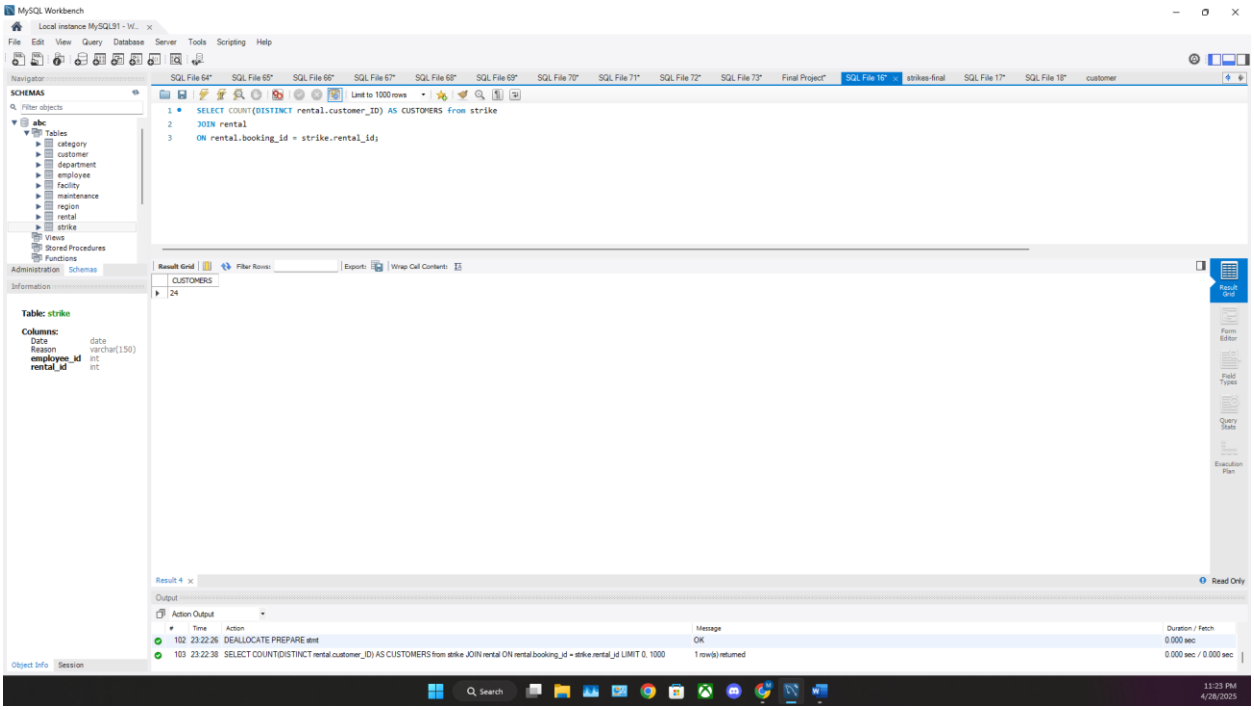
1a. This SQL Statement shows how many customers have received strikes.

SELECT COUNT (DISTINCT rental.customer_ID) AS CUSTOMERS from strike

JOIN rental

ON rental.booking_id = strike.rental_id;



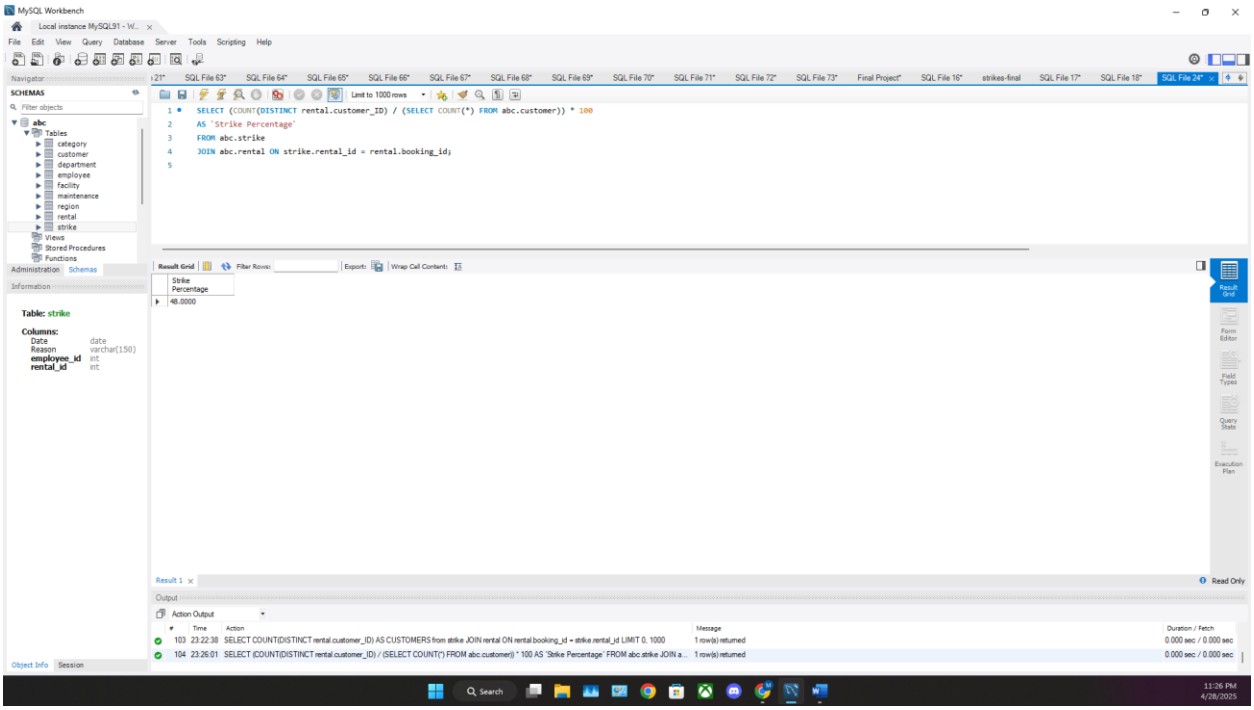
1b. This SQL Statement shows the percentage of customers that have received strikes.

SELECT (COUNT(DISTINCT rental.customer_ID) / (SELECT COUNT(*) FROM abc.customer)) * 100

AS `Strike Percentage`

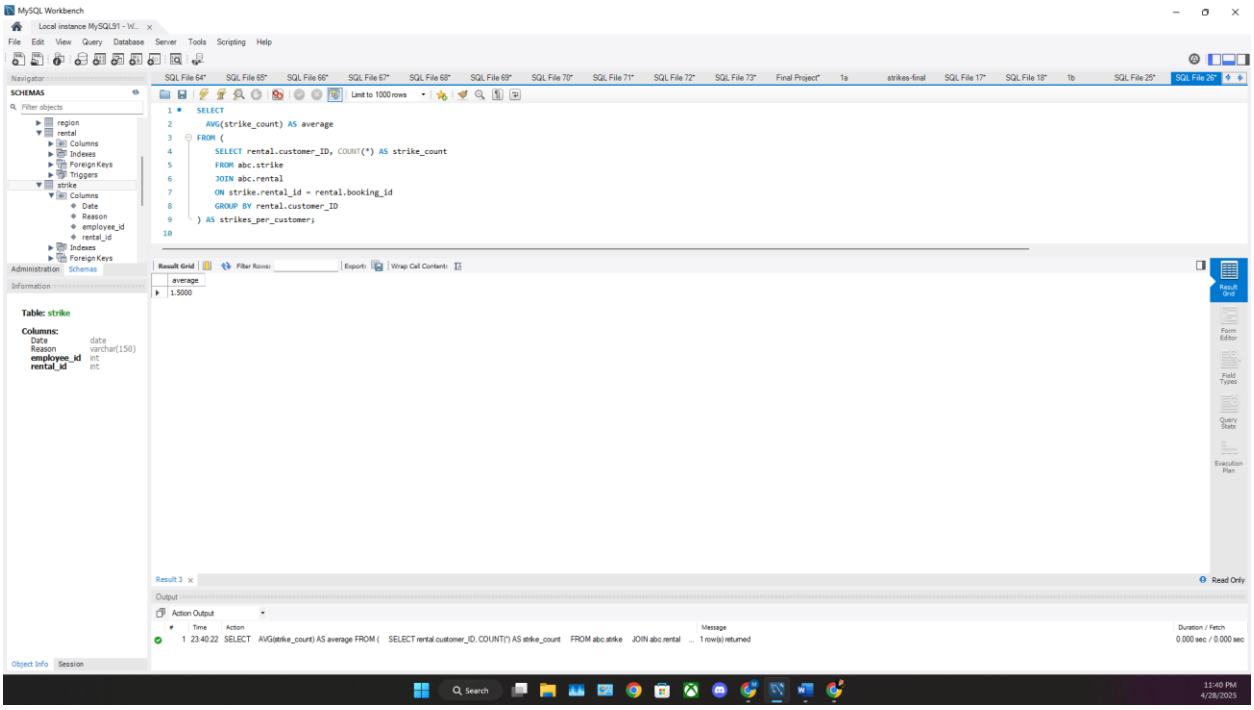
FROM strike

JOIN abc.rental ON strike.rental_id = rental.booking_id;



