

CS 350 Final Project Output of Queries, triggers, stored procedures, and user updates

Matthew Dembny & Nick Koeppen

1) determines the average payment made to employees.

The screenshot shows the MySQL Workbench interface with the SQL editor tab open. The code for query #1 is pasted into the editor:

```
1 # 1) determines the average payment made to employees.
2 • SELECT AVG(payment.PaymentAmount) AS 'AveragePayment' FROM payroll.payment;
3
4 # 2) Determines the number of employees on payroll.
5 • SELECT COUNT(employee.EmployeeID) AS 'NumberOfEmployees' FROM payroll.employee;
6
7 # 3) Determines the number of employees in the student life department.
8 • SELECT COUNT(employee.EmployeeID) AS 'StudentLifeEmployees'
9 FROM employee JOIN job ON employee.JobID = job.JobID JOIN department ON department.DeptID = job.DeptID
10 WHERE department.DeptID = 'SL';
11
12 # 4) Determines the number of expenses that the Teaching Faculty group incures.
13 • SELECT SUM(payment.PaymentAmount) AS 'TTEExpenses'
14 FROM payment JOIN directdeposit ON payment.DirectDepositID = directdeposit.DirectDepositID
15 JOIN employee ON employee.EmployeeID = directdeposit.EmployeeID
```

The results grid shows one row with the value 1637.0000 under the column labeled "AveragePayment".

The output pane shows the execution log with three entries:

#	Time	Action	Message	Duration / Fetch
1	11:57:17	SELECT COUNT(employee.EmployeeID) AS 'NumEmployees', department.Name FROM employee JOIN job ON ...	5 row(s) returned	0.032 sec / 0.000 sec
2	11:58:07	SELECT (@time *job.HourlyRate) AS 'Amount Paid' FROM(SELECT @time := SUM(\$timeworked.Hours) AS Tot...)	1 row(s) returned	0.016 sec / 0.000 sec
3	12:02:51	SELECT AVG(payment.PaymentAmount) AS 'AveragePayment' FROM payroll.payment LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec

2) Determines the number of employees on payroll.

The screenshot shows the MySQL Workbench interface with the SQL editor tab open. The code for query #2 is pasted into the editor:

```
1 # 1) determines the average payment made to employees.
2 • SELECT AVG(payment.PaymentAmount) AS 'AveragePayment' FROM payroll.payment;
3
4 # 2) Determines the number of employees on payroll.
5 • SELECT COUNT(employee.EmployeeID) AS 'NumberOfEmployees' FROM payroll.employee;
6
7 # 3) Determines the number of employees in the student life department.
8 • SELECT COUNT(employee.EmployeeID) AS 'StudentLifeEmployees'
9 FROM employee JOIN job ON employee.JobID = job.JobID JOIN department ON department.DeptID = job.DeptID
10 WHERE department.DeptID = 'SL';
11
12 # 4) Determines the number of expenses that the Teaching Faculty group incures.
13 • SELECT SUM(payment.PaymentAmount) AS 'TTEExpenses'
14 FROM payment JOIN directdeposit ON payment.DirectDepositID = directdeposit.DirectDepositID
15 JOIN employee ON employee.EmployeeID = directdeposit.EmployeeID
```

The results grid shows one row with the value 8 under the column labeled "NumberOfEmployees".

The output pane shows the execution log with four entries:

#	Time	Action	Message	Duration / Fetch
1	11:57:17	SELECT COUNT(employee.EmployeeID) AS 'NumEmployees', department.Name FROM employee JOIN job ON ...	5 row(s) returned	0.032 sec / 0.000 sec
2	11:58:07	SELECT (@time *job.HourlyRate) AS 'Amount Paid' FROM(SELECT @time := SUM(\$timeworked.Hours) AS Tot...)	1 row(s) returned	0.016 sec / 0.000 sec
3	12:02:51	SELECT AVG(payment.PaymentAmount) AS 'AveragePayment' FROM payroll.payment LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
4	12:03:42	SELECT COUNT(employee.EmployeeID) AS 'NumberOfEmployees' FROM payroll.employee LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

3) Determines the number of employees in the student life department.

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** payroll
- Tables:** benefits, department, directdeposit, employee, job, payment, payperiods, timeworked
- Query:**

```

1 # 1) determines the average payment made to employees.
2 • SELECT AVG(payment.PaymentAmount) AS 'AveragePayment' FROM payroll.payment;
3
4 # 2) Determines the number of employees on payroll.
5 • SELECT COUNT(employee.EmployeeID) AS 'NumberOfEmployees' FROM payroll.employee;
6
7 # 3) Determines the number of employees in the student life department.
8 • SELECT COUNT(employee.EmployeeID) AS 'StudentLifeEmployees'
9 FROM employee JOIN job ON employee.JobID = job.JobID JOIN department ON department.DeptID = job.DeptID
10 WHERE department.DeptID = 'SL';
11
12 # 4) Determines the number of expenses that the Teaching Faculty group incures.
13 • SELECT SUM(payment.PaymentAmount) AS 'TFEExpenses'
14 FROM payment JOIN directdeposit ON payment.DirectDepositID = directdeposit.DirectDepositID
15 JOIN employee ON employee.EmployeeID = directdeposit.EmployeeID
16 JOIN job ON employee.JobID = job.JobID
17 JOIN department ON job.DeptID = department.DeptID
18 WHERE department.DeptID = 'TF';
19
20 # 5) Determines all of the teaching faculty with a Health Insurance tier of Silver
21 • SELECT employee.Firstname, employee.Lastname

```
- Result Grid:** StudentLifeEmployees | 1
- Action Output:**

#	Time	Action	Message	Duration / Fetch
3	12:02:51	SELECT AVG(payment.PaymentAmount) AS 'AveragePayment' FROM payroll.payment;	1 row(s) returned	0.015 sec / 0.000 sec
4	12:03:42	SELECT COUNT(employee.EmployeeID) AS 'NumberOfEmployees' FROM payroll.employee;	1 row(s) returned	0.000 sec / 0.000 sec
5	12:04:12	SELECT COUNT(employee.EmployeeID) AS 'StudentLifeEmployees' FROM employee JOIN job ON employee...	1 row(s) returned	0.000 sec / 0.000 sec

4) Determines the number of expenses that the Teaching Faculty group incures.

