

Screenshots of the code and outputs:

The screenshot shows the MySQL Workbench interface with the 'lab2_queries' schema selected. The SQL editor contains the following code:

```
59
60 • INSERT INTO professors (professor_id, first_name, last_name, email)
61 VALUES
62 (1, 'Dr. Huang', 'Xiu', 'huangxiu@scu.edu.cn'),
63 (2, 'Dr. Zhao', 'Jun', 'zhaojun@scu.edu.cn'),
64 (3, 'Dr. Linda', 'Lee', 'lindalee@scu.edu.cn'),
65 (4, 'Dr. Yuan', 'Xuedong', 'yuanxuedong@scu.edu.cn'),
66 (5, 'Dr. Guo', 'Jixiang', 'guojixiang@scu.edu.cn'),
67 (6, 'Dr. Shu', 'Li', 'shuli@scu.edu.cn');
68
69
70 • INSERT INTO enrollments (enrollment_id, student_id, course_id)
71 VALUES
72 (1, 1, 101),
73 (2, 2, 102),
74 (3, 3, 103),
75 (4, 4, 104),
76 (5, 5, 105),
77 (6, 6, 106);
```

The Output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
8	18:18:31	CREATE TABLE professors (professor_id INT PRIMARY KEY, first_name VARCHAR(10...	0 row(s) affected	0.047 sec
9	18:18:47	CREATE TABLE enrollments (enrollment_id INT PRIMARY KEY, student_id INT, co...	0 row(s) affected	0.078 sec
10	18:19:06	INSERT INTO students (student_id, first_name, last_name, email, birth_date) VALUES (1,...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.015 sec
11	18:19:17	INSERT INTO courses (course_id, course_name, credits) VALUES (101, 'Database Mana...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.015 sec
12	18:19:29	INSERT INTO professors (professor_id, first_name, last_name, email) VALUES (1, 'Dr. Hua...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.015 sec
13	18:19:39	INSERT INTO enrollments (enrollment_id, student_id, course_id) VALUES (1, 1, 101), (2...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.000 sec

- TABLES:

The screenshot shows the MySQL Workbench interface with the 'lab2_queries' schema selected. The SQL editor contains the command:

```
77 • SHOW TABLES;
```

The Output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
4	22:04:23	CREATE TABLE professors (professor_id INT PRIMARY KEY, first_name VARCHAR(1...	0 row(s) affected	0.063 sec
5	22:04:26	CREATE TABLE enrollments (enrollment_id INT PRIMARY KEY, student_id INT, co...	0 row(s) affected	0.047 sec
6	22:04:31	INSERT INTO students (student_id, first_name, last_name, email, birth_date) VALUES (1,...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.015 sec
7	22:04:36	INSERT INTO courses (course_id, course_name, credits) VALUES (101, 'Database Mana...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.015 sec
8	22:04:41	INSERT INTO professors (professor_id, first_name, last_name, email) VALUES (1, 'Dr. Hu...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.016 sec
9	22:04:45	INSERT INTO enrollments (enrollment_id, student_id, course_id) VALUES (1, 1, 101), (...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.015 sec
10	22:04:53	SHOW TABLES	4 row(s) returned	0.000 sec / 0.000 sec

The Result Grid shows the output of the SHOW TABLES command:

Tables_in_lab2_queries
courses
enrollments
professors
students

Navigator

SCHEMAS

Filter objects

lab2_queries

Tables

Views

Stored Procedures

Functions

university_db

Administration Schemas

Information

Schema: lab2_queries

Query 1

Limit to 1000 rows

```

72      (2, 2, 102),
73      (3, 3, 103),
74      (4, 4, 104),
75      (5, 5, 105),
76      (6, 6, 106);
77 • SHOW TABLES;
78 • DESCRIBE students;
79 • SELECT * FROM students;
80

```

Result Grid

Field	Type	Null	Key	Default	Extra
student_id	int	NO	PRI	NULL	
first_name	varchar(100)	YES		NULL	
last_name	varchar(100)	YES		NULL	
email	varchar(100)	YES	UNI	NULL	
birth_date	date	YES		NULL	

