

Basi di Dati 2022-2023

Assignment 2: SQL

Termine per la consegna

Il termine tassativo per la consegna dell'Assignment 2 è **giovedì 1° dicembre 2022 alle 23:59 CET**. Nessuna consegna dopo il termine indicato sarà considerata per la valutazione.

Istruzioni

Si consideri il seguente schema di base di dati relazionale (lo script per generarlo è fornito qui sotto):

```
movies (title:string, year:int, director:string, budget:int, gross:int)
    (director) è chiave esterna che fa riferimento a directors
directors (director:string, yearofbirth:int)
directorawards (director:string, year:int, award:string, result:string)
    (director) è chiave esterna che fa riferimento a directors
movieawards (title:string, year:int, award:string, result:string)
    (title, year) è chiave esterna che fa riferimento a movies
```

Precisazioni:

- Gli attributi sottolineati indicano le chiavi primarie.
- Tutti gli attributi sono da considerare come NOT NULL.
- L'attributo `result` può avere solo due valori: 'won' o 'nominated'.
- I valori dell'attributo `award` nella relazione `directorawards` è il nome dell'evento, per esempio 'oscar' o 'cannes'.
- I valori dell'attributo `award` nella relazione `movieawards` consistono di due parti, ovvero l'evento e il tipo di premio (`award`) separati da una virgola. Per esempio: 'oscar, best film', 'golden globe, screenplay', 'oscar, best director'.
- Tutti i premi (awards) memorizzati nella relazione `directorawards` sono del tipo “miglior regista” (best director). Per esempio, ‘oscar’ in `directorawards` è equivalente a 'oscar, best director' in `movieawards`.
- È possibile che un premio per il regista (director award) appaia anche nei premi per il film (movie award). Per esempio, un premio per il regista (director award) ‘oscar’ apparirà in `directorawards` (semplicemente come ‘oscar’) e in `movieawards` (come 'oscar, best director').

Script per creare lo schema della base di dati:

```
CREATE TYPE awardresult AS ENUM  
('won', 'nominated');
```

```
CREATE TABLE directors  
(  
    director varchar(50) NOT NULL,  
    yearofbirth integer NOT NULL,  
    CONSTRAINT "directors_pkey" PRIMARY KEY (director)  
);
```

```
CREATE TABLE movies  
(  
    title varchar(100) NOT NULL,  
    year integer NOT NULL,  
    director varchar(50) NOT NULL,  
    budget integer NOT NULL,  
    gross integer NOT NULL,  
    CONSTRAINT "movies_pkey" PRIMARY KEY (title, year),  
    CONSTRAINT "directorawards_director_fkey" FOREIGN KEY (director)  
        REFERENCES directors(director)  
);
```

```
CREATE TABLE directorawards  
(  
    director varchar(50) NOT NULL,  
    year integer NOT NULL,  
    award varchar(100) NOT NULL,  
    result awardresult NOT NULL,  
    CONSTRAINT "DirectorAwards_pkey" PRIMARY KEY (director, year, award),  
    CONSTRAINT "DirectorAwards_director_fkey" FOREIGN KEY (director)  
        REFERENCES Directors(director)  
);
```

```
CREATE TABLE movieawards  
(  
    title varchar(100) NOT NULL,  
    year integer NOT NULL,  
    award varchar(100) NOT NULL,  
    result awardresult NOT NULL,  
    CONSTRAINT "movieawards_pkey" PRIMARY KEY (title, year, award),  
    CONSTRAINT "movieawards_year_title_fkey" FOREIGN KEY (title, year)  
        REFERENCES movies(title, year)  
);
```

Scrivere la seguenti QUERY in SQL (le query sono intenzionalmente formulate sia in Italiano sia in Inglese):

1. Per ogni regista (director) con più di 50 anni, restituire il profitto minimo, massimo e medio dei suoi film. (Restituire -1 per il profitto minimo, massimo e medio dei registi per i quali non è presente alcun film)

For each director that is older than 50, find the minimum, maximum, and average profit of his/her movies. (Output -1 for minimum, maximum, and average of the directors with no movies).

2. Trovare il film (o i film) più recente/i che ha (hanno) vinto almeno 3 premi.

Find the latest movie(s) that won at least 3 awards.

3. Trovare il film (i film) con più il guadagno più alto e il film (i film) meno costoso/i. (Usare una colonna denominata “*feature*” che accetta come valori rispettivamente “*most profitable*” e “*least expensive*” per distinguere i due casi). (Si noti che il guadagno (profit) indica gli incassi meno le spese).

Find the most profitable movie(s), and the least expensive movie(s). (Use a column named “*feature*” which accepts values 'most profitable' and 'least expensive' respectively to distinguish mentioned cases) (note that profit means the gross that remains on top of budget)

4. Trovare i premi ricevuti dai registi (*director*) che hanno fatto alti incassi negli ultimi 5 anni. (Alti incassi significa incassi superiore a 1 milione). (Restituire in ordine premio, anno, regista (*award, year, director*)).

Find awards received by directors who made a high-grossing movie in the past 5 years. (High-grossing movie is a movie that have made over 1 million gross) (Output award, year, and director)

5. Per ogni film, trovare il suo “tasso di successo”, ovvero la probabilità di ricevere un premio per il quale era stato candidato. (Restituire -1 per i film che non hanno ricevuto alcuna candidatura (*nomination*)). (Il tasso di successo è espresso con rappresentazione decimale).

For each movie, find its award "success-rate", that is, the probability of getting an award it was nominated for. (Output -1 for movies never even nominated) (success-rate is in a decimal representation)

6. Trovare il film (i film) che ha (hanno) vinto il più alto numero di Oscar.

Find the movie(s) that won the largest number of Oscars.