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** payroll
- Tables:** benefits, department, directdeposit, employee, job, payment, payperiods, timeworked
- Query:**

```

7 # 3) Determines the number of employees in the student life department.
8 • SELECT COUNT(employee.EmployeeID) AS 'StudentLifeEmployees'
9 FROM employee JOIN job ON employee.JobID = job.JobID JOIN department ON department.DeptID = job.DeptID
10 WHERE department.DeptID = 'SL';
11
12 # 4) Determines the number of expenses that the Teaching Faculty group incures.
13 • SELECT SUM(payment.PaymentAmount) AS 'TFEExpenses'
14 FROM payment JOIN directdeposit ON payment.DirectDepositID = directdeposit.DirectDepositID
15 JOIN employee ON employee.EmployeeID = directdeposit.EmployeeID
16 JOIN job ON employee.JobID = job.JobID
17 JOIN department ON job.DeptID = department.DeptID
18 WHERE department.DeptID = 'TF';
19
20 # 5) Determines all of the teaching faculty with a Health Insurance tier of Silver
21 • SELECT employee.Firstname, employee.Lastname

```
- Result Grid:** TFEExpenses | 4203
- Action Output:**

#	Time	Action	Message	Duration / Fetch
4	12:03:42	SELECT COUNT(employee.EmployeeID) AS 'NumberOfEmployees' FROM payroll.employee LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	12:04:12	SELECT COUNT(employee.EmployeeID) AS 'StudentLifeEmployees' FROM employee JOIN job ON employee...	1 row(s) returned	0.000 sec / 0.000 sec
6	12:04:38	SELECT SUM(payment.PaymentAmount) AS 'TFEExpenses' FROM payment JOIN directdeposit ON payment.Dr...	1 row(s) returned	0.015 sec / 0.000 sec

5) Determines all of the teaching faculty with a Health Insurance tier of Silver

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, MySQL Model (CS-350 Final Proj..), EER Diagram.
- Navigator:** Schemas (bits, ch9qa, colonial, music, payroll), Tables (benefits, department, directdeposit, employee, job, payment, payperiods, timeworked), Views, Stored Procedures, Functions, Administration, Schemas.
- SQL Editor:** Final Project Queries tab, contains the following SQL code:


```

16 JOIN job ON employee.JobID = job.JobID
17 JOIN department ON job.DeptID = department.DeptID
18 WHERE department.DeptID = 'TF';
19
20 # 5) Determines all of the teaching faculty with a Health Insurance tier of Silver
21 • SELECT employee.FirstName, employee.LastName
22 FROM employee JOIN job ON employee.JobID = job.JobID
23 JOIN benefits ON benefits.BenefitsID = job.BenefitsID
24 JOIN department ON job.DeptID = department.DeptID
25 WHERE benefits.HealthInsuranceTier = 'Silver' AND department.Name = 'Teaching Faculty';
26
27 # 6) Determines the total number of hours that employee "Nicholas Rosasco" worked before the month of October.
28 • SELECT SUM(timeworked.Hours) AS 'TotalHours'
29 FROM employee JOIN timeworked ON employee.EmployeeID = timeworked.EmployeeID
30 JOIN payperiods ON timeworked.PayPeriodID = payperiods.PayPeriodID
      
```
- Result Grid:** Shows the results for the first query:

FirstName	LastName
Nicholas	Rosasco
Timothy	Malchow
- Action Output:** Shows the execution history:

#	Time	Action	Message	Duration / Fetch
5	12:04:12	SELECT COUNT(employee.EmployeeID) AS 'StudentLifeEmployees' FROM employee JOIN job ON employee.JobID = job.JobID	1 row(s) returned	0.000 sec / 0.000 sec
6	12:04:38	SELECT SUM(payment.PaymentAmount) AS 'TTEExpenses' FROM payment JOIN directdeposit ON payment.DirectDepositID = directdeposit.DirectDepositID	1 row(s) returned	0.015 sec / 0.000 sec
7	12:05:11	SELECT employee.FirstName, employee.LastName FROM employee JOIN job ON employee.JobID = job.JobID	2 row(s) returned	0.000 sec / 0.000 sec

6) Determines the total number of hours that employee "Nicholas Rosasco" worked before the month of October.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, MySQL Model (CS-350 Final Proj..), EER Diagram.
- Navigator:** Schemas (bits, ch9qa, colonial, music, payroll), Tables (benefits, department, directdeposit, employee, job, payment, payperiods, timeworked), Views, Stored Procedures, Functions, Administration, Schemas.
- SQL Editor:** Final Project Queries tab, contains the following SQL code:


```

25 WHERE benefits.HealthInsuranceTier = 'Silver' AND department.Name = 'Teaching Faculty';
26
27 # 6) Determines the total number of hours that employee "Nicholas Rosasco" worked before the month of October.
28 • SELECT SUM(timeworked.Hours) AS 'TotalHours'
29 FROM employee JOIN timeworked ON employee.EmployeeID = timeworked.EmployeeID
30 JOIN payperiods ON timeworked.PayPeriodID = payperiods.PayPeriodID
31 WHERE employee.EmployeeID = 1 AND payperiods.EndDate < '2023-10-1';
32
33 # 7) Returns the number of employees in each department with an employee on payroll.
34 • SELECT COUNT(employee.EmployeeID) AS 'NumEmployees', department.Name
35 FROM employee JOIN job ON employee.JobID = job.JobID
36 JOIN benefits ON benefits.BenefitsID = job.BenefitsID
37 JOIN department ON job.DeptID = department.DeptID
38 GROUP BY department.DeptID
39 HAVING COUNT(employee.EmployeeID) > 0
      
```
- Result Grid:** Shows the results for the second query:

TotalHours
160
- Action Output:** Shows the execution history:

#	Time	Action	Message	Duration / Fetch
6	12:04:38	SELECT SUM(payment.PaymentAmount) AS 'TTEExpenses' FROM payment JOIN directdeposit ON payment.DirectDepositID = directdeposit.DirectDepositID	1 row(s) returned	0.015 sec / 0.000 sec
7	12:04:51	SELECT employee.FirstName, employee.LastName FROM employee JOIN job ON employee.JobID = job.JobID	2 row(s) returned	0.000 sec / 0.000 sec
8	12:05:42	SELECT SUM(timeworked.Hours) AS 'TotalHours' FROM employee JOIN timeworked ON employee.EmployeeID = 1	1 row(s) returned	0.000 sec / 0.000 sec

7) Returns the number of employees in each department with an employee on payroll.

