

Esame di Basi di Dati

A.A. 2020/2021 – Appello del 09/06/2021

Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione relativo alle rotatorie stradali localizzate nel territorio nazionale. Di ogni rotatoria interessa le coordinate (identificative), la provincia in cui si trova e gli sbocchi che la rotatoria ha sulle strade che in essa convergono, dove ogni sbocco è caratterizzato da un numero che lo identifica nell'ambito della rotatoria. Ovviamente ogni rotatoria ha almeno due sbocchi in assoluto e non può avere più di due sbocchi sulla stessa strada. È anche di interesse sapere, per ogni rotatoria, quale sbocco (esattamente uno) è quello principale, perchè su esso viene installato un rilevatore di traffico. Di ogni sbocco principale è di interesse conoscere la marca del rilevatore di traffico installato. Di ogni strada interessa il codice (identificativo) ed il tipo. Interessano anche i veicoli che attraversano la rotatoria, ogni volta con lo sbocco dal quale il veicolo entra nella rotatoria, lo sbocco dal quale il veicolo esce dalla rotatoria e la data, l'ora e il minuto in cui il veicolo entra nella rotatoria. Si noti che un veicolo può entrare al più in una rotatoria al minuto. Ogni veicolo è caratterizzato dalla targa (identificativa), la marca ed il modello.

Problema 2

Si richiede di effettuare la progettazione logica partendo dallo schema concettuale prodotto per il problema 1, tenendo conto dell'indicazione che ogni volta che si accede ad una rotatoria interessa conoscere qual è lo sbocco principale, con la marca del rilevatore di traffico installato in tale sbocco.

Problema 3

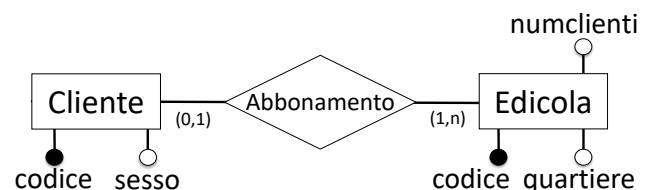
Si consideri una base di dati che comprende la relazione **Lezione**(codice, professore, corso, voto) e la relazione **Corso**(codice, facoltà), dove la prima memorizza il codice (chiave primaria) di ogni lezione erogata nell'anno accademico 2020/21, il professore che ha tenuto la lezione, il corso relativo alla lezione ed il voto che deriva dalle valutazioni date dagli studenti alla lezione stessa, e la seconda memorizza per ogni corso il codice (chiave primaria) e la facoltà al quale il corso appartiene. Sappiamo che le relazioni non contengono valori nulli e che soddisfano il vincolo di foreign key da **corso** di **Lezione** a **codice** di **Corso**. Si scrivano in SQL le seguenti query: (1) Per *ogni* corso, mostrare il codice del corso ed il numero di lezioni di quel corso che sono state erogate. (2) Calcolare *tutte* le coppie $\langle p, f \rangle$ tali che il professore p non ha erogato alcuna lezione di corsi relativi alla facoltà f . (3) Per ogni corso per il quale sono state erogate almeno 6 lezioni, calcolare la media dei voti delle lezioni di quel corso.

Problema 4

Riferendosi alla base di dati del problema 3, si scrivano in algebra relazionale le seguenti query: (1) Calcolare il valore massimo che si trova nel campo "voto" della relazione **Lezione**. (2) Calcolare le facoltà che hanno avuto almeno un corso del quale non è stata erogata alcuna lezione.