7. Trovare i registi più giovani e più anziani che hanno vinto un Oscar come “miglior regista” (si ricordi che questo può essere memorizzato in movieawards o in directorawards). (Restituire il regista e “feature” che accetta i valori ‘youngest’ e ‘oldest’). (Più giovane / più anziano (youngest / oldest) si riferisce al momento di esecuzione della query e non all’età del regista al momento in cui ha ricevuto il premio).

Find the youngest and the oldest directors who won an Oscar for "best director" (recall that it can be recorded in movieawards or directorawards) (Output director, and feature accepts values 'youngest' and 'oldest') (youngest/oldest refers to their current age at the time of executing the query, not the age at the time of receiving the awards)

8. Trovare la percentuale di film che hanno vinto un Oscar tra i film girati negli anni ‘80 (Restituire -1 se nessun film è stato girato negli anni ‘80).

Find the percentage of movies that won an Oscar among all movies made in the 80s. (Output -1 if no movies were made in the 80s.)

9. Trovare i registi che hanno vinto il premio per miglior regista, anche se il film per il quale hanno vinto il premio non ha prodotto margini di guadagno.

Find directors who won the 'best director' award, although the movie for which they won the award was not profitable.

10. Trovare i registi che hanno vinto il premio per miglior regista (best director) tutti gli anni in cui un film di Spielberg ha vinto almeno 3 premi (“Spielberg” è il nome esatto del regista e non una parte del nome).

Find directors who won a 'best director' every year when a Spielberg movie won at least three awards. (“Spielberg” is the exact name of director not part of it)

Istruzioni

- L'assignment deve essere svolto individualmente.
- Scrivere ogni interrogazione (query) in un singolo file di testo (**plain text file**). Quindi qualcosa che è possibile creare con Visual Studio Code, IntelliJ, ecc., oppure semplicemente con il Blocco Note di Windows. **NON USATE WORD, WordPad, OPPURE IL PROGRAMMA TEXTEDIT DI MAC, PERCHÉ IL FORMATO DI FILE USATO DA QUEI PROGRAMMI NON È TESTO SEMPLICE.**
- Ciascun file deve contenere soltanto la query corrispondente.
- Ogni query sarà valutata da uno script automatico che confronterà i risultati delle vostre query con quelli attesi. Ogni query sarà valutata come corretta (+1 point) o sbagliata (+0 points). La valutazione è fatta automaticamente, quindi se la query presenta errori di sintassi, verrà valutata automaticamente con 0.
- La lista di **attributi nel risultato deve essere ordinata come indicato nel testo**. Esempi: “restituire il nome della birra e il suo produttore” deve corrispondere alla query “SELECT name, manufacturer FROM ...”. L'errato ordinamento delle colonne nel risultato produrrà una valutazione di 0 (zero) punti.
- **PER CIASCUNA QUERY, le tuple in output devono essere ordinate in base all'ordine crescente.** Per esempio, se una delle query restituisce una lista di 3 attributi x,y,z, allora le tuple del risultato della query devono essere ordinate rispetto a x, y, z. Cioè, va utilizzato ORDER BY x, y, z
- Quando viene richiesto di restituire informazione su una relazione, scegliere sempre gli attributi che la identificano, ovvero la chiave primaria. Per esempio, “Trovare tutti i registi (directors) che ...” deve essere: “SELECT director FROM directors”. (dove l'attributo “director” è chiave primaria della relazione “directors”)
- I numeri non interi devono essere **troncati** al secondo decimale. Usare il comando TRUNCATE. Per esempio: 0.1274 deve diventare 0.12.
- Gli output che contengono numeri senza decimali non devono contenere alcun decimale. Per esempio, restituire 11 e non 11.00.
- A meno che non sia specificato diversamente, **i risultati di una query devono essere intesi come distinti**. In altri termini, si usi sempre il DISTINCT a meno che non sia esplicitamente richiesto di non usarlo.
- **Non creare tabelle aggiuntive o viste persistenti.** Se proprio lo si ritiene necessario, possono essere create viste temporanee (usando il parametro TEMP).
- Non modificare in alcun modo lo schema esistente.
- Le interrogazioni saranno verificate mediante un database PostgreSQL fornito dall'Università. E' possibile installare una propria istanza di PostgreSQL sul proprio PC per lavorarci localmente, ma si tenga sempre a mente che le query devono funzionare sul database dell'Università. A tale fine riceverete il numero (nome, username e password) del database che è stato riservato per voi.
- Si verifichino le proprie interrogazioni prima di spedirle per la valutazione! Accertarsi che le interrogazioni restituiscano sempre il risultato corretto. Si consideri la possibilità che ci siano casi

particolari. Se si desiderano ottenere dati reali per testare le interrogazioni, è possibile utilizzare dati dal database IMDB.

Consegna

- Creare un file di **testo semplice** per ciascuna query. Il nome del file non è importante. Per il nome, si suggerisce comunque di usare il formato **query_<number>.sql**. Per esempio, la query 8 corrisponderà al file `query_8.sql`
- Caricare singolarmente i 10 file così creati usando il seguente Google Form:
https://docs.google.com/forms/d/e/1FAIpQLScImTQZkDCwQrbO_1xb3HvoIX-6ExbU-VwnUq62V88ivF_qxA/viewform?usp=sf_link
- **La valutazione delle query è eseguita da un software automatico.** Quindi file di testo non validi e altre imprecisioni porteranno a valutare l'elaborato con 0 punti.