

# MySQL - Understanding Database Query Performance Analysis

**Objective:** To understand one of the database query performance analysis techniques using EXPLAIN plan



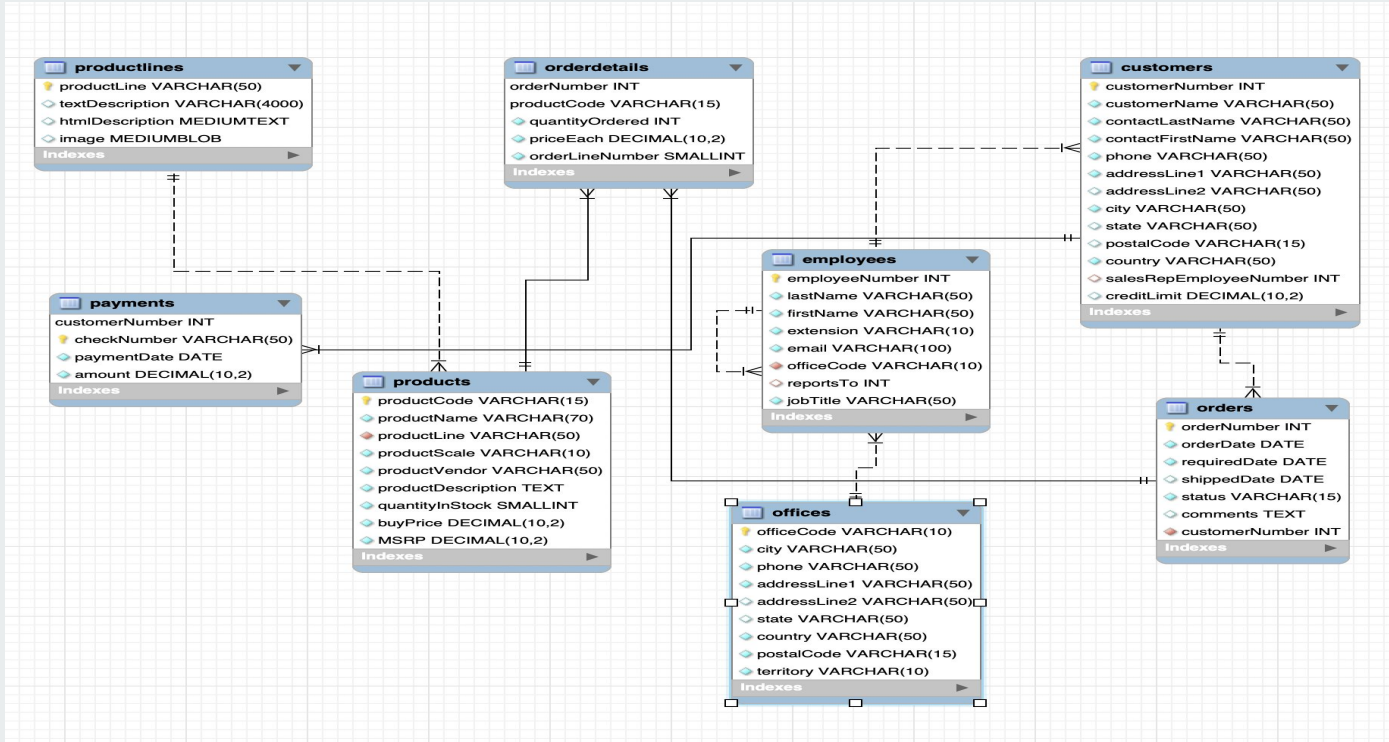
## Table of Contents

- Use a sample database with preloaded entities.
- Consider a use case that will allow us to write a query with multiple entities joined.
- Go through the EXPLAIN plan.
- Multiple ways of EXPLAIN plan execution.
- Understanding each component of EXPLAIN plan output.
- Summary



# MySQL - Understanding Database Query Performance Analysis

Topic: Use a sample database with preloaded entities



# MySQL - Understanding Database Query Performance Analysis

**Topic:** Consider a use case that will allow us to write a query with multiple entities joined

**Use case:** Using the above schema, let's write a query that would produce the customer name, product name, order number, order date, and quantity ordered.

