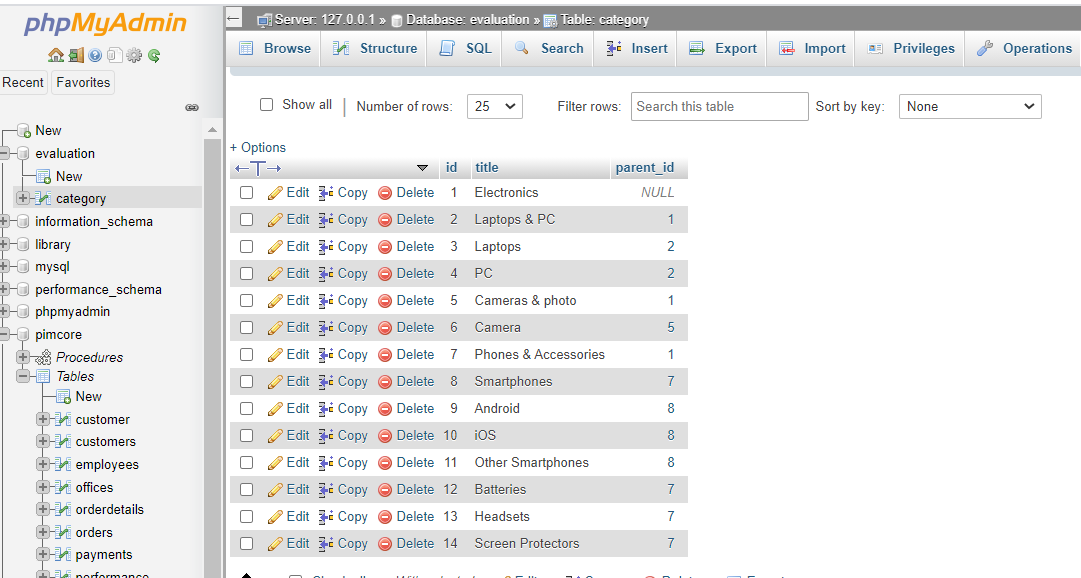
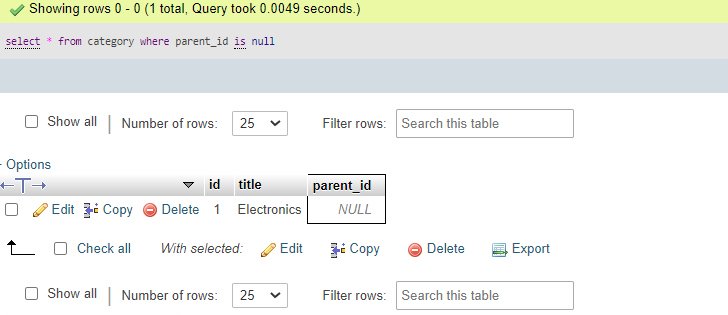
**EVALUATION ASSESSMENT**

**Create a tree structure in MySql table that could store the following tree structure:**

**http://prnt. sc/wcbzth**

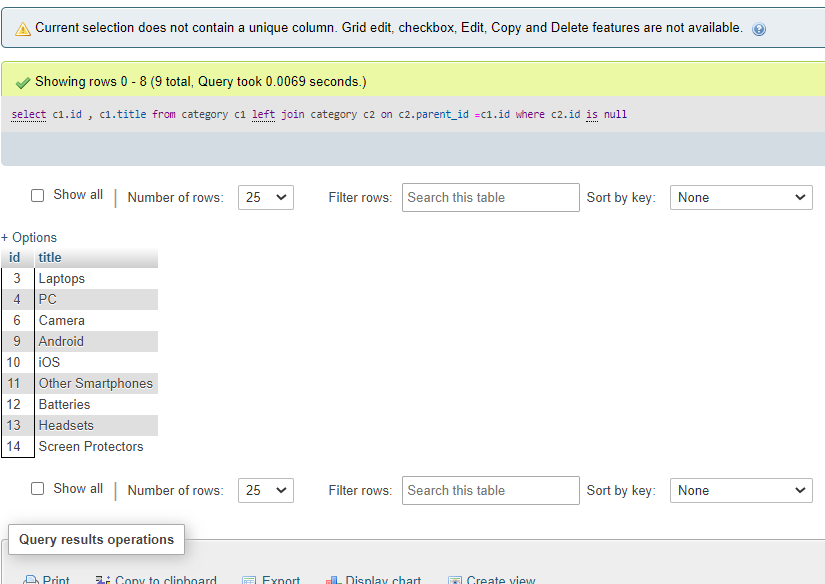
****

**1) Write a query to find the root node.**

**select \* from category where parent\_id IS NULL;**

**2) Write a query to find leaf node.**

**select c1.id , c1.title from category left join category c2 on c2.parent\_id =c1.id where c2.id is null;**

****

3) Write a query to find non-leaf node.

