

# Basi di Dati

*Appello del 7 Giugno 2022 --soluzioni*

## Parte B

### Esercizio 1 (punti 8)

1. CREATE TABLE ACQUISTO(  
    Cliente NUMERIC(5) REFERENCES CLIENTE ON UPDATE CASCADE,  
    Cellulare NUMERIC(5) REFERENCES CELLULARE ON UPDATE CASCADE,  
    DataAcq DATE,  
    ModPagamento VARCHAR(20),  
    PRIMARY KEY(Cliente, Cellulare, DataAcq));  
se viene modificato il codice di un cliente nella tabella CLIENTE (o il codice di un cellulare nella tabella Cellulare) le modifiche vengono propagate alla tabella ACQUISTI. Non viene permessa la cancellazione di un cliente/cellulare se esistono acquisti ad esso relativi.
2. DELETE FROM ACQUISTO WHERE DataAcq < '1/1/2020' DATE AND  
    Cellulare IN (SELECT Codice FROM CELLULARE  
                  WHERE Produttore = 'Huawei');
3. UPDATE CELLULARE SET Prezzo = 0.9\*Prezzo  
    WHERE Serie = 'S1';

### Esercizio 2 (punti 12)

1. SELECT CF, Cognome FROM Cliente JOIN ACQUISTO ON CF = Cliente  
    WHERE Cellulare IN (SELECT Codice FROM CELLULARE  
                        WHERE Produttore = 'Samsung')  
    INTERSECT  
    SELECT CF, Cognome FROM Cliente JOIN ACQUISTO ON CF = Cliente  
    WHERE Cellulare IN (SELECT Codice FROM CELLULARE  
                        WHERE Produttore = 'Xiaomi');
2. SELECT Cellulare, Modello FROM CELLULARE JOIN ACQUISTO  
    ON Codice = Cellulare  
    WHERE Cliente IN (SELECT CF FROM CLIENTE  
                      WHERE EXTRACT(YEAR FROM DataNascita) > 2000)  
    GROUP BY Cellulare, Modello  
    HAVING COUNT(DISTINCT(Cliente)) >= 6;
3. SELECT Codice, Modello FROM Cellulare WHERE Colore = 'Argento'  
    AND Prezzo >= ALL(SELECT Prezzo FROM Cellulare WHERE Colore = 'Argento');
4. SELECT CF, DataNascita FROM CLIENTE WHERE CF NOT IN  
    (SELECT Cliente FROM ACQUISTO WHERE ModPagamento = 'Paypal');

### Esercizio 3 (punti 10)

1. L'attributo multivalore a9 viene ristrutturato sostituendolo con una entità chiamata a9, con un unico attributo a9.1 identificatore. Inoltre, viene creata una associazione A4 che lega E3 ed a9. La cardinalità di E3 ed a9 rispetto alla nuova associazione è (1,n). In base al carico di lavoro:
  - a. La gerarchia viene ristrutturata eliminando le entità figlie. Gli attributi a4, a4, ed a6 diventano attributi opzionali (ovvero cardinalità (0,1)) dell'entità E1. All'entità E1 viene aggiunto un attributo tipo con cardinalità (1,1).

- b. L'attributo a11 viene ristrutturato eliminando i sotto-attributi a12 ed a13.
  - c. Vincoli derivanti dalla ristrutturazione: tipo assume uno di due valori E11 e E12. Se tipo = E11, a4 ed a5 sono non nulli. Se tipo = E12 a4 ed a5 sono nulli.
2. E1(a2,a3,a4<sub>0</sub>,a5<sub>0</sub>,a6<sub>0</sub>,tipo,a1,a10<sup>E2</sup>) E2(a10,a11) E3(a7,a8) a9(a9.1) E4(a10,a11,a2,a7<sup>E3</sup>) A2(a10<sup>E2</sup>, a7<sup>E3</sup>) A4(a7<sup>E3</sup>,a9.1<sup>a9</sup>)
  3. ALTER TABLE E1 ADD CONSTRAINT TipoOK CHECK Tipo in ('E11','E12');  
ALTER TABLE E1 ADD CONSTRAINT Gerarchia CHECK((Tipo = 'E11' and a4 IS NOT NULL AND a5 IS NOT NULL) OR (Tipo = 'E12' AND a4 IS NULL AND a5 IS NULL));