

firstly create the database :-

The screenshot shows the MySQL Workbench interface. In the SQL editor, the command `create database join2;` is being typed. The output pane shows the result of the command: `1 row(s) affected` in 0.063 seconds. A tooltip in the top right corner says: "Automatic context help is disabled. Use the toolbar or manually get help for this." The status bar at the bottom indicates the current cursor position.

```
1 -- firstly create the database
2
3 • create database join2;
```

Action Output	#	Time	Action	Message	Duration / Fetch
	1	10:22:40	create database join2	1 row(s) affected	0.063 sec

use the join2 database :-

The screenshot shows the MySQL Workbench interface. In the SQL editor, the command `use join2;` is being typed. The output pane shows the result of the command: `0 row(s) affected` in 0.031 seconds. A tooltip in the top right corner says: "Current cursor position or click to toggle automatic help." The status bar at the bottom indicates the current cursor position.

```
5 -- use the join2 database
6
7 • use join2;
```

Action Output	#	Time	Action	Message	Duration / Fetch
	1	10:22:40	create database join2	1 row(s) affected	0.063 sec
	2	10:24:34	use join2	0 row(s) affected	0.031 sec

create the table students :-

The screenshot shows the MySQL Workbench interface. In the SQL editor, the command `CREATE TABLE students(` is being typed, followed by the table structure. The output pane shows the results of the CREATE TABLE command: `0 row(s) affected` in 0.156 seconds. A tooltip in the top right corner says: "Context Help Snippets". The status bar at the bottom indicates the current cursor position.

```
9 -- create the table students
10
11 • ◇ CREATE TABLE students(
12     student_id INT PRIMARY KEY,
13     name VARCHAR(50),
14     course_id INT
15 );
```

Action Output	#	Time	Action	Message	Duration / Fetch
	1	10:22:40	create database join2	1 row(s) affected	0.063 sec
	2	10:24:34	use join2	0 row(s) affected	0.031 sec
	3	10:32:08	CREATE TABLE students(student_id INT PRIMARY KEY, name VARCHAR(50), course_id INT)	0 row(s) affected	0.156 sec

create the table courses :-

The screenshot shows the MySQL Workbench interface. In the SQL editor, the command `CREATE TABLE courses(` is being typed, followed by the table structure. The output pane shows the results of the CREATE TABLE command: `0 row(s) affected` in 0.047 seconds. A tooltip in the top right corner says: "Context Help Snippets". The status bar at the bottom indicates the current cursor position.

```
17
18 -- create the table courses
19
20 • ◇ CREATE TABLE courses(
21     course_id INT PRIMARY KEY,
22     course_name VARCHAR(50)
23 );
```

Action Output	#	Time	Action	Message	Duration / Fetch
	1	10:22:40	create database join2	1 row(s) affected	0.063 sec
	2	10:24:34	use join2	0 row(s) affected	0.031 sec
	3	10:32:08	CREATE TABLE students(student_id INT PRIMARY KEY, name VARCHAR(50), course_id INT)	0 row(s) affected	0.156 sec
	4	10:36:06	CREATE TABLE courses(course_id INT PRIMARY KEY, course_name VARCHAR(50))	0 row(s) affected	0.047 sec

create the table payments :-

The screenshot shows the MySQL Workbench interface. In the SQL editor, the following SQL code is written:

```
-- create the table payments
CREATE TABLE payments(
    payment_id INT PRIMARY KEY,
    student_id INT,
    amount INT
);
```

The code is highlighted with syntax coloring. The output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	10:22:40	create database join2	1 row(s) affected	0.063 sec
2	10:24:34	use join2	0 row(s) affected	0.031 sec
3	10:32:08	CREATE TABLE students(student_id INT PRIMARY KEY, name VARCHAR(50), course_id INT)	0 row(s) affected	0.156 sec
4	10:36:06	CREATE TABLE courses(course_id INT PRIMARY KEY, course_name VARCHAR(50))	0 row(s) affected	0.047 sec
5	10:48:35	CREATE TABLE payments(payment_id INT PRIMARY KEY, student_id INT, amount INT)	0 row(s) affected	0.031 sec

show all the tables in the databases join2 :-

The screenshot shows the MySQL Workbench interface. In the SQL editor, the following SQL code is written:

```
-- show all the tables in the databases join2
SHOW TABLES;
```

The code is highlighted with syntax coloring. The output pane shows the execution results:

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

Tables_in_join2
courses
payments
students

Result 1 x | Output: | Action Output | # | Time | Action | Message | Duration / Fetch

