

SECTIONS 10, 11, 12 & 13

Section 10

In this lesson, we explored the INSERT statement.

Exercise:

Select ten records from the “titles” table to get a better idea about its content.

Then, in the same table, insert information about employee number 9999901. State that he/she is a “Senior Engineer”, who has started working in this position on October 1st, 1997.

At the end, sort the records from the “titles” table in descending order to check if you have successfully inserted the new record.

Solution:

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'employees' expanded, showing tables like 'titles'. The main editor window contains the following SQL code:

```
324 # Select ten records from the "titles" table to get a better idea about its content. #
325 select * from titles
326 limit 10;
```

The 'Result Grid' below the editor shows the results of the first query, displaying columns: emp_no, title, from_date, to_date. The results are sorted by emp_no in descending order.

emp_no	title	from_date	to_date
10001	Senior Engineer	1986-06-26	9999-01-01
10002	Staff	1996-08-03	9999-01-01
10003	Senior Engineer	1995-12-03	9999-01-01
10004	Engineer	1986-12-01	1995-12-01
10004	Senior Engineer	1995-12-01	9999-01-01
10005	Senior Staff	1996-09-12	9999-01-01
10005	Staff	1989-09-12	1996-09-12
10006	Senior Engineer	1990-08-05	9999-01-01
10007	Senior Staff	1996-02-11	9999-01-01
10007	Staff	1989-02-11	1996-02-11

The 'Output' window at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
238	12:04:17	insert into employees (emp_no, birth_date, first_name, last_name, gender, hire_date) values ...	1 row(s) affected	0.328 sec
239	12:05:15	select * from employees order by emp_no desc	300025 row(s) returned	0.063 sec / 0.515 sec
240	12:10:28	select * from titles limit 10	10 row(s) returned	0.109 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: employees employees employees

SCHEMAS

Filter objects

employees

Tables

departments

dept_emp

dept_manager

employees

salaries

titles

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

Table: salaries

Columns:

emp_no int PK

salary int

from_date date PK

to_date date

Object Info Session

SQL File 5* employees.employees

Limit to 1000 rows

```

328 /* Then, in the same table, insert information about employee number 9999901.
329 State that he/she is a "Senior Engineer", who has started working in this position on October 1st, 1997.
330 At the end, sort the records from the "titles" table in descending order to check if you have successfully inserted the
331 */
332 insert into titles ( emp_no, title, from_date)
333 values ('9999901', 'Senior Engineer', '1997-09-01')
334 ;
335 select * from titles
336 order by emp_no desc
337 ;

```

Result Grid

emp_no	title	from_date	to_date
9999901	Senior Engineer	1997-09-01	9999-01-01
4999999	Engineer	1997-11-30	9999-01-01
4999998	Staff	1993-12-27	1998-12-27
4999998	Senior Staff	1998-12-27	9999-01-01
4999997	Senior Engineer	1992-08-29	9999-01-01
4999997	Engineer	1987-08-30	1992-08-29
4999996	Senior Engineer	2002-05-13	9999-01-01
4999996	Engineer	1996-05-13	2002-05-13

titles 125 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
241	12:23:34	insert into titles (emp_no, title, from_date) values ('9999901', 'Senior Engineer', '1997-09-01')	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (employee...	0.156 sec
242	12:24:59	insert into titles (emp_no, title, from_date) values ('9999901', 'Senior Engineer', '1997-09-01')	1 row(s) affected	0.141 sec
243	12:26:44	select * from titles order by emp_no desc	443309 row(s) returned	0.344 sec / 1.109 sec

Context Help Snippets

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

Exercise:

Insert information about the individual with employee number 9999901 into the "dept_emp" table. He/She is working for department number 5, and has started work on October 1st, 1997; her/his contract is for an indefinite period of time.

Hint: Use the date '9999-01-01' to designate the contract is for an indefinite period.

Solution:

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: employees employees employees

SCHEMAS

Filter objects

employees

Tables

departments

dept_emp

dept_manager

employees

salaries

titles

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

Table: salaries

Columns:

emp_no int PK

salary int

from_date date PK

to_date date

Object Info Session

SQL File 5* employees.employees

Limit to 1000 rows

```

339 select * from dept_emp;
340
341 insert into dept_emp (emp_no, dept_no, from_date, to_date)
342 values ('9999901', 'd005', '1997-09-01', '9999-01-01')
343 ;
344 select * from dept_emp
345 order by emp_no desc
346 limit 10
347 ;
348

```

Result Grid

emp_no	dept_no	from_date	to_date
9999901	d005	1997-09-01	9999-01-01
4999999	d004	1997-11-30	9999-01-01
4999998	d002	1993-12-27	9999-01-01
4999997	d005	1987-08-30	9999-01-01
4999996	d004	1996-05-13	9999-01-01
4999995	d004	1997-06-02	9999-01-01
4999994	d004	1993-02-22	1993-10-27
4999993	d004	1997-04-07	9999-01-01

dept_emp 128 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
246	12:37:26	select * from dept_emp	331603 row(s) returned	0.062 sec / 0.422 sec
247	12:38:22	insert into dept_emp (emp_no, dept_no, from_date, to_date) values ('9999901', 'd005', '1997-09-01', '9999-01-01')	1 row(s) affected	0.172 sec
248	12:39:14	select * from dept_emp order by emp_no desc limit 10	10 row(s) returned	0.000 sec / 0.000 sec

Context Help Snippets

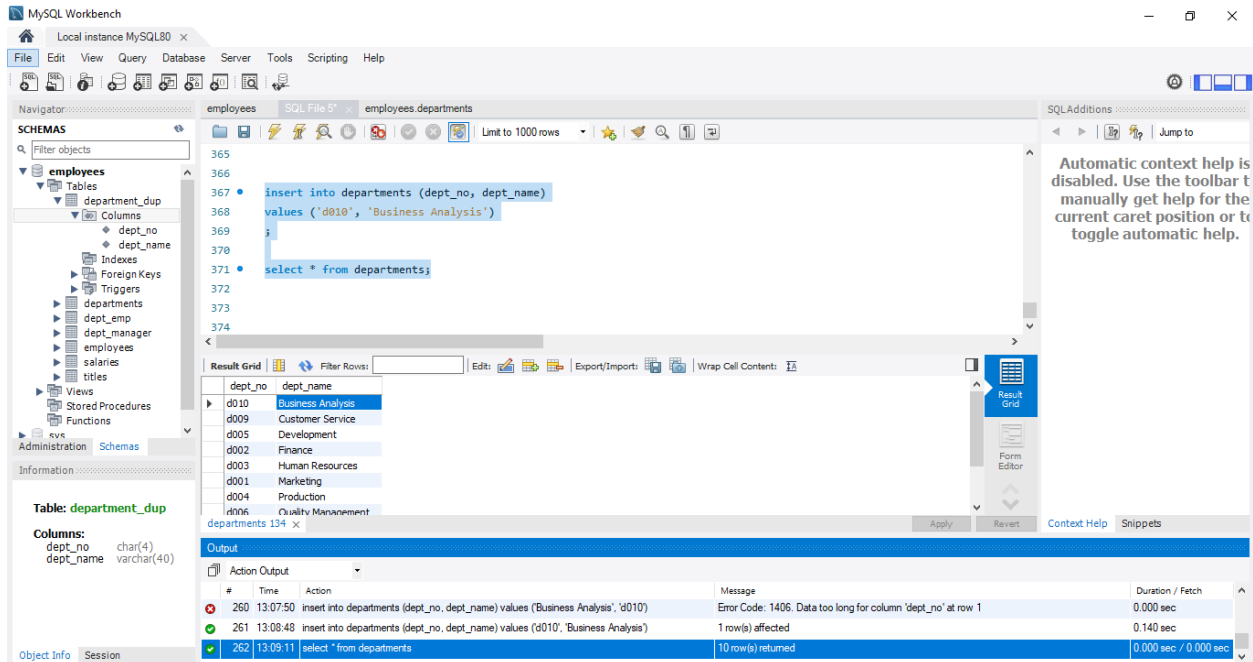
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Exercise:

Create a new department called “Business Analysis”. Register it under number ‘d010’.

Hint: To solve this exercise, use the “departments” table.

Solution:



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following code:

```
365
366
367 insert into departments (dept_no, dept_name)
368 values ('d010', 'Business Analysis')
369 ;
370
371 select * from departments;
```

The result grid shows the following data:

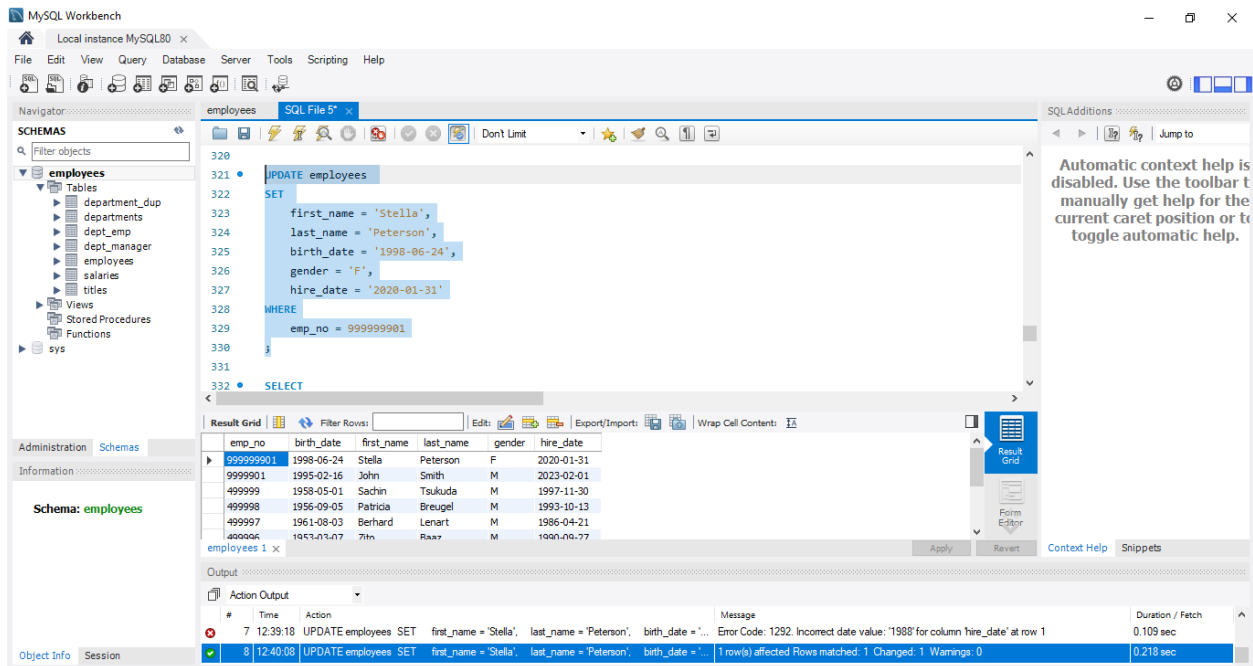
dept_no	dept_name
d010	Business Analysis
d009	Customer Service
d005	Development
d002	Finance
d003	Human Resources
d001	Marketing
d004	Production
d006	Quality Management

The output pane shows the following messages:

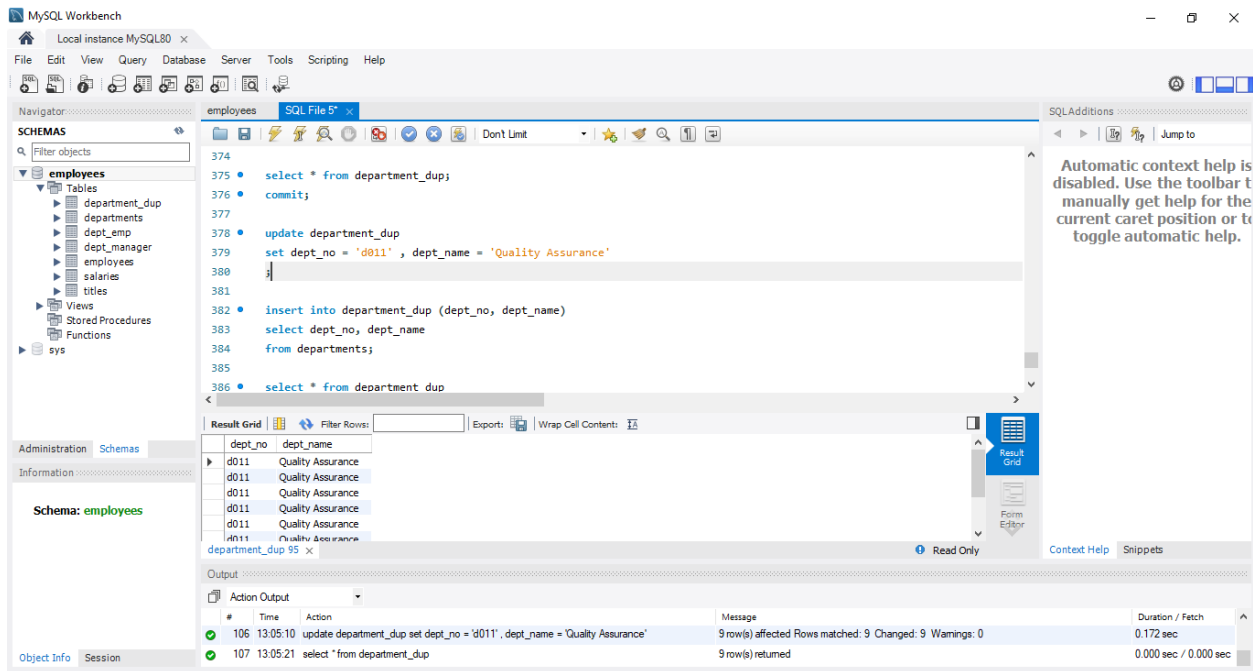
#	Time	Action	Message	Duration / Fetch
260	13:07:50	insert into departments (dept_no, dept_name) values ('Business Analysis', 'd010')	Error Code: 1406. Data too long for column 'dept_no' at row 1	0.000 sec
261	13:08:48	insert into departments (dept_no, dept_name) values ('d010', 'Business Analysis')	1 row(s) affected	0.140 sec
262	13:09:11	select * from departments	10 row(s) returned	0.000 sec / 0.000 sec

Section 11:

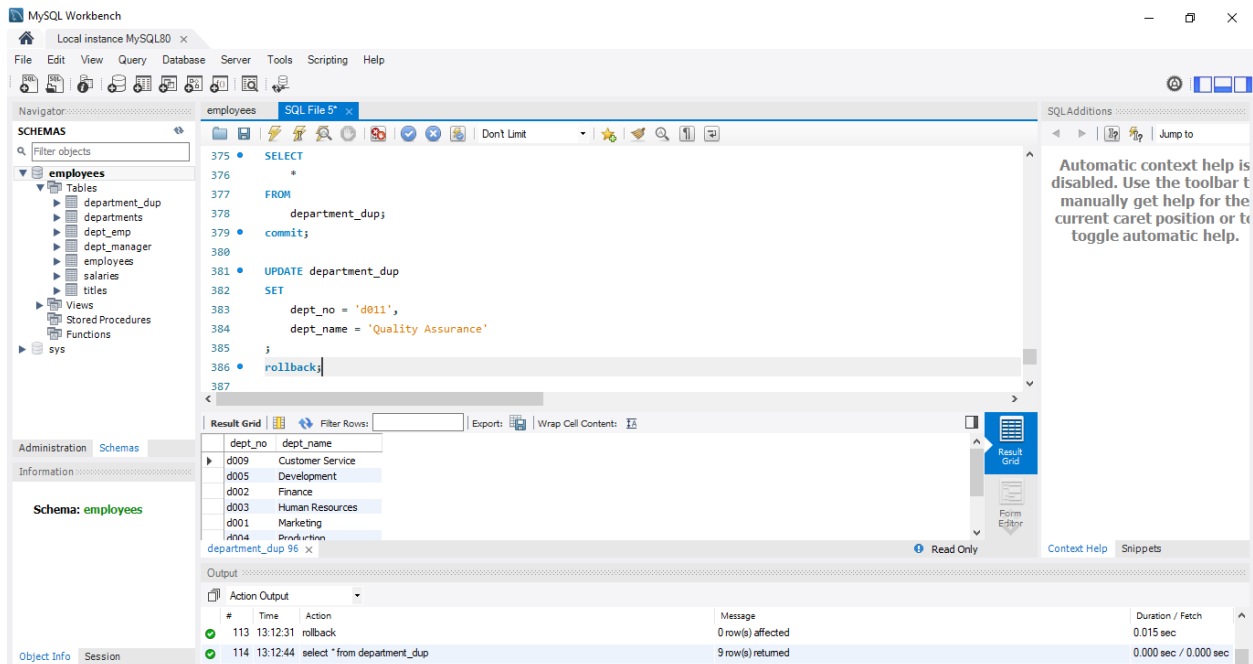
In this lesson, we learned the UPDATE statement.



I was updating my department duplicate table that comprises nine rows and two columns. Unfortunately, I forgot to add the WHERE condition in my code.



As a result, all my rows are identical. So I used ROLLBACK to take me back and make the necessary changes.

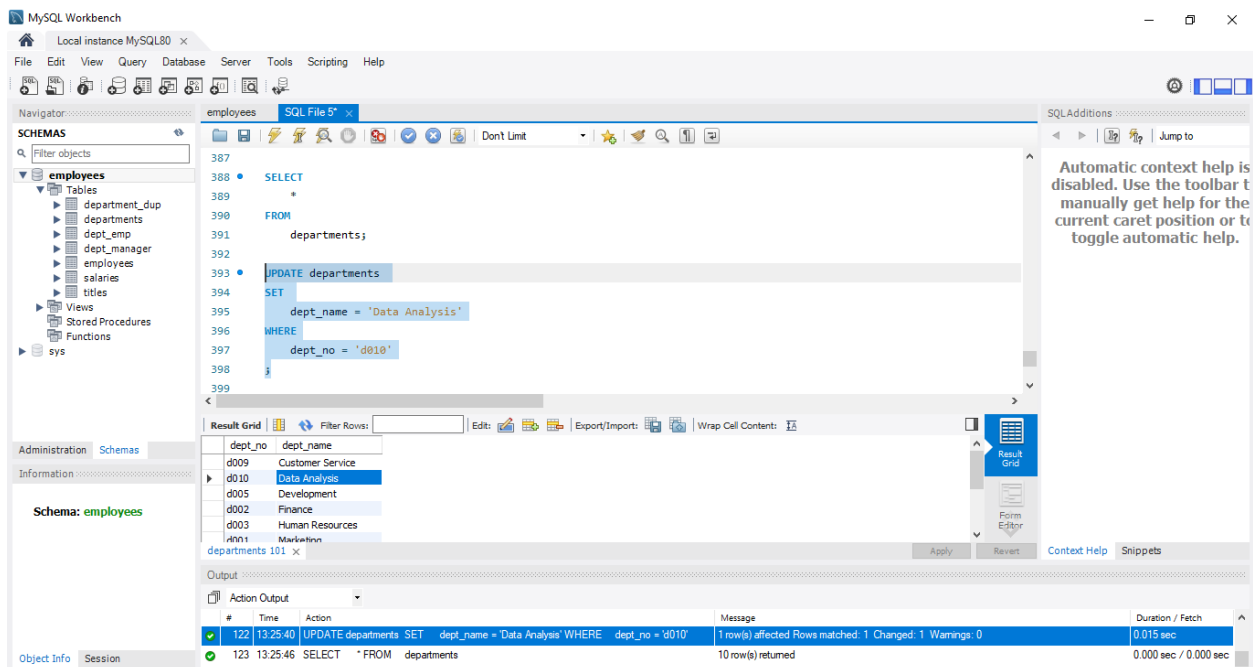


Exercise:

Change the “Business Analysis” department name to “Data Analysis”.

Hint: To solve this exercise, use the “departments” table.

Solution:



SECTION 12:

Here we explored the DELETE statement.

Exercise:

Remove the department number 10 record from the “departments” table.

Solution:

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'employees' schema selected. The main editor window contains the following SQL query:

```
select * from departments;
```

Below the query editor, the 'Result Grid' shows the output of the query:

dept_no	dept_name
d009	Customer Service
d010	Data Analysis
d005	Development
d002	Finance
d003	Human Resources
d001	Marketing

The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
129	11:48:48	select * from employees where emp_no = '999999901'	0 row(s) returned	0.000 sec / 0.000 sec
130	12:01:04	select * from departments	10 row(s) returned	0.000 sec / 0.000 sec

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'employees' schema selected. The main editor window contains the following SQL query:

```
SELECT
*
FROM
departments;

# Remove the department number 10 record from the "departments" table. #

DELETE FROM departments
WHERE
dept_no = 'd010'
```

Below the query editor, the 'Result Grid' shows the output of the query:

dept_no	dept_name
d009	Customer Service
d005	Development
d002	Finance
d003	Human Resources
d001	Marketing

The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
131	12:06:59	DELETE FROM departments WHERE dept_no = 'd010'	1 row(s) affected	0.000 sec
132	12:07:06	SELECT * FROM departments	9 row(s) returned	0.000 sec / 0.000 sec

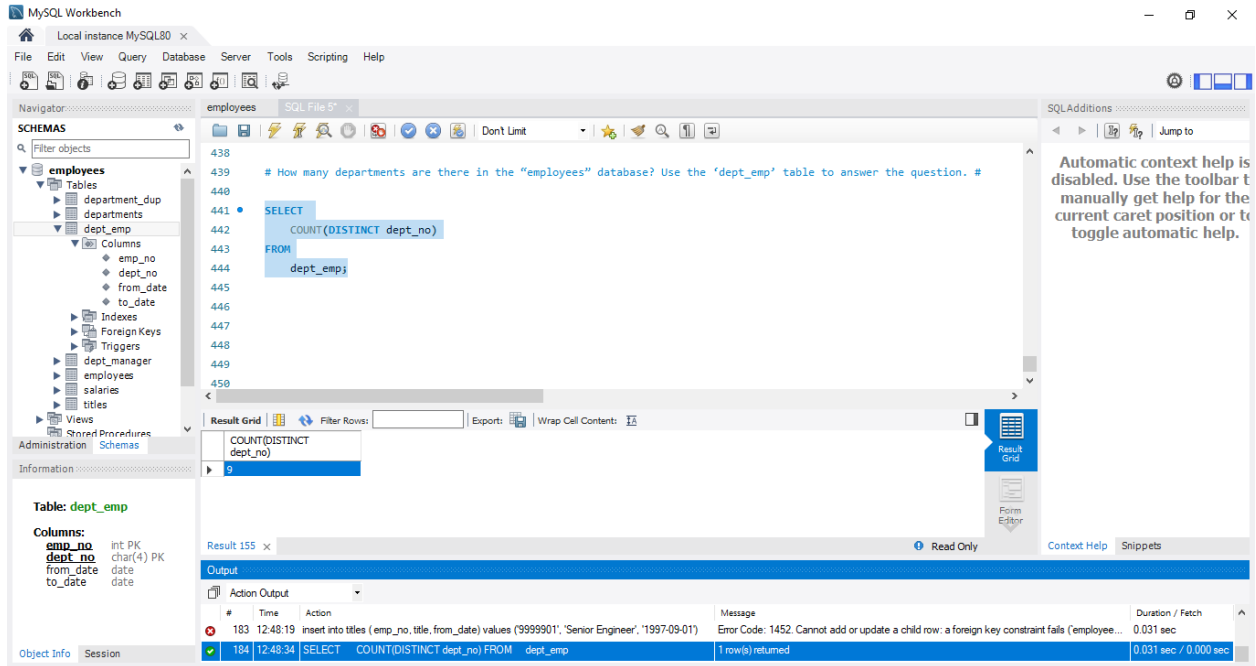
SECTION 13:

This section explores different aggregate functions, including COUNT(), SUM(), MIN(), MAX(), AVG(), etc.

Exercise:

How many departments are there in the “employees” database? Use the ‘dept_emp’ table to answer the question.

Solution:



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'employees' database schema, with the 'dept_emp' table selected. The main editor window contains the following SQL query:

```
SELECT  
  COUNT(DISTINCT dept_no)  
FROM  
  dept_emp;
```

The 'Result Grid' at the bottom shows the query result:

COUNT(DISTINCT dept_no)
9

The bottom status bar shows the query execution details:

#	Time	Action	Message	Duration / Fetch
183	12:48:19	insert into titles (emp_no, title, from_date) values ('9999901', 'Senior Engineer', '1997-09-01')	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('employee...	0.031 sec
184	12:48:34	SELECT COUNT(DISTINCT dept_no) FROM dept_emp	1 row(s) returned	0.031 sec / 0.000 sec

Our next question is, how much money the firm spends on the salaries of the employees? For this, we will use the SUM function.

MySQL Workbench interface showing a query in the SQL Editor:

```

446 SELECT
447     SUM(salary)
448 FROM
449     salaries
450
451
452
453
454
455
456
457

```

The Result Grid shows the output of the query:

SUM(salary)
61678125784

The Information tab shows the structure of the salaries table:

Table: salaries

Columns:

- emp_no: int PK
- salary: int
- from_date: date PK
- to_date: date

The Output tab shows the execution details:

#	Time	Action	Message	Duration / Fetch
184	12:48:34	SELECT COUNT(DISTINCT dept_no) FROM dept_emp	1 row(s) returned	0.031 sec / 0.000 sec
185	12:53:01	SELECT SUM(salary) FROM salaries	1 row(s) returned	0.562 sec / 0.000 sec

It turns out the firm is spending more than 61 billion. That's a huge amount!

Exercise:

What is the total amount of money spent on salaries for all contracts starting after the 1st of January 1997?

Solution:

MySQL Workbench interface showing a query in the SQL Editor:

```

451
452
453 # What is the total amount of money spent on salaries for all contracts starting after the 1st of January 1997? #
454 SELECT
455     *
456 FROM
457     salaries;
458
459
460
461
462
463

```

The Result Grid shows the output of the query:

emp_no	salary	from_date	to_date
10001	60117	1986-06-26	1987-06-26
10001	62102	1987-06-26	1988-06-25
10001	66074	1988-06-25	1989-06-25
10001	66596	1989-06-25	1990-06-25
10001	66961	1990-06-25	1991-06-25
10001	71704	1991-06-25	1992-06-24

The Output tab shows the execution details:

#	Time	Action	Message	Duration / Fetch
185	12:53:01	SELECT SUM(salary) FROM salaries	1 row(s) returned	0.562 sec / 0.000 sec
186	12:59:19	select * from salaries	967330 row(s) returned	0.000 sec / 1.047 sec

MySQL Workbench interface showing a SQL query and its results.

Query:

```

453 # What is the total amount of money spent on salaries for all contracts starting after the 1st of January 1997? #
454 SELECT
455 *
456 FROM
457 salaries;
458
459 SELECT
460 SUM(salary)
461 FROM
462 salaries
463 WHERE
464 from_date > '1997-01-01'
465

```

Result Grid:

SUM(salary)
31909143195

Table: salaries

Columns:

- emp_no: int PK
- salary: int
- from_date: date PK
- to_date: date

Output:

#	Time	Action	Message	Duration / Fetch
186	12:59:19	select * from salaries	967330 row(s) returned	0.000 sec / 1.047 sec
187	13:02:37	SELECT SUM(salary) FROM salaries WHERE from_date > '1997-01-01'	1 row(s) returned	0.656 sec / 0.000 sec

Now, the question arises what is the highest salary the company offers? Thus,

MySQL Workbench interface showing a SQL query and its results.

Query:

```

465 ;
466
467 SELECT
468 MAX(salary)
469 FROM
470 salaries;
471
472
473
474
475
476
477

```

Result Grid:

MAX(salary)
158220

Table: salaries

Columns:

- emp_no: int PK
- salary: int
- from_date: date PK
- to_date: date

Output:

#	Time	Action	Message	Duration / Fetch
187	13:02:37	SELECT SUM(salary) FROM salaries WHERE from_date > '1997-01-01'	1 row(s) returned	0.656 sec / 0.000 sec
188	13:10:20	SELECT MAX(salary) FROM salaries	1 row(s) returned	0.500 sec / 0.000 sec

The result shows the highest salary offered is \$158,220 annually.

And what is the lowest salary in the company?

