

# How to Add a Column to the Beginning of a MySQL Table

Authored by  
**mohammed looti**

January 6, 2026

## RECOMMENDED CITATION

mohammed looti (2026). *How to Add a Column to the Beginning of a MySQL Table*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=124728>

To add a column in the first position in MySQL, you can use the "ALTER TABLE" command followed by the "ADD COLUMN" keyword. Within the parentheses, specify the name and data type of the new column. To add it in the first position, use the "FIRST" keyword after the column name. This will insert the new column at the beginning of the table, shifting all other columns to the right. Additionally, you can specify other options such as default values or constraints for the new column.

You can use the following syntax in MySQL to add a column to the first position in a table:

```
ALTER TABLE athletes ADD COLUMN rebounds INT NOT NULL FIRST;
```

This particular example adds an integer column named **rebounds** to the first position of the table named **athletes**.

**Note:** The **FIRST** statement at the end of the line tells MySQL that the new column should be in the first position of the table.

The following example shows how to use this syntax in practice.

## Example: Add Column in First Position in MySQL

Suppose we create the following table named **athletes** that contains information about various basketball players:

```
-- create table
```

```
CREATE TABLE athletes (  
athleteID INT PRIMARY KEY,  
team TEXT NOT NULL,  
points INT NOT NULL  
);
```

```
-- insert rows into table
```

```
INSERT INTO athletes VALUES (0001, 'Mavs', 22);  
INSERT INTO athletes VALUES (0002, 'Warriors', 14);  
INSERT INTO athletes VALUES (0003, 'Nuggets', 37);  
INSERT INTO athletes VALUES (0004, 'Lakers', 19);  
INSERT INTO athletes VALUES (0005, 'Celtics', 26);
```

```
-- view all rows in table
```

```
SELECT * FROM athletes;
```

**Output:**

```
+-----+-----+-----+
| athleteID | team | points |
+-----+-----+-----+
| 1 | Mavs | 22 |
| 2 | Warriors | 14 |
| 3 | Nuggets | 37 |
| 4 | Lakers | 19 |
| 5 | Celtics | 26 |
+-----+-----+-----+
```

Suppose that we would like to add a new column to the table named **rebounds** in the first position of the table.

We can use the following syntax to do so:

```
-- add rebounds column to first position in table
ALTER TABLE athletes ADD COLUMN rebounds INT NOT NULL FIRST;

-- view all rows in updated table
SELECT * FROM athletes;
```

**Output:**

```
+-----+-----+-----+-----+
| rebounds | athleteID | team | points |
+-----+-----+-----+-----+
| 0 | 1 | Mavs | 22 |
| 0 | 2 | Warriors | 14 |
| 0 | 3 | Nuggets | 37 |
| 0 | 4 | Lakers | 19 |
| 0 | 5 | Celtics | 26 |
+-----+-----+-----+-----+
```

Notice that the new **rebounds** column is in the first position of the table.

You can also add multiple columns to the front of the table by using the following syntax:

```
-- add rebounds and assists columns to front positions in table
```

```
ALTER TABLE athletes
ADD COLUMN assists INT NOT NULL FIRST,
ADD COLUMN rebounds INT NOT NULL FIRST;
```

```
-- view all rows in updated table
SELECT * FROM athletes;
```

**Output:**

```
+-----+-----+-----+-----+-----+
| rebounds | assists | athleteID | team | points |
+-----+-----+-----+-----+-----+
| 0 | 0 | 1 | Mavs | 22 |
| 0 | 0 | 2 | Warriors | 14 |
| 0 | 0 | 3 | Nuggets | 37 |
| 0 | 0 | 4 | Lakers | 19 |
| 0 | 0 | 5 | Celtics | 26 |
+-----+-----+-----+-----+-----+
```

Notice that the **rebounds** and **assists** columns have both been added to the front of the table.

Note that the new columns will appear in the reverse order that you specified them.

For example, we specified **rebounds** as the last new column to add but it appears in the first position of the table.

The following tutorials explain how to perform other common tasks in MySQL: