

8WeekSQLCHALLENGE

FROM VIEWS TO PURCHASES: A COMPREHENSIVE ANALYSIS OF ONLINE SHOPPING PATTERNS

Fatih Sahin

---

8WEEKSQLCHALLENGE.COM  
CASE STUDY # 6



# CliqueBait

ATTENTION CAPTURING

DATAWITHDANNY.COM

<https://8weeksqlchallenge.com/case-study-6/>

---

SEPTEMBER 2023

---

# CONTENTS

<b>Introduction</b>	<b>3</b>
<b>Problem Statement</b>	<b>4</b>
Creating Schema and Tables	5
Entity Relationship Diagram	7
Data Structure	7
Entity Relationship Diagram	8
<b>Case Study Questions</b>	<b>9</b>
A.DIGITAL ANALYSIS	9
1. How many users are there?	9
2. How many cookies does each user have on average?	9
3. What is the unique number of visits by all users per month?	9
4. What is the number of events for each event type?	10
5. What is the percentage of visits which have a purchase event?	10
6. What is the percentage of visits which view the checkout page but do not have a purchase event?	10
7. What are the top 3 pages by number of views?	11
8. What is the number of views and cart adds for each product category?	11
9. What are the top 3 products by purchases?	12
B. PRODUCT FUNNEL ANALYSIS	13
1. How many times was each product viewed?	14
2. How many times was each product added to cart?	14
3. How many times was each product added to a cart but not purchased (abandoned)?	15
4. How many times was each product purchased?	15
5. Which product had the most views, cart adds and purchases?	16
6. Which product was most likely to be abandoned?	16
7. Which product had the highest view to purchase percentage?	16
8. What is the average conversion rate from view to cart add?	17
C.CAMPAIGNS ANALYSIS	17
1. Generate a table that has 1 single row for every unique visit_id record and has the following columns:	17
<b>Key Findings and Observations</b>	<b>19</b>
<b>CONCLUSION</b>	<b>22</b>

---

## Introduction

This report sheds light on the findings from the "Clique Bait Attention Capturing" case study, an integral part of the '8WeekSQLChallenge' curated by Danny Ma. This particular case study offers a deep dive into the digital landscape of Clique Bait, a renowned online platform known for its unique products and engaging user experience. Taking on this challenge allowed me to simulate the role of a data analyst at Clique Bait, unearthing insights into user behavior, product preferences, and campaign effectiveness.

Beginning my exploration with the "Clique Bait Digital Challenge" was akin to embarking on a digital expedition, navigating through vast amounts of data that resonated with real-world user interactions, product analytics, and campaign results. Those keen on understanding the intricacies of this challenge can delve deeper at

<https://8weeksqlchallenge.com/case-study-6/>

Such challenges transcend mere SQL exercises; they emulate genuine business situations, nudging analysts to extract meaningful, actionable insights from a sea of data points.

For this analysis, I employed the MS SQL Server Management Studio, a versatile platform, perfect for querying and dissecting data with precision. Every insight and observation detailed in this report stems from meticulous analysis, fortified by the proficiency I've gained through my SQL journey.

The dual purpose of this report is to highlight the crucial takeaways from the Clique Bait dataset and to underscore the profound capabilities of SQL in unearthing business insights. It's my sincere hope that this case study not only provides valuable insights but also stands as a testament to the pivotal role of data analysis in shaping the digital businesses of today.

---

## Problem Statement

Clique Bait, an innovative player in the digital realm, finds itself at a transformative crossroads. Guided by the principles of engaging user experiences and robust digital campaigns, the platform aims to deeply understand its **user's behavior, product interactions**, and the **effectiveness of its campaigns**. However, before embarking on this analytical journey, there was an imperative step of refining and preparing the data:

- **Data Consistency:** Ensured that all event types and identifiers adhered to a uniform categorization system, facilitating a seamless event-based analysis.
- **Temporal Metrics Augmentation:** Enriched the dataset with calculated metrics like visit durations, session lengths, and interactivity times, enhancing the granularity of user interactions.
- **User Segmentation:** Introduced new columns that categorized users based on their engagement levels, preferences, and patterns, aiming for a nuanced user behavior analysis.
- **User Journey Metrics:** Calculated metrics like average interactions per visit and bounce rates, offering a deeper perspective into user journey dynamics.