MySQL Workbench interface showing a query executed on the 'employees' database. The query is:

```
SELECT MIN(salary) FROM salaries;
```

The result grid displays the minimum salary as 38735. The output pane shows the query execution details:

#	Time	Action	Message	Duration / Fetch
188	13:10:20	SELECT MAX(salary) FROM salaries	1 row(s) returned	0.500 sec / 0.000 sec
189	13:12:58	SELECT MIN(salary) FROM salaries	1 row(s) returned	0.484 sec / 0.000 sec

It is \$38,735 per annum.

Exercise:

1. Which is the lowest employee number in the database?
2. Which is the highest employee number in the database?

Solution:

MySQL Workbench interface showing two queries executed on the 'employees' database. The first query is:

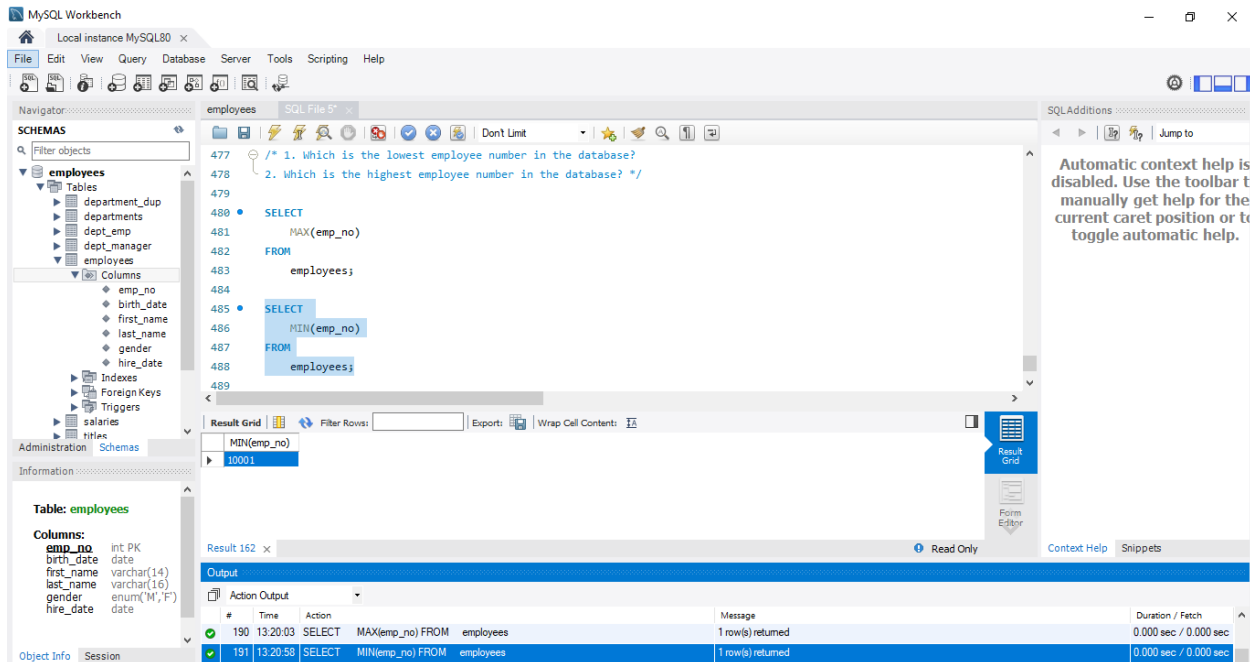
```
SELECT MIN(salary) FROM salaries;
```

The second query is:

```
SELECT MAX(emp_no) FROM employees;
```

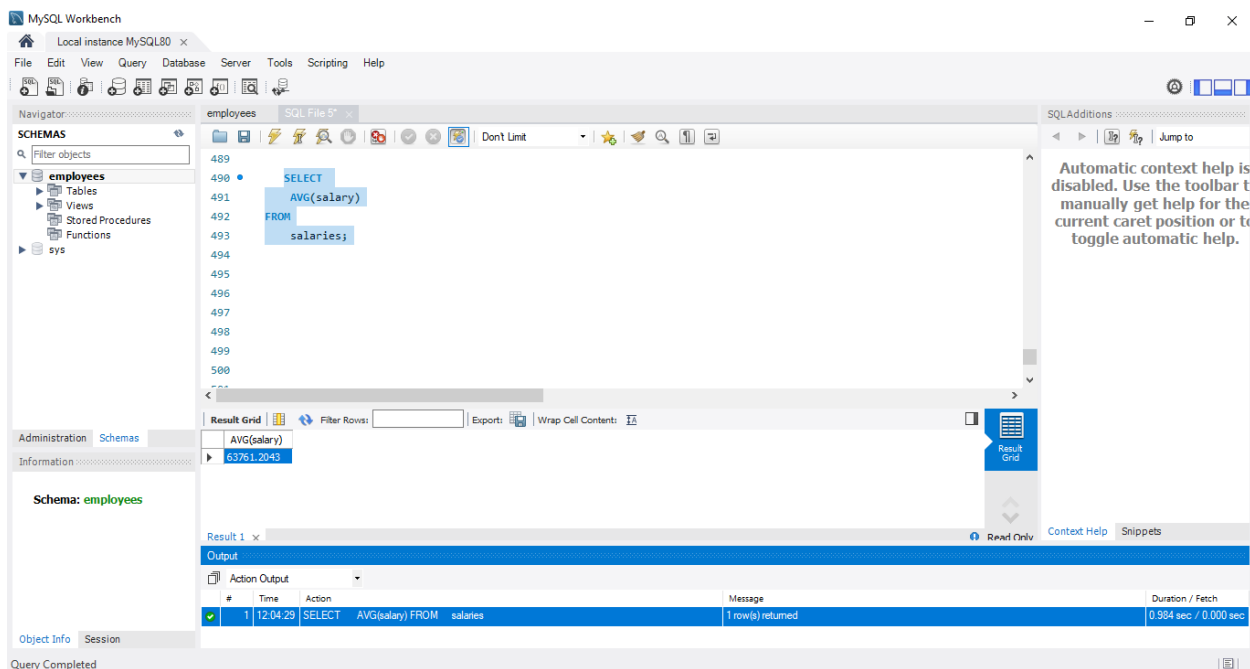
The result grid displays the maximum employee number as 999999901. The output pane shows the query execution details:

#	Time	Action	Message	Duration / Fetch
188	13:12:58	SELECT MIN(salary) FROM salaries	1 row(s) returned	0.484 sec / 0.000 sec
190	13:20:03	SELECT MAX(emp_no) FROM employees	1 row(s) returned	0.000 sec / 0.000 sec



Let's say we want to calculate the average annual price of a contract in the employees database.

In other words, our question is, what is the annual average salary the company's employees receive?



It turns out the average annual salary is \$63,761.2043.

Exercise:

What is the average annual salary paid to employees who started after the 1st of January 1997?

Solution:

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: employees SQL File 5*

SCHMAS

Filter objects

employees

Tables

department_dup

departments

dept_emp

dept_manager

employees

salaries

titles

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

Schema: employees

Result Grid

avg(salary)

63572.1765

Result 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	12:10:08	select * from salaries	967330 row(s) returned	0.000 sec / 1.485 sec
3	12:11:23	select avg(salary) from salaries where from_date = '1997-01-01'	1 row(s) returned	0.469 sec / 0.000 sec

Formatted 1 statements.

