**Query 1: Use INNER JOIN to Display the First and Last Names, as Well as the Address, of Each Staff Member**

**Tables Involved:** staff, address

SELECT

s.first\_name,

s.last\_name,

a.address

FROM

staff s

INNER JOIN

address a ON s.address\_id = a.address\_id;

**/\* Use `INNER JOIN` to display first name, last name and the amount for each staff member.**

**Tables Involved:** **`staff` and `payment\*/**

SELECT

s.first\_name,

s.last\_name,

p.amount

FROM

staff s

INNER JOIN

payment p ON s.staff\_id = p.staff\_id;

**Query 2: Use INNER JOIN to Display the Total Amount Rung Up by Each Staff Member in August of 2005**

**Tables Involved:** staff, payment

SELECT

s.first\_name,

s.last\_name,

SUM(p.amount) AS total\_amount

FROM

staff s

INNER JOIN

payment p ON s.staff\_id = p.staff\_id

WHERE

MONTH(p.payment\_date) = 8

AND YEAR(p.payment\_date) = 2005

GROUP BY

s.staff\_id, s.first\_name, s.last\_name;

**Query 4:**  Use `INNER JOIN` to display first name, last name, payment date and

the amount for each staff member in August of 2005.

**Tables Involved:** staff, payment

SELECT

s.first\_name,

s.last\_name,

p.amount,

p.payment\_date

FROM

staff s

INNER JOIN

payment p ON s.staff\_id = p.staff\_id

WHERE

MONTH(p.payment\_date) = 8

AND YEAR(p.payment\_date) = 2005

**Query 5: List Each Film and the Number of Actors Who Are Listed for That Film**

**Tables Involved:** film\_actor, film

SELECT

f.title,

COUNT(fa.actor\_id) AS number\_of\_actors

FROM

film f

INNER JOIN

film\_actor fa ON f.film\_id = fa.film\_id

GROUP BY

f.film\_id, f.title;

**Query 6: How Many Copies of the Film Hunchback Impossible Exist in the Inventory System?**

**Tables Involved:** inventory, film

SELECT

COUNT(i.inventory\_id) AS number\_of\_copies

FROM

inventory i

INNER JOIN

film f ON i.film\_id = f.film\_id

WHERE

f.title = 'Hunchback Impossible';

**Query 7** Using the tables `payment` and `customer` and the `INNER JOIN` command, list payments by each customer. List the customers alphabetically by last name

**SELECT**

**c.first\_name,**

**c.last\_name,**

**(p.amount) AS total\_paid**

**FROM**

**customer c**

**INNER JOIN**

**payment p ON c.customer\_id = p.customer\_id;**

**Query 8: List the Total Paid by Each Customer, Ordered Alphabetically by Last Name**

**Tables Involved:** payment, customer

SELECT

c.first\_name,

c.last\_name,

SUM(p.amount) AS total\_paid

FROM

customer c

INNER JOIN

payment p ON c.customer\_id = p.customer\_id

GROUP BY

c.customer\_id, c.first\_name, c.last\_name

ORDER BY

c.last\_name, c.first\_name;

**Query 9: Which Actor Has Appeared in the Most Films?**

**Tables Involved:** actor, film\_actor

SELECT

a.first\_name,

a.last\_name,

COUNT(fa.film\_id) AS film\_count

FROM

actor a

INNER JOIN

film\_actor fa ON a.actor\_id = fa.actor\_id

GROUP BY

a.actor\_id, a.first\_name, a.last\_name

ORDER BY

film\_count DESC

LIMIT 1;

**Query 10: Find the Most Common Actor Surname**

**Tables Involved:** actor

SELECT

last\_name,

COUNT(\*) AS frequency

FROM

actor

GROUP BY

last\_name

ORDER BY

frequency DESC

LIMIT 1;

**Query 11: List the Last Names of Actors, as Well as How Many Actors Have That Last Name**

**Tables Involved:** actor

SELECT

last\_name,

COUNT(\*) AS actor\_count

FROM

actor

GROUP BY

last\_name

ORDER BY

actor\_count DESC;

**Query 12: Which Last Names Are Not Repeated?**

**Tables Involved:** actor

SELECT

last\_name

FROM

actor

GROUP BY

last\_name

HAVING

COUNT(\*) = 1;

**Query 13: Which Last Names Appear More Than Once?**

**Tables Involved:** actor

SELECT

last\_name,

COUNT(\*) AS occurrence

FROM

actor

GROUP BY

last\_name

HAVING

COUNT(\*) > 1;

These queries cover a wide range of operations, including joins, aggregation, and filtering, to help students learn and practice SQL with the Sakila database.