

## 03FYZ TECNICHE DI PROGRAMMAZIONE

Esercitazione di Laboratorio 05 – 15 Aprile 2015

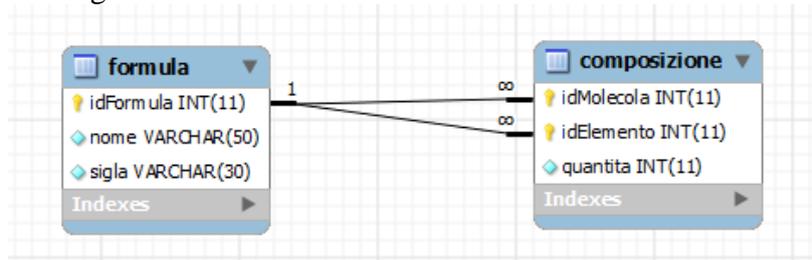
### Obiettivi dell'esercitazione:

- Database
- Controller e Gestione Eventi in FXML
- DAO

**Esame del 2014/01/23, esercizio 1** – In natura esiste un numero finito di **elementi** chimici (es. ossigeno, ferro, zolfo, ... raccolti nella tavola periodica degli elementi), che si possono combinare in un numero pressoché infinito di combinazioni per formare delle **molecole** più complesse. Per comodità, chiamiamo “formula chimica” o più semplicemente “**formula**”, un qualsiasi **elemento** oppure una qualsiasi **molecola**.

Ciascuna formula è caratterizzata da un nome (es. “Oxygen” oppure “Carbon dioxide”) e da una sigla (es. “O” oppure “CO2”). Se la formula è relativa ad un elemento, non ci sono altre informazioni, mentre se la formula è relativa ad una molecola, allora è disponibile anche la **composizione** di tale molecola. Con “composizione” si intende l’insieme degli elementi di cui la molecola è composta, indicando per ciascuno la relativa quantità. Ad esempio, per “Carbon dioxide”, la composizione sarà: “Oxygen 2, Carbon 1”. Si precisa che in questo contesto ci interessa solamente la “formula bruta”, ossia il numero di atomi componenti, e non la struttura molecolare.

Le informazioni relative agli elementi ed alle molecole più utilizzate sono rappresentate nella base dati avente la struttura seguente.



Nella tabella ‘composizione’, la chiave esterna idMolecola rappresenta la molecola di cui stiamo descrivendo la composizione, mentre la chiave esterna idElemento rappresenta il singolo elemento di cui stiamo specificando la quantità. Per meglio comprendere la struttura del database si può analizzare l’esempio seguente, nel quale si riporta la composizione della formula CO2 (idFormula=3) che è composta da quantità 2 di idElemento=1 (ossia O) e quantità 1 di idElemento=2 (ossia C).

molecole.formula: 3 righe totali (circa)

| idFormula | nome           | sigla |
|-----------|----------------|-------|
| 1         | Oxygen         | O     |
| 2         | Carbon         | C     |
| 3         | Carbon Dioxide | CO2   |

molecole.composizione: 4 righe totali (circa)

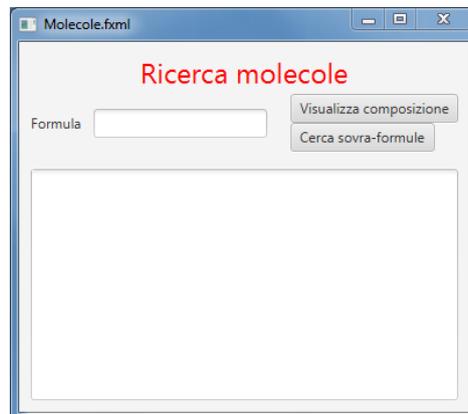
| idMolecola | idElemento | quantita |
|------------|------------|----------|
| 1          | 1          | 1        |
| 2          | 2          | 1        |
| 3          | 1          | 2        |
| 3          | 2          | 1        |

Si intende costruire un’applicazione JavaFX che permetta di interrogare tale base dati, ed estrarre informazioni a proposito delle molecole.

