

# **Basi di Dati**

## *12 Gennaio 2023*

### **PARTE B: SOLUZIONI**

#### **Esercizio 1 (punti 8)**

1. CREATE TABLE PRENOTAZIONI (Cliente CHAR(16) REFERENCES CLIENTI(CF)  
ON UPDATE CASCADE,  
Bici NUMERIC(6) REFERENCES BICICLETTE(Id)  
ON UPDATE CASCADE,  
Data DATE NOT NULL,  
PRIMARY KEY(Cliente, Bici));  
  
Integrità referenziale: non è consentita la cancellazione di un cliente/bici se ci sono prenotazioni associate. Se viene modificato il CF di un cliente in CLIENTE (o l'id di una bicicletta in BICICLETTE) la modifica viene propagata alla tabella PRENOTAZIONI.
2. UPDATE BICICLETTE SET Costo\_orario = 0,9\* Costo\_orario  
WHERE Id NOT IN (SELECT Bici FROM PRENOTAZIONI  
WHERE Data >= '2/1/2022' DATE);
3. ALTER TABLE BICICLETTE ADD CONSTRAINT CostoOk  
CHECK(Costo\_orario BETWEEN 2 AND 5);
4. ALTER TABLE CLIENTI ADD COLUMN email VARCHAR(20);

#### **Esercizio 2 (punti 12)**

1. SELECT CF, Cognome FROM CLIENTI, PRENOTAZIONI, BICICLETTE  
WHERE Cliente = CF AND Bici = Id AND Colore = 'rosso'  
INTERSECT  
SELECT CF, Cognome FROM CLIENTI, PRENOTAZIONI, BICICLETTE  
WHERE Cliente = CF AND Bici = Id AND Colore = 'verde';
2. SELECT Modello FROM BICICLETTE JOIN PRENOTAZIONI ON (Id = Bici)  
GROUP BY Modello
3. HAVING COUNT(\*) >= ALL (SELECT COUNT(\*) FROM BICICLETTE JOIN  
PRENOTAZIONI ON (Id = Bici)  
GROUP BY Modello);
4. SELECT Bici FROM PRENOTAZIONI  
MINUS  
SELECT Bici FROM PRENOTAZIONI JOIN CLIENTI ON (CF = Cliente)  
WHERE Indirizzo NOT LIKE '%Milano%';
5. SELECT Cliente, AVG(Costo\_orario)
6. FROM BICICLETTE JOIN PRENOTAZIONI ON (Id = Bici)  
GROUP BY Cliente  
HAVING COUNT(\*) >= 3;

#### **Esercizio 3 (punti 10)** **Si veda file allegato)**