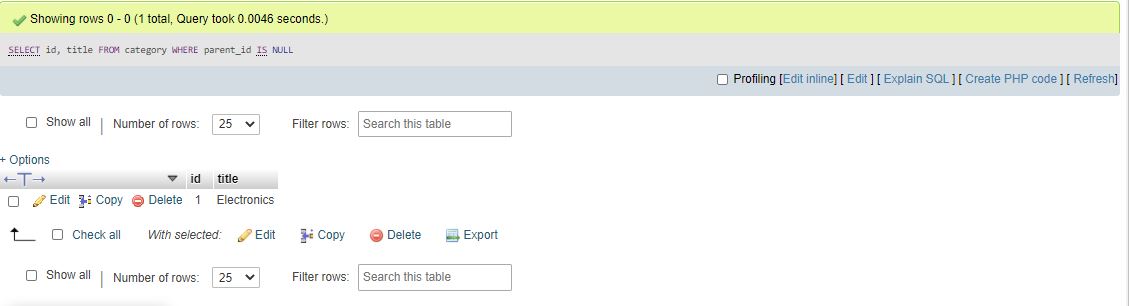
##### **DOCUMENTATION**

**Question 1: Write a query to find the root node?**

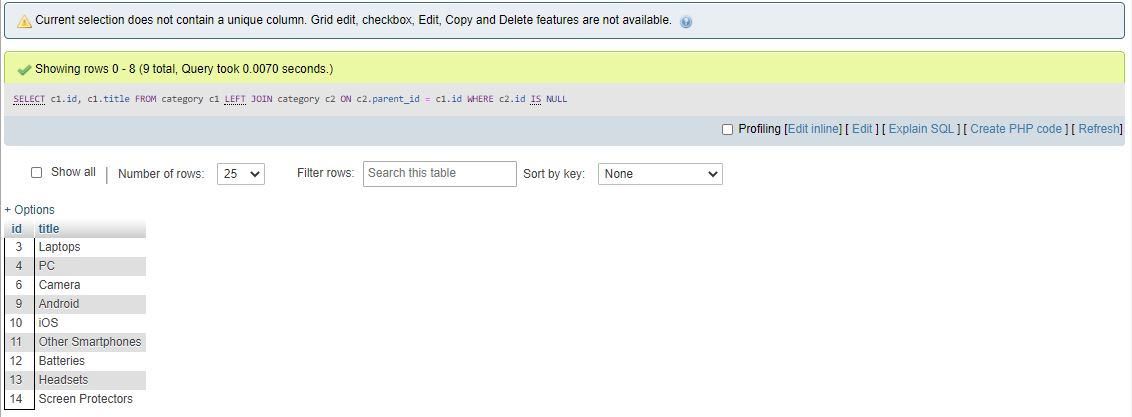
**Ans:-**

SELECT id, title FROM category WHERE parent\_id IS NULL;



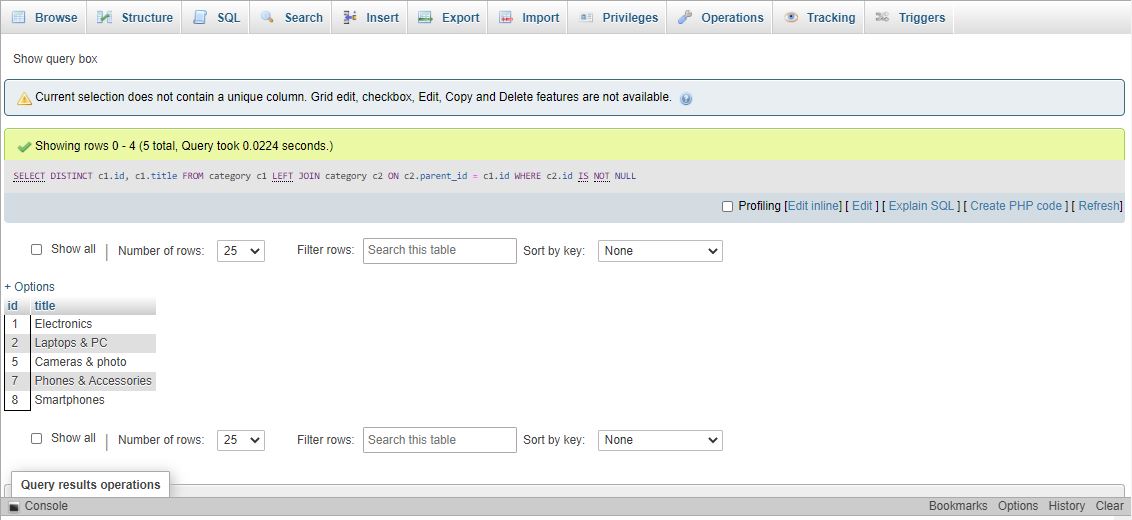
**Question2: write a query to find the leaf node?**

SELECT c1.id, c1.title FROM category c1 LEFT JOIN category c2 ON c2.parent\_id = c1.id WHERE c2.id IS NULL;



