

# ITINERE\_11\_2022

## 👍 Da completare, mancanti:

- 2 query tema A
- tema B

## Tema\_A

TESSERE\_FEDELTA(numero, data\_emissione, cognome, nome)

PRODOTTI(codice, categoria, descrizione, prezzo\_unitario)

SCONTRINI(codice, data, cassa, progr, tessera<sub>fk</sub>\*)

RIGHE\_SCONTRINI(scontrino<sub>fk</sub>, progr\_riga, prodotto<sub>fk</sub>, quantita)

PAGAMENTI\_ELETTRONICI(scontrino<sub>fk</sub>, tipo\_carta, numero\_carta)

- Scrivere l'istruzione DDL per la definizione della relazione **RIGHE\_SCONTRINI**; la quantità acquistata deve essere un numero compreso tra 0.01 e 10.00 (precisione centesimo di unità).

```
CREATE TABLE righe_scontrini (  
    scontrino INTEGER NOT NULL,  
    prog_riga INTEGER NOT NULL,  
    prodotto INTEGER NOT NULL,  
    quantita NUMERIC(2,2) NOT NULL,  
    PRIMARY KEY (scontrino , pro_riga)  
    FOREIGN KEY (scontrino) REFERENCES scontrini (codice) ,  
    FOREIGN KEY (prodotto) REFERENCES prodotti (codice),  
    CHECK (quantita BETWEEN 0.01 AND 10.00)  
);
```

- Modificare la relazione **PRODOTTI** per aumentare del 10% i prezzi dei prodotti di categoria "alta gastronomia".

```
UPDATE prodotti  
    SET prezzo_unitario = prezzo_unitario * 1.1  
    WHERE categoria = 'alta gastronomia' ;
```

- Estrarre l'elenco delle tessere fedeltà per le quali non è stato registrato nessuno scontrino negli ultimi 30 giorni, ordinandole per cognome e nome.

```
SELECT ts.codice , ts.cognome , ts.cognome  
FROM tessere_fedelta ts  
WHERE ts.codice NOT IN ( SELECT s.tessera  
                        FROM scontrini s  
                        WHERE current_date = current_data < 30 )  
ORDER BY ts.cognome , ts.nome
```

## Tema\_B

## Domande a risposta aperta

- Date le relazioni  $R(\underline{A}, B^*, C)$  e  $S(\underline{D}, E, F^*)$ , dove  $\#R = n$  e  $\#S = m$ , quante ennuple compongono il risultato della query

```
SELECT * FROM R LEFT OUTER JOIN S ON A = D?
```

La cardinalità corrisponde al numero di  $n$ -uple di  $R$ :  $n$ .

- Fornire una istanza della tabella  $R(A, B)$  per la quale la query

```
SELECT COUNT(A), COUNT(B) FROM R
```

calcola due valori diversi.

$A$	$B$
mario	rossi
NULL	neri

NOTA: nella consegna i due attributi non sono stati indicati annullabili.

Tuttavia la risposta rimane valida.

- Data la relazione  $R(A, B, C)$ , la query

```
SELECT COUNT(*), B*C AS PROD FROM R ORDER BY B*C
```

è errata; per quale motivo? Come deve essere corretta?

Vedere file [SELECT](#)

E' errata perché la query utilizza una funzione aggregata ( `COUNT` ) senza l'operatore `GROUP BY` per raccogliere i risultati.

```
SELECT COUNT(*), B*C AS PROD
FROM R
GROUP BY B*C
ORDER BY B*C
```

- Cosa differenzia la proiezione dell'algebra relazionale rispetto a quella implementata in SQL?

La proiezione `PROJ` dell'algebra relazionale ha la caratteristica di eliminare di default, le  $n$ -uple duplicate. La sua implementazione in SQL invece richiede l'aggiunta della clausola `DISTINCT`.