With the data in its optimal state, the investigative challenges for the business took shape:

- **User Interaction Analysis:** Delve into how users navigate and interact with the platform, identifying potential areas of improvement in UI/UX.
- **Product Preference Assessment:** Understand which products captivate user attention the most, guiding inventory and marketing decisions.
- **Campaign Effectiveness:** Evaluate the performance of various campaigns in terms of impressions, clicks, and conversions, aiming to refine future campaign strategies.
- **Platform Dynamics:** Understand user behavior across different devices and platforms, offering insights into platform-specific optimizations.
- **User Conversion Insights:** Analyze the funnel from views to cart additions and eventually to purchases, identifying bottlenecks and opportunities for increased conversions.

---

## Creating Schema and Tables

All datasets exist within the `clique_bait` database schema.

```
1 CREATE SCHEMA clique_bait;
2
3 CREATE TABLE clique_bait.event_identifier (
4   "event_type" INTEGER,
5   "event_name" VARCHAR(13)
6 );
7
8 INSERT INTO clique_bait.event_identifier
9   ("event_type", "event_name")
10 VALUES
11   ('1', 'Page View'),
12   ('2', 'Add to Cart'),
13   ('3', 'Purchase'),
14   ('4', 'Ad Impression'),
15   ('5', 'Ad Click');
16
17 CREATE TABLE clique_bait.campaign_identifier (
18   "campaign_id" INTEGER,
19   "products" VARCHAR(3),
20   "campaign_name" VARCHAR(33),
21   "start_date" TIMESTAMP,
22   "end_date" TIMESTAMP
23 );
24
25 INSERT INTO clique_bait.campaign_identifier
26   ("campaign_id", "products", "campaign_name", "start_date", "end_date")
27 VALUES
28   ('1', '1-3', 'BOGOF - Fishing For Compliments', '2020-01-01', '2020-01-14'),
29   ('2', '4-5', '25% Off - Living The Lux Life', '2020-01-15', '2020-01-28'),
30   ('3', '6-8', 'Half Off - Treat Your Shellfish', '2020-02-01', '2020-03-31');
31
32 CREATE TABLE clique_bait.page_hierarchy (
33   "page_id" INTEGER,
34   "page_name" VARCHAR(14),
35   "product_category" VARCHAR(9),
36   "product_id" INTEGER
37 );
38
39 INSERT INTO clique_bait.page_hierarchy
40   ("page_id", "page_name", "product_category", "product_id")
41 VALUES
42   ('1', 'Home Page', null, null),
43   ('2', 'All Products', null, null),
44   ('3', 'Salmon', 'Fish', '1'),
45   ('4', 'Kingfish', 'Fish', '2'),
46   ('5', 'Tuna', 'Fish', '3'),
47   ('6', 'Russian Caviar', 'Luxury', '4'),
48   ('7', 'Black Truffle', 'Luxury', '5'),
49   ('8', 'Abalone', 'Shellfish', '6'),
50   ('9', 'Lobster', 'Shellfish', '7'),
51   ('10', 'Crab', 'Shellfish', '8'),
52   ('11', 'Oyster', 'Shellfish', '9'),
53   ('12', 'Checkout', null, null),
54   ('13', 'Confirmation', null, null);
55
56 CREATE TABLE clique_bait.users (
57   "user_id" INTEGER,
58   "cookie_id" VARCHAR(6),
59   "start_date" TIMESTAMP
60 );
61
62 INSERT INTO clique_bait.users
63   ("user_id", "cookie_id", "start_date")
64 VALUES
65   ('1', 'c4ca42', '2020-02-04'),
66   ('2', 'c81e72', '2020-01-18'),
67   ('3', 'eccbc8', '2020-02-21'),
68   ('4', 'a87ff6', '2020-02-22'),
69   ('5', 'e4da3b', '2020-02-01'),
70   ('6', '167909', '2020-01-25'),
71   ('7', '8f14e4', '2020-02-09'),
72   ('8', 'c9f0f8', '2020-02-12');
```

```

1830 ( '303', '104033', '2020-03-14' ),
1831 ( '492', '5d8f47', '2020-03-26' ),
1832 ( '365', '6f4e34', '2020-02-03' ),
1833 ( '313', '0ddc8e', '2020-04-15' ),
1834 ( '140', 'c9bfa9', '2020-03-14' ),
1835 ( '421', 'c686c1', '2020-01-29' ),
1836 ( '248', '42ca74', '2020-03-12' ),
1837 ( '208', '56fd2f', '2020-04-10' ),
1838 ( '392', '863bdd', '2020-02-15' ),
1839 ( '24', 'f4b5e8', '2020-02-10' ),
1840 ( '191', 'e83169', '2020-03-21' ),