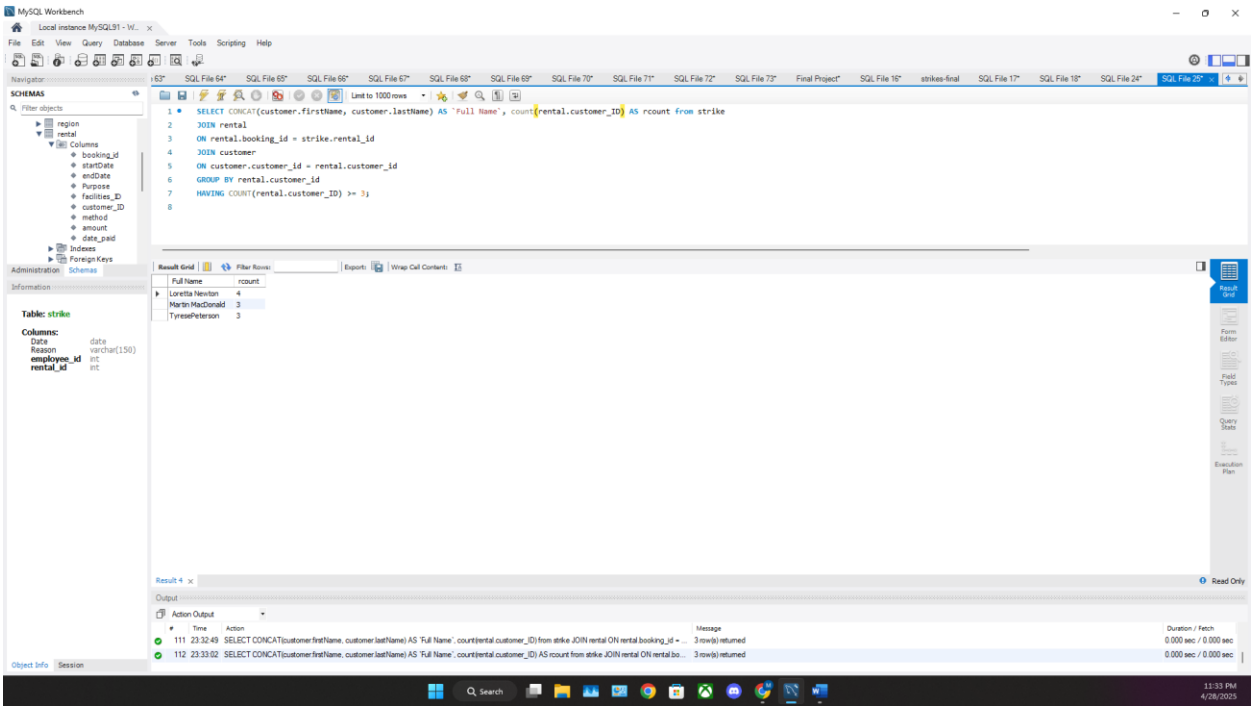
1c. This SQL Statement shows the average number of strikes.

```
SELECT
  AVG(strike_count) AS average
FROM (
    SELECT rental.customer_ID, COUNT(*) AS strike_count
    FROM abc.strike
    JOIN abc.rental
    ON strike.rental_id = rental.booking_id
    GROUP BY rental.customer_ID
) AS strikes_per_customer;
```



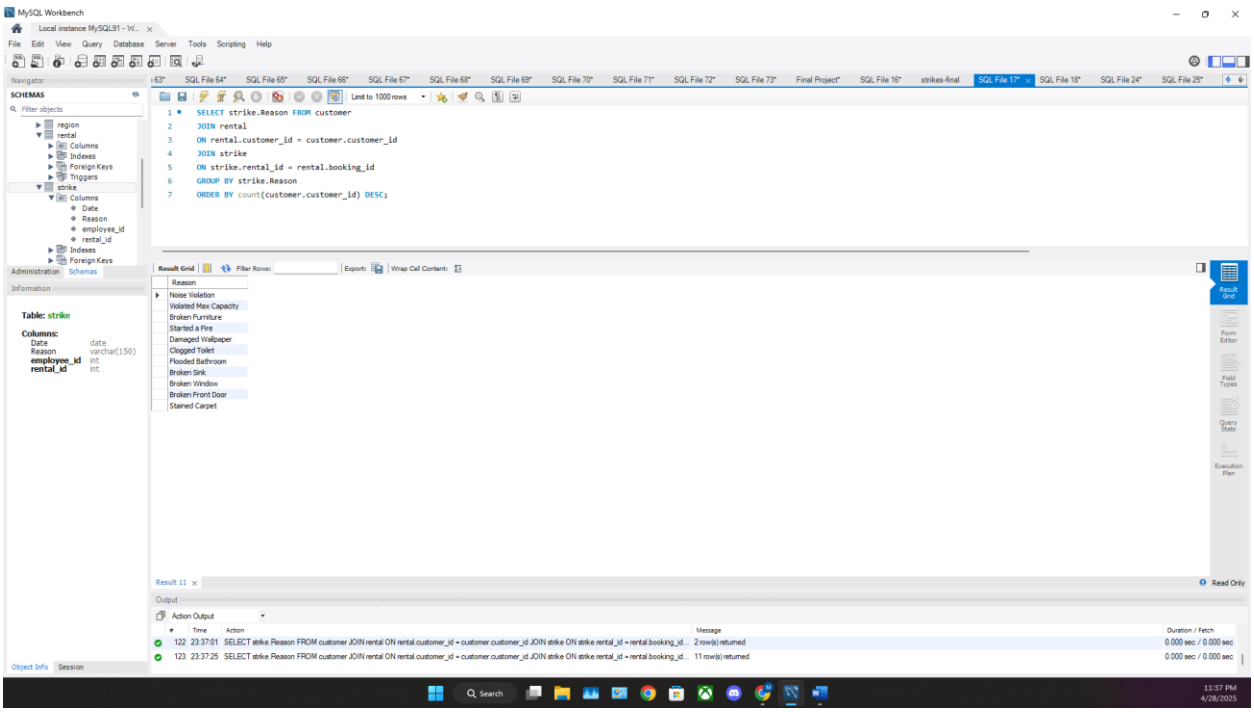
1d. This SQL Statement shows how many customers have 3 or more strikes.

```
SELECT CONCAT(customer.firstName, customer.lastName) AS `Full Name`,
count(rental.customer_ID) AS rcount from strike
JOIN rental
ON rental.booking_id = strike.rental_id
JOIN customer
ON customer.customer_id = rental.customer_id
GROUP BY rental.customer_id
HAVING COUNT(rental.customer_ID) >= 3;
```