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Final Project Queries* | job payperiods timeworked INSERT INTO FINAL PROJECT Create Final Project
SCHEMAS
Filter objects
bita chga colonial music payroll
Tables
department directdeposit employee job payment payperiods timeworked
Views
Stored Procedures Functions Administration Schemas
Information
No object selected
Result Grid Filter Rows Export Wrap Cell Content
Result 10 x
Output
Action Output
# Time Action Message Duration / Fetch
8 12:05:42 SELECT SUM(timeworked.Hours) AS 'TotalHours' FROM employee JOIN timeworked ON employee.EmployeeID = timeworked.EmployeeID 1 row(s) returned 0.000 sec / 0.000 sec
9 12:06:07 SELECT COUNT(employee.EmployeeID) AS 'NumEmployees', department.Name FROM employee JOIN job ON employee.JobID = job.JobID JOIN department ON job.DepartmentID = department.DepartmentID GROUP BY department.DepartmentID HAVING COUNT(employee.EmployeeID) > 0 5 row(s) returned 0.000 sec / 0.000 sec
10 12:06:13 ORDER BY COUNT(employee.EmployeeID);
Object Info Session
Query Completed
12:06 PM 11/17/2023

```

NumEmployees	Name
1	Student Life
1	Information Technology
1	Presidents Office
2	Facilities Management
3	Teaching Faculty

8) Returns the employee's name, job title, and benefits information where their HSA benefit is greater than 1000, ordered by HSA amount.

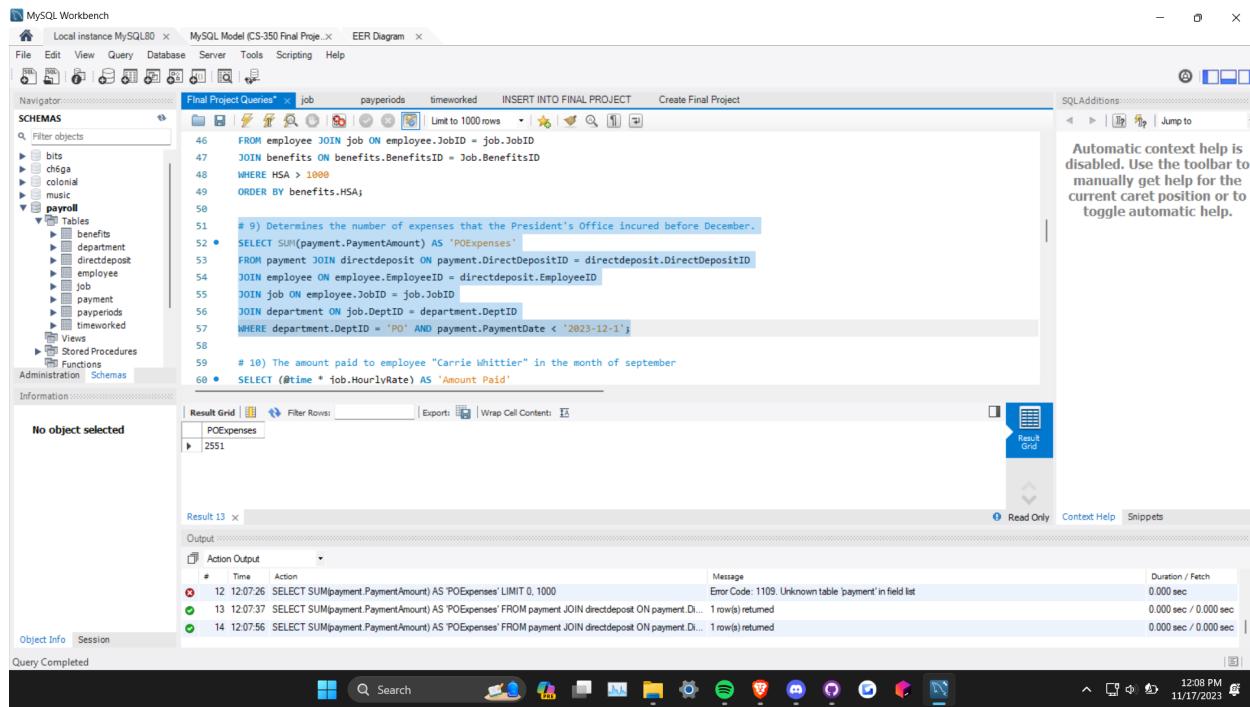
```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Final Project Queries* | job payperiods timeworked INSERT INTO FINAL PROJECT Create Final Project
SCHEMAS
Filter objects
bita chga colonial music payroll
Tables
department directdeposit employee job payment payperiods timeworked
Views
Stored Procedures Functions Administration Schemas
Information
No object selected
Result Grid Filter Rows Export Wrap Cell Content
Result 11 x
Output
Action Output
# Time Action Message Duration / Fetch
9 12:06:07 SELECT SUM(timeworked.Hours) AS 'TotalHours' FROM employee JOIN timeworked ON employee.EmployeeID = timeworked.EmployeeID 1 row(s) returned 0.000 sec / 0.000 sec
10 12:06:13 SELECT COUNT(employee.EmployeeID) AS 'NumEmployees', department.Name FROM employee JOIN job ON employee.JobID = job.JobID JOIN department ON job.DepartmentID = department.DepartmentID GROUP BY department.DepartmentID HAVING COUNT(employee.EmployeeID) > 0 5 row(s) returned 0.000 sec / 0.000 sec
11 12:06:51 ORDER BY benefits.HSA;
Object Info Session
Query Completed
12:07 PM 11/17/2023

```

FirstName	LastName	Title	HSA	HealthInsuranceTier
Carrie	Whitner	Director of Student Life	2000	Gold
Gregg	Johnson	Dean	4000	Gold
Jose	Padilla	President	4000	Platinum

9) Determines the number of expenses that the President's Office incurred before December.



The screenshot shows the MySQL Workbench interface with the SQL editor tab open. The query window contains the following code:

```
FROM employee JOIN job ON employee.JobID = job.JobID
JOIN benefits ON benefits.BenefitsID = Job.BenefitsID
WHERE HSA > 1000
ORDER BY benefits.HSA
```

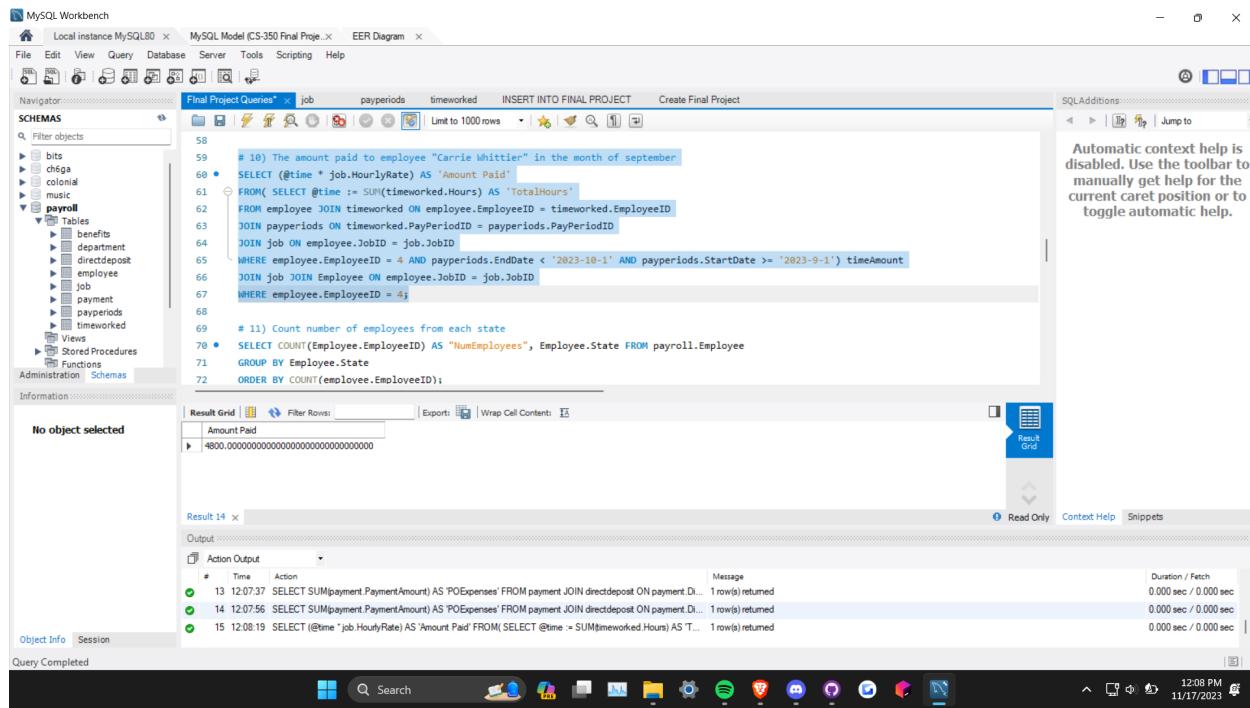
Below this, another section of the query is highlighted:

```
# 9) Determines the number of expenses that the President's Office incurred before December.
SELECT SUM(payment.PaymentAmount) AS 'POExpenses'
FROM payment JOIN directdeposit ON payment.DirectDepositID = directdeposit.DirectDepositID
JOIN employee ON employee.EmployeeID = directdeposit.EmployeeID
JOIN job ON employee.JobID = job.JobID
JOIN department ON job.DeptID = department.DeptID
WHERE department.DeptID = 'PO' AND payment.PaymentDate < '2023-12-1';
```

The result grid shows a single row with the value 2551. The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
12	12:07:26	SELECT SUM(payment.PaymentAmount) AS 'POExpenses' LIMIT 0, 1000	Error Code: 1109. Unknown table 'payment' in field list	0.000 sec
13	12:07:37	SELECT SUM(payment.PaymentAmount) AS 'POExpenses' FROM payment JOIN directdeposit ON payment.D... 1 row(s) returned		0.000 sec / 0.000 sec
14	12:07:56	SELECT SUM(payment.PaymentAmount) AS POExpenses' FROM payment JOIN directdeposit ON payment.D... 1 row(s) returned		0.000 sec / 0.000 sec

10) The amount paid to employee "Carrie Whittier" in the month of September



The screenshot shows the MySQL Workbench interface with the SQL editor tab open. The query window contains the following code:

```
# 10) The amount paid to employee "Carrie Whittier" in the month of september
SELECT (@time := SUM(timeworked.Hours)) AS 'TotalHours'
FROM SELECT @time := SUM(timeworked.Hours) AS 'TotalHours'
FROM employee JOIN timeworked ON employee.EmployeeID = timeworked.EmployeeID
JOIN payperiods ON timeworked.PayPeriodID = payperiods.PayPeriodID
JOIN job ON employee.JobID = job.JobID
WHERE employee.EmployeeID = 4 AND payperiods.EndDate < '2023-10-1' AND payperiods.StartDate >= '2023-9-1' timeAmount
JOIN job JOIN Employee ON employee.JobID = job.JobID
WHERE employee.EmployeeID = 4;
```

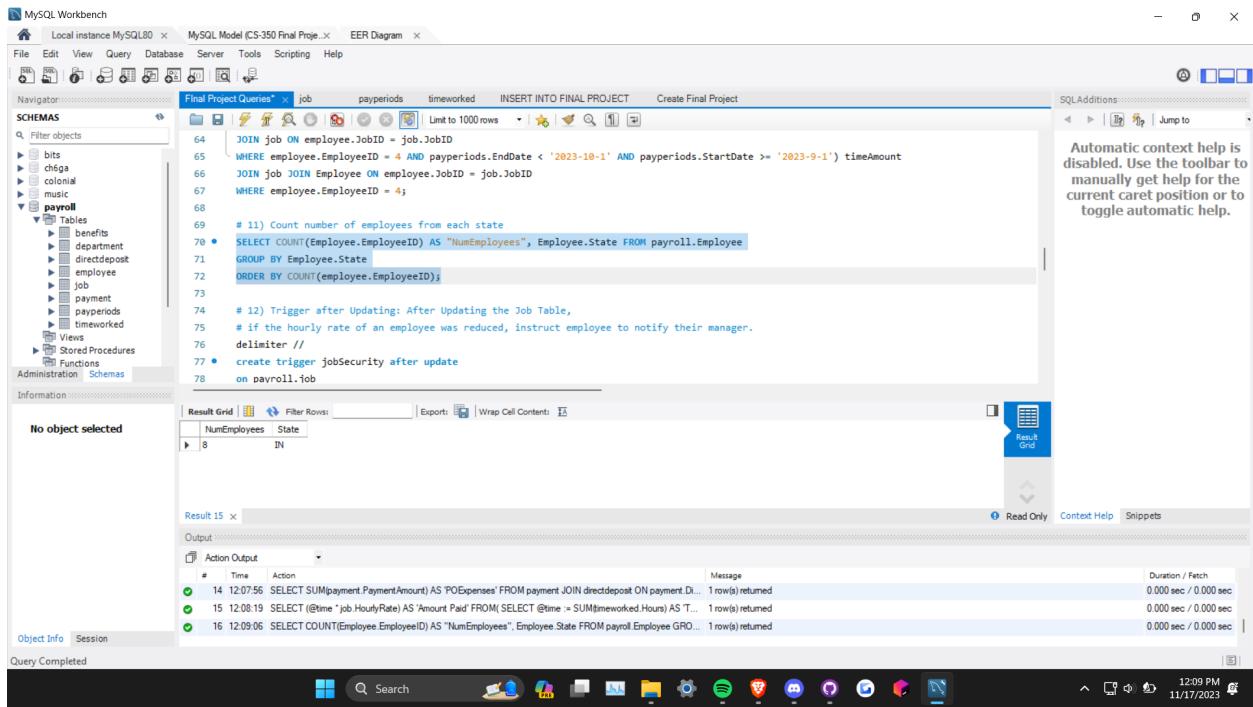
Below this, another section of the query is highlighted:

```
# 11) Count number of employees from each state
SELECT COUNT(Employee.EmployeeID) AS "NumEmployees", Employee.State FROM payroll.Employee
GROUP BY Employee.State
ORDER BY COUNT(employee.EmployeeID);
```

The result grid shows a single row with the value 4800. The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
13	12:07:37	SELECT SUM(payment.PaymentAmount) AS 'POExpenses' FROM payment JOIN directdeposit ON payment.D... 1 row(s) returned		0.000 sec / 0.000 sec
14	12:07:56	SELECT SUM(payment.PaymentAmount) AS 'POExpenses' FROM payment JOIN directdeposit ON payment.D... 1 row(s) returned		0.000 sec / 0.000 sec
15	12:08:19	SELECT (@time := SUM(timeworked.Hours)) AS 'TotalHours'		0.000 sec / 0.000 sec

11) Count the number of employees from each state



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80, MySQL Model (CS-350 Final Proj...), EER Diagram.
- Toolbar:** Standard MySQL Workbench tools.
- Navigator:** Schemas (bits, chsga, colonial, music, payroll). The payroll schema is expanded, showing tables: benefits, department, directdeposit, employee, job, payment, payperiods, timeworked, and views.
- SQL Editor:**