Result 2

Output

Action Output

#	Time	Action	Message
5	22:04:26	CREATE TABLE enrollments (enrollment_id INT PRIMARY KEY, student_id INT, co...	0 row(s) affected
6	22:04:31	INSERT INTO students (student_id, first_name, last_name, email, birth_date) VALUES (1,...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0
7	22:04:36	INSERT INTO courses (course_id, course_name, credits) VALUES (101, 'Database Man...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0

Object Info Session

- STUDENTS TABLE:

Navigator

SCHEMAS

Filter objects

lab2_queries

Tables

Views

Stored Procedures

Functions

university_db

Administration Schemas

Information

Schema: lab2_queries

Query 1

Limit to 1000 rows

```

71      (1, 1, 101),
72      (2, 2, 102),
73      (3, 3, 103),
74      (4, 4, 104),
75      (5, 5, 105),
76      (6, 6, 106);
77 • SHOW TABLES;
78 • DESCRIBE students;
79 • SELECT * FROM students;
80

```

Result Grid

student_id	first_name	last_name	email	birth_date
1	Li	Wei	liwei@gmail.com	2000-04-10
2	Wang	Fang	wangfang@gmail.com	1999-08-23
3	Chen	Li	chenli@gmail.com	2001-11-02
4	Chuan	Yao	chuanyao@gmail.com	2000-06-15
5	Fei	Jia	feijia@gmail.com	2001-02-25
6	Tang	Lizhen	tanglizhen@gmail.com	2002-11-05

students 3

Output

Action Output

#	Time	Action	Message
10	22:04:53	SHOW TABLES	4 row(s) returned
11	22:05:40	DESCRIBE students	5 row(s) returned
12	22:06:21	SELECT * FROM students LIMIT 0, 1000	6 row(s) returned

Object Info Session

- COURSE TABLE:

Navigator: Query 1 x

SCHEMAS

Filter objects

lab2_queries

Tables

Views

Stored Procedures

Functions

university_db

Administration Schemas

Information

Schema: lab2_queries

Query 1 x

Limit to 1000 rows

```

72 (2, 2, 102),
73 (3, Execute the statement under the keyboard cursor
74 (4, 4, 104),
75 (5, 5, 105),
76 (6, 6, 106);
77 SHOW TABLES;
78 DESCRIBE students;
79 SELECT * FROM students;
80 SELECT * FROM courses;

```

Result Grid

course_id	course_name	credits
101	Database Management	3
102	Data Structures	4
103	Operating Systems	4
104	System Level Programming	3
105	Computer Networks	4
106	Artificial Intelligence	3
HULL	HULL	HULL

courses 4 x

Output

Action Output

#	Time	Action	Message
11	22:05:40	DESCRIBE students	5 row(s) returned
12	22:06:21	SELECT * FROM students LIMIT 0, 1000	6 row(s) returned
13	22:08:12	SELECT * FROM courses LIMIT 0, 1000	6 row(s) returned

Object Info Session

- PROFESSORS TABLE:

Navigator: Query 1 x

SCHEMAS

Filter objects

lab2_queries

Tables

Views

Stored Procedures

Functions

university_db

Administration Schemas

Information

Schema: lab2_queries

Query 1 x

Limit to 1000 rows

```

73 (3, 3, 103),
74 (4, 4, 104),
75 (5, 5, 105),
76 (6, 6, 106);
77 SHOW TABLES;
78 DESCRIBE students;
79 SELECT * FROM students;
80 SELECT * FROM courses;
81 SELECT * FROM professors;

```

Result Grid

professor_id	first_name	last_name	email
1	Dr. Huang	Xiu	huangxiu@scu.edu.cn
2	Dr. Zhao	Jun	zhaojun@scu.edu.cn
3	Dr. Linda	Lee	lindalee@scu.edu.cn
4	Dr. Yuan	Xuedong	yuanxuedong@scu.edu.cn
5	Dr. Guo	Jixiang	guojixiang@scu.edu.cn
6	Dr. Shu	Li	shuli@scu.edu.cn
HULL	HULL	HULL	HULL

professors 5 x

Output

Action Output

#	Time	Action	Message
12	22:06:21	SELECT * FROM students LIMIT 0, 1000	6 row(s) returned
13	22:08:12	SELECT * FROM courses LIMIT 0, 1000	6 row(s) returned
14	22:09:08	SELECT * FROM professors LIMIT 0, 1000	6 row(s) returned

Object Info Session

- ENROLLMENTS TABLE:

The screenshot shows a database management interface. On the left, the 'SCHEMAS' pane displays a tree view with 'lab2_queries' selected. The main query editor shows a SQL query with line numbers 74 to 83. Below the query editor, the 'Result Grid' displays the contents of the 'enrollments' table. The table has four columns: 'enrollment_id', 'student_id', 'course_id', and 'enrollment_date'. The data rows are numbered 1 to 6. Below the table, the 'Output' pane shows a log of actions and their results.

enrollment_id	student_id	course_id	enrollment_date
1	1	101	2025-04-03 22:04:45
2	2	102	2025-04-03 22:04:45
3	3	103	2025-04-03 22:04:45
4	4	104	2025-04-03 22:04:45
5	5	105	2025-04-03 22:04:45
6	6	106	2025-04-03 22:04:45

#	Time	Action	Message
13	22:08:12	SELECT * FROM courses LIMIT 0, 1000	6 row(s) returned
14	22:09:08	SELECT * FROM professors LIMIT 0, 1000	6 row(s) returned
15	22:09:58	SELECT * FROM enrollments LIMIT 0, 1000	6 row(s) returned

- INNER JOIN:

The screenshot shows a database management interface. The main query editor displays a SQL query with line numbers 82 to 91. The query uses an INNER JOIN to combine data from the 'students' and 'courses' tables based on the 'enrollments' table. Below the query editor, the 'Result Grid' displays the results of the query. The table has three columns: 'first_name', 'last_name', and 'course_name'. The data rows are numbered 1 to 6.

```

82 • SELECT * FROM enrollments;
83
84 -- this is INNER JOIN to get list of students and their enrolled courses
85 • SELECT students.first_name, students.last_name, courses.course_name
86 FROM students
87 INNER JOIN enrollments ON students.student_id = enrollments.student_id
88 INNER JOIN courses ON enrollments.course_id = courses.course_id;
89
90 -- this is LEFT JOIN to get all students and their enrolled courses (including those not enrolled in any course)
91 • SELECT students.first_name, students.last_name, courses.course_name

```

first_name	last_name	course_name
Li	Wei	Database Management
Wang	Fang	Data Structures
Chen	Li	Operating Systems
Chuan	Yao	System Level Programming
Fei	Jia	Computer Networks
Tang	Lizhen	Artificial Intelligence

- LEFT JOIN:

Limit to 1000 rows

```

87 INNER JOIN enrollments ON students.student_id = enrollments.student_id
88 INNER JOIN courses ON enrollments.course_id = courses.course_id;
89
90 -- this is LEFT JOIN to get all students and their enrolled courses
91 • SELECT students.first_name, students.last_name, courses.course_name
92 FROM students
93 LEFT JOIN enrollments ON students.student_id = enrollments.student_id
94 LEFT JOIN courses ON enrollments.course_id = courses.course_id;
95
96 -- UPDATE: Update the email of Dr. Zhao Jun

```

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

first_name	last_name	course_name
Li	Wei	Database Management
Wang	Fang	Data Structures
Chen	Li	Operating Systems
Chuan	Yao	System Level Programming
Fei	Jia	Computer Networks
Tang	Lizhen	Artificial Intelligence

Result 12 x Read Only

- UPDATE:

Query 1 x

Limit to 1000 rows

```

93 LEFT JOIN enrollments ON students.student_id = enrollments.student_id
94 LEFT JOIN courses ON enrollments.course_id = courses.course_id;
95
96 -- this is to UPDATE the email of Dr. Zhao Jun
97 • UPDATE professors
98 SET email = 'zhaojun_1001@scu.edu.cn'
99 WHERE professor_id = 2;
100