**Query:**

```
SELECT
    c.customerName AS "Customer Name",
    p.productName AS "Product Name",
    o.orderNumber AS "Order Number",
    o.orderDate AS "Order Date",
    od.quantityOrdered AS "Quantity Ordered"
FROM
    products p, orderdetails od, orders o, customers c
WHERE od.productCode = p.productCode
AND od.orderNumber = o.orderNumber
AND c.customerNumber = o.customerNumber;
```



# MySQL - Understanding Database Query Performance Analysis

**Topic:** Go through the EXPLAIN plan



*The EXPLAIN statement provides information about how MySQL executes statements:*

- EXPLAIN works with SELECT, DELETE, INSERT, REPLACE, and UPDATE statements. (Only SELECT will be discussed)
- When EXPLAIN is used with an explainable statement, MySQL displays information from the optimizer about the statement execution plan. That is, MySQL explains how it would process the statement, including information about how tables are joined and in which order.
- EXPLAIN is useful for examining queries involving partitioned tables. (Not covered in this tutorial)
- The FORMAT option can be used to select the output format. TRADITIONAL presents the output in tabular format. This is the default if no FORMAT option is present. JSON format displays the information in JSON format

**Topic:** Multiple ways of EXPLAIN plan execution.

- Using MySQL Workbench:



# MySQL - Understanding Database Query Performance Analysis

Topic: Multiple ways of EXPLAIN plan execution.

- Using default command: `EXPLAIN <SELECT statement>`

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' panel is open, showing the 'classicmodels' database selected. A red arrow points to the 'classicmodels' schema with the text 'Select Schema'. The main editor displays a SQL query starting with 'EXPLAIN SELECT'. A red arrow points to the query text with the label 'Query'. Below the query, the 'Result Grid' shows the EXPLAIN output in tabular format. A red arrow points to the output table with the text 'EXPLAIN output in tabular format'. The output table has columns: id, select\_ty..., table, partitions, type, possible\_keys, key, key\_len, ref, rows, filtered, and Extra. The first row shows a SIMPLE query on table 'o' with a PRIMARY key on 'customerNumber'.

```
20  /*
21  #1. Use EXPLAIN before SELECT or allowed DML to see the output in tabular format
22  */
23  EXPLAIN SELECT
24    c.customerName AS "Customer Name",
25    p.productName AS "Product Name",
26    o.orderNumber AS "Order Number",
27    o.orderDate AS "Order Date",
28    od.quantityOrdered AS "Quantity Ordered"
29  FROM
30    products p, orderdetails od, orders o, customers c
31  WHERE od.productCode = p.productCode
32        AND od.orderNumber = o.orderNumber
33        AND c.customerNumber = o.customerNumber;
34
35  -- Query ends here
36
```

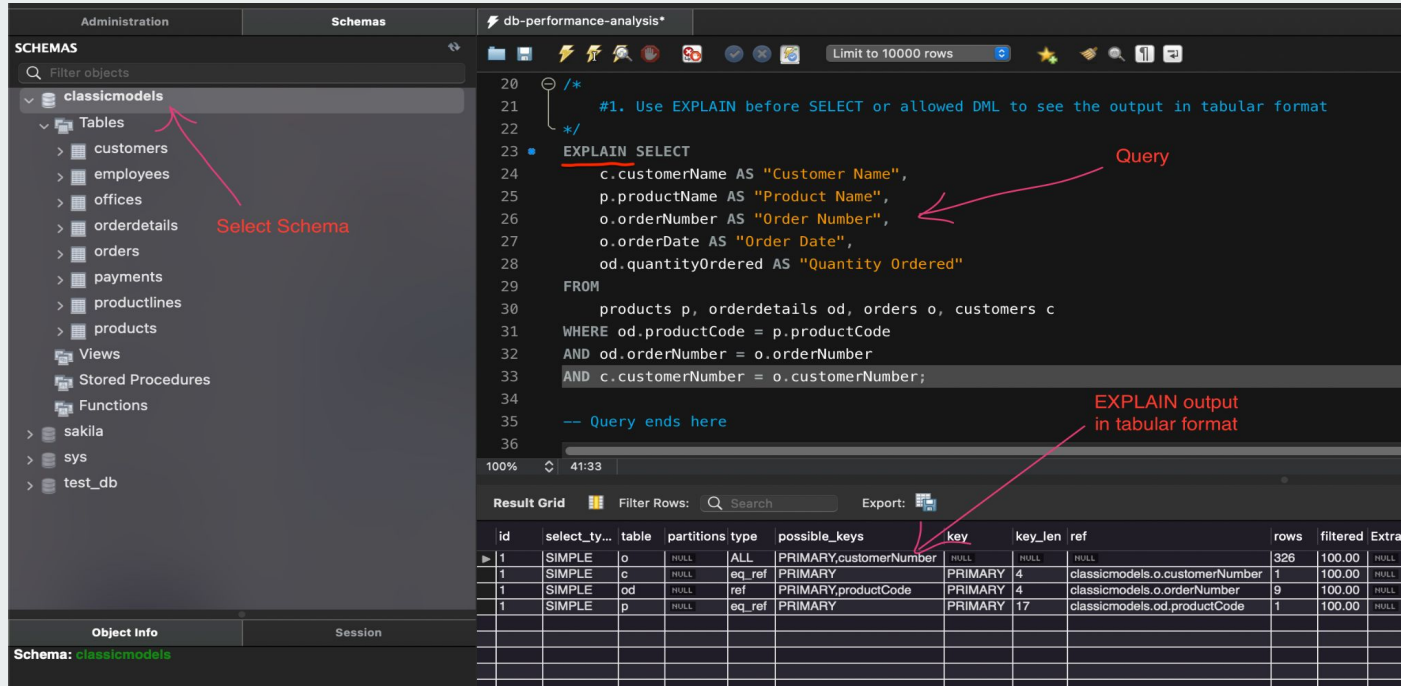
id	select_ty...	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	o	HULL	ALL	PRIMARY:customerNumber	HULL	HULL	HULL	326	100.00	HULL
1	SIMPLE	c	HULL	eq_ref	PRIMARY	PRIMARY	4	classicmodels.o.customerNumber	1	100.00	HULL
1	SIMPLE	od	HULL	ref	PRIMARY:productCode	PRIMARY	4	classicmodels.o.orderNumber	9	100.00	HULL
1	SIMPLE	p	HULL	eq_ref	PRIMARY	PRIMARY	17	classicmodels.od.productCode	1	100.00	HULL



