Introduction to MySQL JOINs
One of the most common tasks of MySQL database analysts and
administrators
is extracting records from several tables according to specific conditions.

It could be quite a sophisticated operation requiring
you to write and execute multiple
queries. Or, you can use JOINs.

JOINs help you fetch data from several tables, which share a
common field,
 and get a single result. As MySQL tables are connected logically and linked
with a common key value,
 JOINs can provide you with the data you need from several tables by
using just one query.

 Why MySQL JOINs are helpful:

Speed
JOINs allow you to use a
single query instead of many simple queries run one by one.
 The result delivered is the
same.

Effectiveness
In MySQL, JOIN clauses perform much better because they work through
indexing.

Less load
JOINs help users reduce the number of data transfers between a MySQL
server and a target application.
 Respectively, it reduces the server load.

Database
developers, admins, and analysts use JOINs every day — to combine sets of data,
 process and
find data patterns in vast databases, normalize table structures, etc. Therefore,
 anyone who
works with databases as part of their functional role needs a solid understanding of how to
apply this clause.

This white paper is created to equip you with the best professional
practices for using JOINs
and help you master them quickly and easily.

What are the
supported types of JOINs in MySQL?
MySQL supports the following JOIN types: INNER JOIN, OUTER
JOIN (further divided into LEFT JOIN and RIGHT JOIN),
 SELF JOIN, and CROSS JOIN. The type of
the JOIN defines how tables are related in a query.
You can join more than two
tables.

INNER JOIN
Compares all rows from all tables specified in a query with each other
and
 returns records with matching values as a new result table.

 LEFT OUTER JOIN
Returns
the result table with all rows from the left table and only those rows from the right table
that match the
JOIN condition (non-matching rows are returned with NULLs).

RIGHT OUTER
JOIN
Returns the result table with all rows from the right table and only those rows from the
left table where the
 JOIN condition is fulfilled (non-matching rows are returned
with

NULLs).

CROSS JOIN
Combines each row from one table with each row from another table and
thus returns a new result table with
 all possible row combinations
from each table

FULL
OUTER JOIN
MySQL does not directly support FULL OUTER JOIN that returns both matching and
non-matching rows from the joined tables,
but you can combine LEFT and RIGHT OUTER JOINs to
achieve the same result.

SELF JOIN
Compares a row with other rows within the same table or
extracts hierarchical
data - table aliases should be used
instead of repeating the same
table name in a query.