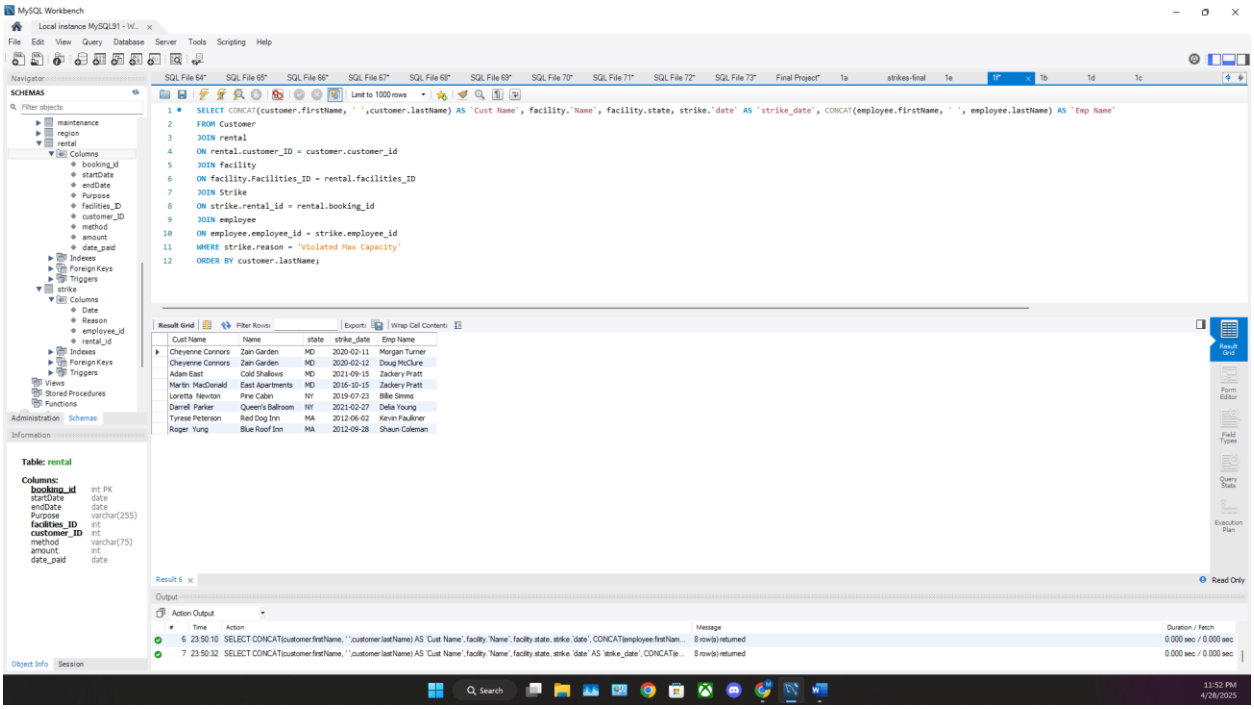
1e. This SQL Statement shows the most common reasons for strikes ordered by most common to least.

```
SELECT strike.Reason FROM customer
JOIN rental
ON rental.customer_id = customer.customer_id
JOIN strike
ON strike.rental_id = rental.booking_id
GROUP BY strike.Reason
ORDER BY count(customer.customer_id) DESC;
```



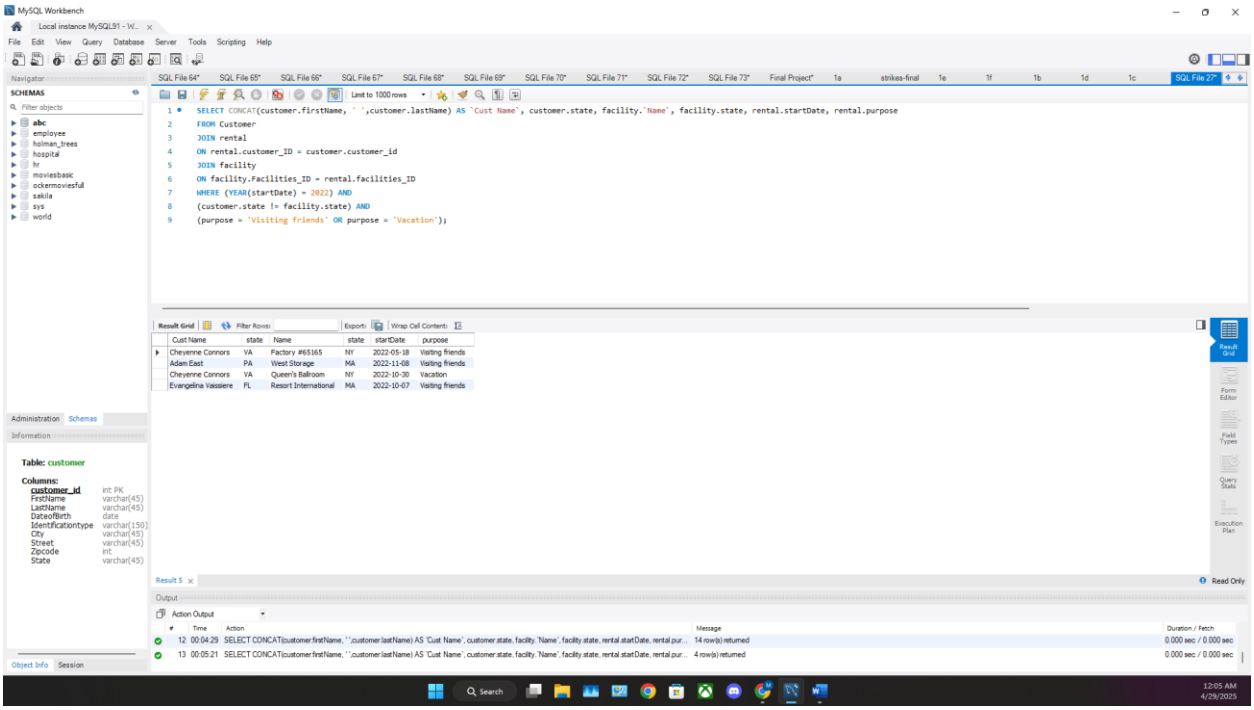
1f. This SQL Statement shows the customer name, facility name, state, strike date, and Employee name where the employee reported Max Capacity, sorted by customer last name

```
SELECT CONCAT(customer.firstName, ' ',customer.lastName) AS `Cust Name`,
facility.`Name`, facility.state, strike.`date` AS `strike_date`, CONCAT(employee.firstName, ' ',
employee.lastName) AS `Emp Name`
FROM Customer
JOIN rental
ON rental.customer_ID = customer.customer_id
JOIN facility
ON facility.Facilities_ID = rental.facilities_ID
JOIN Strike
ON strike.rental_id = rental.booking_id
JOIN employee
ON employee.employee_id = strike.employee_id
WHERE strike.reason = 'Violated Max Capacity'
ORDER BY customer.lastName;
```



1g. This SQL statement shows the Customer name, state, facility name, state, rental start date, and the purpose of renting during the year 2022 where the customer and facility state aren't the same and the purpose of renting is either visiting friends or vacationing.

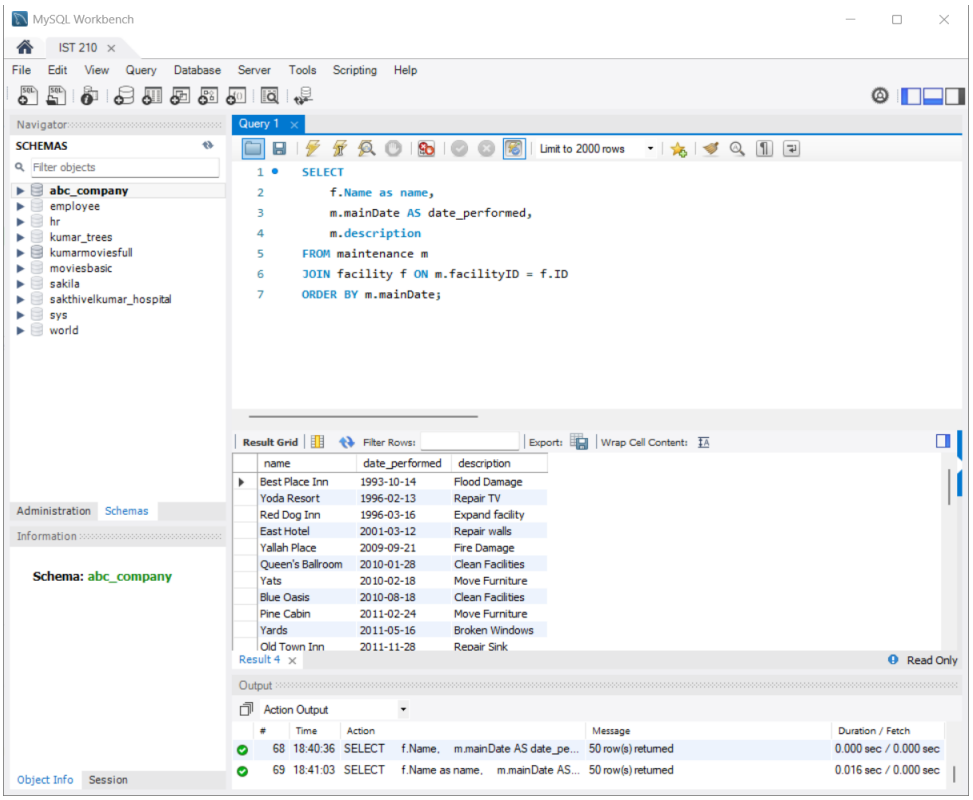
```
SELECT CONCAT(customer.firstName, ' ',customer.lastName) AS `Cust Name`, customer.state, facility.`Name`, facility.state, rental.startDate, rental.purpose
FROM Customer
JOIN rental
ON rental.customer_ID = customer.customer_id
JOIN facility
ON facility.Facilities_ID = rental.facilities_ID
WHERE (YEAR(startDate) = 2022) AND
(customer.state != facility.state) AND
(purpose = 'Visiting friends' OR purpose = 'Vacation');
```



2a.

