TASK

1) Write a query to find the root node.

ANSWER-  **SELECT id,title FROM categories WHERE parent\_id IS NULL;**

2) Write a query to find leaf node.

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

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

ANSWER- SELECT DISTINCT( c1.id), c1.title FROM categories c1 INNER JOIN

categories c2 ON c2.parent\_id = c1.id WHERE c2.id IS NOT NULL;

4) Write a query to find the path of each node.

ANSWER- WITH RECURSIVE categories\_path (id, title, path) AS

(

SELECT id, title, title as path FROM categories WHERE parent\_id IS NULL

UNION ALL

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

FROM categories\_path AS cp JOIN categories AS c ON cp.id = c.parent\_id)

SELECT \* FROM categories\_path ORDER BY path;

5) Write a function to calculate node level.

ANSWER- **WITH** **RECURSIVE** categories\_path (**id**, title, lvl) **AS**

(

**SELECT** **id**, title, 0 lvl

**FROM** **categories**

**WHERE** parent\_id **IS** NULL

**UNION** **ALL**

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

**FROM** categories\_path **AS** cp **JOIN** **categories** **AS** c

**ON** cp.id = c.parent\_id

)

**SELECT** \* **FROM** categories\_path

**ORDER** **BY** lvl;

6) Write a procedure to get the immediate children.

ANSWER- CREATE DEFINER=`root`@`localhost` PROCEDURE `child`(IN `cid` INT) NOT DETERMINISTIC NO SQL SQL SECURITY DEFINER SELECT id, title FROM categories WHERE parent\_id = cid;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | |  | |  |  | https://mail.google.com/mail/u/0/images/cleardot.gif  https://mail.google.com/mail/u/0/images/cleardot.gif |
| |  | | --- | | https://mail.google.com/mail/u/0/images/cleardot.gif | | | |