SELECT distinct c1.id, c1.title FROM category c1 JOIN category c2 ON c2.parent\_id = c1.id WHERE c1.id IS NOT NULL and c1.parent\_id is NOT NULL 

4) Write a query to find the path of each node. e.g. http://prnt.sc/wcc4bg

WITH RECURSIVE 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)

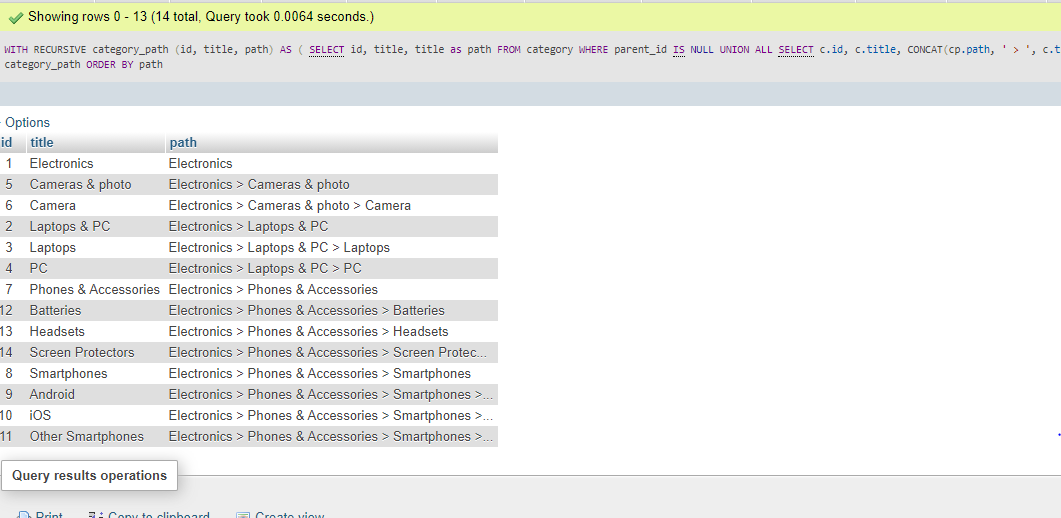
FROM category\_path AS cp JOIN category AS c

ON cp.id = c.parent\_id

)

SELECT \* FROM Path

ORDER BY path;



5) Write a function to calculate node level. e.g. Electronics is at 0 level, Camera is on level 2 and iOs is on level 3.

WITH RECURSIVE category\_path (id, title, level) AS

(

SELECT id, title, 0 level

FROM category

WHERE parent\_id IS NULL

UNION ALL

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

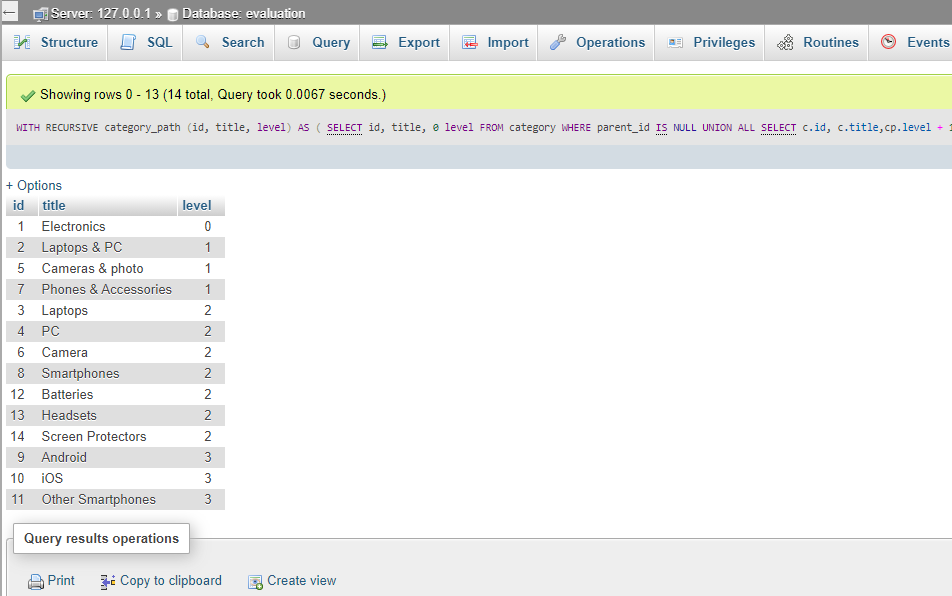
FROM category\_path AS cp JOIN category AS c

ON cp.id = c.parent\_id

)

SELECT \* FROM category\_path

ORDER BY level



6) Write a procedure to get the immediate children.

Create procedure abc() begin select id,title,parent\_id from category where parent\_id=1 end;