1841 ( '429', '8313df', '2020-03-06' ),
1842 ( '25', '46dd2f', '2020-03-29' ),
1843 ( '94', '59511b', '2020-03-22' ),
1844 ( '49', 'd345a8', '2020-02-23' ),
1845 ( '211', 'a26e03', '2020-02-20' ),
1846 ( '64', '87a4ba', '2020-03-18' );
1847
1848 CREATE TABLE clique_bait.events (
1849 "visit_id" VARCHAR(6),
1850 "cookie_id" VARCHAR(6),
1851 "page_id" INTEGER,
1852 "event_type" INTEGER,
1853 "sequence_number" INTEGER,
1854 "event_time" TIMESTAMP
1855 );
1856
1857 INSERT INTO clique_bait.events
1858 ("visit_id", "cookie_id", "page_id", "event_type", "sequence_number", "event_time")
1859 VALUES
1860 ('ccf365', 'c4ca42', '1', '1', '1', '2020-02-04 19:16:09.182546'),
1861 ('ccf365', 'c4ca42', '2', '1', '2', '2020-02-04 19:16:17.358191'),
1862 ('ccf365', 'c4ca42', '6', '1', '3', '2020-02-04 19:16:58.454669'),
1863 ('ccf365', 'c4ca42', '9', '1', '4', '2020-02-04 19:16:58.609142'),
1864 ('ccf365', 'c4ca42', '9', '2', '5', '2020-02-04 19:17:51.72942'),
1865 ('ccf365', 'c4ca42', '10', '1', '6', '2020-02-04 19:18:11.605815'),
1866 ('ccf365', 'c4ca42', '10', '2', '7', '2020-02-04 19:19:10.570786'),
1867 ('ccf365', 'c4ca42', '11', '1', '8', '2020-02-04 19:19:46.911728'),
1868 ('ccf365', 'c4ca42', '11', '2', '9', '2020-02-04 19:20:45.27469'),
1869 ('ccf365', 'c4ca42', '12', '1', '10', '2020-02-04 19:20:52.307244'),
1870 ('ccf365', 'c4ca42', '13', '3', '11', '2020-02-04 19:21:26.242563'),
1871 ('d58cbd', 'c81e72', '1', '1', '1', '2020-01-18 23:40:54.761906'),
1872 ('d58cbd', 'c81e72', '2', '1', '2', '2020-01-18 23:41:06.391027'),
1873 ('d58cbd', 'c81e72', '4', '1', '3', '2020-01-18 23:42:02.213001'),
1874 ('d58cbd', 'c81e72', '4', '2', '4', '2020-01-18 23:42:02.370046'),
1875 ('d58cbd', 'c81e72', '5', '1', '5', '2020-01-18 23:42:44.717024'),
1876 ('d58cbd', 'c81e72', '5', '2', '6', '2020-01-18 23:43:11.121855'),
1877 ('d58cbd', 'c81e72', '7', '1', '7', '2020-01-18 23:43:25.806239'),
1878 ('d58cbd', 'c81e72', '8', '1', '8', '2020-01-18 23:43:40.537995'),
1879 ('d58cbd', 'c81e72', '8', '2', '9', '2020-01-18 23:44:14.026393'),
1880 ('d58cbd', 'c81e72', '10', '1', '10', '2020-01-18 23:44:22.103768'),
1881 ('d58cbd', 'c81e72', '10', '2', '11', '2020-01-18 23:45:00.004781'),
1882 ('d58cbd', 'c81e72', '12', '1', '12', '2020-01-18 23:45:38.186554'),
1883 ('9a2f24', 'eccbc8', '1', '1', '1', '2020-02-21 03:19:10.032455'),
1884 ('9a2f24', 'eccbc8', '4', '1', '2', '2020-02-21 03:19:24.677901'),
1885 ('9a2f24', 'eccbc8', '4', '2', '3', '2020-02-21 03:19:48.146489'),
1886 ('9a2f24', 'eccbc8', '7', '1', '4', '2020-02-21 03:20:13.39183'),
1887 ('9a2f24', 'eccbc8', '7', '2', '5', '2020-02-21 03:20:13.869733'),
1888 ('9a2f24', 'eccbc8', '10', '1', '6', '2020-02-21 03:20:45.854556'),
1889 ('9a2f24', 'eccbc8', '11', '1', '7', '2020-02-21 03:21:20.335104'),
1890 ('9a2f24', 'eccbc8', '12', '1', '8', '2020-02-21 03:21:43.262109'),
1891 ('9a2f24', 'eccbc8', '13', '3', '9', '2020-02-21 03:22:22.501245'),
1892 ('7caba5', 'a87ff6', '1', '1', '1', '2020-02-22 17:49:37.646174'),
1893 ('7caba5', 'a87ff6', '4', '1', '2', '2020-02-22 17:50:23.736729'),
1894 ('7caba5', 'a87ff6', '5', '1', '3', '2020-02-22 17:50:26.878153'),
1895 ('7caba5', 'a87ff6', '5', '2', '4', '2020-02-22 17:51:17.963956'),
1896 ('7caba5', 'a87ff6', '8', '1', '5', '2020-02-22 17:51:53.119277'),
1897 ('7caba5', 'a87ff6', '9', '1', '6', '2020-02-22 17:52:21.958526'),
1898 ('7caba5', 'a87ff6', '9', '2', '7', '2020-02-22 17:52:56.251508'),
1899 ('f61ed7', 'e4da3b', '1', '1', '1', '2020-02-01 06:30:39.766168'),
1900 ('f61ed7', 'e4da3b', '2', '1', '2', '2020-02-01 06:30:50.502402'),
1901 ('f61ed7', 'e4da3b', '4', '1', '3', '2020-02-01 06:31:07.364658'),