This query shows all the maintenance performed at the facilities since the beginning of 2020. It shows which facilities (name) required maintenance, what type of maintenance or repair (description) was performed as well as date when each repair was performed.

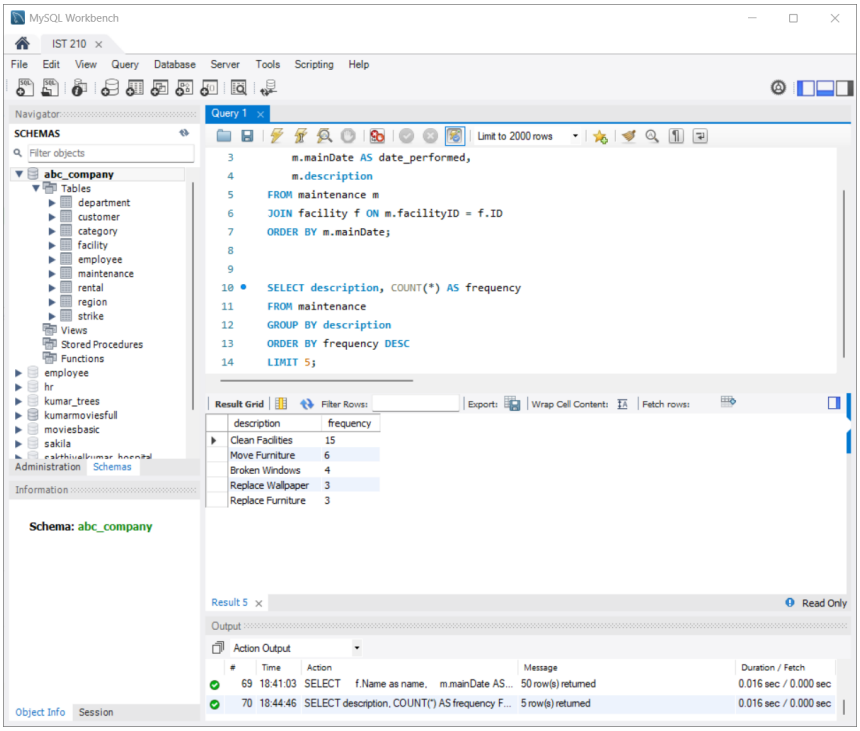
```
SELECT
    f.Name as name,
    m.mainDate AS date_performed,
    m.description
FROM maintenance m
JOIN facility f ON m.facilityID = f.ID
ORDER BY m.mainDate;
```



2b.

This query shows the top five types of maintenace and their frequency. It identifies the most common types of maintenance or repair that are being performed across all facilities as well as the frequency of the maintenance.

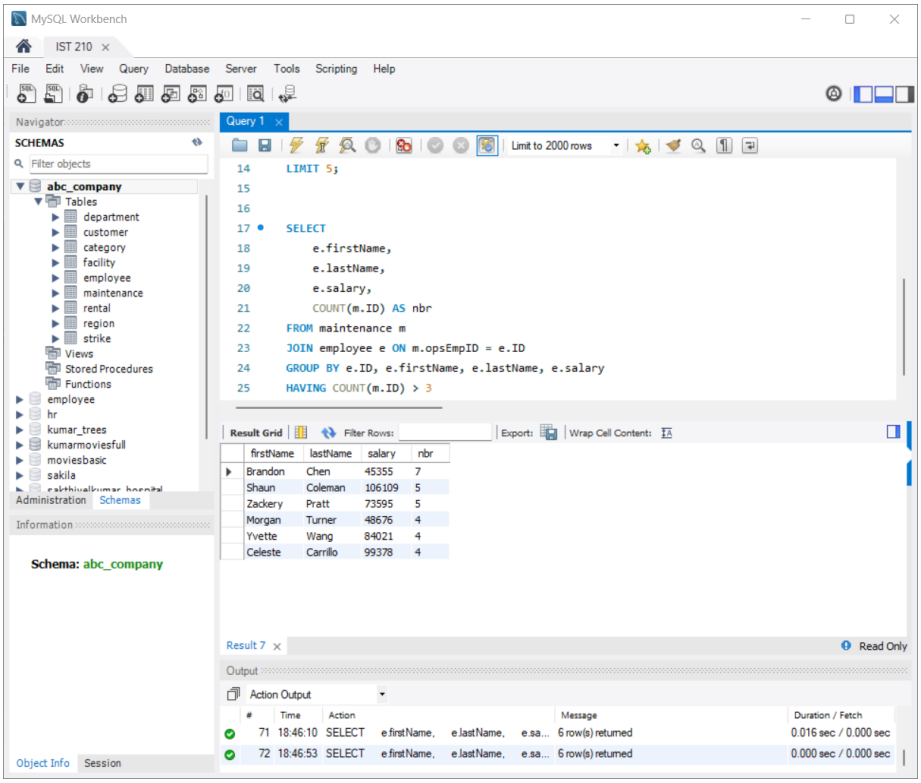
```
SELECT description, COUNT(*) AS frequency
FROM maintenance
GROUP BY description
ORDER BY frequency DESC
LIMIT 5;
```



2 c.

This query shows the employes who have performed more than three maintenance jobs and their salaries. It shows the employees first and last name, salary and the number of times each of them has performed maintenance.

```
SELECT
    e.firstName,
    e.lastName,
    e.salary,
    COUNT(m.ID) AS nbr
FROM maintenance m
JOIN employee e ON m.opsEmpID = e.ID
GROUP BY e.ID, e.firstName, e.lastName, e.salary
HAVING COUNT(m.ID) > 3
ORDER BY nbr DESC;
```



2d.

This query shows the total and average salary of the maintenance employees by year since 2018. It analyzes the salary expenses related to the maintenance employees by year as the results show the total and average salary.

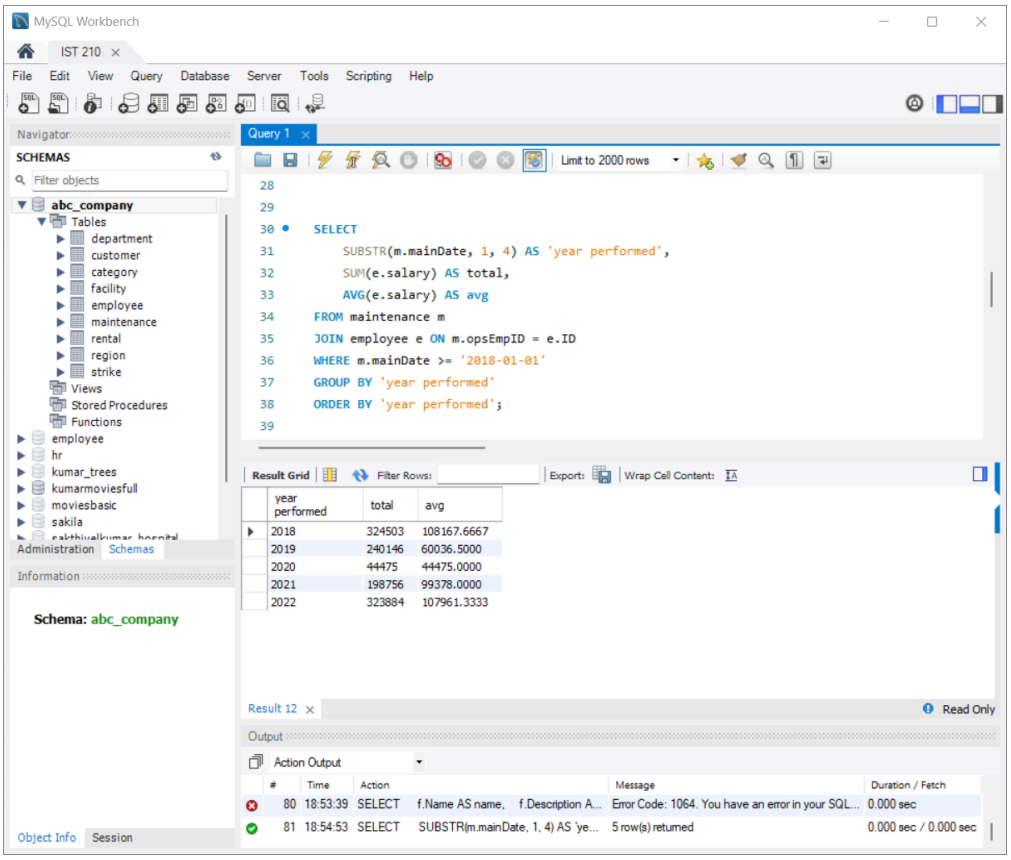
```
SELECT
    (m.mainDate, 1, 4) AS 'year performed',
    SUM(e.salary) AS total,
    AVG(e.salary) AS avg
FROM maintenance m
```