```

Output

Action Output

#	Time	Action	Message
✓ 10	22:04:53	SHOW TABLES	4 row(s) returned
✓ 11	22:05:40	DESCRIBE students	5 row(s) returned
✓ 12	22:06:21	SELECT * FROM students LIMIT 0, 1000	6 row(s) returned
✓ 13	22:08:12	SELECT * FROM courses LIMIT 0, 1000	6 row(s) returned
✓ 14	22:09:08	SELECT * FROM professors LIMIT 0, 1000	6 row(s) returned
✓ 15	22:09:58	SELECT * FROM enrollments LIMIT 0, 1000	6 row(s) returned
✓ 16	22:12:12	SELECT students.first_name, students.last_name, courses.course_name FROM students IN...	6 row(s) returned
✓ 17	22:12:33	SELECT students.first_name, students.last_name, courses.course_name FROM students IN...	6 row(s) returned
✓ 18	22:12:36	SELECT students.first_name, students.last_name, courses.course_name FROM students IN...	6 row(s) returned
✓ 19	22:12:42	SELECT students.first_name, students.last_name, courses.course_name FROM students IN...	6 row(s) returned
✓ 20	22:12:48	SELECT students.first_name, students.last_name, courses.course_name FROM students IN...	6 row(s) returned
✓ 21	22:13:56	SELECT students.first_name, students.last_name, courses.course_name FROM students LE...	6 row(s) returned
✓ 22	22:15:24	UPDATE professors SET email = 'zhaojun_1001@scu.edu.cn' WHERE professor_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
✓ 23	22:15:30	UPDATE professors SET email = 'zhaojun_1001@scu.edu.cn' WHERE professor_id = 2	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0

- DELETE:

```

101 -- DELETE this is to remove enrollment for student with student_id = 3
102 • DELETE FROM enrollments
103 WHERE student_id = 3;
104

```

Output

Action Output

#	Time	Action	Message
✓ 15	22:09:58	SELECT * FROM enrollments LIMIT 0, 1000	6 row(s) returned
✓ 16	22:12:12	SELECT students.first_name, students.last_name, courses.course_name FROM students IN...	6 row(s) returned
✓ 17	22:12:33	SELECT students.first_name, students.last_name, courses.course_name FROM students IN...	6 row(s) returned
✓ 18	22:12:36	SELECT students.first_name, students.last_name, courses.course_name FROM students IN...	6 row(s) returned
✓ 19	22:12:42	SELECT students.first_name, students.last_name, courses.course_name FROM students IN...	6 row(s) returned
✓ 20	22:12:48	SELECT students.first_name, students.last_name, courses.course_name FROM students IN...	6 row(s) returned
✓ 21	22:13:56	SELECT students.first_name, students.last_name, courses.course_name FROM students LE...	6 row(s) returned
✓ 22	22:15:24	UPDATE professors SET email = 'zhaojun_1001@scu.edu.cn' WHERE professor_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
✓ 23	22:15:30	UPDATE professors SET email = 'zhaojun_1001@scu.edu.cn' WHERE professor_id = 2	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0
✓ 24	22:16:48	DELETE FROM enrollments WHERE student_id = 3	1 row(s) affected

AGGREGATION:

```

104
105 -- Aggregation to count the number of students enrolled per course
106 • SELECT courses.course_name, COUNT(enrollments.student_id) AS student_count
107 FROM courses
108 LEFT JOIN enrollments ON courses.course_id = enrollments.course_id
109 GROUP BY courses.course_name
110 HAVING COUNT(enrollments.student_id) > 1;
111

```

course_name	student_count
-------------	---------------

Result 13 x

• SUBQUERY:

Query 1

```

112 -- Subquery to get students who are NOT enrolled in any course
113 • SELECT first_name, last_name
114 FROM students
115 WHERE student_id NOT IN (
116 SELECT student_id
117 FROM enrollments
118 )

```

first_name	last_name
Chen	Li

students 14 x

Output

#	Time	Action	Message
✓ 1	22:25:43	SELECT first_name, last_name FROM students WHERE student_id NOT IN (SELECT stud...	1 row(s) returned