

# Seafood E-Commerce Analytics with PostgreSQL

## Mô tả dữ liệu:


- Bảng **page\_hierarchy**: Trang web bán các sản phẩm: Salmon, Kingfish, Tuna, Russian Caviar, Black Truffle, Abalone, Lobster, Crab, Oyster với product\_id từ 1 đến 9. Mỗi sản phẩm thuộc vào 1 trong 3 category: Fish, Luxusy, Shellfish.
- Bảng **users**: Khi một người dùng truy cập vào trang web, hệ thống sẽ tạo hoặc đọc một cookie trên trình duyệt của họ (cookie\_id).
- Bảng **event\_identifier**: lưu trữ thông tin về event\_type và event\_name gồm Page View, Add to Cart, Purchase, Ad Impression, Ad Click.
- Bảng **campaign\_identifier**: hiển thị thông tin về 3 chiến dịch mà website đã chạy
- Bảng **events**: mỗi lần user truy cập vào trang web (cookie\_id), bảng events sẽ lưu lại những trang nào user đó đã truy cập (page\_id), hành vi của user đó (event\_type), thời gian (event\_time), và sequence\_number cho biết thứ tự thực hiện các hành vi của user.

## Thực hiện các yêu cầu:

### 1. Tỷ lệ phần trăm lượt truy cập có sự kiện mua hàng là bao nhiêu?

```
SELECT
    ROUND(COUNT(DISTINCT CASE WHEN ei.event_name = 'Purchase'
                                THEN e.visit_id END) * 100.0 / COUNT(DISTINCT visit_id), 2)
        AS purchase_percent
FROM events e
JOIN event_identifier ei ON e.event_type = ei.event_type;
```


Output:

	<b>purchase_percent</b> numeric 
1	49.86

## 2. Tỷ lệ phần trăm lượt truy cập xem trang thanh toán nhưng không có sự kiện mua hàng là bao nhiêu?

```
SELECT ROUND(COUNT(DISTINCT e.visit_id)* 100.0 / (SELECT COUNT(DISTINCT visit_id)
                                                    FROM events), 2)
                                                    AS checkout_no_buy_percent
FROM events e
JOIN page_hierarchy p ON e.page_id = p.page_id
JOIN event_identifier ei ON e.event_type = ei.event_type
WHERE p.page_name = 'Checkout'
      AND ei.event_name = 'Page View'
      AND e.visit_id NOT IN ( SELECT DISTINCT e2.visit_id
                              FROM events e2
                              JOIN event_identifier ei2 ON e2.event_type = ei2.event_type
                              WHERE ei2.event_name = 'Purchase'
                              );
```



Output:

	checkout_no_buy_percent 
1	9.15

## 3. Ba trang có số lượt xem nhiều nhất là những trang nào?

```
SELECT ph.page_name,
       COUNT(*) AS highest_views
FROM page_hierarchy ph
JOIN events e ON ph.page_id = e.page_id
JOIN event_identifier ei ON e.event_type = ei.event_type
WHERE ei.event_name = 'Page View'
GROUP BY ph.page_name
ORDER BY highest_views DESC
LIMIT 3;
```

Output:

	page_name 	highest_views 
1	All Products	3174
2	Checkout	2103
3	Home Page	1782

#### 4. Số lượt xem và số lần thêm vào giỏ hàng cho từng danh mục sản phẩm là bao nhiêu?

```
SELECT ph.product_category,  
       COUNT(CASE WHEN ei.event_name = 'Page View' THEN 1 END) AS view_count,  
       COUNT(CASE WHEN ei.event_name = 'Add to Cart' THEN 1 END) AS add_to_cart  
FROM page_hierarchy ph  
JOIN events e ON ph.page_id = e.page_id  
JOIN event_identifier ei ON e.event_type = ei.event_type  
WHERE ph.product_category IS NOT NULL  
GROUP BY ph.product_category;
```

Output:

	product_category character varying (9)	view_count bigint	add_to_cart bigint
1	Luxury	3032	1870
2	Shellfish	6204	3792
3	Fish	4633	2789

#### 5. Ba sản phẩm có số lượt mua nhiều nhất là gì?

```
WITH purchased_id AS(  
    SELECT DISTINCT e.visit_id  
    FROM events e  
    JOIN event_identifier ei ON e.event_type = ei.event_type  
    WHERE ei.event_name = 'Purchase'  
)  
SELECT ph.page_name AS products,  
       COUNT(*) AS purchased  
FROM page_hierarchy ph  
JOIN events e ON ph.page_id = e.page_id  
JOIN event_identifier ei ON e.event_type = ei.event_type  
JOIN purchased_id pi ON e.visit_id = pi.visit_id  
WHERE ei.event_name = 'Add to Cart'  
       AND ph.product_id IS NOT NULL  
GROUP BY ph.page_name  
ORDER BY purchased DESC  
LIMIT 3;
```

Output:

	products character varying (14)	purchased bigint
1	Lobster	754
2	Oyster	726
3	Crab	719