```

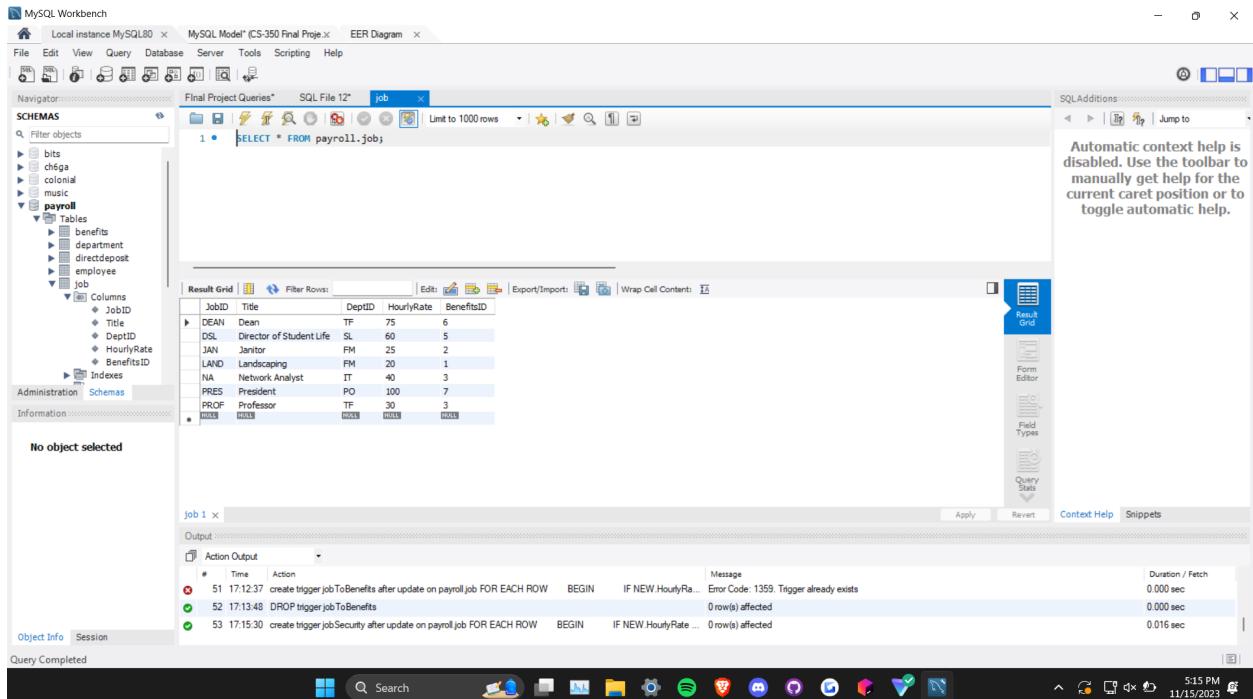
64 JOIN job ON employee.JobID = job.JobID
65 WHERE employee.EmployeeID = 4 AND payperiods.EndDate < '2023-10-1' AND payperiods.StartDate >= '2023-9-1') timeAmount
66 JOIN job JOIN Employee ON employee.JobID = job.JobID
67 WHERE employee.EmployeeID = 4;
68
69 # 11) Count number of employees from each state
70 • SELECT COUNT(Employee.EmployeeID) AS "NumEmployees", Employee.State FROM payroll.Employee
71 GROUP BY Employee.State
72 ORDER BY COUNT(employee.EmployeeID);
73
74 # 12) Trigger after Updating: After Updating the Job Table,
75 # if the hourly rate of an employee was reduced, instruct employee to notify their manager.
76 delimiter //
77 • create trigger jobSecurity after update
    on payroll.job
    
```
- Result Grid:** Shows the output of the query:

NumEmployees	State
8	IN
- Action Output:** Shows the history of actions:

Time	Action	Message	Duration / Fetch
14 12:07:56	SELECT SUM(payment.PaymentAmount) AS 'P0Expenses' FROM payment JOIN directdeposit ON payment.DIRECTDEPOSITID = directdeposit.ID	1 row(s) returned	0.000 sec / 0.000 sec
15 12:08:19	SELECT (@time := job.HourlyRate) AS 'Amount Paid' FROM SELECT @time := SUM(meworked.Hours) AS TotalHours, job.HourlyRate AS AmountPaid FROM payroll.job	1 row(s) returned	0.000 sec / 0.000 sec
16 12:09:06	SELECT COUNT(Employee.EmployeeID) AS "NumEmployees", Employee.State FROM payroll.Employee GROUP BY Employee.State	1 row(s) returned	0.000 sec / 0.000 sec

12) Trigger after Updating: After Updating the Job Table, # if the hourly rate of an employee was reduced, instruct the employee to notify their manager.

Job Table Before



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80, MySQL Model (CS-350 Final Proj...), EER Diagram.
- Toolbar:** Standard MySQL Workbench tools.
- Navigator:** Schemas (bits, chsga, colonial, music, payroll). The payroll schema is expanded, showing tables: benefits, department, directdeposit, employee, job, payment, payperiods, timeworked, and views. The job table is selected.
- SQL Editor:**

```

1 • SELECT * FROM payroll.job;

```
- Result Grid:** Shows the output of the query:

JobID	Title	DeptID	HourlyRate	BenefitsID
DEAN	Dean	TF	75	6
DSL	Director of Student Life	SL	60	5
JAN	Janitor	FM	25	2
LAND	Landscape	FM	20	1
NA	Network Analyst	IT	40	3
PRES	President	PO	100	7
PROF	Professor	TF	30	3
HALL	Hall	HALL	HALL	HALL
- Action Output:** Shows the history of actions:

Time	Action	Message	Duration / Fetch
51 17:12:37	create trigger jobToBenefits after update on payroll.job FOR EACH ROW BEGIN IF NEW.HourlyRate < OLD.HourlyRate THEN	Error Code: 1359. Trigger already exists	0.000 sec
52 17:13:48	DROP trigger jobToBenefits	0 row(s) affected	0.000 sec
53 17:15:30	create trigger jobSecurity after update on payroll.job FOR EACH ROW BEGIN IF NEW.HourlyRate < OLD.HourlyRate THEN	0 row(s) affected	0.016 sec

When Attempting to decrease the payroll amount it requests manager approval.

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** payroll
- Table:** job
- Code (SQL File 12*):**

```

70 • SELECT COUNT(Employee.EmployeeID) AS "NumEmployees", Employee.State FROM payroll.Employee
71 GROUP BY Employee.State
72 ORDER BY COUNT(Employee.EmployeeID);
73
74 # 12) Trigger after Updating: After Updating the Job Table,
75 # if the hourly rate of an employee was reduced, instruct employee to notify their manager.
76 delimiter //
77 • create trigger jobSecurity after update
78 on payroll.job
79 FOR EACH ROW
80     BEGIN
81         IF NEW.HourlyRate < OLD.HourlyRate THEN
82             SIGNAL SQLSTATE '50002' SET MESSAGE_TEXT = 'Notify Manager of Payroll decrease!';
83         END IF;
84     END;
85 // delimiter;
86
87
88 UPDATE job
89 SET job.HourlyRate = 35
90 WHERE job.HourlyRate = 40
91 
```
- Output:**

Action	Time	Action	Message	Duration / Fetch
Drop trigger	52 17:13:48	jobToBenefits	0 row(s) affected	0.000 sec
Create trigger	53 17:15:30	jobSecurity after update on payroll.job FOR EACH ROW	BEGIN IF NEW.HourlyRate ...	0.016 sec
Update job	54 17:16:01	SET job.HourlyRate = 35 WHERE job.HourlyRate = 40	Error Code: 1644. Notify Manager of Payroll decrease!	0.016 sec

13) Trigger before Deleting: Before Deleting a row in the payperiods table, all associated rows in the timeworked table are deleted.

PayPeriods and Timeworked BEFORE

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** payroll
- Table:** payperiods
- Code (SQL File 12*):**