```

---

## Entity Relationship Diagram

### Data Structure

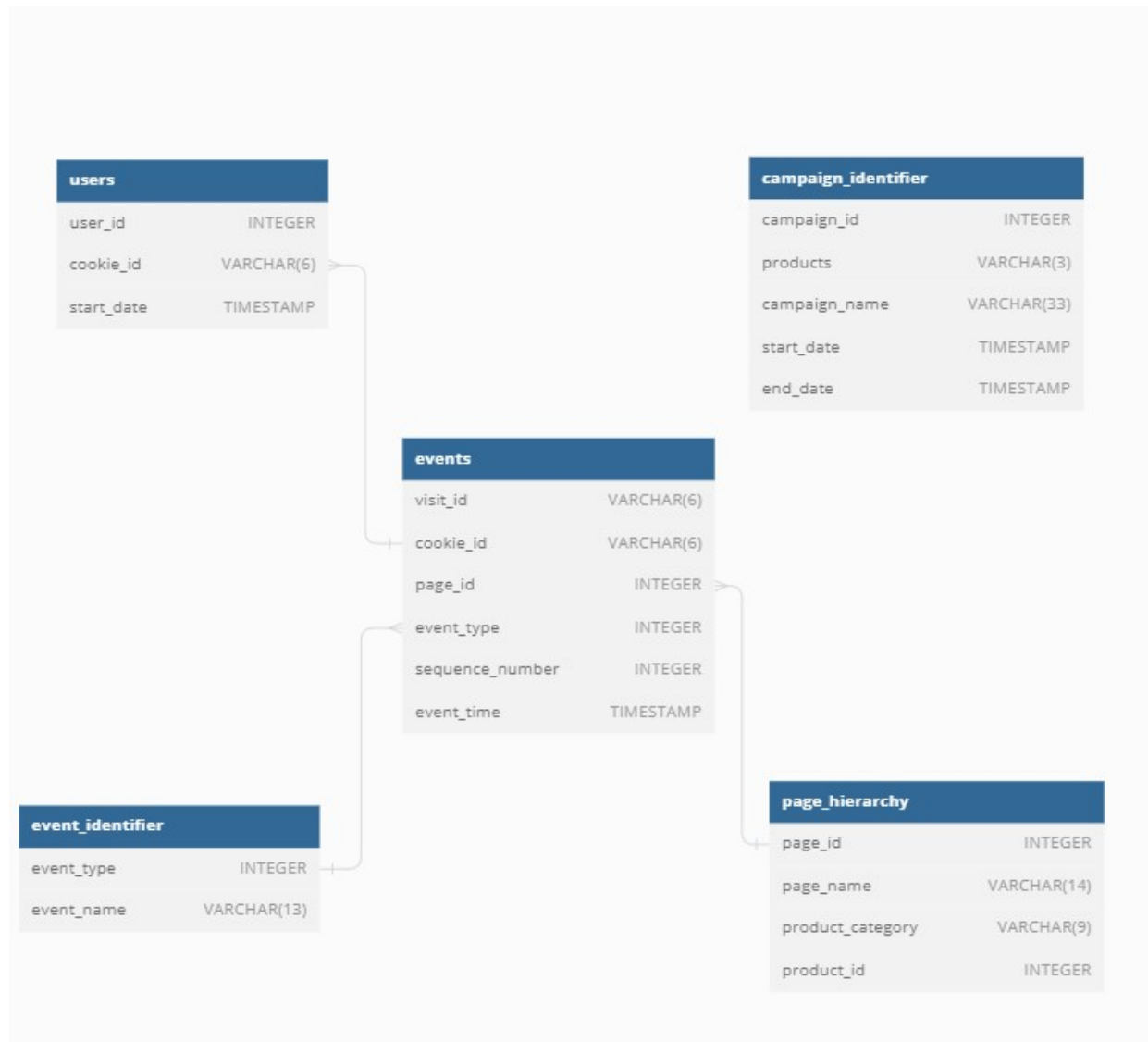
Clique Bait employs a sophisticated data management system to harness the potential of its vast digital interactions and events. The core of this system revolves around tables capturing user interactions, events, and campaign details. Here's a delineation of the tables:

1. **clique\_bait.users:** This foundational table is the heartbeat of Clique Bait's user data. It encompasses pivotal details such as:
  - **user\_id:** A unique identifier for each user.
  - **cookie\_id:** The cookie associated with the user, facilitating session tracking.
  - **event\_time:** Timestamp denoting when a specific interaction took place.
2. **clique\_bait.events:** This table captures the essence of user interactions on the platform. It comprises columns that articulate crucial event information like:
  - **event\_type:** The specific type of event, e.g., page view, cart addition, or purchase.
  - **page\_id:** Identifier for the specific page where the event occurred.
  - **visit\_id:** A unique identifier representing a user's session or visit.
3. **clique\_bait.page\_hierarchy:** Serving as a bridge between products and events, this table provides context to the events table by detailing:
  - **page\_name:** The name or title of the page.
  - **product\_id:** If applicable, the unique identifier of the product associated with the page.
  - **product\_category:** Broad categorization of products, such as Shellfish or Fish.
4. **clique\_bait.event\_identifier:** This auxiliary table aids in translating the event\_type codes into more readable formats by offering:
  - **event\_name:** A descriptive name for each event type, like "Page View" or "Purchase".
5. **clique\_bait.campaign\_identifier:** As Clique Bait ventures into digital campaigns, this table captures the nuances of each campaign, including:

- **campaign\_name:** The title or name of the campaign.
- **start\_date & end\_date:** The time frame during which the campaign was active.

With the synergy of these tables, especially the interaction between clique\_bait.events and clique\_bait.page\_hierarchy, Clique Bait stands poised to derive holistic insights from its data. The tables' interrelationships not only enable a comprehensive understanding of user interactions but also set the stage for advanced analytics, segmentation, and campaign optimization.

## Entity Relationship Diagram





---

# Case Study Questions

## A.DIGITAL ANALYSIS

1. How many users are there?

```
SELECT COUNT (DISTINCT (user_id)) AS total_users  
FROM clique_bait.users
```

	total_users
1	500

2. How many cookies does each user have on average?

```
WITH totals AS(  
  SELECT  
    COUNT (DISTINCT (user_id)) as total_users,  
    COUNT(cookie_id) as total_cookies  
  FROM clique_bait.users  
)  
select (total_cookies/total_users) as average_cookies_per_user  
from totals
```

	average_cookies_per_user
1	3

3. What is the unique number of visits by all users per month?

```
SELECT  
  DATEPART (month,event_time) as month,  
  COUNT (DISTINCT(visit_id)) total_visits  
FROM clique_bait.events  
GROUP BY DATEPART (month,event_time)  
ORDER BY DATEPART (month,event_time)
```

	month	total_visits
1	1	876
2	2	1488
3	3	916
4	4	248
5	5	36

---

4. What is the number of events for each event type?

```
SELECT
    e.event_type,
    ei.event_name,
    COUNT( DISTINCT(e.event_time)) as number_events
FROM clique_bait.events e
JOIN clique_bait.event_identifier ei
ON ei.event_type=e.event_type
GROUP BY e.event_type,event_name
ORDER BY event_type
```

	event_type	event_name	number_events
1	1	Page View	20928
2	2	Add to Cart	8451
3	3	Purchase	1777
4	4	Ad Impression	876
5	5	Ad Click	702