#	Time	Action	Message	Duration / Fetch
3	10:32:08	CREATE TABLE students(student_id INT PRIMARY KEY, name VARCHAR(50), course_id INT)	0 row(s) affected	0.156 sec
4	10:36:06	CREATE TABLE courses(course_id INT PRIMARY KEY, course_name VARCHAR(50))	0 row(s) affected	0.047 sec
5	10:48:35	CREATE TABLE payments(payment_id INT PRIMARY KEY, student_id INT, amount INT)	0 row(s) affected	0.031 sec
6	10:50:05	SHOW TABLES	3 row(s) returned	0.015 sec / 0.000 sec

insert values into students table :-

The screenshot shows the MySQL Workbench interface. In the SQL editor, the following SQL code is written:

```
-- insert values into students table
INSERT INTO students VALUES
(1,"Swapnil",101),
(2,"Vijay",102),
(3,"Ram",101),
(4,"Seeta",NULL),
(5,"Angel",NULL),
(6,"Geeta",NULL);
```

The code is highlighted with syntax coloring. The output pane shows the execution results:

Output: | Action Output | # | Time | Action | Message | Duration / Fetch

#	Time	Action	Message	Duration / Fetch
7	10:54:58	INSERT INTO students values (1,"Amit",101), (1,"Amit",101), (1,"Amit",101), (2,"Vijay",102), (3,"Ram",101), (4,"Seeta",NULL), (5,"An...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
8	10:58:06	INSERT INTO students VALUES (1,"Swapnil",101), (2,"Vijay",102), (3,"Ram",101), (4,"Seeta",NULL), (5,"An...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.031 sec

insert values into courses table :-

```
50
51 -- insert values into courses table
52
53 • INSERT INTO courses VALUES
54   (101,"Data Science"),
55   (102,"Machine Learning"),
56   (103,"Data Analytics");
57
```

Output

#	Time	Action	Message	Duration / Fetch
8	10:58:06	INSERT INTO courses VALUES (101,"Data Science"),(102,"Machine Learning"),(103,"Data Analytics")	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.031 sec
9	11:02:03	INSERT INTO courses VALUES (101,"Data Science"),(102,"Machine Learning"),(103,"Data Analytics")	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec

insert values into payments table :-

```
58
59 -- insert values into payments table
60
61 • INSERT INTO payments VALUES
62   (1,1,5000),
63   (2,2,6000),
64   (3,1,2000);
65
```

Output

#	Time	Action	Message	Duration / Fetch
9	11:02:03	INSERT INTO payments VALUES (1,1,5000),(2,2,6000),(3,1,2000)	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec
10	11:05:58	INSERT INTO payments VALUES (1,1,5000),(2,2,6000),(3,1,2000)	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec

select all values from table students :-

```
-- select all value from table students
66
67
68 • select * from students;
```

Output

student_id	name	course_id
1	Swapnil	101
2	Vijay	102
3	Ram	101
4	Seeta	NULL
5	Angel	NULL

Output

#	Time	Action	Message	Duration / Fetch
11	11:05:58	INSERT INTO payments VALUES (1,1,5000),(2,2,6000),(3,1,2000)	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec
11	11:12:17	select * from students LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

select all values from table courses :-

```
-- select all value from table courses
71
72
73 • select * from courses;
```

Output

course_id	course_name
101	Data Science
102	Machine Learning
103	Data Analytics
NULL	NULL

Output

#	Time	Action	Message	Duration / Fetch
11	11:12:17	select * from courses LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
12	11:12:57	select * from courses LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

select all values from table payments :-

```
76 -- select all values from table payments
77
78 • SELECT * FROM courses
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Context Help | Snippets

course_id	course_name
101	Data Science
102	Machine Learning
103	Data Analytics
NULL	NULL

courses 4 x

Action Output

#	Time	Action	Message	Duration / Fetch
12	11:12:57	selected * from courses LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
13	11:14:29	SELECT * FROM courses LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Join/inner join :-

```
80
81 -- now joins
82 -- join is also called inner join
83 -- it gives only matching condition
84 -- join
85 • SELECT s.name , c.course_name
86   from students as s
87   join courses as c on s.course_id = c.course_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Context Help | Snippets

name	course_name
Swapnil	Data Science
Vijay	Machine Learning
Ram	Data Science

Result 5 x

Action Output

#	Time	Action	Message	Duration / Fetch
13	11:14:29	SELECT * FROM courses LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
14	11:20:45	SELECT s.name , c.course_name from students as s join courses as c on s.course_id = c.course_id LIMIT 0, ...	3 row(s) returned	0.000 sec / 0.000 sec