# MySQL - Understanding Database Query Performance Analysis

Topic: Multiple ways of EXPLAIN plan execution.

- Using default command: `EXPLAIN <SELECT statement>`



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'classicmodels' database selected. The main editor displays a query starting with `EXPLAIN SELECT`. The query selects columns from the 'customers', 'products', 'orders', and 'orderdetails' tables. The EXPLAIN output is shown in a tabular format at the bottom.

**Query:**

```
20  /*
21  #1. Use EXPLAIN before SELECT or allowed DML to see the output in tabular format
22  */
23  EXPLAIN SELECT
24      c.customerName AS "Customer Name",
25      p.productName AS "Product Name",
26      o.orderNumber AS "Order Number",
27      o.orderDate AS "Order Date",
28      od.quantityOrdered AS "Quantity Ordered"
29  FROM
30      products p, orderdetails od, orders o, customers c
31  WHERE od.productCode = p.productCode
32  AND od.orderNumber = o.orderNumber
33  AND c.customerNumber = o.customerNumber;
34
35  -- Query ends here
36
```

**EXPLAIN output in tabular format:**

id	select_ty...	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	o	HULL	ALL	PRIMARY:customerNumber	HULL	HULL	HULL	326	100.00	HULL
1	SIMPLE	c	HULL	eq_ref	PRIMARY	PRIMARY	4	classicmodels.o.customerNumber	1	100.00	HULL
1	SIMPLE	od	HULL	ref	PRIMARY:productCode	PRIMARY	4	classicmodels.o.orderNumber	9	100.00	HULL
1	SIMPLE	p	HULL	eq_ref	PRIMARY	PRIMARY	17	classicmodels.od.productCode	1	100.00	HULL





