

# Compito di Basi di dati

23 giugno 2021

## Esercizio 1:

Sia dato il seguente schema relazionale relativo a spettacoli e stagioni teatrali:

*OPERA*(*IdOpera*, *Titolo*, *Compagnia*, *Regista*, *AnnoProduzione*);

*STAGIONE*(*NomeTeatro*, *Anno*);

*CARTELLONE*(*Opera*, *Teatro*, *Anno*).

Si assuma che ogni opera teatrale sia individuata univocamente da un identificatore e sia caratterizzata da un titolo, una compagnia, un regista (si assuma che ogni opera abbia un unico regista) e un anno di produzione. Si assuma che un regista possa dirigere più di un'opera. Si ammetta la possibilità che vengano realizzate opere teatrali diverse con lo stesso titolo (con la stessa compagnia o con compagnie diverse e con lo stesso regista o con un regista diverso), ma si escluda che possano essere prodotte lo stesso anno. Si assuma che ogni teatro sia identificato dal suo nome (se necessario, il nome del teatro comprenderà il nome della città ove si trova; ad esempio, Teatro Stabile di Torino). Ogni stagione teatrale sia caratterizzata dal teatro ove si tiene e dall'anno in cui si tiene. Infine, il cartellone specifichi le opere offerte da un certo teatro un certo anno. Si assuma che una stessa opera possa essere presente in più cartelloni dello stesso anno o di anni diversi. In particolare, una stessa opera possa essere offerta da un certo teatro più anni, ossia possa essere inserita in cartellone da quel teatro più anni.

Definire preliminarmente le chiavi primarie, le eventuali altre chiavi candidate e, se ve ne sono, le chiavi esterne delle relazioni date. Successivamente, formulare opportune interrogazioni in SQL che permettano di determinare (senza usare l'operatore CONTAINS e usando solo se e quando necessario le funzioni aggregate):

- le opere che nel 2020 sono state inserite nel cartellone di esattamente 2 teatri;
- i teatri che nella stagione 2018 avevano in cartellone tutte le opere presenti nel cartellone del Teatro Stabile di Torino di quell'anno.

## Esercizio 2:

Si immagini di dover progettare una base di dati che raccoglie informazioni sugli esopianeti (pianeti con caratteristiche simili alla Terra) finora scoperti e sulle stelle attorno alle quali essi orbitano.

La base di dati comprende le tabelle **Stelle** e **Esopianeti**.

La tabella **Stelle** ha due attributi:

- nome**, che è una stringa di al più 50 caratteri ed è la chiave primaria;
- classificazione**, che è una stringa di al più 50 caratteri, non nulla.

La tabella **Esopianeti** ha quattro attributi:

- nome**, che è una stringa di al più 50 caratteri ed è la chiave primaria;
- la **stella** attorno alla quale orbita, che è una stringa di al più 50 caratteri, non nulla, ed è una chiave esterna che fa riferimento alla chiave primaria della tabella **Stelle**;
- il parametro **ESI** (*Earth Similarity Index*), che è un valore decimale, non nullo, compreso tra 0 (“completamente diverso dalla Terra”) ed 1 (“uguale alla Terra”);
- la **distanza** dalla Terra (espressa in *anni luce*), che è un valore decimale maggiore di 0 o *nullo* (nel caso non si sappia la distanza o il pianeta sia la Terra stessa).

Si scriva il codice SQL per creare e popolare le seguenti tabelle:

Esopianeti			
nome	stella	ESI	distanza
terra	sole	1	-
trappist-1 d	trappist-1	0.87	39
proxima b	proxima centaury	0.87	4.22
teegarden b	teegarden	0.93	12.6

Stelle	
nome	classificazione
sole	nana gialla
teegarden	nana rossa
trappist-1	nana rossa
proxima centaury	nana rossa
antares	supergigante rossa

Si ipotizzi che gli scienziati abbiano appena fatto dei calcoli più precisi rivelando che la corretta distanza dalla Terra del pianeta **teegarden b** è 12.5 anni luce. Si scriva il codice SQL che effettua tale modifica.

Si scriva il codice SQL per ordinare la tabella **Esopianeti** in ordine decrescente rispetto all'attributo ESI, escludendo la Terra. A parità di ESI, ordinare rispetto alla distanza crescente.

Si consideri il seguente vincolo: un esopianeta non può orbitare attorno ad una stella che è una supergigante rossa. Quali operazioni, e su quali tabelle, possono violare questo vincolo? Si scelga una di queste operazioni e si scriva un trigger SQL che eviti tale violazione.

### Esercizio 3:

Sia dato il seguente insieme di requisiti relativi ad una base di dati per la gestione di un insieme di officine tutte di proprietà di una stessa azienda.

- Ogni officina sia caratterizzata da un nome, che la identifica univocamente, un indirizzo, un recapito telefonico, un'email, un certo insieme (non vuoto) di dipendenti, che lavorano presso di essa, e un direttore. Si assuma che ogni dipendente possa lavorare presso una e una sola officina e che ogni officina abbia uno ed un solo direttore, ma un direttore possa dirigere più di un'officina. Di ogni officina si voglia sapere il numero di dipendenti che vi lavorano.
- Dei dipendenti e dei direttori si registrino il codice fiscale, la residenza e uno o più numeri di telefono. Dei dipendenti vengano memorizzati gli anni di servizio presso l'azienda. Dei direttori si registri la data di nascita. Si assuma che il direttore non sia necessariamente un dipendente dell'azienda, ma possa esserlo.
- Ogni riparazione sia effettuata da una ed una sola officina e riguardi un unico veicolo. Di ogni riparazione vengano registrati un codice, che la identifica univocamente all'interno delle riparazioni effettuate dall'officina, data e ora di accettazione del veicolo e, nel caso di riparazioni terminate, data e ora di riconsegna del veicolo.
- Dei veicoli vengano memorizzati il modello, il tipo, la targa, l'anno di immatricolazione e il proprietario. Si assuma, per semplicità, che ogni veicolo abbia uno ed un solo proprietario. Dei proprietari di veicoli si registrino il codice fiscale, la residenza e uno o più numeri di telefono.

Si definisca uno schema Entità-Relazioni che descriva il contenuto informativo del sistema, illustrando con chiarezza le eventuali assunzioni fatte. Lo schema dovrà essere completato con attributi ragionevoli per ciascuna entità (identificando le possibili chiavi) e relazione. Vanno specificati accuratamente i vincoli di cardinalità e partecipazione di ciascuna relazione. Si definiscano anche eventuali regole di gestione (regole di derivazione e vincoli di integrità) necessarie per codificare alcuni dei requisiti attesi del sistema.

### Esercizio 4:

In riferimento alla nozione di serializzabilità rispetto alle viste, si stabilisca se possono esistere (i) degli schedule seriali non serializzabili e (ii) degli schedule serializzabili non seriali. In caso di risposta positiva, si fornisca un esempio; in caso di risposta negativa, si spieghi il perché.

Successivamente, si stabilisca se il seguente schedule appartiene o meno a VSR, CSR, 2PL e 2PL stretto:

- $s : r_0(x), r_3(y), w_1(x), r_2(z), r_1(x), w_0(x), r_2(y), w_1(x), w_3(z)$ .