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

Exercise:

Round the average amount of money spent on salaries for all contracts that started after the 1st of January 1997 to a precision of cents.

Solution:

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: employees SQL File 5*

SCHMAS

Filter objects

employees

Tables

department_dup

departments

dept_emp

dept_manager

employees

salaries

titles

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

Schema: employees

Result Grid

ROUND(AVG(salary) , 2)

63572.18

Result 7 x

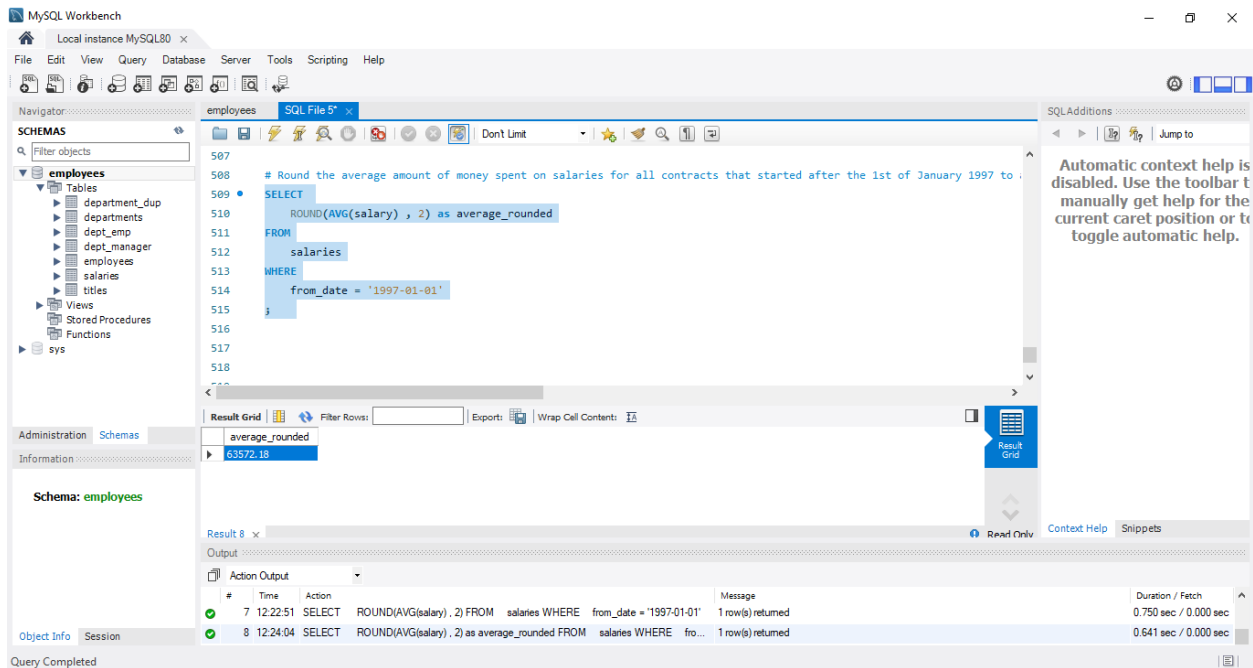
Output

Action Output

#	Time	Action	Message	Duration / Fetch
6	12:22:37	SELECT AVG(salary) FROM salaries WHERE from_date = '1997-01-01'	1 row(s) returned	0.625 sec / 0.000 sec
7	12:22:51	SELECT ROUND(AVG(salary) , 2) FROM salaries WHERE from_date = '1997-01-01'	1 row(s) returned	0.750 sec / 0.000 sec

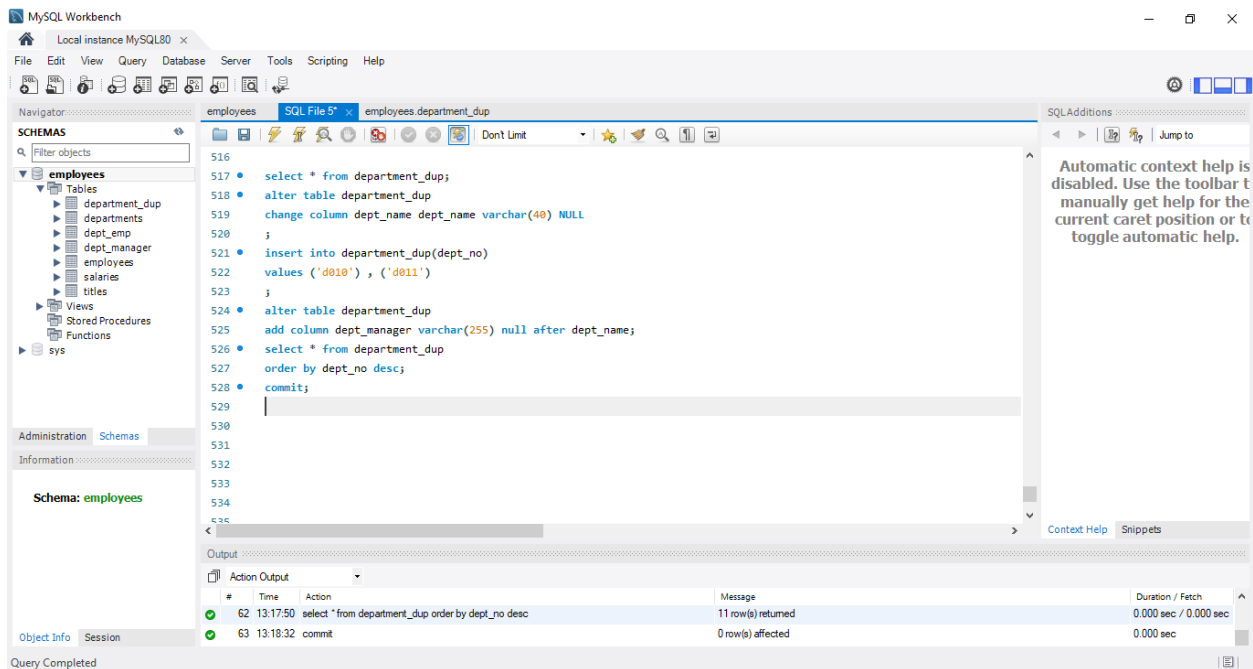
Query Completed

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



Next, we have the advanced SQL functions in the toolbox of SQL. Experts include IF NULL() and COALESCE() in this toolbox. They are utilized when your data table has scattered null values, and you want to replace the null values with another value.

Therefore, we will first modify the department_dup table so that it serves the objectives of the following video, in which we will use IF NULL() and COALESCE().