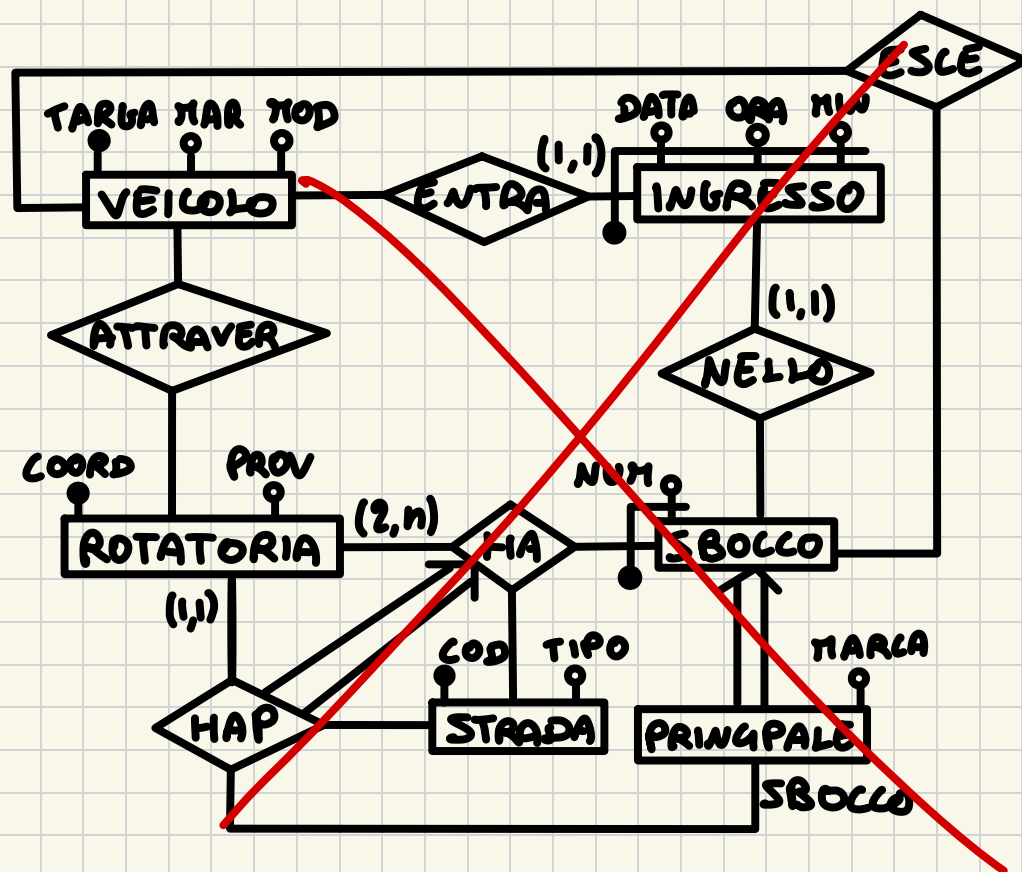
Problema 5

Nello schema concettuale S mostrato qui a destra, tutti gli attributi tranne **numclienti** sono di tipo "varchar" e l'attributo **numclienti** (di tipo intero) indica, per ogni istanza di **Edicola**, il numero di clienti abbonati a quell'edicola (come specificato dal vincolo esterno). Si sa a priori che non si eseguiranno inserimenti e cancellazione di "abbonamenti" nel prossimo anno, che quando si accede ad un cliente si vuole sempre sapere l'eventuale edicola alla quale è abbonato e quando si accede ad una edicola si vuole sempre sapere il numero di clienti abbonati a quell'edicola. Si chiede di eseguire la progettazione logica partendo dallo schema concettuale S esprimendo lo schema relazionale risultante in SQL.



Vincolo esterno: per ogni istanza l di S , per ogni e in $Istanze(l, Edicola)$, il valore n tale che $\langle e, n \rangle$ è in $Istanze(l, numclienti)$ è pari al numero di istanze di **Cliente** collegate ad e dalla relazione **Abbonamento**.

PROBLEMA 1



PROBLEMA 3

- 1)

```
SELECT C.CODICE, COUNT(*) AS NUM
FROM CORSO C JOIN LEZIONE L ON L.CORSO = C.CODICE
GROUP BY C.CODICE

UNION

SELECT C.CODICE, 0 AS NUM
FROM CORSO C
WHERE C.CODICE NOT IN (SELECT CORSO
                        FROM LEZIONE)
```
- 2)

```
SELECT L.PROFESSORE, C.FACOLTÀ
FROM LEZIONE L, CORSO C
WHERE (L.PROFESSORE, C.FACOLTÀ) NOT IN
      (SELECT L.PROFESSORE, C.FACOLTÀ
       FROM LEZIONE L JOIN CORSO C
        ON L.CORSO = C.CODICE)
```
- 3)

```
SELECT L.CORSO, AVG(L.VOTO)
FROM LEZIONE L
GROUP BY L.CORSO
HAVING COUNT(*) ≥ 6
```

PROBLEMA 4

1) $PROJ_{VOTO}(LEZIONE) - PROJ_{VOTO}(LEZIONE JOIN$
 $VOTO < V$
 REN
 $C \leftarrow CODICE(LEZIONE)$
 $P \leftarrow PROF$
 $LS \leftarrow CORSO$
 $V \leftarrow VOTO$

2)