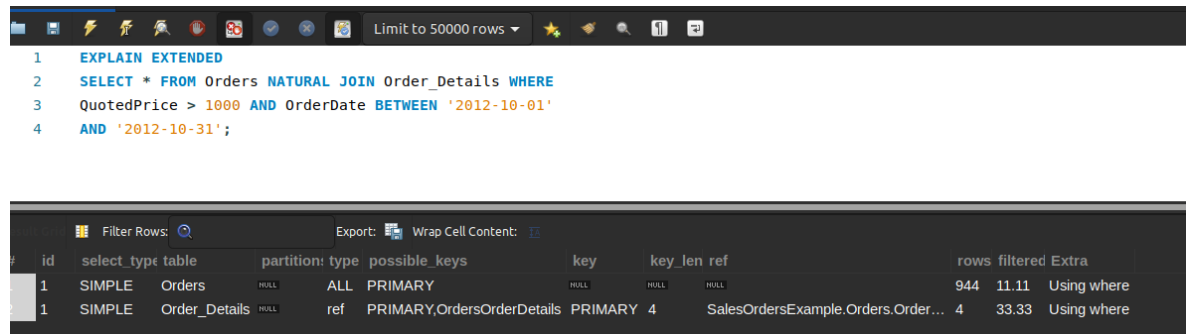


Fundamentals of Data Management

2019HS2 | 101624964 | Jimmy Trac

Credit Task 10.2.1



The screenshot shows a database query editor with a toolbar at the top. The SQL query is as follows:

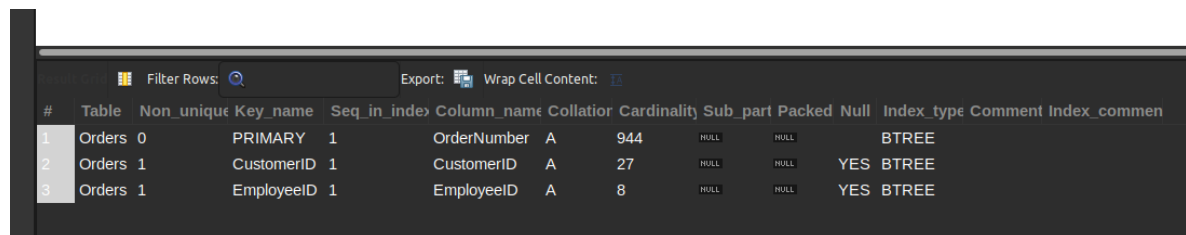
```
1 EXPLAIN EXTENDED
2 SELECT * FROM Orders NATURAL JOIN Order_Details WHERE
3 QuotedPrice > 1000 AND OrderDate BETWEEN '2012-10-01'
4 AND '2012-10-31';
```

Below the query, the execution plan is displayed in a table format:

#	id	select_type	table	partition:	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	1	SIMPLE	Orders		ALL	PRIMARY				944	11.11	Using where
2	1	SIMPLE	Order_Details		ref	PRIMARY,OrdersOrderDetails	PRIMARY	4	SalesOrdersExample.Orders.Order...	4	33.33	Using where

- At first, the table `ORDERS` is joined with `ORDER_DETAILS` using a natural join.

Checking the indexes within `Orders` gives us:



The screenshot shows a database index view for the `Orders` table. The table has the following columns: `#`, `Table`, `Non_unique`, `Key_name`, `Seq_in_index`, `Column_name`, `Collation`, `Cardinality`, `Sub_part`, `Packed`, `Null`, `Index_type`, `Comment`, and `Index_commen`.

#	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_commen
1	Orders	0	PRIMARY	1	OrderNumber	A	944				BTREE		
2	Orders	1	CustomerID	1	CustomerID	A	27			YES	BTREE		
3	Orders	1	EmployeeID	1	EmployeeID	A	8			YES	BTREE		

- Given the Cardinality of the system is 944, similar to the first row in the `EXPLAIN EXTENDED`, it shows us that the orders were associated with `Order_Details` based on the primary key.
 - This means that the DBMS goes through each `OrderNumber` (as that is the key used) to find which are from October
- From those rows, the DBMS then searches for the ones over 1000, returning 4 rows.