The screenshot shows the MySQL Workbench interface with a SQL script in the editor. The script creates a new table `department_dup` by copying data from `department`, then adds a new column `dept_manager` and updates some data. The result grid shows the data for `department_dup`.

```

517 • select * from department_dup;
518 • alter table department_dup
519   change column dept_name dept_name varchar(40) NULL
520   ;
521 • insert into department_dup(dept_no)
522   values ('d010') , ('d011')
523   ;
524 • alter table department_dup
525   add column dept_manager varchar(255) null after dept_name;
526 • select * from department_dup
527   order by dept_no desc;
528 • commit;

```

dept_no	dept_name	dept_manager
d006	Quality Management	NULL
d008	Research	NULL
d007	Sales	NULL
d010	NULL	NULL
d011	NULL	NULL

The output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
63	13:18:32	commit	0 row(s) affected	0.000 sec
64	12:32:42	select * from department_dup	11 row(s) returned	0.016 sec / 0.000 sec

Now, here two of the department name fields are null. We want to replace them with department name not provided instead of null, so

The screenshot shows the MySQL Workbench interface with a SQL script that updates the `dept_name` for rows where it is null. The result grid shows the updated data.

```

526 • select * from department_dup
527   order by dept_no desc;
528 • commit;
529
530 • select dept_no,
531   ifnull(dept_name , 'Department_Name not provided') as dept_name
532   from department_dup;

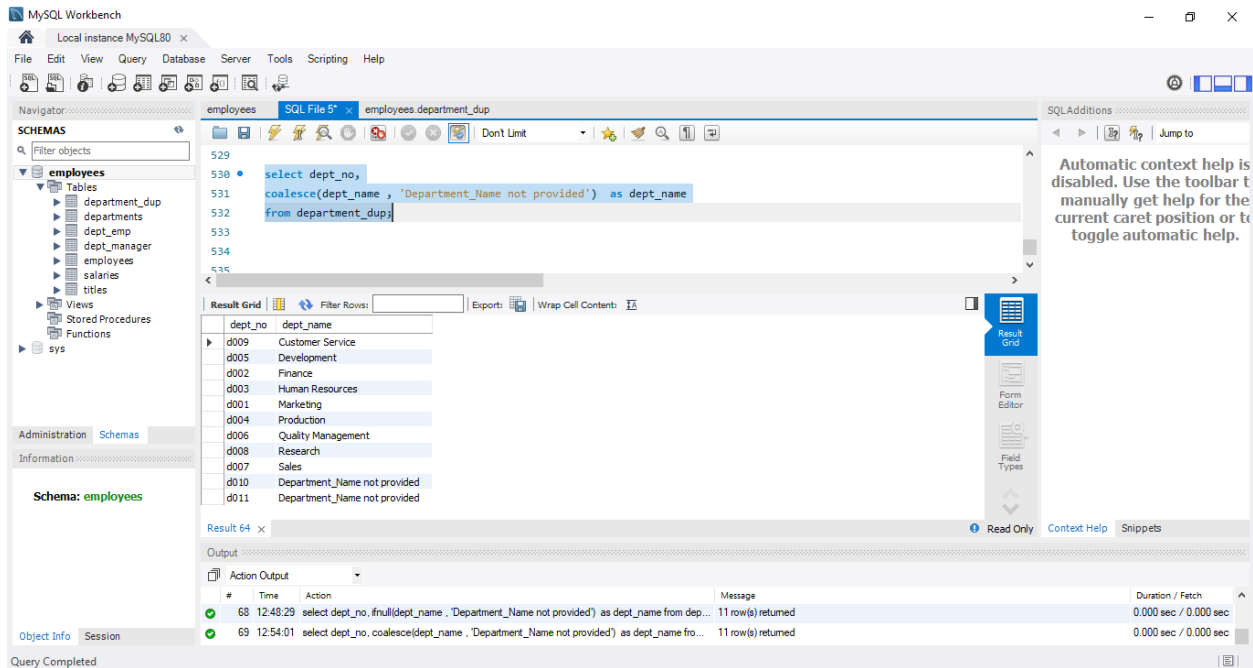
```

dept_no	dept_name
d009	Customer Service
d005	Development
d002	Finance
d003	Human Resources
d001	Marketing
d004	Production
d006	Quality Management
d008	Research
d007	Sales
d010	Department_Name not provided
d011	Department_Name not provided

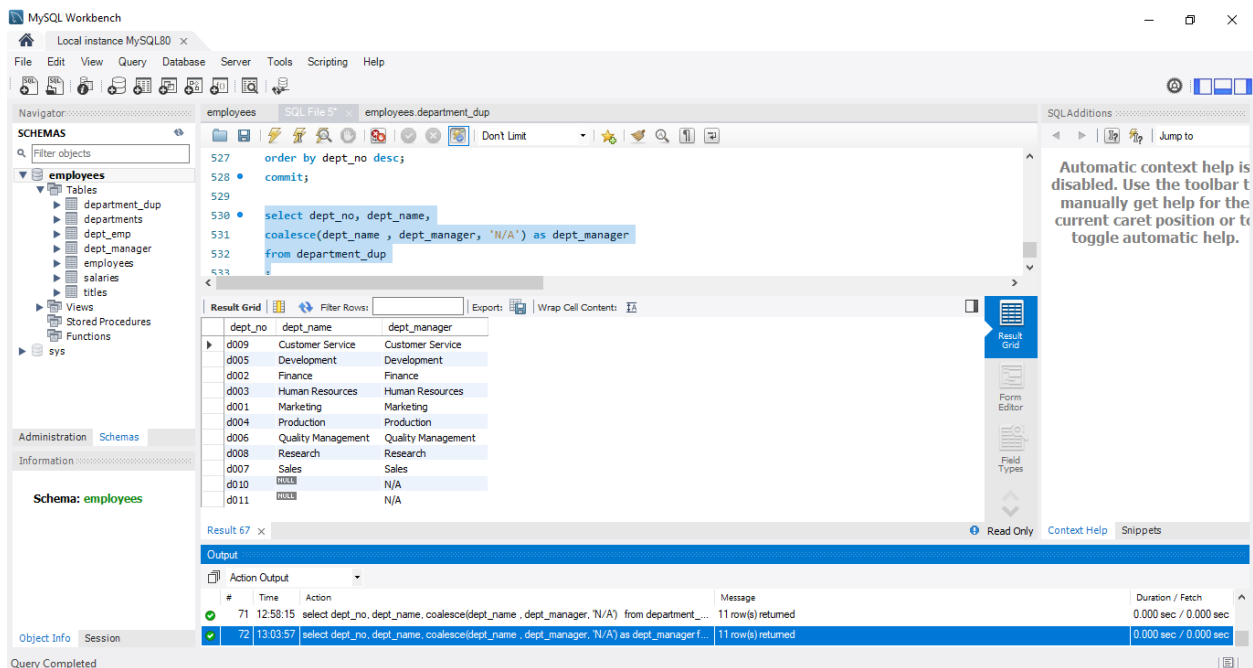
The output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
67	12:46:21	select dept_no, ifnull(dept_name , 'Department_Name not provided') as Department_Names ...	11 row(s) returned	0.000 sec / 0.000 sec
68	12:48:29	select dept_no, ifnull(dept_name , 'Department_Name not provided') as dept_name from dep ...	11 row(s) returned	0.000 sec / 0.000 sec

Note: COALESCE() = IFNULL() if there are two arguments.



