These are some of the more difficult questions. I decided to map out the tables. All my qyeries are without aliases because I’m not familiar with that yet.

Jeff Olson

**7e. Display the most frequently rented movies in descending order.**

Three Table Join:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table 3: rental** | | **Table 2: inventory** | | **Table 3: film** | |
| **Id** | **type** | **Id** | **type** | **Id** | **type** |
| rental\_id |  | inventory\_id | key | film\_id | key |
| rental\_date |  | film\_id | key | title |  |
| inventory\_id | key | store\_id |  | description |  |
| customer\_id |  | last\_update |  | release\_year |  |
| return\_date |  |  |  | language\_id |  |
| staff\_id |  |  |  | original\_language\_id |  |
| last\_update |  |  |  | rental\_duration |  |
|  |  |  |  | rental\_rate |  |
|  |  |  |  | length |  |
|  |  |  |  | replacement\_cost |  |
|  |  |  |  | rating |  |
|  |  |  |  | special\_features |  |
|  |  |  |  | last\_update |  |

mySQL query:

**SELECT film.title, inventory.inventory\_id, COUNT(rental.inventory\_id)**

**FROM film**

**INNER JOIN inventory ON inventory.film\_id = film.film\_id**

**JOIN rental ON rental.inventory\_id=inventory.inventory\_id**

**GROUP BY film.title**

**ORDER BY COUNT(rental.inventory\_id) DESC;**

**-- 7f. Write a query to display how much business, in dollars, each store brought in.**

Tables to Join:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table 3: staff** | | | **Table 4: payment** | | |
| **Id** | **type** | **Id** | | **type** |
| staff\_id | key | payment\_id | |  |
| first\_name |  | customer\_id | |  |
| last\_name |  | staff\_id | |  |
| address\_id |  | rental\_id | | key |
| picture |  | amount | |  |
| Email |  | payment\_date | |  |
| store\_id | group | last\_update | |  |
| active |  |  | |  |
| username |  |  | |  |
| password |  |  | |  |
| last\_update |  |  | |  |

**Only need tables three and 5 from above**

**SELECT staff.store\_id, sum(payment.amount)**

**FROM staff**

**LEFT JOIN payment ON staff.staff\_id=payment.staff\_id**

**GROUP BY staff.store\_id;**

**ORDER BY SUM(payment.amount) DESC;**

**7g. Write a query to display for each store its store ID, city, and country.**

**Tables to Join:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 2: inventory** | | **Table 2: store** | | **Table 3: city** | | **Table 4: country** | |
| **Id** | **type** | **Id** | **type** | **Id** | **type** | **Id** | **type** |
| inventory\_id | key | store\_id | key | city\_id | key | country\_id | key |
| film\_id | key | manager\_staff\_id |  | City |  | Country |  |
| store\_id |  | address\_id | key | country\_id | key | last\_update |  |
| last\_update |  | last\_update |  | last\_update |  |  |  |

**Only need tables three and 5 from above**

**SELECT staff.store\_id, sum(payment.amount)**

**FROM staff**

**LEFT JOIN payment ON staff.staff\_id=payment.staff\_id**

**GROUP BY staff.store\_id;**

**ORDER BY SUM(payment.amount) DESC;**

**7h. List the top five genres in gross revenue in descending order.**

**-- use the following tables: category, film\_category, inventory, payment, and rental.**

**Tables to Join:**

|  |  |
| --- | --- |
| **Table 1: inventory** | |
| **Id** | **type** |
| inventory\_id | key |
| film\_id | key |
| store\_id |  |

|  |  |  |
| --- | --- | --- |
| **Table 2: payment** | | |
| **Id** | **type** |
| payment\_id |  |
| customer\_id |  |
| staff\_id |  |
| rental\_id | key |
| amount |  |
| payment\_date |  |

|  |  |
| --- | --- |
| **Table 3: rental** | |
| **Id** | **type** |
| rental\_id | key |
| rental\_date |  |
| inventory\_id | key |
| customer\_id |  |
| return\_date |  |
| staff\_id |  |

|  |  |
| --- | --- |
| **Table 4: category** | |
| **Id** | **type** |
| category\_id | key |
| name |  |

|  |  |
| --- | --- |
| **Table 5: film\_category** | |
| **Id** | **type** |
| film\_id |  |
| category\_id | key |

**SELECT category.name, sum(payment.amount)**

**FROM category -- table 4**

**JOIN film\_category ON film\_category.category\_id=category.category\_id -- table 5**

**JOIN inventory ON film\_category.film\_id = inventory.film\_id -- table 1**

**JOIN rental ON rental. inventory\_id = inventory.inventory\_id --table 3**

**JOIN payment ON rental. rental\_id =payment.rental\_id -- table 2**

**GROUP BY category.name -- group by**

**ORDER BY sum(payment.amount) DESC;**