6. Sử dụng một truy vấn SQL duy nhất - tạo một bảng đầu ra mới có các chi tiết sau:

- Mỗi sản phẩm được xem bao nhiêu lần?
- Mỗi sản phẩm được thêm vào giỏ hàng bao nhiêu lần?
- Mỗi sản phẩm được thêm vào giỏ hàng nhưng không được mua (bị bỏ rơi) bao nhiêu lần?
- Mỗi sản phẩm được mua bao nhiêu lần?

```
CREATE TABLE product_events AS(
  WITH purchased_id AS(
    SELECT DISTINCT e.visit_id
    FROM events e
    JOIN event_identifier ei ON e.event_type = ei.event_type
    WHERE ei.event_name = 'Purchase'
  )
  SELECT ph.page_name AS product_name,
    COUNT(CASE WHEN ei.event_name = 'Page View' THEN 1 END) AS product_views,
    COUNT(CASE WHEN ei.event_name = 'Add to Cart' THEN 1 END) AS add_to_cart,
    COUNT(CASE WHEN ei.event_name = 'Add to Cart'
      AND pi.visit_id IS NULL THEN 1 END) AS add_cart_no_purchase,
    COUNT(CASE WHEN ei.event_name = 'Add to Cart'
      AND pi.visit_id IS NOT NULL THEN 1 END) AS purchased
  FROM page_hierarchy ph
  JOIN events e ON ph.page_id = e.page_id
  JOIN event_identifier ei ON e.event_type = ei.event_type
  LEFT JOIN purchased_id pi ON e.visit_id = pi.visit_id
  WHERE ph.product_id IS NOT NULL
  GROUP BY ph.page_name
);
```

Output:

	product_name character varying (14) 🔒	product_views bigint 🔒	add_to_cart bigint 🔒	add_cart_no_purchase bigint 🔒	purchased bigint 🔒
1	Abalone	1525	932	233	699
2	Oyster	1568	943	217	726
3	Salmon	1559	938	227	711
4	Crab	1564	949	230	719
5	Tuna	1515	931	234	697
6	Lobster	1547	968	214	754
7	Kingfish	1559	920	213	707
8	Russian Caviar	1563	946	249	697
9	Black Truffle	1469	924	217	707

7. Hãy tạo một bảng khác để tổng hợp thêm dữ liệu tương tự như câu 6 nhưng lần này là cho từng danh mục sản phẩm thay vì từng sản phẩm riêng lẻ.

```
CREATE TABLE category_events AS(
    SELECT  ph.product_category AS product_category,
            SUM(pe.product_views) AS category_views,
            SUM(pe.add_to_cart) AS add_to_cart,
            SUM(pe.add_cart_no_purchase) AS add_cart_no_purchase,
            SUM(pe.purchased) AS purchased
    FROM page_hierarchy ph
    JOIN product_events pe ON ph.page_name = pe.product_name
    WHERE ph.product_id IS NOT NULL
    GROUP BY ph.product_category
);
```

Output:

	product_category character varying (9) 🔒	category_views numeric 🔒	add_to_cart numeric 🔒	add_cart_no_purchase numeric 🔒	purchased numeric 🔒
1	Luxury	3032	1870	466	1404
2	Shellfish	6204	3792	894	2898
3	Fish	4633	2789	674	2115

Sử dụng 2 bảng mới từ câu 6 và câu 7 của bạn - trả lời các câu hỏi sau:

8. Sản phẩm nào có nhiều lượt xem, thêm vào giỏ hàng và mua nhất?

```
SELECT  (SELECT product_name FROM product_events
          ORDER BY product_views DESC LIMIT 1) AS highest_views,
        (SELECT product_name FROM product_events
          ORDER BY add_to_cart DESC LIMIT 1) AS highest_add_to_cart,
        (SELECT product_name FROM product_events
          ORDER BY purchased DESC LIMIT 1) AS highest_purchased;
```

Output:

	highest_views character varying (14) 🔒	highest_add_to_cart character varying (14) 🔒	highest_purchased character varying (14) 🔒
1	Oyster	Lobster	Lobster

9. Sản phẩm nào có khả năng bị bỏ rơi (thêm vào giỏ hàng nhưng không được mua) nhiều nhất?

```
SELECT product_name
FROM product_events
ORDER BY add_cart_no_purchase DESC
LIMIT 1;
```



Output:

	product_name character varying (14) 
1	Russian Caviar

10. Sản phẩm nào có tỷ lệ phần trăm lượt xem thành mua (view to purchase) cao nhất?

```
SELECT product_name,
       ROUND(purchased * 100.0 / product_views, 2) AS view_to_purchase_percent
FROM product_events
ORDER BY view_to_purchase_percent DESC
LIMIT 1;
```

Output:

	product_name character varying (14) 	view_to_purchase_percent numeric 
1	Lobster	48.74

11. Tỷ lệ chuyển đổi trung bình từ lượt xem thành thêm vào giỏ hàng (from view to cart add) là bao nhiêu?

```
SELECT ROUND(SUM(add_to_cart)/SUM(category_views), 3)
       AS avg_view_to_add_cart_rate
FROM category_events;
```

Output:

	avg_view_to_add_cart_rate numeric 
1	0.609

12. Tỷ lệ chuyển đổi trung bình từ thêm vào giỏ hàng thành mua (from cart add to purchase) là bao nhiêu?

```
SELECT ROUND(SUM(purchased)/SUM(add_to_cart), 3)
        AS avg_view_to_add_cart_rate
FROM category_events;
```

Output:

	avg_view_to_add_cart_rate 
1	0.759