```

1 • SELECT * FROM payroll.payperiods;
 
```
- Output:**

PayPeriodID	StartDate	EndDate
1	2023-09-01	2023-09-15
2	2023-09-15	2023-09-29
3	2023-09-29	2023-10-13
4	2023-10-13	2023-10-27
5	2023-10-27	2023-11-10
- Action Output:**

Action	Time	Message	Duration / Fetch
Create trigger	53 17:15:30	create trigger jobSecurity after update on payroll.job FOR EACH ROW BEGIN IF NEW.HourlyRate ...	0.016 sec
Update job	54 17:16:01	SET job.HourlyRate = 35 WHERE job.HourlyRate = 40	0.016 sec
Select payperiods	55 17:20:43	SELECT * FROM payroll.payperiods LIMIT 0, 1000	0.000 sec / 0.000 sec

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Local instance MySQL80 MySQL Model (CS-350 Final Proj.x) EER Diagram

Final Project Queries* SQL File 12* job payperiods timeworked

1 • `SELECT * FROM payroll.timeworked;`

Result Grid

EmployeeID	PayPeriodID	Hours
1	1	80
1	2	80
1	3	80
2	1	40
2	2	40
2	3	40
3	1	60
3	2	60
3	3	60
4	1	40
4	2	40
4	3	40
5	1	40
5	2	40

timeworked 1 ×

Action Output

#	Time	Action	Message	Duration / Fetch
54	17:16:01	UPDATE job SET job.HourlyRate = 35 WHERE job.HourlyRate = 40	Error Code: 1644. Notify Manager of Payroll decrease!	0.016 sec
55	17:20:43	SELECT * FROM payroll.payperiods LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
56	17:20:57	SELECT * FROM payroll.timeworked LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

5:21 PM 11/15/2023

PayPeriods and TimeWorked AFTER delete

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Local instance MySQL80 MySQL Model (CS-350 Final Proj.x) EER Diagram

Final Project Queries* SQL File 12* job payperiods timeworked

1 • `SELECT * FROM payroll.payperiods;`

Result Grid

PayPeriodID	StartDate	EndDate
1	2023-09-01	2023-09-15
2	2023-09-15	2023-09-29
3	2023-09-29	2023-10-13
4	2023-10-13	2023-10-27

payperiods 2 ×

Action Output

#	Time	Action	Message	Duration / Fetch
57	17:22:35	DELETE FROM payperiods WHERE payperiods.PayPeriodID = 5	1 row(s) affected	0.000 sec
58	17:22:51	SELECT * FROM payroll.payperiods LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
59	17:22:54	SELECT * FROM payroll.timeworked LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

5:23 PM 11/15/2023

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas: payroll

Final Project Queries* SQL File 12* job payperiods timeworked

```
1 • SELECT * FROM payroll.timeworked;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

EmployeeID	PayPeriodID	Hours
2	1	40
2	2	40
2	3	40
3	1	60
3	2	60
3	3	60
4	1	40
4	2	40
4	3	40
5	1	40
5	2	40
5	3	40

timeworked 2 x

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
57	17:22:35	DELETE FROM payperiods WHERE payperiods.PayPeriodID = 5	1 row(s) affected	0.000 sec
58	17:22:51	SELECT * FROM payroll.payperiods LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
59	17:22:54	SELECT * FROM payroll.timeworked LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

5:23 PM 11/15/2023

14) Stored Procedure to get the sum of all expenses for a given department (param is DeptID) for the current calendar year

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas: payroll

Final Project Queries* SQL File 12* job payperiods timeworked

```
112
113 JOIN employee ON employee.EmployeeID = directdeposit.EmployeeID
114 JOIN job ON employee.JobID = job.JobID
115 JOIN department ON job.DeptID = department.DeptID
116 WHERE department.DeptID = pDept AND YEAR(payment.PaymentDate) = YEAR(CURDATE());
117 end //
118 • call GetSpendingForDepartment("PO");
119
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Expenses
2551

Result 1 x

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
59	17:22:54	SELECT * FROM payroll.timeworked LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec
60	17:28:03	create procedure GetSpendingForDepartment(IN pDept VARCHAR(4)) begin SELECT SUM(payment.PaymentAmount) AS Expenses FROM payroll.timeworked WHERE timeworked.DeptID = pDept; END;	Error Code: 1304. PROCEDURE GetSpendingForDepartment already exists	0.000 sec
61	17:28:08	call GetSpendingForDepartment("PO")	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

5:28 PM 11/15/2023

15) Testing Create User statements:

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** payroll
- SQL Editor:**

```

113 JOIN job ON employee.JobID = job.JobID
114 JOIN department ON job.DeptID = department.DeptID
115 WHERE department.DeptID = pDept AND YEAR(payment.PaymentDate) = YEAR(CURDATE())
116 end //
117 delimiter ;
118 • call GetSpendingForDepartment("PO");
119
# 15) Create New Users w/ GRANT and REVOKE
120 • CREATE USER IF NOT EXISTS nkoepen@localhost
121 • IDENTIFIED BY "Koeppen1";
122 • CREATE USER IF NOT EXISTS mdemby@localhost
123 • IDENTIFIED BY "Dembny1";
124
125
126 • GRANT ALL ON payroll TO nkoepen@localhost;
127 • GRANT ALL ON payroll TO mdemby@localhost;
128 • SHOW GRANTS FOR nkoepen@localhost;
129 • SHOW GRANTS FOR mdemby@localhost;
130
131 • REVOKE UPDATE,DELETE ON bits FROM nkoepen@localhost;
132 • SHOW GRANTS FOR nkoepen@localhost;
133 • REVOKE UPDATE,DELETE ON bits FROM mdemby@localhost;
134 • SHOW GRANTS FOR mdemby@localhost;

```
- Output:**
 - Action Output:

Time	Action	Message	Duration / Fetch
61 17:28:08	Action	1 row(s) returned	0.000 sec / 0.000 sec
62 17:30:09	CREATE USER IF NOT EXISTS nkoepen@localhost IDENTIFIED BY "Koeppen1"	0 row(s) affected, 1 warning(s): 3163 Authorization ID 'nkoepen'@'localhost' already exists.	0.016 sec
63 17:30:15	CREATE USER IF NOT EXISTS mdemby@localhost IDENTIFIED BY "Dembny1"	0 row(s) affected	0.031 sec

Showing Grants before privileges are revoked

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** payroll
- SQL Editor:**