JOIN employee e ON m.opsEmpID = e.ID

WHERE m.mainDate >= '2018-01-01'

GROUP BY 'year performed'

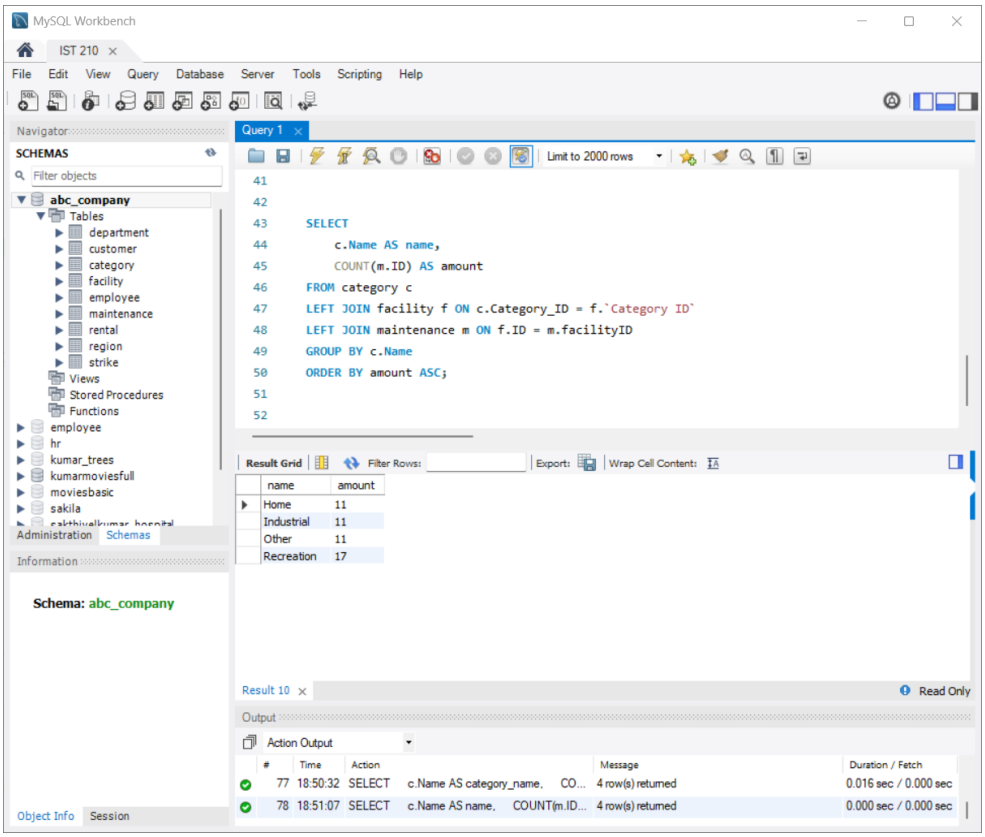
ORDER BY 'year performed';



2e.

This query shows the facility type that recives the least amount maintenance. It shows the type of facility and the number of times it receives maintenace ordered by the amount to see the least amount.

```
SELECT
    c.Name AS name,
    COUNT(m.ID) AS amount
FROM category c
LEFT JOIN facility f ON c.Category_ID = f.`Category ID`
LEFT JOIN maintenance m ON f.ID = m.facilityID
GROUP BY c.Name
ORDER BY amount ASC;
```

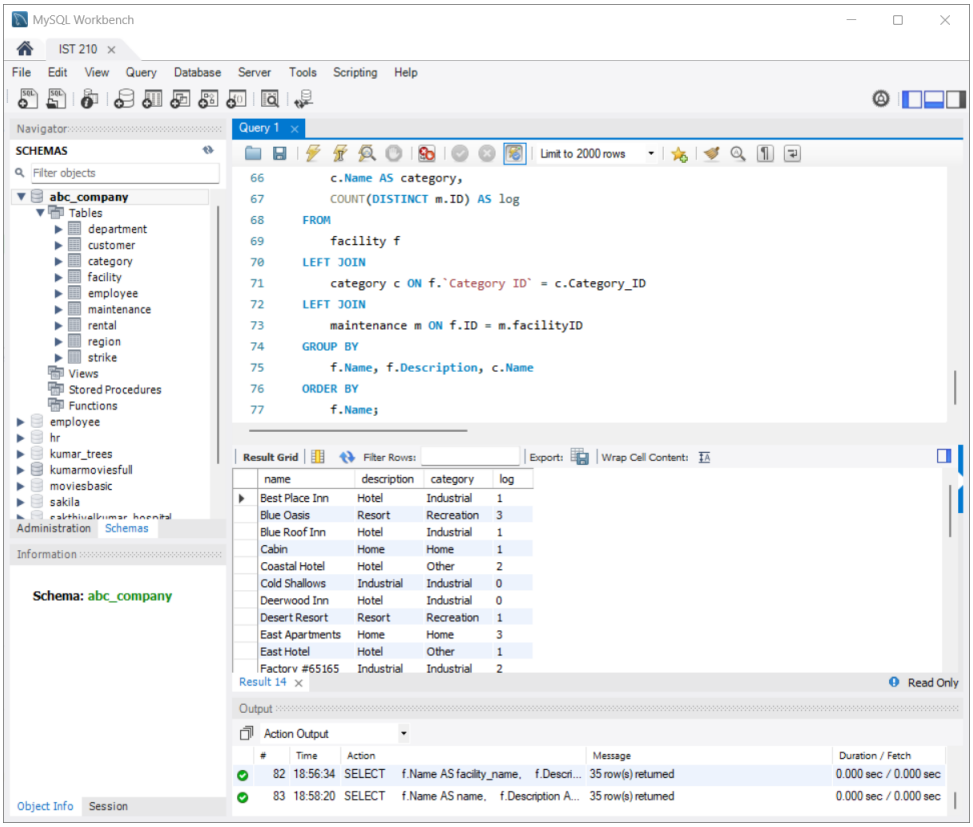


2f.

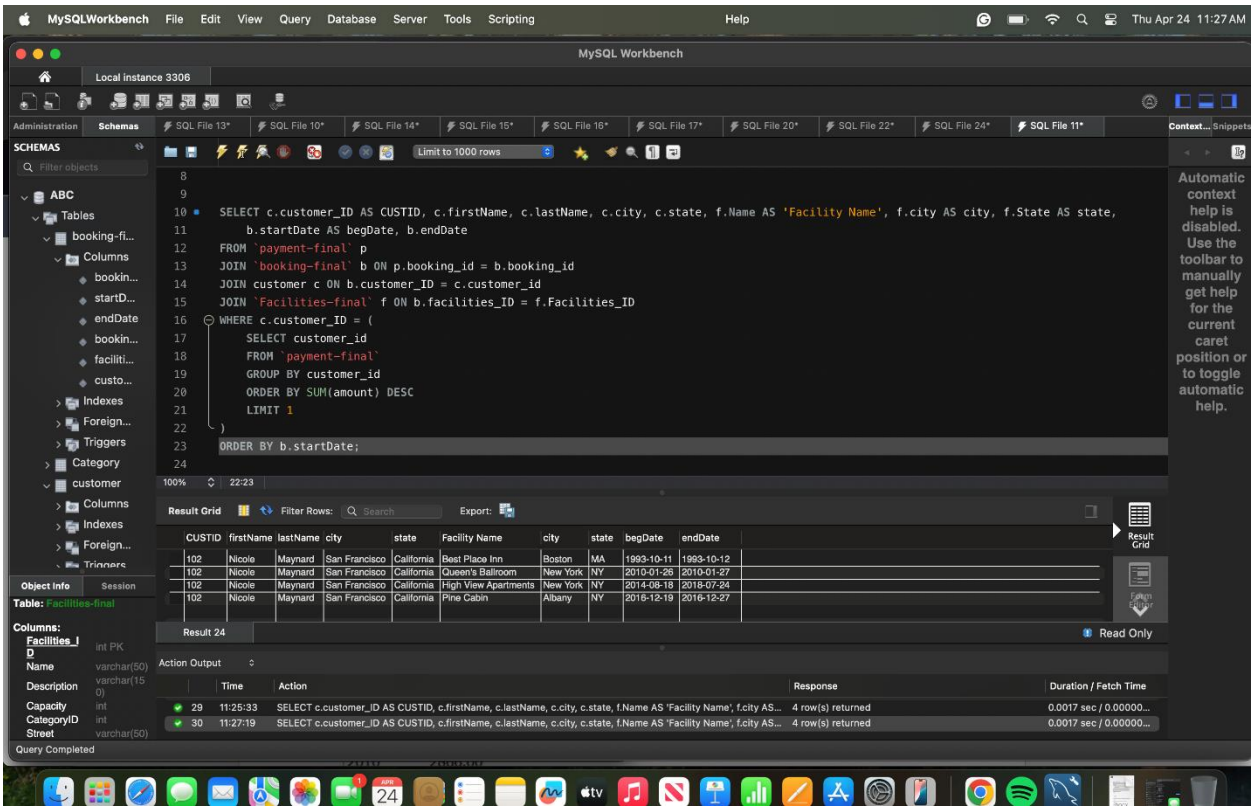
This query shows a comprehensive overview of each facility. It shows the name of the facility, the type, category and number of bookings.

```
SELECT
    f.Name AS name,
    f.Description AS description,
    c.Name AS category,
```