**Question3: write a query to find the non-leaf nodes?**

SELECT DISTINCT c1.id, c1.title FROM category c1 LEFT JOIN category c2 ON c2.parent\_id = c1.id WHERE c2.id IS NOT NULL;



**Question 4: write a query to find the path of each node?**

WITH RECURSIVE node\_path (id, title, path) AS

(

SELECT id, title, title as path

FROM category

WHERE parent\_id IS NULL

UNION ALL

SELECT c.id, c.title, CONCAT(cp.path, ' > ', c.title)

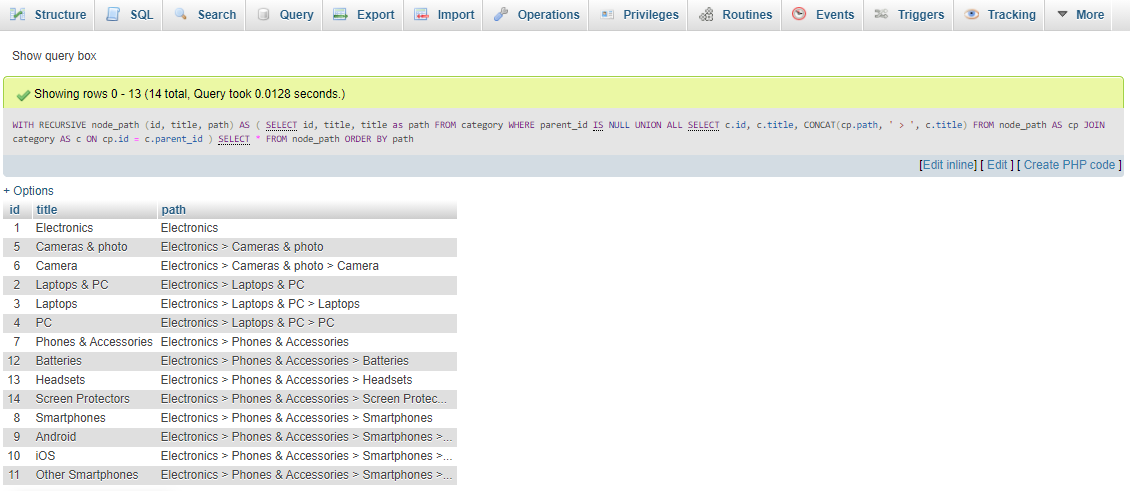
FROM node\_path AS cp JOIN category AS c

ON cp.id = c.parent\_id

)

SELECT \* FROM node\_path

ORDER BY path;



**QUESTION5: WRITE A FUNCTION TO CALCULATE NODE LEVEL**

WITH RECURSIVE node\_level (id, title, lvl) AS

(

SELECT id, title, 0 lvl

FROM category

WHERE parent\_id IS NULL

UNION ALL

SELECT c.id, c.title,cp.lvl + 1

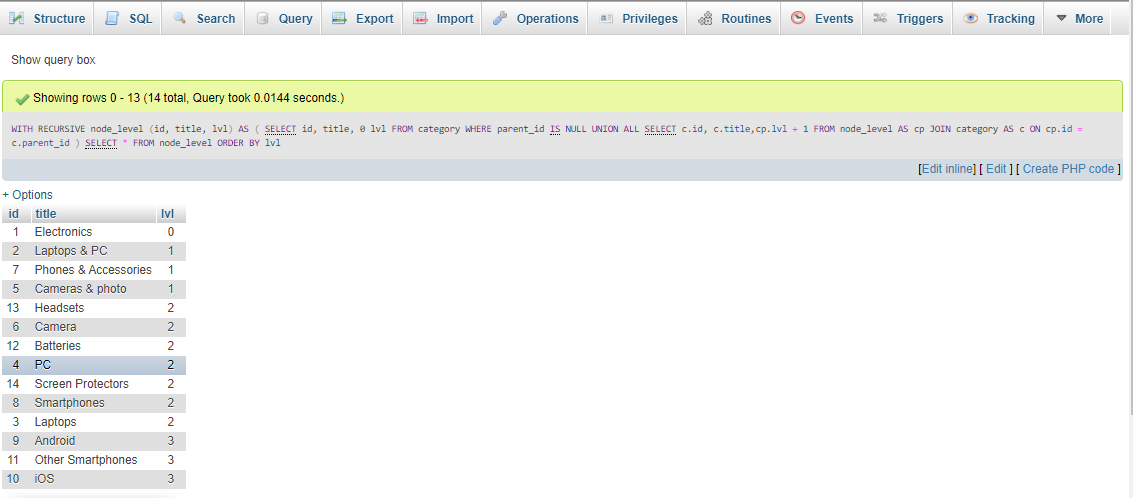
FROM node\_level AS cp JOIN category AS c

ON cp.id = c.parent\_id

)

SELECT \* FROM node\_level

ORDER BY lvl;



**QUESTION 6: WRITE A PROCDEDURE TO GET IMMEDIATE CHILDREN**

DELIMITER $$

CREATE PROCEDURE getCustomers()

BEGIN

SELECT id, title FROM category WHERE parent\_id = 1;

END$$

DELIMITER ;

