LIBR 554: Final Group Project

Kyla McCallum, Bryn Shaffer, Stephanie Hohn

Query One

User: Library patrons looking for movies that they would like and understand.

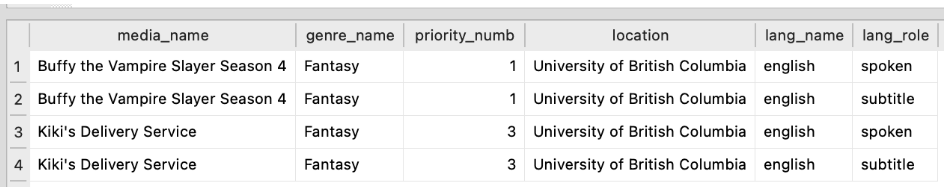
Purpose: Find media items that are in the fantasy genre and either spoken or subtitled in English.

Required layout for the result set: media_name, genre_name, priority_numb, location

SQL statement:

```
SELECT m.media_name, g.genre_name, mg.priority_numb, m.location,  l.lang_name,
l.lang_role
FROM media_genre mg
JOIN media m ON mg.media_ID = m.media_ID
JOIN genre g ON mg.genre_ID = g.genre_ID
JOIN Media_Lang ml ON ml.media_ID = m.media_ID
JOIN language l ON l.lang_ID = ml.lang_ID
WHERE g.genre_name = "Fantasy" AND lang_name = "english"
ORDER BY mg.priority_numb ASC
```

Result set generated by your query in your database:



	media_name	genre_name	priority_numb	location	lang_name	lang_role
1	Buffy the Vampire Slayer Season 4	Fantasy	1	University of British Columbia	english	spoken
2	Buffy the Vampire Slayer Season 4	Fantasy	1	University of British Columbia	english	subtitle
3	Kiki's Delivery Service	Fantasy	3	University of British Columbia	english	spoken
4	Kiki's Delivery Service	Fantasy	3	University of British Columbia	english	subtitle

---

Query Two

User: Users interested in discovering what actors and directors produced media in a VHS format. These could be library patrons, or administrators seeking to understand the scope of Videomatica media currently on VHS.

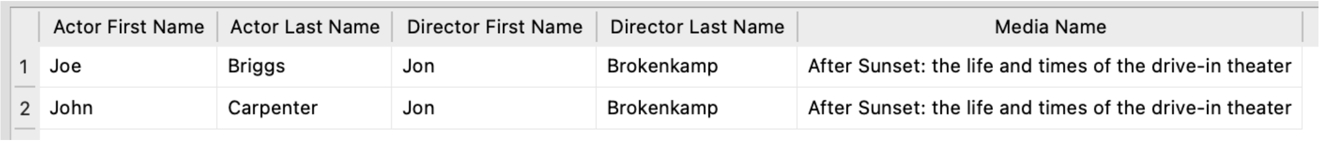
Purpose: To view further information about actors and directors that have Videomatica media on VHS.

Required layout for the result set: Actor First Name, Actor Last Name, Director First Name, Director Last Name, Media Name

SQL statement:

```
CREATE VIEW vhs_media AS
SELECT a.first_name as 'Actor First Name', a.last_name as 'Actor Last Name', d.first_name as
'Director First Name', d.last_name as 'Director Last Name', m.media_name as 'Media Name'
FROM media m
JOIN media_actor ma ON m.media_ID = ma.media_ID
JOIN actor a ON a.actor_ID = ma.actor_ID
JOIN media_director md ON md.media_ID = m.media_ID
JOIN director d ON md.director_ID = d.director_ID
WHERE m.format = "VHS"
ORDER BY a.last_name ASC;

SELECT * FROM  vhs_media
```

Result set generated by your query in your data:



	Actor First Name	Actor Last Name	Director First Name	Director Last Name	Media Name
1	Joe	Briggs	Jon	Brokenkamp	After Sunset: the life and times of the drive-in theater
2	John	Carpenter	Jon	Brokenkamp	After Sunset: the life and times of the drive-in theater

---

Query Three

User: An admin who is looking to acquire new media items.

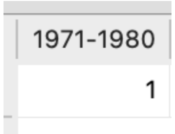
Purpose: The admin might want to understand the release year of their current collection. This is a stored procedure so that the admin can check the status of their collection repeatedly.

Required layout for the result set: min_year - max_year

```
DELIMITER //

CREATE PROCEDURE view_release_year(IN min_year INT, IN max_year INT)
BEGIN
        SET @column_heading = CONCAT(min_year, '-', max_year);
        SET @sql = CONCAT('SELECT COUNT(CASE WHEN year BETWEEN ',
        min_year, ' AND ', max_year, ' THEN media_ID END) AS "', @column_heading, '"
        FROM media;');
        PREPARE stmt FROM @sql;
        EXECUTE stmt;
        DEALLOCATE PREPARE stmt;
```

```
END //

DELIMITER ;

CALL view_release_year (1971,1980)
```

Result set generated by your query in your database:



1971-1980
1

Query Four

User: An admin who is looking to acquire new media items.

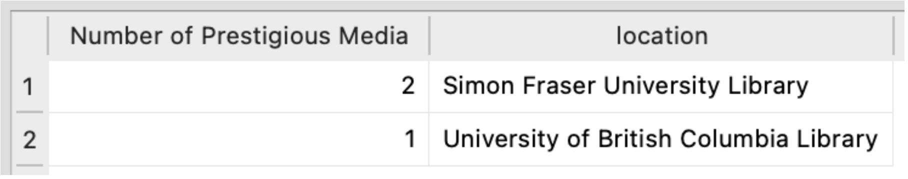
Purpose: The admin might want to see how many of their media items have awards. They could compare this to other locations to see if their collection includes many more or many less prestigious media items.

Required layout for the result set: Number of Prestigious Media, location

SQL statement:

```
SELECT COUNT(media.media_ID) AS 'Number of Prestigious Media', media.location
FROM media
JOIN media_award ma
ON ma.media_ID = media.media_ID
GROUP BY location
HAVING COUNT(ma.media_ID)>0;
```

Result set generated by your query in your database:



	Number of Prestigious Media	location
1	2	Simon Fraser University Library
2	1	University of British Columbia Library

Query Five

User: When an admin adds a new item to the Videomatica collection, a log message is created noting the date of the addition.

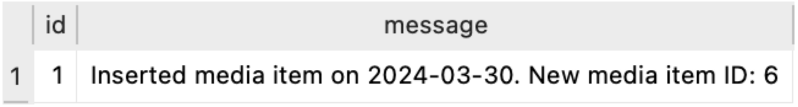
Purpose: To create a record of when new Videomatica items came into the collection.

Required layout for the result set: log_messages.id, log_messages.message

SQL statement:

```
CREATE TRIGGER IF NOT EXISTS new_media_log

AFTER INSERT ON media FOR EACH ROW

BEGIN

INSERT INTO log_messages (message) VALUES ('Inserted media item on ' ||
CURRENT_DATE || '. New media item ID: ' || NEW.media_ID);

END;

INSERT INTO media (media_ID, media_name, year, format, location, series_name)

VALUES ("6", "Sex : the Annabel Chong story", "1972", "DVD", "Simon Fraser University
Library", "standalone");

SELECT id, message

FROM log_messages;
```

Result set generated by your query in your database:



	id	message
1	1	Inserted media item on 2024-03-30. New media item ID: 6