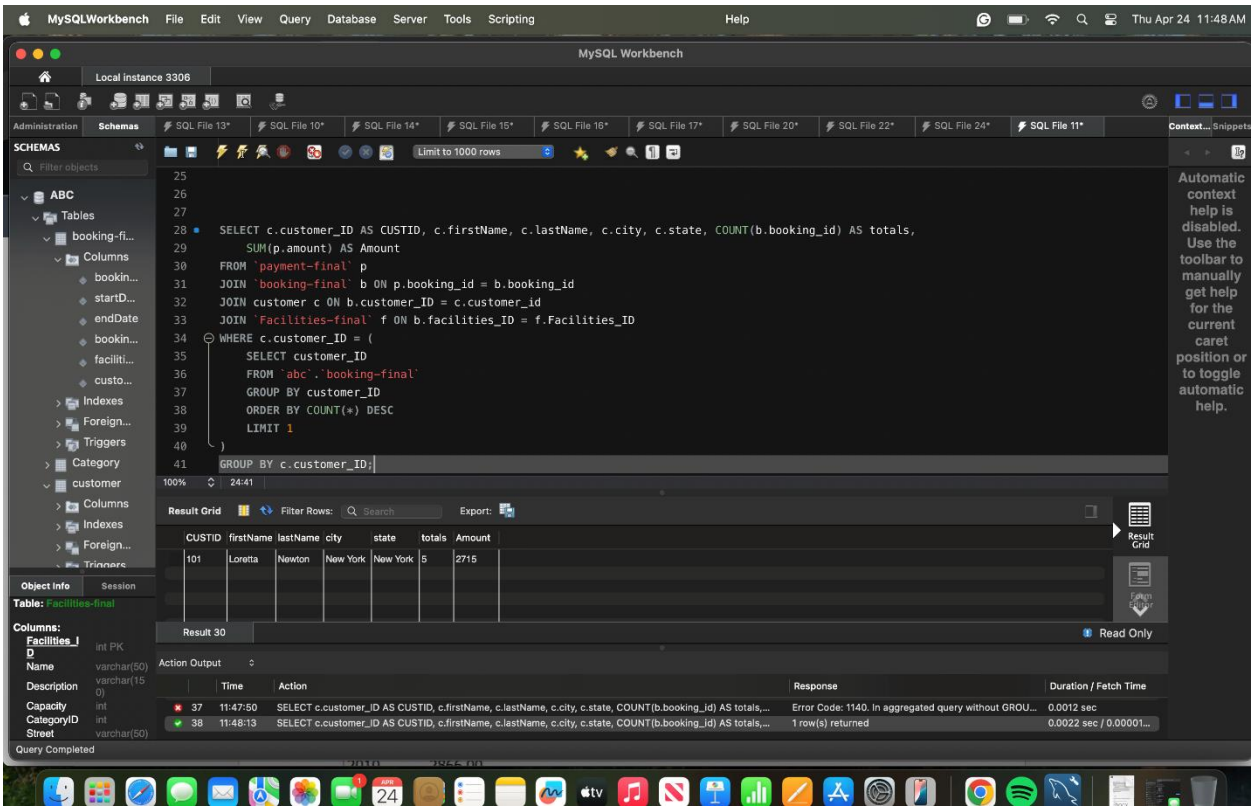
```
    COUNT(DISTINCT m.ID) AS log
FROM
    facility f
LEFT JOIN
    category c ON f.`Category ID` = c.Category_ID
LEFT JOIN
    maintenance m ON f.ID = m.facilityID
GROUP BY
    f.Name, f.Description, c.Name
ORDER BY
    f.Name;
```



3a. This query shows the customer that spent the most in rentals, which was Nicole Maynard. Her properties are located in Boston and New York. This was in 1993, 2010, 2014, and 2016. The facility names are Best Place Inn, Queen's Ballroom, High View Apartments, and Pine Cabin. Code: SELECT c.customer_ID AS CUSTID, c.firstName, c.lastName, c.city, c.state, f.Name AS 'Facility Name', f.city AS city, f.State AS state, b.startDate AS begDate, b.endDate FROM payment-final p JOIN booking-final b ON p.booking_id = b.booking_id JOIN customer c ON b.customer_ID = c.customer_id JOIN Facilities-final f ON b.facilities_ID = f.Facilities_ID WHERE c.customer_ID = (SELECT customer_id FROM payment-final GROUP BY customer_id ORDER BY SUM(amount) DESC LIMIT 1) ORDER BY b.startDate;



3b. The customer who has the most rentals and the amount is Loretta Newton. Code: SELECT c.customer_ID AS CUSTID, c.firstName, c.lastName, c.city, c.state, COUNT(b.booking_id) AS totals, SUM(p.amount) AS Amount FROM payment-final p JOIN booking-final b ON p.booking_id = b.booking_id JOIN customer c ON b.customer_ID = c.customer_id JOIN Facilities-final f ON b.facilities_ID = f.Facilities_ID WHERE c.customer_ID = (SELECT customer_ID FROM abc.booking-final GROUP BY customer_ID ORDER BY COUNT(*) DESC LIMIT 1) GROUP BY c.customer_ID;



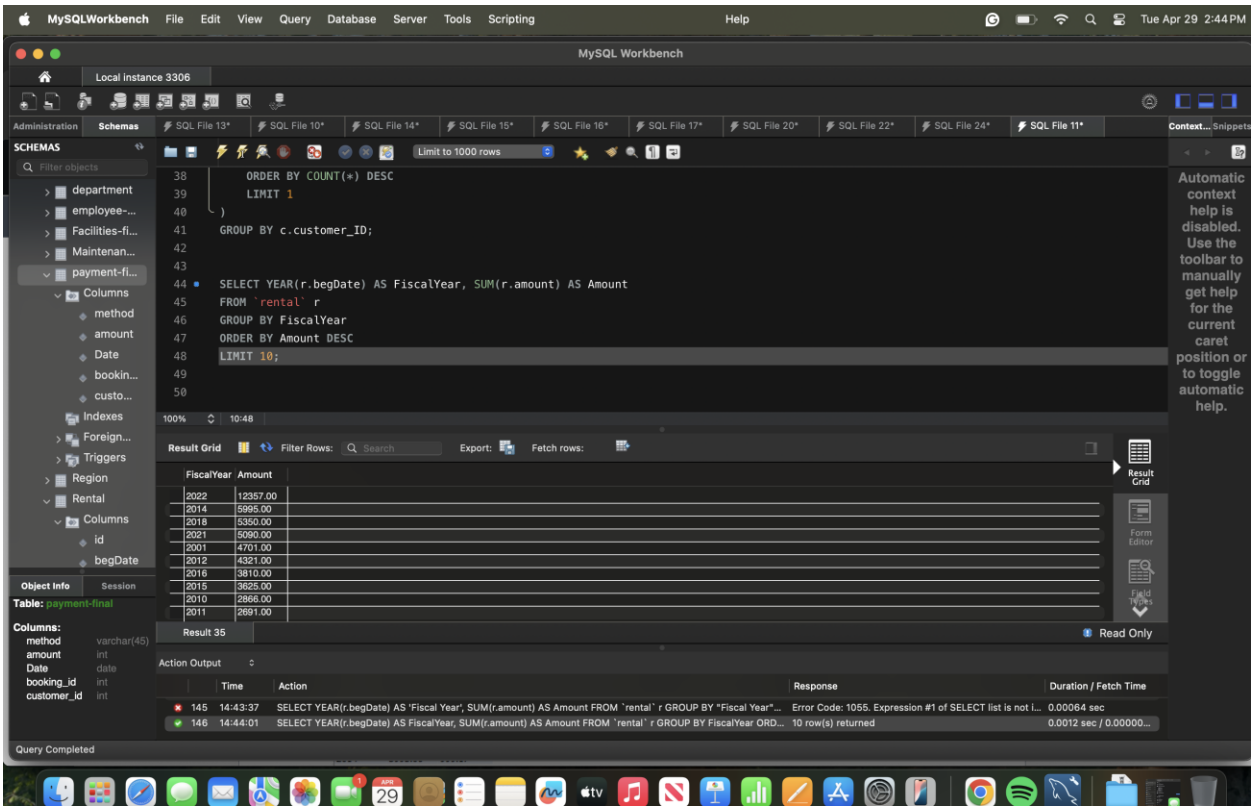
3c. This shows the top 10 fiscal years total earnings, with 2022 being the highest, and 2011 being the lowest. Code: SELECT YEAR(r.begDate) AS FiscalYear, SUM(r.amount) AS Amount