```

125
126 • GRANT ALL ON payroll TO nkoepen@localhost;
127 • GRANT ALL ON payroll TO mdemby@localhost;
128 • SHOW GRANTS FOR nkoepen@localhost;
129 • SHOW GRANTS FOR mdemby@localhost;
130
131 • REVOKE UPDATE,DELETE ON bits FROM nkoepen@localhost;
132 • SHOW GRANTS FOR nkoepen@localhost;

```
- Result Grid:**

Grants for nkoepen@localhost
GRANT USAGE ON *.* TO `nkoepen`@`localhost`;
GRANT ALL PRIVILEGES ON `payroll`.* TO `nkoepen`@`localhost`;
- Output:**
 - Action Output:

Time	Action	Message	Duration / Fetch
64 17:31:15	GRANT ALL ON payroll TO nkoepen@localhost	0 row(s) affected	0.000 sec
65 17:31:15	GRANT ALL ON payroll TO mdemby@localhost	0 row(s) affected	0.000 sec
66 17:31:19	SHOW GRANTS FOR nkoepen@localhost	2 row(s) returned	0.015 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80 x MySQL Model* (CS-350 Final Proj... x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigators: Final Project Queries x SQL File 12* job payperiods timeworked

Limit to 1000 rows

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

SCHEMAS

- bits
- ch6ga
- colonial
- music
- payroll**
 - tables
 - benefits
 - department
 - directdeposit
 - employee
 - job
 - payment
 - payperiods
 - timeworked
- Views
- Stored Procedures
- Functions
- sakila

Administration Schemas

No object selected

Result 2 x

Output:

#	Time	Action	Message	Duration / Fetch
64	17:31:15	GRANT ALL ON payroll TO nkoepen@localhost;	0 row(s) affected	0.000 sec
65	17:31:15	GRANT ALL ON payroll TO mdebnby@localhost;	0 row(s) affected	0.000 sec
66	17:31:19	SHOW GRANTS FOR nkoepen@localhost;	2 row(s) returned	0.015 sec / 0.000 sec

Object Info Session

Query Completed

5:31 PM 11/15/2023

The screenshot shows the MySQL Workbench interface with the 'Final Project Queries' tab active. In the 'SCHEMAS' tree, the 'payroll' schema is expanded, showing its tables: bits, department, directdeposit, employee, job, payment, payperiods, and timeworked. The 'Results' pane displays the output of several SQL statements. The first two statements grant all privileges on the 'payroll' schema to 'nkoepen@localhost' and 'mdebnby@localhost'. The third statement shows grants for 'nkoepen@localhost', listing the granted privileges. The bottom section shows the execution history with three entries: the grants, and then a 'SHOW GRANTS FOR nkoepen@localhost;' command which returns two rows.

Showing Grants after Privileges are revoked

MySQL Workbench

Local instance MySQL80 x MySQL Model* (CS-350 Final Proj... x EER Diagram x

File Edit View Query Database Server Tools Scripting Help

Navigators: Final Project Queries x SQL File 12* job payperiods timeworked

Limit to 1000 rows

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

SCHEMAS

- bits
- ch6ga
- colonial
- music
- payroll**
 - tables
 - benefits
 - department
 - directdeposit
 - employee
 - job
 - payment
 - payperiods
 - timeworked
- Views
- Stored Procedures
- Functions
- sakila

Administration Schemas

No object selected

Result 14 x

Output:

#	Time	Action	Message	Duration / Fetch
79	17:32:59	REVOKE UPDATE,DELETE ON bits FROM nkoepen@localhost;	Error Code: 1147. There is no such grant defined for user 'nkoepen' on host 'localhost' on table 'bits'	0.000 sec
80	17:33:25	REVOKE UPDATE,DELETE ON payroll FROM nkoepen@localhost;	0 row(s) affected	0.016 sec
81	17:33:25	SHOW GRANTS FOR nkoepen@localhost;	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

5:33 PM 11/15/2023

This screenshot shows the MySQL Workbench interface after revoking privileges from the user 'nkoepen@localhost'. The 'Final Project Queries' tab is active, and the 'SCHEMAS' tree shows the 'payroll' schema with its tables. The 'Results' pane displays the execution history. The first entry is a failed 'REVOKE' statement for the 'bits' table, indicating that the grant does not exist. The second entry shows the successful revocation of update and delete privileges on the 'payroll' table for 'nkoepen@localhost'. The third entry shows the grants again, but this time it only lists the 'SHOW GRANTS FOR nkoepen@localhost;' command, which returns two rows.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Final Project Queries* SQL File 12* job payperiods timeworked

Schemas

- bits
- ch6ga
- colonial
- music
- payroll
 - benefits
 - department
 - directdeposit
 - employee
 - job
 - payment
 - payperiods
 - timeworked
- sakila

No object selected

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 14 x Read Only Context Help Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
79	17:32:59	REVOKE UPDATE,DELETE ON bits FROM rkoepen@localhost	Error Code: 1147. There is no such grant defined for user 'rkoepen' on host 'localhost' on table 'bits'	0.000 sec
80	17:33:25	REVOKE UPDATE,DELETE ON payroll FROM rkoepen@localhost	0 row(s) affected	0.016 sec
81	17:33:25	SHOW GRANTS FOR rkoepen@localhost	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

5:34 PM 11/15/2023