[\text{IdBici}, \text{TimestampInizio}]))) \}$$

Datalog

ALMENODUE(IDISCRITTO):- VIAGGIO(BICI1,_,_,_, IDISCRITTO,_), VIAGGIO(BICI2,_,_,_,IDISCRITTO,_),
BICI1<>BICI2.

ALMENODUE(IDISCRITTO):- VIAGGIO(,_TS1,_,_, IDISCRITTO,_), VIAGGIO(,_TS2,_,_,IDISCRITTO,_),
TS1<>TS2.

SOLOUNO(ID, N, C):- ISCRITTO(ID,N,C,_,_), VIAGGIO(____,ID,KM), \neg ALMENODUE(ID), KM<5.
?- SOLOUNO(x,y,z).

B. Interrogazioni SQL (10 punti)

1. Trovare gli iscritti che non hanno mai effettuato viaggi più lunghi di 10 chilometri. (3 punti)

```
SELECT Iscritto.*  
FROM Iscritto I JOIN Viaggio V ON I.IdIscritto=V.IdIscritto  
WHERE IdIscritto NOT IN ( SELECT IdIscritto  
                        FROM Viaggio  
                        WHERE KmPercorsi<=10)
```

2. Trovare le bici che, tra quelle che in totale hanno percorso almeno 1000 chilometri, sono state danneggiate il maggior numero di volte. (3.5 punti)

```
CREATE VIEW Almeno1000 (Bici) AS (  
    SELECT IdBici  
    FROM Viaggio  
    GROUP BY IdBici  
    HAVING SUM(KmPercorsi)>=1000  
)  
  
SELECT IdBici  
FROM Danno  
WHERE IdBici IN ( SELECT *  
                FROM Almeno1000)  
GROUP BY IdBici  
HAVING COUNT(*) >= ALL ( SELECT COUNT(*)  
                        FROM Danno  
                        WHERE IdBici IN ( SELECT *  
                                    FROM Almeno1000)  
                        GROUP BY IdBici )
```

3. Trovare per ciascun utente la percorrenza chilometrica media dei viaggi di durata inferiore a 1 ora. (3.5 punti)

```
SELECT IdIscritto, AVG(KmPercorsi)  
FROM Viaggio  
WHERE timestampdiff(hour, TimestampInizio, TimeStampFine)<1  
GROUP BY IdIscritto
```

Nota: ogni DBMS fornisce una sintassi propria per calcolare la differenza tra timestamp. Nella soluzione è stata utilizzata la sintassi del DBMS MySQL.

C. DDL: Definizione di schemi e vincoli (3 punti)

1. Specificare in SQL la creazione della tabella VIAGGIO, 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 Viaggio (  
    IdBici CHAR(25),  
    TimestampInizio TIMESTAMP,  
    TimestampFine TIMESTAMP,  
    PosizioneInizio VARCHAR(100),  
    PosizioneFine VARCHAR(100),  
    IdIscritto INT REFERENCES Iscritto(IdIscritto) ON UPDATE CASCADE ON DELETE SET NULL,  
    KmPercorso REAL,  
    PRIMARY KEY (IdBici, TimestampInizio)  
);
```

2. Specificare in SQL il vincolo che verifica che nessuna bici risulti utilizzata per un viaggio mentre è danneggiata. (2 punti)

```
CREATE ASSERTION VerificaBici CHECK (  
    NOT EXISTS ( SELECT *  
        FROM Viaggio V JOIN Danno D ON V.IdBici=D.IdBici  
        WHERE (V.TimestampInizio BETWEEN D.TimestampInizio AND D.TimestampFine) OR  
            (V.TimestampFine BETWEEN D.TimestampInizio AND D.TimestampFine) OR  
            (V.TimestampInizio <= D.TimestampInizio AND V.TimestampFine >= D.TimestampFine)  
    ) );
```

D. Progettazione di una base di dati (13 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 primarie e i vincoli di integrità referenziale.

L'agenzia YouthTravel organizza vacanze studio per ragazzi. Durante l'iscrizione di ogni ragazzo vengono memorizzati i dati anagrafici, indirizzo, recapito telefonico, le eventuali allergie (nome allergene e descrizione delle precauzioni necessarie), gli hobby. Ogni iscritto prenota la vacanza, caratterizzata da un identificativo, città di permanenza, data di partenza, numero di settimane ed esprime una preferenza per il tipo di alloggio (college o famiglia). In caso di vacanza in college occorre specificare se si desidera avere una camera singola o condivisa, per la vacanza in famiglia è possibile inoltre specificare il nome di un amico che andrebbe nella stessa famiglia. Ogni vacanza prevede anche una o più gite, per le quali si conoscono la destinazione, il costo, il numero di ore e una descrizione. Dei college si conoscono il nome, l'indirizzo e le attività proposte (nome e descrizione). Per ogni famiglia ospitante si memorizzano i dati del capo famiglia, il numero di componenti il nucleo familiare, presenza di animali domestici, il numero di camere disponibili, il numero di bagni e la distanza dal centro città. Prima della partenza, ogni iscritto riceve conferma del nome del college o della famiglia in cui soggiornerà e del nome dell'eventuale compagno con cui condividerà la stanza nel college o in famiglia. La vacanza può essere pagata con carta di credito o bancomat (e si deve

tenere traccia della scelta). Al termine del viaggio ogni ragazzo riceve un certificato che specifica il livello raggiunto (ad esempio B2, C1) nella lingua straniera studiata durante la vacanza.

E. Teoria (2 punti)

Descrivere brevemente le proprietà ACID garantite da un DBMS.

NOTE

- **Non è ammessa la consultazione di libri e appunti.**
- **Alla domanda di teoria occorre rispondere con parole proprie, in modo che i docenti possano valutare il livello di comprensione dell'argomento.**