

NOME
COGNOME
MATRICOLA

ESERCIZIO 1 (12 punti)

Dato il seguente schema relazionale, che modella i dati di un'applicazione che tiene traccia delle crociere nel Mediterraneo.

CROCIERA(Id, CostoTotale, NomeCompagnia)
 TRAGITTO(IdCrociera, NomeLocalità, Ordine)
 LOCALITA(Nome, Stato, NumAbitanti)
 RECENSIONE(Codice, IdCrociera, Voto)

Con i seguenti vincoli di integrità referenziale:

TRAGITTO.IdCrociera → CROCIERA.Id
 TRAGITTO.NomeLocalità → LOCALITA.Nome
 RECENSIONE.IdCrociera → CROCIERA.Id

- Ordine è un numero progressivo; indica la sequenza delle località che formano il tragitto: se pari a 1, indica la prima località visitata (partenza); se pari a 2, indica la seconda località, ... l'ultimo valore della sequenza indica la località destinazione.
Es. Se nella tabella TRAGITTO sono presenti queste tre righe: <1, Bari, 1>, <1, Tirana, 2>, <1, Atene, 3>, la crociera con ID 1 include: Bari (partenza), Tirana, ed Atene (destinazione)

a) (1 pt) Indicare il numero di righe e numero di colonne (valori esatti oppure range [min:max]) prodotti dalla seguenti query:

- 1)

```
SELECT *
FROM CROCIERA, TRAGITTO
WHERE ((IdCrociera = Id) AND (CostoTotale > 2000))
```
- 2)

```
SELECT Id, COUNT(*)
FROM CROCIERA, TRAGITTO
WHERE (IdCrociera = Id)
GROUPBY Id
```
- 3)

```
SELECT *
FROM CROCIERA, TRAGITTO
WHERE (NomeCompagnia = "CCS")
```

Sapendo che CROCIERA ha 10 righe, e TRAGITTO 30 righe.

b) (2 pt) Scrivere la query che determina il nome della località di destinazione della Crociera con id pari a 10.

c) (3 pt) Scrivere in **SQL** la query che restituisce Nome e Stato delle località che sono incluse in almeno 2 tragitti di crociera operate da compagnie diverse.

d) (3 pt) Scrivere in **SQL** la query che restituisce il numero delle crociere che NON includono nel tragitto delle località francesi [**VINCOLO**: Usare la clausola EXISTS].

e) (3 pt) Scrivere in **SQL** la query che restituisce Id e NomeCompagnia della crociera/e che ha/hanno il costo più basso tra quelle che partono da Venezia.

ESERCIZIO 2 (12 punti)

Si vuole progettare una base di dati a supporto di un sistema intelligente di raccolta e gestione dei rifiuti urbani della città di Bologna. La città è suddivisa in quartieri, ognuno con un nome (“Navile”), un’estensione ed un campo **#numcontenitori** (*ridondanza concettuale*, vedi sotto). Il sistema di raccolta dei rifiuti è formato da un insieme di contenitori (“cassonetti”): ogni contenitore dispone di un codice univoco, una latitudine, una longitudine, una capacità massima, ed è dislocato in un solo quartiere. Si considerano due sotto-tipologie di contenitori: contenitori per l’indifferenziato e contenitori per l’umido. I primi dispongono dei seguenti campi: larghezza, lunghezza ed altezza. I secondi hanno una data e nome del modello. Esistono contenitori che non ricadono in alcuna delle tipologie sovra-elencate. Inoltre, alcuni contenitori dispongono di un sensore che misura la temperatura esterna ed il livello di riempimento del contenitore stesso: ogni sensore ha un id univoco ed una descrizione. Si vuole tenere traccia dei dati (temperatura ed %riempimento) prodotti dal sensore nel tempo. Inoltre, si vogliono gestire i dati degli utenti residenti. Ogni utente dispone di codice fiscale, nome, cognome, anno di nascita, e risiede in uno specifico quartiere di Bologna. Alcuni utenti (ma non tutti, in quanto si assume che il servizio sia ancora in fase di sperimentazione), dispongono di una tessera, con id e data di rilascio, per la raccolta dell’indifferenziato. I contenitori dell’indifferenziato sono utilizzabili solo mediante la tessera: si vuole tenere traccia degli eventi di apertura del contenitore da parte di un utente. Infine, il sistema è composto da una flotta di veicoli che si occupano della raccolta dei rifiuti. Ogni veicolo dispone di una targa, un nome del modello, una o più foto, uno stato (fermo/in servizio). Si vuole tenere traccia del tragitto operato da ogni veicolo: un tragitto ha un orario di inizio, un orario di fine, e consiste in un elenco di contenitori da visitare, in uno specifico ordine. In caso di riempimento oltre una certa soglia, il sensore di un contenitore invierà una notifica di allarme. Una notifica dispone di una data, un testo, ed è raccolta -al massimo- da un solo veicolo. E’ possibile che la notifica non sia raccolta/gestita da alcun veicolo.

a) (6pt) Costruire il modello Entità-Relazione (E-R) della base di dati.

b) (4pt) Tradurre il modello E-R nel modello logico relazionale, preferendo (NELLA TRADUZIONE DI EVENTUALI GENERALIZZAZIONI, E SOLO IN QUESTE) la soluzione che **minimizzi il numero di valori NULL**. Indicare i vincoli di integrità referenziale tra gli attributi dello schema.

c) (2pt) Indicare se la ridondanza concettuale #numcontenitori debba **essere mantenuta o eliminata**, sulla base delle seguenti operazioni sui dati:

- Contare il numero di contenitori presenti in uno specifico quartiere (Batch, 1 volta/mese).
 - Aggiungere un contenitore ad un quartiere (Batch, 1 volta/mese).
 - Dato uno specifico veicolo, contare il numero di contenitori visitati durante un tragitto (Interattiva, 10 volte/mese).
- Tabella dei volumi.** 6 contenitori per quartiere, 10 contenitori per tragitto, w_I (peso operazioni interattive)=1, w_B (peso operazioni batch)=0.5, alpha = 2.

ESERCIZIO 3 (6 punti)

Data la tabella:

GRUPPO LETTURA BIBLIOTECA(TitoloLibro, Id, Collocazione, Professione, AnnoNascita, CodUtente, StatoCopia, NumeroMembri)

con le seguenti dipendenze funzionali:

TitoloLibro → Collocazione StatoCopia

Id → TitoloLibro

CodUtente → Professione

Id → CodUtente

CodUtente Professione → AnnoNascita

Id TitoloLibro → NumeroMembri

a) (2 pt) Indicare una chiave ed una superchiave della tabella. **Giustificare la risposta.**

b) (3 pt) La tabella è in forma normale di Boyce e Codd (FNBC)? La tabella è in terza forma normale (3FN)? La tabella è in seconda forma normale (2FN)?

Giustificare le risposte.

c) (1 pt) Nel caso in cui la tabella non sia in 3FN, **normalizzarla.**

NOTA: E' consentito ridenominare le colonne es. usando lettere dell'alfabeto.