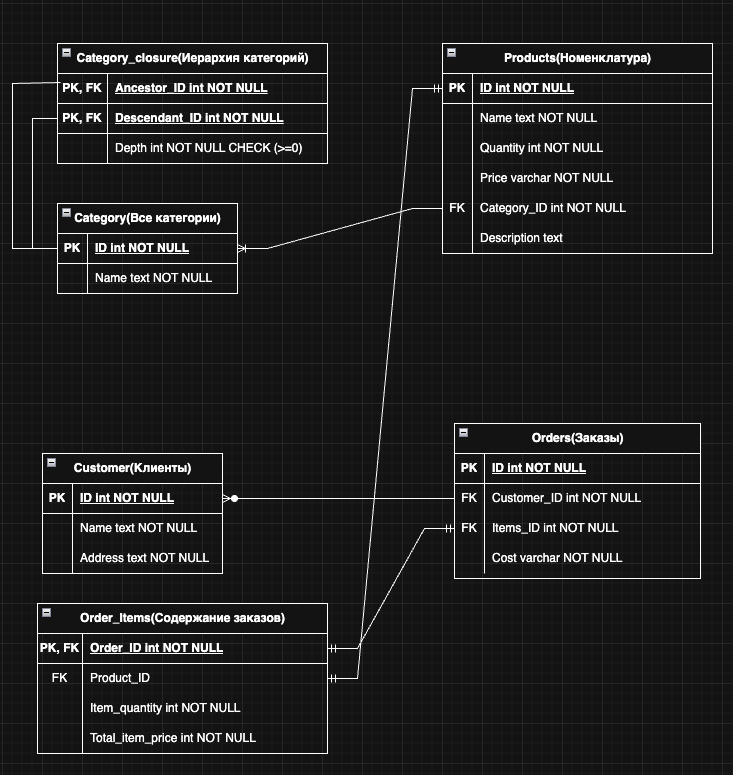
1. Для дерева категорий я выбираю подход Closure Table.

Для того чтобы предусмотреть возможность делать заказы из разного набора товаров заведена отдельная таблица Order Items.



2. 1

SELECT

c.id as customer\_id,

c.first\_name || ' ' || c.last\_name as customer\_name,

c.email,

COUNT(DISTINCT o.id) as total\_orders,

COUNT(oi.id) as total\_items,

SUM(oi.quantity) as total\_quantity,

SUM(oi.total\_price) as total\_amount,

AVG(oi.total\_price) as avg\_order\_amount,

FROM customer c

LEFT JOIN orders o ON c.id = o.customer\_id

LEFT JOIN order\_items oi ON o.id = oi.order\_id

GROUP BY c.id, c.first\_name, c.last\_name, c.email

ORDER BY total\_amount DESC NULLS LAST;

2.2

SELECT

parent.id as category\_id,

parent.name as category\_name,

COUNT(child.id) as direct\_children\_count

FROM category parent

LEFT JOIN category\_closure cc ON parent.id = cc.ancestor\_id

LEFT JOIN category child ON cc.descendant\_id = child.id

WHERE cc.depth = 1 -- Прямые потомки (первый уровень вложенности)

GROUP BY parent.id, parent.name

ORDER BY direct\_children\_count DESC;

2.3.1

CREATE OR REPLACE VIEW top\_5\_products\_last\_month\_optimized AS

WITH last\_month AS (

SELECT

DATE\_TRUNC('month', CURRENT\_DATE - INTERVAL '1 month') as start\_date,

DATE\_TRUNC('month', CURRENT\_DATE) as end\_date

),

product\_sales AS (

SELECT

oi.product\_id,

SUM(oi.quantity) as total\_quantity,

COUNT(DISTINCT oi.order\_id) as orders\_count,

SUM(oi.total\_price) as total\_revenue

FROM order\_items oi

JOIN orders o ON oi.order\_id = o.id

CROSS JOIN last\_month lm

WHERE o.order\_date >= lm.start\_date

AND o.order\_date < lm.end\_date

AND o.status IN ('confirmed', 'processing', 'shipped', 'delivered')

GROUP BY oi.product\_id

),

category\_mapping AS (

SELECT

cc.descendant\_id as category\_id,

c.name as first\_level\_name

FROM category\_closure cc

JOIN category c ON cc.ancestor\_id = c.id

WHERE cc.depth = 1

)

SELECT

p.id as product\_id,

p.name as product\_name,

COALESCE(cm.first\_level\_name, 'Без категории') as first\_level\_category,

ps.total\_quantity as total\_quantity\_sold,

ps.orders\_count,

ps.total\_revenue,

RANK() OVER (ORDER BY ps.total\_quantity DESC) as rank\_by\_quantity

FROM product\_sales ps

JOIN product p ON ps.product\_id = p.id

LEFT JOIN category\_mapping cm ON p.category\_id = cm.category\_id

ORDER BY ps.total\_quantity DESC

LIMIT 5;

2.3.2

Для оптимизации можно использовать такие инструменты как партиции(например по месяцам или дням, для этого нужно добавить даты в заказы).

Можно сделать дополнительную витрину или таблицу с нужными метриками по товарам чтобы делать запросы не напрямую в базу.

Если запросов очень много нужно масштабировать базу, то есть использовать шардирование(Легко делается в Greenplum например)

mount,