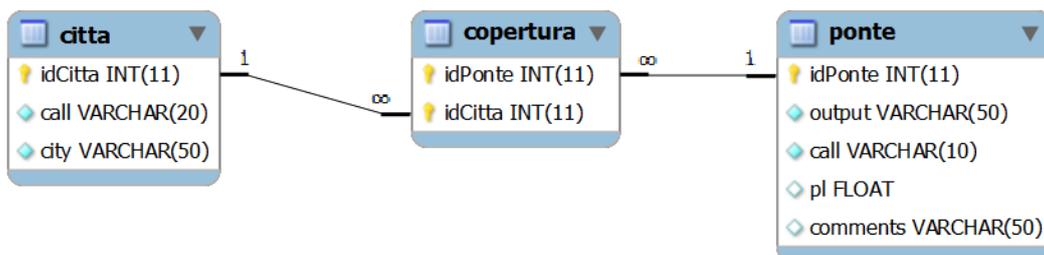
L’applicazione dovrà svolgere la seguente funzione:

1. Permettere all’utente di inserire il nome di una formula (elemento o molecola) e visualizzarne la sigla e la composizione. Ad esempio, se l’input è “Carbon dioxide” occorre visualizzare “CO2. Composizione: Oxygen 2, Carbon 1”. L’ordine degli elementi nella

composizione non è rilevante. È richiesto di gestire tutti i possibili errori di immissione dei dati.



**Esame del 2014/09/12, esercizio 2** – Si consideri un data set che rappresenta la copertura di una serie di città da parte di una rete di ponti radio. Ogni ponte radio trasmette su una certa frequenza (output) ed il suo segnale è in grado di coprire una o più città. La stessa città può essere coperta da più di un ponte radio.



Si intende costruire un'applicazione JavaFX che permetta di interrogare tale base dati, e calcolare informazioni a proposito del funzionamento della rete.

L'applicazione dovrà svolgere le seguenti funzioni:

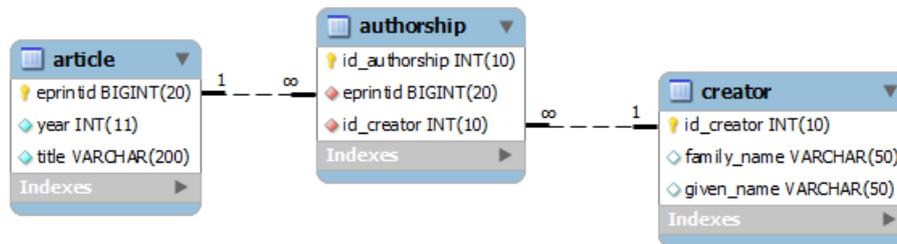
1. Permettere all'utente di selezionare il nome di due città da altrettanti menu a tendina (in ordine alfabetico) e visualizzare l'elenco dei ponti radio che coprono entrambe le città, ammesso che ne esistano. Le città devono essere distinte tra di loro.



**Esame del 2013/09/10, esercizio 3** – Nelle comunità scientifiche è talvolta necessario analizzare la produzione scientifica dei ricercatori. A tal fine si utilizza il meccanismo denominato “peer review” in cui un articolo è letto e valutato da altri ricercatori. I revisori sono scelti fra chi non ha mai collaborato direttamente con gli autori dell’articolo. Disponendo di una base dati delle pubblicazioni, quest’ultimo requisito si traduce nello scegliere i revisori fra coloro che non sono mai stati co-autori di uno degli autori dell’articolo, e che non hanno mai avuto un co-autore comune ad uno degli autori dell’articolo.

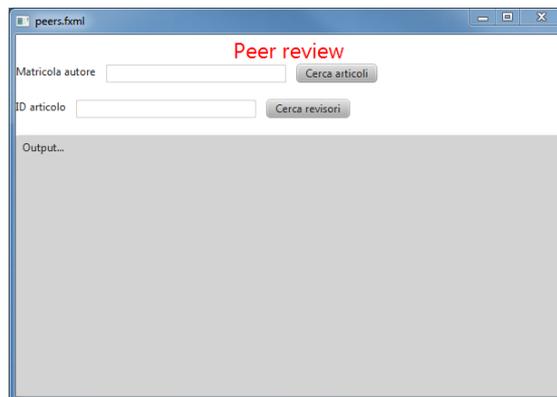
Ad esempio, “The History of Cold Fusion” di Martin Fleischmann non potrà essere valutato né da chi Stanley Pons (che ha scritto un articolo insieme a Fleischmann), né da Kevin Ashley (che ha scritto un articolo insieme a Stanley Pons).

Si intende costruire un’applicazione JavaFX che permetta di estrarre delle statistiche a partire dalle pubblicazioni presenti nel database “Porto” del Politecnico di Torino. Lo schema del database su cui lavorare è riportato in figura:



L’applicazione dovrà svolgere la seguente funzione:

1. Permettere all’utente di inserire il numero di matricola di uno degli autori (si presume che l’utente lo conosca) e, alla pressione del bottone “Cerca articoli”, visualizzare tutti gli articoli dell’autore specificato, in ordine decrescente di anno di pubblicazione, riportando per ciascuno anno, titolo ed identificatore numerico. È necessario riconoscere e gestire ogni eventuale errore di immissione.



La figura seguente riporta la struttura dei Bean che dovranno essere implementati:

