

Compito di Basi di Dati

Il compito è suddiviso in due parti: la prima si focalizza sulla formulazione delle query, la seconda sulla progettazione delle basi di dati. Ovviamente la prima parte è valida per il recupero del primo compito, la seconda per il recupero del secondo compito.

Gli studenti interessati a recuperare un solo compito devono restituire lo scritto una volta trascorse le prime due ore.

Prima Parte

Si consideri il seguente schema, relativo a Pizzerie, Pizzaioli, Clienti e Recensioni alle Pizzerie:

```
TABLE Pizzerie {
  Nome TEXT NOT NULL ,
  PartitaIVA TEXT NOT NULL UNIQUE,
  Telefono INTEGER NOT NULL UNIQUE,
  Indirizzo TEXT NOT NULL ,
  Città TEXT NOT NULL ,
  PRIMARY KEY(PartitaIVA)
}
```

```
TABLE Pizzaioli {
  Nome TEXT NOT NULL ,
  Cognome TEXT NOT NULL ,
  CodFisc TEXT NOT NULL UNIQUE,
  Indirizzo TEXT NOT NULL ,
  CittàResidenza TEXT NOT NULL ,
  CAP INTEGER NOT NULL ,
  PRIMARY KEY(CodFisc),
  FOREIGN KEY(Pizzeria)
  REFERENCES Pizzerie(PartitaIVA)
}
```

```
TABLE Clienti {
  Nickname TEXT NOT NULL UNIQUE,
  Nome TEXT NOT NULL ,
  Cognome TEXT NOT NULL ,
  Et  INTEGER NOT NULL ,
  CittàResidenza TEXT NOT NULL ,
  PRIMARY KEY(Nickname)
}
```

```
TABLE Recensioni {
  IDrecensione INTEGER NOT NULL UNIQUE,
  Voto INTEGER NOT NULL ,
  Data INTEGER NOT NULL ,
  PRIMARY KEY(IDrecensione),
  FOREIGN KEY(Pizzeria)
  REFERENCES Pizzerie(PartitaIVA),
  FOREIGN KEY(Cliente)
  REFERENCES Clienti(Nickname)
}
```

Produrre le seguenti interrogazioni in SQL:

1. Elenco dei nomi delle pizzerie che non hanno ricevuto alcuna valutazione;

```
SELECT Pext.Nome
FROM Pizzerie AS Pext
WHERE Pext.PartitaIVA NOT IN (SELECT P.PartitaIVA
                              FROM Pizzerie AS P JOIN Recensioni R
                              ON P.PartitaIVA=R.Pizzeria)
```

2. Elenco dei Codici Fiscali dei pizzaioli che lavorano per pizzerie per le quali le recensioni ricevute abbiano tutte un voto superiore a 3;

```
SELECT Pext.CodFisc
FROM Pizzerie AS Pext
WHERE Pext.PartitaIVA NOT IN (SELECT P.PartitaIVA
                              FROM Pizzerie AS P JOIN Recensioni R
                              ON P.PartitaIVA=R.Pizzeria)
```

3. Et  media dei Clienti di Pisa che hanno dato una valutazione superiore a 4 alla pizzeria avente come numero di telefono 050571268;
4. Nome e cognome dei pizzaioli che lavorano nelle pizzerie che hanno avuto il voto medio pi  alto;
5. Elenco delle pizzerie ordinate per numero di telefono

Seconda Parte

Si considerino i seguenti fatti riguardanti una base di dati relativa alla gestione di un polo didattico nel quale si svolgono lezioni universitarie.

Il polo è composto da aule, ogni aula è rappresentata da un identificativo univoco. Ogni aula ha una certa capienza (espressa in posti a sedere) e un certo numero di lavagne. Un'aula può avere o non avere: un proiettore, un microfono e una presa di rete. Nelle aule vengono tenuti i corsi universitari. Ogni corso ha un identificatore univoco, uno o più docenti di riferimento, un numero predeterminato di lezioni. Tutte le lezioni di un corso avvengono nella medesima aula. I corsi sono frequentati nelle aule dagli studenti, di ogni studente interessa il nome, il cognome, la matricola, la data di iscrizione all'università, l'eventuale status di studente lavoratore e di studente fuori corso. Uno studente segue più corsi e può seguire un corso più di una volta.

Si definisca lo schema concettuale, ristrutturato e logico. Nella progettazione **NON** si utilizzi alcuna relazione ternaria e si tenga presente che il committente è interessato a poter ricavare dalla base di dati almeno le seguenti informazioni:

1. la capienza totale del polo;
2. per ogni corso:
 - (a) numero di studenti iscritti
 - (b) il numero di studenti che hanno già seguito un certo corso almeno altre 2 volte;
3. per ogni aula: i corsi ospitati;
4. per ogni studente: il numero di volte che ha seguito un determinato corso;

BONUS: per ogni aula: il numero massimo di studenti che frequentano l'aula;

Per ogni punto dell'elenco precedente si illustri in che modo la base di dati progettata si presta a fornire l'informazione richiesta.

Esempi:

- è possibile recuperare l'informazione richiesta al punto XY utilizzando la tabella A congiunta alla tabella B, ed effettuando una *count()* sull'attributo abc;
- per fornire l'informazione richiesta al punto YZ è sufficiente aggiungere un attributo alla tabella K;

Note Metodologiche: le eventuali tabelle di supporto **devono** avere come nome la giustapposizione dei nomi delle tabelle congiunte in ordine alfabetico. es. due tabelle dal nome Proprietari e Moto avrebbero come supporto la tabella MotoProprietari.