5. What is the percentage of visits which have a purchase event?

```
SELECT
    ROUND((SELECT COUNT(DISTINCT visit_id) FROM clique_bait.events WHERE event_type = 3) * 100.0 /
    (SELECT COUNT(DISTINCT visit_id) FROM clique_bait.events),3) AS purchase_percentage;
```

	purchase_percentage
1	49.8600000000000

6. What is the percentage of visits which view the checkout page but do not have a purchase event?

```
SELECT
    (SELECT COUNT(DISTINCT e.visit_id)
    FROM clique_bait.events e
    WHERE e.page_id = 12 AND
    e.visit_id NOT IN (SELECT DISTINCT visit_id FROM clique_bait.events WHERE event_type = 3)) * 100.0 /
    (SELECT COUNT(DISTINCT visit_id) FROM clique_bait.events WHERE page_id = 12) AS no_purchase_checkout_percentage;
```

	no_purchase_checkout_percentage
1	15.501664289110

---

7. What are the top 3 pages by number of views?

```
SELECT TOP 3
    page_id,
    COUNT(event_type) AS number_views
FROM clique_bait.events
WHERE event_type=1
GROUP BY page_id
ORDER BY number_views desc;
```

	page_id	number_views
1	2	3174
2	12	2103
3	1	1782

8. What is the number of views and cart adds for each product category?

```
SELECT
    ph.product_category,
    SUM( CASE WHEN ei.event_name='Page View' THEN 1 ELSE 0 END) AS page_view,
    SUM( CASE WHEN ei.event_name='Add to Cart' THEN 1 ELSE 0 END) AS Add_Card_view

FROM clique_bait.page_hierarchy ph
JOIN clique_bait.events e
ON e.page_id=ph.page_id
JOIN clique_bait.event_identifier ei
ON ei.event_type=e.event_type
WHERE ph.product_category IS NOT null
GROUP BY product_category
ORDER BY page_view DESC;
```

	product_category	page_view	Add_Card_view
1	Shellfish	6204	3792
2	Fish	4633	2789
3	Luxury	3032	1870

---

## 9. What are the top 3 products by purchases?

```
WITH purchases AS (  
    SELECT visit_id  
    FROM clique_bait.events  
    WHERE event_type = 3  
)  
SELECT TOP 3  
    ph.page_name,  
    SUM(  
        CASE WHEN e.event_type = 2 THEN 1 ELSE 0 END ) AS purchased  
FROM clique_bait.page_hierarchy AS ph  
    JOIN clique_bait.events AS e ON e.page_id = ph.page_id  
    JOIN purchases AS gp ON e.visit_id = gp.visit_id  
WHERE ph.product_category IS NOT NULL  
    AND ph.page_id NOT IN('1', '2', '12', '13')  
    AND gp.visit_id = e.visit_id  
GROUP BY ph.page_name  
ORDER BY purchased DESC
```

	page_name	purchased
1	Lobster	754
2	Oyster	726
3	Crab	719

## B. PRODUCT FUNNEL ANALYSIS

Using a single SQL query - **create a new output table** which has the following details:

- How many times was each product viewed?
- How many times was each product added to cart?
- How many times was each product added to a cart but not purchased (abandoned)?
- How many times was each product purchased?

```
WITH ProductViews AS (
    SELECT
        page_id,
        COUNT(visit_id) AS view_count
    FROM
        clique_bait.events
    WHERE
        event_type = 1
    GROUP BY
        page_id
),
ProductCartAdd AS (
    SELECT
        page_id,
        COUNT(visit_id) AS cart_add_count
    FROM
        clique_bait.events
    WHERE
        event_type = 2
    GROUP BY
        page_id
),
ProductAbandoned AS (
    SELECT
        e.page_id,
        COUNT(e.visit_id) AS abandoned_count
    FROM
        clique_bait.events e
    WHERE
        e.event_type = 2 AND NOT EXISTS (
            SELECT 1
            FROM clique_bait.events ev2
            WHERE ev2.visit_id = e.visit_id AND ev2.event_type = 3
        )
    GROUP BY
        e.page_id
),
ProductPurchased AS (
    SELECT
        e.page_id,
        SUM(CASE WHEN event_type = 2 THEN 1 ELSE 0 END) AS purchased_from_cart
    FROM clique_bait.page_hierarchy AS ph
    JOIN clique_bait.events AS e ON ph.page_id = e.page_id
    WHERE ph.product_id IS NOT NULL AND EXISTS(
        SELECT visit_id
        FROM clique_bait.events
        WHERE event_type = 3
        AND e.visit_id = visit_id
    )
    GROUP BY e.page_id
)
-- Final Select Statement
SELECT
    ph.product_id,
    ph.page_name,
    ISNULL(pv.view_count, 0) AS view_count,
    ISNULL(pca.cart_add_count, 0) AS cart_add_count,
    ISNULL(pa.abandoned_count, 0) AS abandoned_count,
    ISNULL(pp.purchased_from_cart, 0) AS purchase_count
INTO
    ProductAnalysis
FROM
    clique_bait.page_hierarchy ph
LEFT JOIN
    ProductViews pv ON ph.page_id = pv.page_id
LEFT JOIN
    ProductCartAdd pca ON ph.page_id = pca.page_id
LEFT JOIN
    ProductAbandoned pa ON ph.page_id = pa.page_id
LEFT JOIN
    ProductPurchased pp ON ph.page_id = pp.page_id
WHERE
    ph.product_id IS NOT NULL;
```