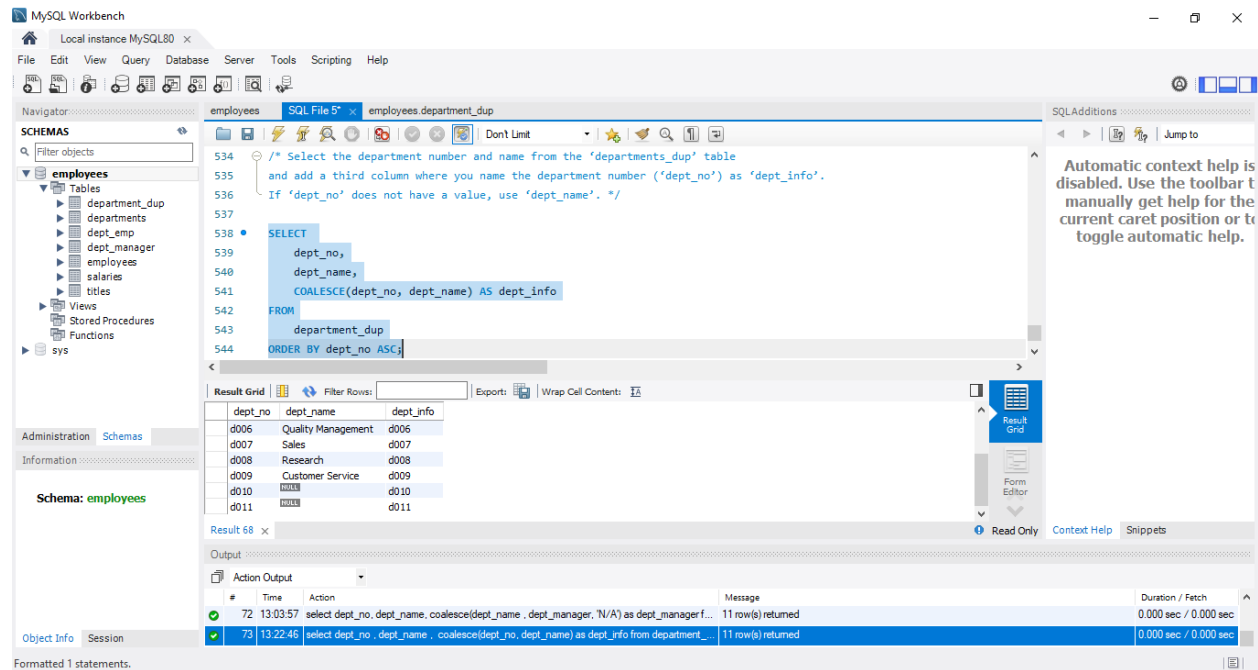
Using COALESCE() when there are three arguments.



Exercise:

Select the department number and name from the 'departments_dup' table and add a third column where you name the department number ('dept_no') as 'dept_info'. If 'dept_no' does not have a value, use 'dept_name'.

Solution:



The screenshot shows the MySQL Workbench interface. The SQL editor contains a query that selects department numbers and names from the 'departments_dup' table, using COALESCE to handle missing values. The results are displayed in a table with columns dept_no, dept_name, and dept_info.

```
534 /* Select the department number and name from the 'departments_dup' table
535 and add a third column where you name the department number ('dept_no') as 'dept_info'.
536 If 'dept_no' does not have a value, use 'dept_name'. */
537
538 SELECT
539     dept_no,
540     dept_name,
541     COALESCE(dept_no, dept_name) AS dept_info
542 FROM
543     department_dup
544 ORDER BY dept_no ASC;
```

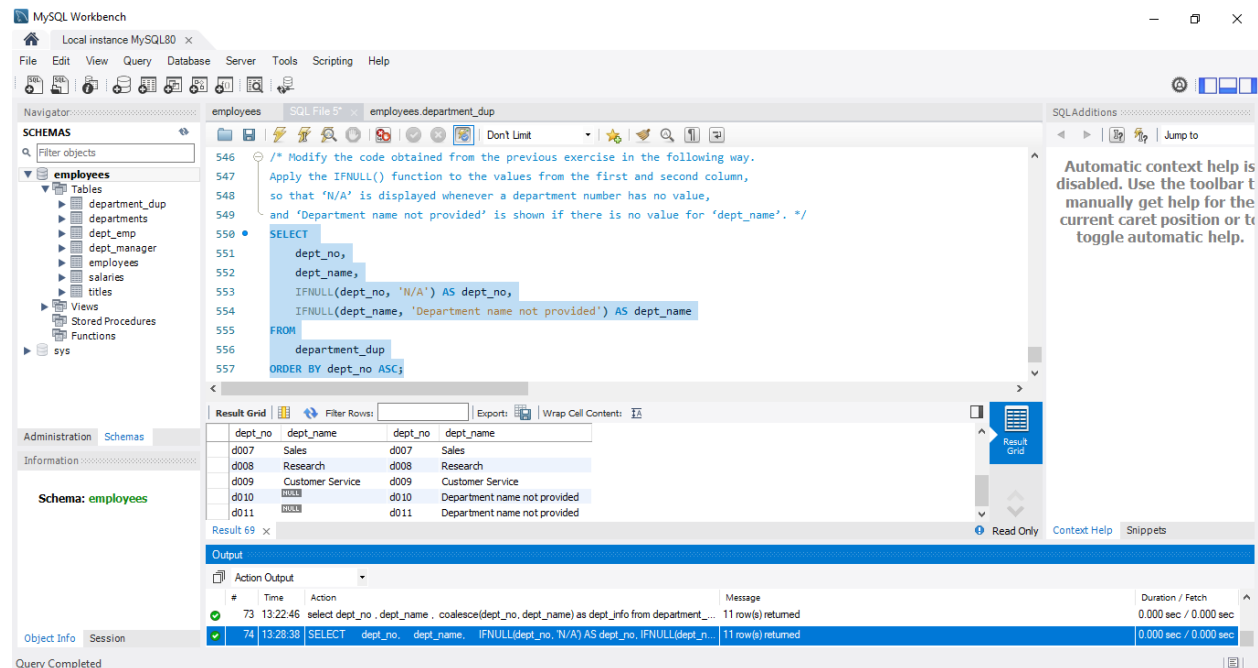
dept_no	dept_name	dept_info
d006	Quality Management	d006
d007	Sales	d007
d008	Research	d008
d009	Customer Service	d009
d010		d010
d011		d011

The output pane shows the execution of the query, with a message indicating that 11 rows were returned.

Exercise:

Modify the code obtained from the previous exercise in the following way. Apply the IFNULL() function to the values from the first and second columns so that 'N/A' is displayed whenever a department number has no value, and 'Department name not provided' is shown if there is no value for 'dept_name'.

Solution:



The screenshot shows the MySQL Workbench interface with a modified SQL query. The query uses IFNULL() to handle missing values in the first and second columns. The results are displayed in a table with columns dept_no, dept_name, dept_no, and dept_name.

```
546 /* Modify the code obtained from the previous exercise in the following way.
547 Apply the IFNULL() function to the values from the first and second column,
548 so that 'N/A' is displayed whenever a department number has no value,
549 and 'Department name not provided' is shown if there is no value for 'dept_name'. */
550
551 SELECT
552     dept_no,
553     dept_name,
554     IFNULL(dept_no, 'N/A') AS dept_no,
555     IFNULL(dept_name, 'Department name not provided') AS dept_name
556 FROM
557     department_dup
558 ORDER BY dept_no ASC;
```

dept_no	dept_name	dept_no	dept_name
d007	Sales	d007	Sales
d008	Research	d008	Research
d009	Customer Service	d009	Customer Service
d010		d010	Department name not provided
d011		d011	Department name not provided

The output pane shows the execution of the query, with a message indicating that 11 rows were returned.