FROM `rental` r

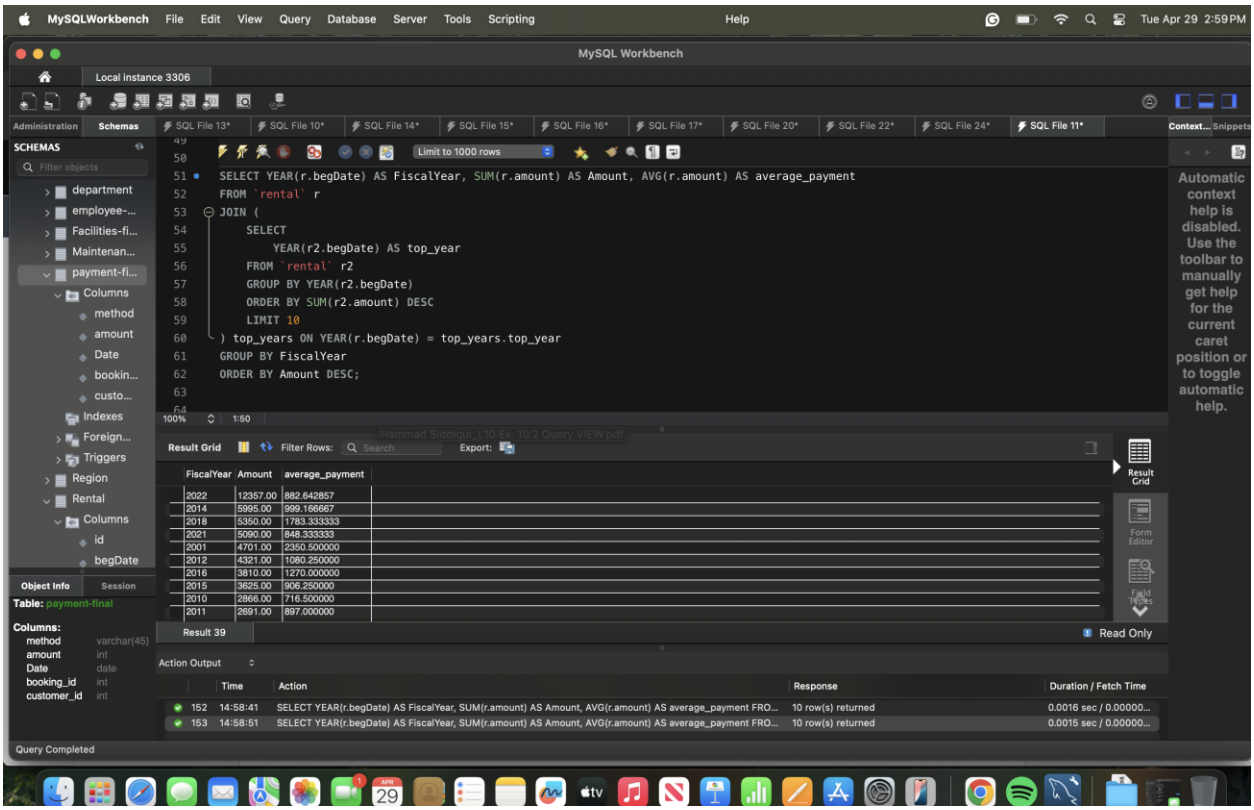
GROUP BY FiscalYear

ORDER BY Amount DESC

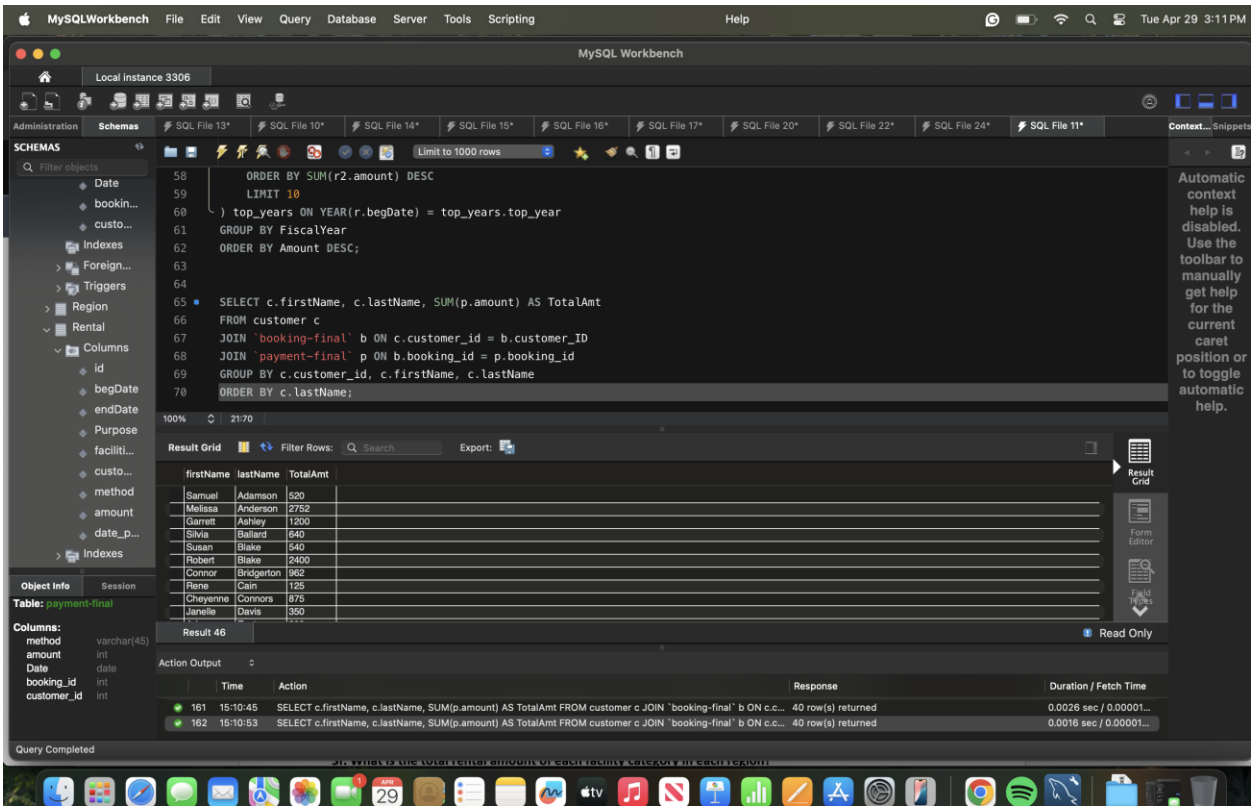
LIMIT 10;



