Homework 7 MySQL

|  |  |  |
| --- | --- | --- |
|  | Display the first and last names of all actors from the table actor. | |
|  | SELECT first\_name, last\_name FROM actor; | |
|  | Display the first and last name of each actor in a single column in upper case letters. Name the column Actor Name. | |
|  | SELECT CONCAT(first\_name, ' ', last\_name) AS Actor\_Name FROM actor;  You need to find the ID number, first name, and last name of an actor, of whom you know only the first name, "Joe." What is one query would you use to obtain this information? | |
|  |  | |
| SElECT actor\_id, first\_name, last\_name FROM actor WHERE first\_name = 'joe';  Find all actors whose last name contain the letters GEN: | |
| ` |
|  | Select first\_name, last\_name From actor WHERE last\_name LIKE '%GEN%';  Find all actors whose last names contain the letters LI. This time, order the rows by last name and first name, in that order: | |
|  |  | |
|  | SELECT last\_name, first\_name FROM actor WHERE last\_name LIKE '%LI';  2d. Using IN, display the country\_id and country columns of the following countries: Afghanistan, Bangladesh, and China: | |
|  |  | |
|  | SELECT country\_id, country FROM country WHERE country IN ('Afghanistan', 'Bangladesh', 'China'); | |
|  | You want to keep a description of each actor. You don't think you will be performing queries on a description, so create a column in the table actor named description and use the data type BLOB (Make sure to research the type BLOB, as the difference between it and VARCHAR are significant). | |
|  | ALTER TABLE actor ADD middle\_name VARCHAR(255) AFTER first\_name; | |
|  |  | |
|  | ALTER TABLE actor MODIFY middle\_name BLOB; | |
|  |  | |
|  | ALTER TABLE actor DROP COLUMN middle\_name; | |
|  | List the last names of actors, as well as how many actors have that last name. | |
|  | SELECT last\_name, COUNT(last\_name) AS name\_count FROM actor | |
|  | GROUP BY last\_name; | |
|  |  | |
|  | List last names of actors and the number of actors who have that last name, but only for names that are shared by at least two actors  SELECT last\_name, COUNT(last\_name) AS name\_count FROM actor | |
|  | GROUP BY last\_name | |
|  | HAVING COUNT(last\_name) > 1; | |
|  |  | |
|  | The actor HARPO WILLIAMS was accidentally entered in the actor table as GROUCHO WILLIAMS. Write a query to fix the record.  UPDATE actor | |
|  | SET first\_name = 'HARPO' | |
|  | WHERE first\_name = 'GROUCHO' AND last\_name = 'WILLIAMS'; | |
|  | Perhaps we were too hasty in changing GROUCHO to HARPO. It turns out that GROUCHO was the correct name after all! In a single query, if the first name of the actor is currently HARPO, change it to GROUCHO. | |
|  | UPDATE actor | |
|  | SET first\_name = 'GROUCHO' | |
|  | WHERE first\_name = 'HARPO'; | |
|  | 5a. You cannot locate the schema of the address table. Which query would you use to re-create it? | |
|  | CREATE TABLE address2 | |
|  | ( address\_id smallint UNSIGNED NOT NULL AUTO\_INCREMENT, address VARCHAR(50), | |
|  | address2 VARCHAR(50), district VARCHAR(20), city\_id smallint UNSIGNED, | |
|  | postal\_code VARCHAR(10), phone VARCHAR(20), location GEOMETRY, last\_update TIMESTAMP ); | |
|  |  | |
|  | Use JOIN to display the first and last names, as well as the address, of each staff member. Use the tables staff and address:  SELECT staff.first\_name, staff. last\_name, address.adress | |
|  | FROM staff JOIN address ON staff.address\_id = address.adress\_id; | |
|  | Use JOIN to display the total amount rung up by each staff member in August of 2005. Use tables staff and payment. | |
|  | SELECT staff.first\_name, staff.last\_name, SUM(payment.amount) AS total\_sales | |
|  | FROM staff JOIN payment ON staff.staff\_id = payment.staff\_id | |
|  | WHERE payment.payment\_date LIKE '2005-08%' | |
|  | GROUP BY username; | |
|  | list each film and the number of actors who are listed for that film. Use tables film\_actor and film. Use inner join. | |
|  | SELECT title, COUNT(actor\_id) AS actors\_in\_film | |