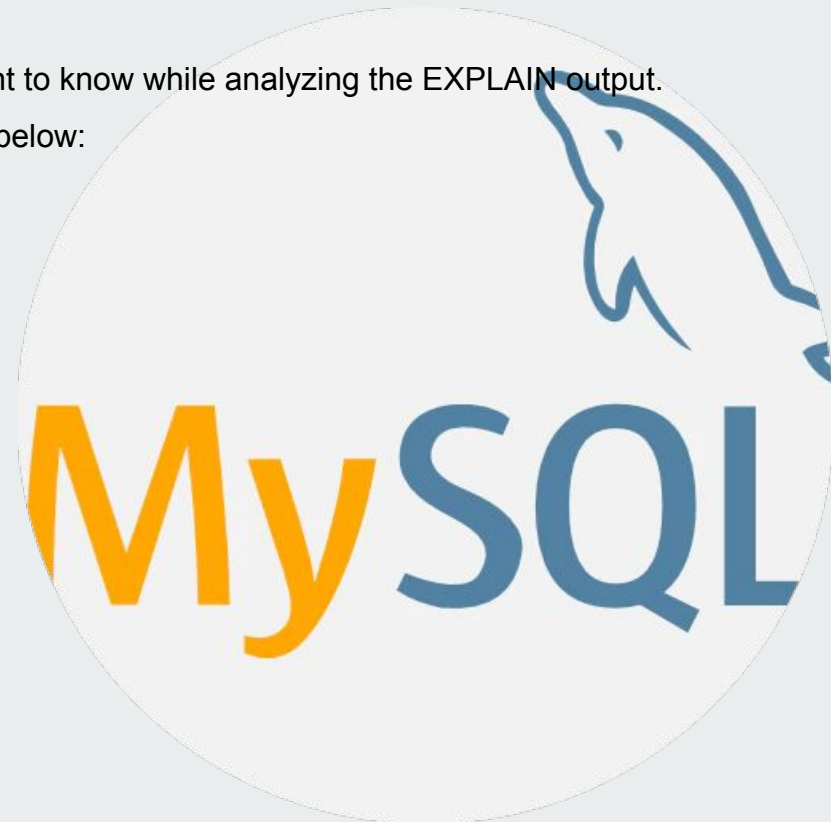
# MySQL - Understanding Database Query Performance Analysis

**Topic:** Understanding each component of EXPLAIN plan output.

There are various terminologies/types/columns that are important to know while analyzing the EXPLAIN output.

The most commonly used ones that I find useful are mentioned below:

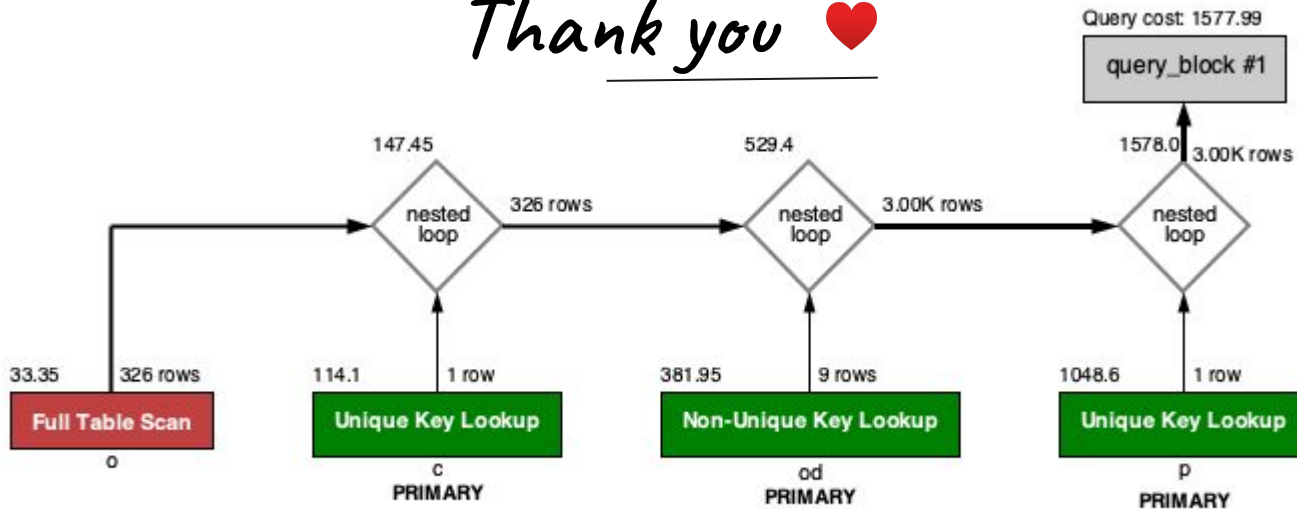
- key
- key\_len
- ref
- eq\_ref
- NLJ (Nested-loop Join) algorithm





# MySQL - Understanding Database Query Performance Analysis

*Thank you* ❤️



Got question or need help? Please follow any of the below options:

Option 1: Raise your detailed request through <https://www.learnandshare.live/contact>

Option 2: Please join my slack channel (learnandshare.live) for a quick response

***Happy learning and sharing...!***

<https://www.learnandshare.live/>