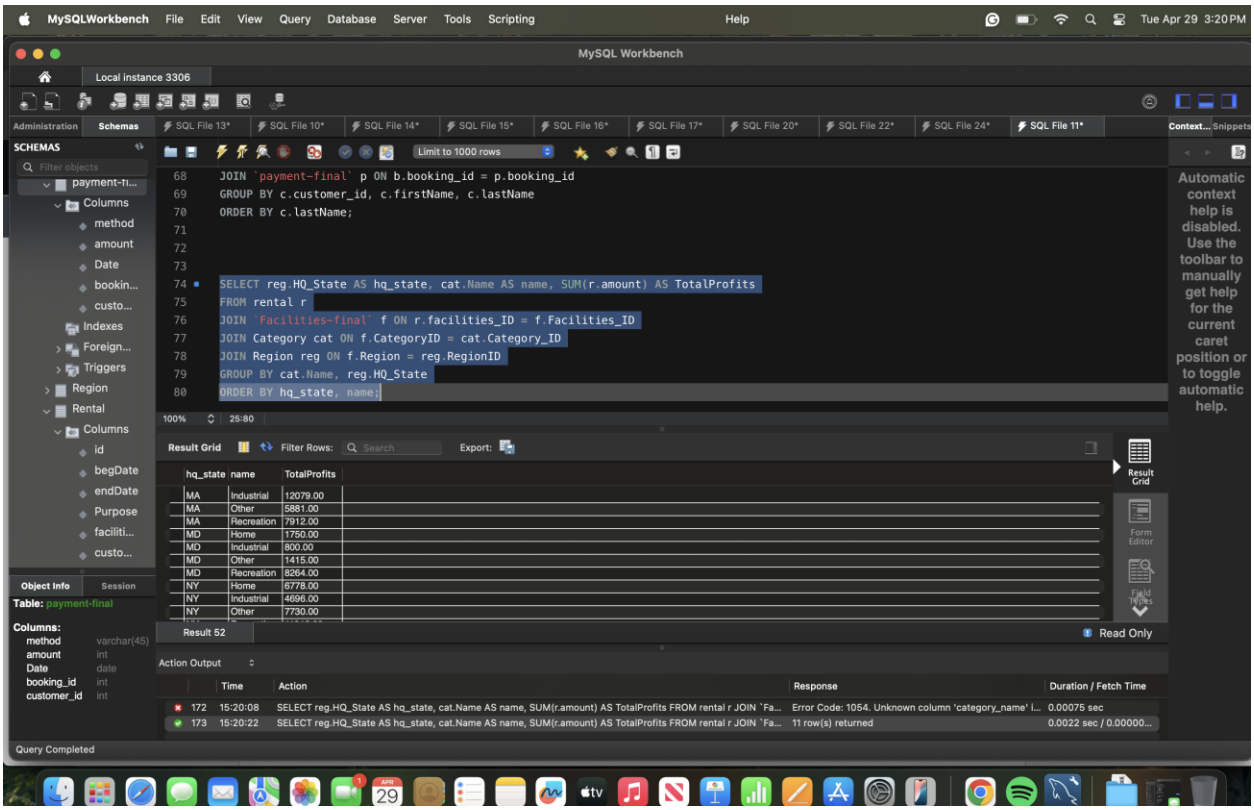
3d. This shows the average payment in those top 10 fiscal years. Code: SELECT YEAR(r.begDate) AS FiscalYear, SUM(r.amount) AS Amount, AVG(r.amount) AS average_payment FROM rental r JOIN (SELECT YEAR(r2.begDate) AS top_year FROM rental r2 GROUP BY YEAR(r2.begDate) ORDER BY SUM(r2.amount) DESC LIMIT 10) top_years ON YEAR(r.begDate) = top_years.top_year GROUP BY FiscalYear ORDER BY Amount DESC;



3e. This shows how much money each customer has spent on bookings. Code: SELECT c.firstName, c.lastName, SUM(p.amount) AS TotalAmt FROM customer c JOIN booking-final b ON c.customer_id = b.customer_ID JOIN payment-final p ON b.booking_id = p.booking_id GROUP BY c.customer_id, c.firstName, c.lastName ORDER BY c.lastName;

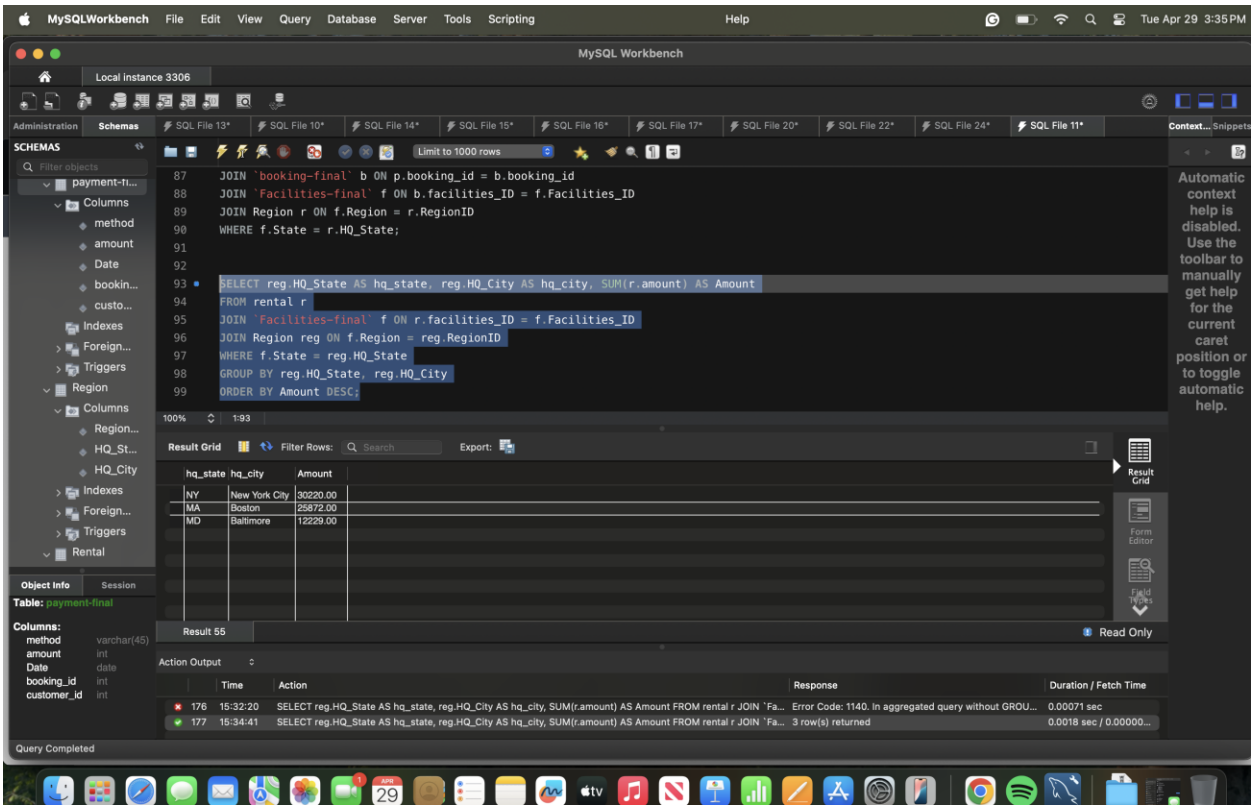


3f. This shows the total rental amount of each facility category in each region, with the industrial facility in MA being the highest. Code: SELECT reg.HQ_State AS hq_state, cat.Name AS name, SUM(r.amount) AS TotalProfits FROM rental r JOIN Facilities-final f ON r.facilities_ID = f.Facilities_ID JOIN Category cat ON f.CategoryID = cat.Category_ID JOIN Region reg ON f.Region = reg.RegionID GROUP BY cat.Name, reg.HQ_State ORDER BY hq_state, name;



3g. This shows the profit of the rental facility where the location is the same as the headquarters.

Code: SELECT reg.HQ_State AS hq_state, reg.HQ_City AS hq_city, SUM(r.amount) AS Amount FROM rental r JOIN Facilities-final f ON r.facilities_ID = f.Facilities_ID JOIN Region reg ON f.Region = reg.RegionID WHERE f.State = reg.HQ_State GROUP BY reg.HQ_State, reg.HQ_City ORDER BY Amount DESC;


MySQLWorkbench File Edit View Query Database Server Tools Scripting Help Tue Apr 29 3:35 PM
MySQL Workbench
Local instance 3306
SQL File 13* SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 17* SQL File 20* SQL File 22* SQL File 24* SQL File 11*
SCHEMAS
Limit to 1000 rows
87 JOIN `booking-final` b ON p.booking_id = b.booking_id
88 JOIN `Facilities-final` f ON b.facilities_ID = f.Facilities_ID
89 JOIN Region r ON f.Region = r.RegionID
90 WHERE f.State = r.HQ_State;
91
92
93 SELECT reg.HQ_State AS hq_state, reg.HQ_City AS hq_city, SUM(r.amount) AS Amount
94 FROM rental r
95 JOIN `Facilities-final` f ON r.facilities_ID = f.Facilities_ID
96 JOIN Region reg ON f.Region = reg.RegionID
97 WHERE f.State = reg.HQ_State
98 GROUP BY reg.HQ_State, reg.HQ_City
99 ORDER BY Amount DESC;
Result Grid Filter Rows: Search Export:
hq_state hq_city Amount
NY New York City 30220.00
MA Boston 25872.00
MD Baltimore 12229.00
Result 55 Read Only
Action Output
Time Action Response Duration / Fetch Time
176 15:32:20 SELECT reg.HQ_State AS hq_state, reg.HQ_City AS hq_city, SUM(r.amount) AS Amount FROM rental r JOIN `Fa... Error Code: 1140. In aggregated query without GROU... 0.00071 sec
177 15:34:41 SELECT reg.HQ_State AS hq_state, reg.HQ_City AS hq_city, SUM(r.amount) AS Amount FROM rental r JOIN `Fa... 3 row(s) returned 0.0018 sec / 0.00000...
Query Completed
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.