---

1. How many times was each product viewed?

```
SELECT
    product_id,
    page_name,
    view_count
FROM [clique_bait].[dbo].[ProductAnalysis]
```

	product_id	page_name	view_count
1	1	Salmon	1559
2	2	Kingfish	1559
3	3	Tuna	1515
4	4	Russian C...	1563
5	5	Black Tru...	1469
6	6	Abalone	1525
7	7	Lobster	1547
8	8	Crab	1564
9	9	Oyster	1568

2. How many times was each product added to cart?

```
SELECT
    product_id,
    page_name,
    cart_add_count
FROM [clique_bait].[dbo].[ProductAnalysis]
```

	product_id	page_name	cart_add_count
1	1	Salmon	938
2	2	Kingfish	920
3	3	Tuna	931
4	4	Russian Caviar	946
5	5	Black Truffle	924
6	6	Abalone	932
7	7	Lobster	968
8	8	Crab	949
9	9	Oyster	943

---

3. How many times was each product added to a cart but not purchased (abandoned)?

```
SELECT
    product_id,
    page_name,
    abandoned_count
FROM [clique_bait].[dbo].[ProductAnalysis]
```

	product_id	page_name	abandoned_count
1	1	Salmon	227
2	2	Kingfish	213
3	3	Tuna	234
4	4	Russian Caviar	249
5	5	Black Truffle	217
6	6	Abalone	233
7	7	Lobster	214
8	8	Crab	230
9	9	Oyster	217

4. How many times was each product purchased?

```
SELECT
    product_id,
    page_name,
    purchase_count
FROM [clique_bait].[dbo].[ProductAnalysis]
```

	product_id	page_name	purchase_count
1	1	Salmon	711
2	2	Kingfish	707
3	3	Tuna	697
4	4	Russian Caviar	697
5	5	Black Truffle	707
6	6	Abalone	699
7	7	Lobster	754
8	8	Crab	719
9	9	Oyster	726

---

5. Which product had the most views, cart adds and purchases?

```
SELECT TOP 1
    page_name,
    view_count
FROM dbo.ProductAnalysis
ORDER BY view_count DESC;
```

	page_name	view_count
1	Oyster	1568

6. Which product was most likely to be abandoned?

```
SELECT TOP 1
    page_name,
    abandoned_count
FROM dbo.ProductAnalysis
ORDER BY abandoned_count DESC;
```

	page_name	abandoned_count
1	Russian Caviar	249

7. Which product had the highest view to purchase percentage?

```
SELECT TOP 1
    page_name,
    (purchase_count *100 /view_count) AS view_to_purchase_percentage
FROM dbo.ProductAnalysis
ORDER BY view_to_purchase_percentage DESC;
```

	page_name	view_to_purchase_percentage
1	Black Truffle	48



---

## 8. What is the average conversion rate from view to cart add?

```
WITH totals AS (  
  SELECT  
    SUM(cart_add_count) AS cart_add_total,  
    SUM(view_count) AS view_total  
  FROM dbo.ProductAnalysis  
)  
SELECT  
  CONVERT(DECIMAL(5, 2), (100.0 * cart_add_total / view_total)) AS conversion_rate_percentage  
FROM totals;  
  
--What is the average conversion rate from cart add to purchase?  
WITH totals AS (  
  SELECT  
    SUM(purchase_count) AS purchase_total,  
    SUM(cart_add_count) AS cart_add_total  
  FROM dbo.ProductAnalysis  
)  
SELECT  
  CONVERT(DECIMAL(5, 2), (100.0 * purchase_total / cart_add_total)) AS conversion_rate_percentage  
FROM totals;
```