Right join - All from right table and left table value not then use null :-

```
89
90 -- Right join - All from right table and left table value not then use null
91
92 • SELECT s.name , c.course_name
93   FROM students AS s
94   RIGHT JOIN courses AS c ON s.course_id = c.course_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Context Help | Snippets

name	course_name
Ram	Data Science
Swapnil	Data Science
Vijay	Machine Learning
NULL	Data Analytics

Result 6 x

Action Output

#	Time	Action	Message	Duration / Fetch
14	11:20:45	SELECT s.name , c.course_name from students as s join courses as c on s.course_id = c.course_id LIMIT 0, ...	3 row(s) returned	0.000 sec / 0.000 sec
15	11:24:20	SELECT s.name , c.course_name FROM students AS s RIGHT JOIN courses AS c ON s.course_id = c.cours...	4 row(s) returned	0.000 sec / 0.000 sec

```

97    -- Right join - All from right table and left table value not then use null
98
99 •  SELECT c.course_name , s.name
100   FROM students AS s
101  RIGHT JOIN courses AS c ON s.course_id = c.course_id;
102

```

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

course_name	name
Data Science	Ram
Data Science	Swapnil
Machine Learning	Vijay
Data Analytics	NULL

Action Output

#	Time	Action	Message	Duration / Fetch
15	11:24:20	SELECT s.name , c.course_name FROM students AS s RIGHT JOIN courses AS c ON s.course_id = c.course_id;	4 row(s) returned	0.000 sec / 0.000 sec
16	11:29:42	SELECT c.course_name , s.name FROM students AS s RIGHT JOIN courses AS c ON s.course_id = c.course_id;	4 row(s) returned	0.000 sec / 0.000 sec

Left join - All selected column from left table and right side table value not then use null:-

```

104
105  -- Left join - All selected column from left table and right side table value not then use null
106
107 •  SELECT s.name , c.course_name
108   FROM students AS s
109  LEFT JOIN courses AS c ON s.course_id = c.course_id;
110

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid | Form Editor | Read Only | Context Help | Snippets

name	course_name
Swapnil	Data Science
Vijay	Machine Learning
Ram	Data Science
Seeta	NULL
Angel	NULL
Geeta	NULL

Action Output

#	Time	Action	Message	Duration / Fetch
18	11:36:16	SELECT s.name , p.amount FROM students AS s LEFT JOIN payments AS p ON s.student_id = p.student_id ...	0 row(s) returned	0.015 sec / 0.000 sec
19	11:38:42	SELECT s.name , c.course_name FROM students AS s LEFT JOIN courses AS c ON s.course_id = c.course_id;	6 row(s) returned	0.000 sec / 0.000 sec

full Outer join = combine the left + right join :-

```

112  -- full join = combine the left + right join
113
114 •  SELECT s.name , c.course_name
115   FROM students AS s
116  LEFT JOIN courses AS c ON s.course_id = c.course_id
117
118 UNION
119
120 SELECT c.course_name , s.name
121   FROM students AS s
122  RIGHT JOIN courses AS c ON s.course_id = c.course_id;
123

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid | Form Editor | Read Only | Context Help | Snippets

name	course_name
Swapnil	Data Science
Vijay	Machine Learning
Ram	Data Science
Seeta	NULL
Angel	NULL
Geeta	NULL
Data Science	Ram
Data Science	Swapnil
Machine Learning	Vijay

Action Output

#	Time	Action	Message	Duration / Fetch
19	11:38:42	SELECT s.name , c.course_name FROM students AS s LEFT JOIN courses AS c ON s.course_id = c.course_id;	6 row(s) returned	0.000 sec / 0.000 sec
20	11:49:31	SELECT s.name , c.course_name FROM students AS s LEFT JOIN courses AS c ON s.course_id = c.course_id;	10 row(s) returned	0.016 sec / 0.000 sec

i want those student we not paid amount :-

use left join :-

```
124
125 -- i want those student we not paid amount
126 -- use left join
127
128 • SELECT s.name , p.amount
129   FROM students AS s
130   LEFT JOIN payments AS p ON s.student_id = p.student_id
131 WHERE p.amount is null;
132
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Read Only | Context Help | Snippets

name	amount
Ram	NULL
Seeta	NULL
Angel	NULL
Greta	NULL

Action Output

#	Time	Action	Message	Duration / Fetch
19	11:38:42	SELECT s.name , c.course_name FROM students AS s LEFT JOIN courses AS c ON s.course_id = c.course_id ...	6 row(s) returned	0.000 sec / 0.000 sec
20	11:49:31	SELECT s.name , c.course_name FROM students AS s LEFT JOIN courses AS c ON s.course_id = c.course_id ...	10 row(s) returned	0.016 sec / 0.000 sec
21	11:59:37	SELECT s.name , p.amount FROM students AS s LEFT JOIN payments AS p ON s.student_id = p.student_id ...	Error Code: 1146. Table 'join2.payments' doesn't exist	0.031 sec
22	12:00:08	SELECT s.name , p.amount FROM students AS s LEFT JOIN payments AS p ON s.student_id = p.student_id ...	4 row(s) returned	0.000 sec / 0.000 sec