|  | FROM film\_actor INNER JOIN film ON film\_actor.film\_id = film.film\_id | |
|  | GROUP BY title; | |
|  |  | |
|  | 6d. How many copies of the film Hunchback Impossible exist in the inventory system?  SELECT title, COUNT(inventory.film\_id) AS copies | |
|  | FROM inventory INNER JOIN film ON inventory.film\_id = film.film\_id | |
|  | WHERE title = 'Hunchback Impossible' | |
|  | GROUP BY title; | |
|  |  | |
|  | SELECT first\_name, last\_name, SUM(amount) AS total\_paid | |
|  | FROM payment JOIN custome ON customer.customer\_id = payment.customer\_id | |
|  | GROUP BY last\_name; | |
|  | The music of Queen and Kris Kristofferson have seen an unlikely resurgence. As an unintended consequence, films starting with the letters K and Q have also soared in popularity. Use subqueries to display the titles of movies starting with the letters K and Q whose language is English. | |
|  | SELECT title FROM film | |
|  | WHERE title LIKE 'K%' OR title LIKE 'Q%' | |
|  | AND language\_id=(SELECT language\_id FROM language WHERE name='English'); | |
|  |  | |
|  | 7b. Use subqueries to display all actors who appear in the film Alone Trip.  SELECT first\_name, last\_name | |
|  | FROM actor | |
|  | WHERE actor\_id IN | |
|  | (SELECT actor\_id | |
|  | FROM film\_actor | |
|  | WHERE film\_id IN | |
|  | (SELECT film\_id | |
|  | FROM film | |
|  | WHERE title = 'ALONE TRIP')); | |
|  | 7c. You want to run an email marketing campaign in Canada, for which you will need the names and email addresses of all Canadian customers. Use joins to retrieve this information. | |
|  | select first\_name, last\_name, email | |
|  | FROM customer | |
|  | JOIN adress ON customer.adress\_id=adress.adress\_id | |
|  | JOIN city ON address.city\_id=city.city\_id | |
|  | JOIN country ON city.country\_id=country.country.id | |
|  | WHERE country = 'Canada'; | |
|  | Sales have been lagging among young families, and you wish to target all family movies for a promotion. Identify all movies categorized as family films. | |
|  | SELECT title, name | |
|  | FROM film | |
|  | JOIN film\_category ON film.film\_id = film\_category.film\_id | |
|  | JOIN category ON film\_category.category\_id = category.category\_id | |
|  | WHERE category.name = 'Family'; | |
|  |  | |
|  | SELECT title, COUNT(rental\_id) as 'Times Rented' | |
|  | FROM film f | |
|  | JOIN inventory i on i.film\_id = f.film\_id | |
|  | JOIN rental r ON r.inventory\_id = i.inventory\_id | |
|  | GROUP BY title | |
|  | ORDER BY COUNT(rental\_id) DESC; | |
|  | Write a query to display for each store its store ID, city, and country | |
|  | SELECT store.store\_id, SUM(amount) AS 'Total sales' FROM store | |
|  | JOIN staff on store.store\_id = payment.staff\_id | |
|  | JOIN payment ON staff.staff\_id = payment.staff\_id | |
|  | GROUP BY store\_id; | |
|  | Write a query to display for each store its store ID, city, and country. | |
|  | SELECT store\_id, address, city, country | |
|  | FROM store s | |
|  | JOIN adress a ON s.adress\_id = a.adress\_id | |
|  | JOIN city c ON c.city\_id = a.city\_id | |
|  | JOIN country ON county.country\_id = c.country\_id; | |
|  |  | |
|  | SELECT name, SUM(amount) as 'Gross' FROM category c | |
|  | JOIN film\_category fc ON fc.category\_id = c.categpory\_id | |
|  | JOIN inventory ON inventory.film\_id=film\_category.film\_id | |
|  | JOIN rental r ON r.inventory\_id = i.inventory\_id | |
|  | JOIN payment ON payment.rental\_id=rental.rental\_id | |
|  | GROUP BY name | |
|  | ORDER BY SUM(amount) DESC LIMIT 5; | |
|  |  | |
|  | CREATE VIEW TOP\_5 AS | |
|  | SELECT name, SUM(amount) as 'Gross' FROM category c | |
|  | JOIN film\_catefory fc ON fc.category\_id = fc.film\_id | |
|  | JOIN inventory i ON i.film\_id = i.inventory\_id | |
|  | JOIN rental r ON r.inventory\_id = i.inventory\_id | |
|  | JOIN payment ON payment.rental\_id=rental.rental\_id | |
|  | GROUP BY name | |
|  | ORDER BY SUM (amount) DESC LIMIT 5; | |
|  |  | |
|  | SELECT \* FROM TOP\_5; | |
|  |  | |
|  | DROP VIEW TOP\_5; | |