	conversion_rate_percentage
1	60.93

## C.CAMPAIGNS ANALYSIS

1. Generate a table that has 1 single row for every unique visit\_id record and has the following columns:
  - **user\_id**
  - **visit\_id**
  - **visit\_start\_time**: the earliest event\_time for each visit
  - **page\_views**: count of page views for each visit
  - **cart\_adds**: count of product cart add events for each visit
  - **purchase**: 1/0 flag if a purchase event exists for each visit
  - **campaign\_name**: map the visit to a campaign if the visit\_start\_time falls between the start\_date and end\_date
  - **impression**: count of ad impressions for each visit
  - **click**: count of ad clicks for each visit

```
SELECT
    u.user_id,
    e.visit_id,
    MIN(e.event_time) AS visit_start_time,
    SUM(CASE WHEN e.event_type=1 THEN 1 ELSE 0 END) AS page_views,
    SUM(CASE WHEN e.event_type=2 THEN 1 ELSE 0 END) AS cart_adds,
    MAX(CASE WHEN e.event_type=3 THEN 1 ELSE 0 END) AS purchase,
    ci.campaign_name,
    SUM(CASE WHEN e.event_type=4 THEN 1 ELSE 0 END) AS impression,
    SUM(CASE WHEN e.event_type=5 THEN 1 ELSE 0 END) AS click
INTO
    CampaignAnalysis
FROM
    clique_bait.users u
JOIN clique_bait.events e ON u.cookie_id = e.cookie_id
LEFT JOIN clique_bait.campaign_identifier ci
    ON e.event_time BETWEEN ci.start_date AND ci.end_date
GROUP BY
    u.user_id,
    e.visit_id,
    ci.campaign_name;
```

---

## Key Findings and Observations

### User Dynamics:

The platform boasts a healthy **user base**, with **500 unique users** engaging with the system. This indicates a substantial **interaction volume** that can offer rich insights.

### User-Session Correlation:

On average, each user has **3 cookies** associated with them. This suggests that users tend to visit the platform from different devices or browsers, which underscores the importance of **tracking users over sessions** for accurate analytics.

### Monthly Engagement Trends:

The months of **February and March** witnessed the highest number of visits. However, there is a significant decline observed in **April and May**, hinting at possible **seasonal trends** or external factors affecting user interactions.

### Event Dynamics:

The most frequent event on the platform is the '**Page View**' with over **20,000 occurrences**, closely followed by '**Add to Cart**' events. This suggests that users are actively browsing products, but not all views are converting to cart additions or purchases.

### Conversion Insights:

Almost **half of the visits** result in a purchase, indicating a **high conversion rate**. Yet, there's a **15.5% chance** that users who view the checkout page decide against making a purchase. This points to potential areas of improvement in the **checkout process** or user experience.

---

### Top Pages & Products:

The **Oyster** page witnessed the highest views, suggesting it's a popular product or has prominent placement on the platform. Moreover, when it comes to purchases, products like **Lobster, Oyster, and Crab** are leading, indicating their strong appeal to the user base.

### Product Categories:

The **Shellfish category** dominates both in terms of page views and cart additions, followed by **Fish** and **Luxury** categories. This provides valuable input for **inventory management** and marketing efforts.

### Product Abandonment:

The product '**Russian Caviar**' experienced the highest cart abandonment. This could be due to various reasons, such as pricing, user reviews, or other external factors, and warrants further investigation.

### Conversion Rates:

There's a notable **60.93% chance** that a page view will convert into a cart addition. Furthermore, once a product is added to the cart, it has a **75.93% probability** of being purchased. These metrics are vital for assessing the efficacy of **product placements, recommendations**, and overall user interface.

### Campaign Dynamics :

The dataset suggests complex interplays between **user visits, impressions, clicks**, and **purchases**, all potentially influenced by various campaigns. Understanding these dynamics is crucial for optimizing **marketing ROI** and tailoring future campaigns.

---

In conclusion, while Clique Bait demonstrates promising user engagement and conversion rates, there's always room for optimization. Insights derived from **event dynamics**, **product preferences**, and **abandonment rates** can guide tweaks in user experience, **marketing strategies**, and product placements, ensuring **sustained growth** and user satisfaction.

---

## CONCLUSION

This case study shows how online shopping works. Every time someone looks at a product, adds it to their cart, or buys it, it tells us a little bit about what they like and don't like.

I used Microsoft SQL Server Management Studio to look at all the data. With SQL, I could see patterns and stories in the numbers. Every piece of information helped me understand more about online shopping.

At the end of the day, using SQL helped me see how important it is to pay attention to the details. The things I found out can help businesses make better decisions and understand their customers more.

This study is based on real problems and questions from Danny's many years of working in places like banks and supermarkets. Campaign analysis, or studying how people act online, is very important today. Being good at it can really help in jobs like marketing or retail. This study shows just how powerful it can be.