

# Documentation for SeaFoodKart Data Analysis

---

## About SeaFoodKart

SeaFoodKart is an innovative online seafood store founded by Ramesh, who is also the CEO and part of a digital data analytics team. SeaFoodKart focuses on providing high-quality seafood products delivered directly to customers' doorsteps. The company's mission is to enhance customer experience, optimize operations, and drive sales through quality, convenience, and sustainability.

## About the Data

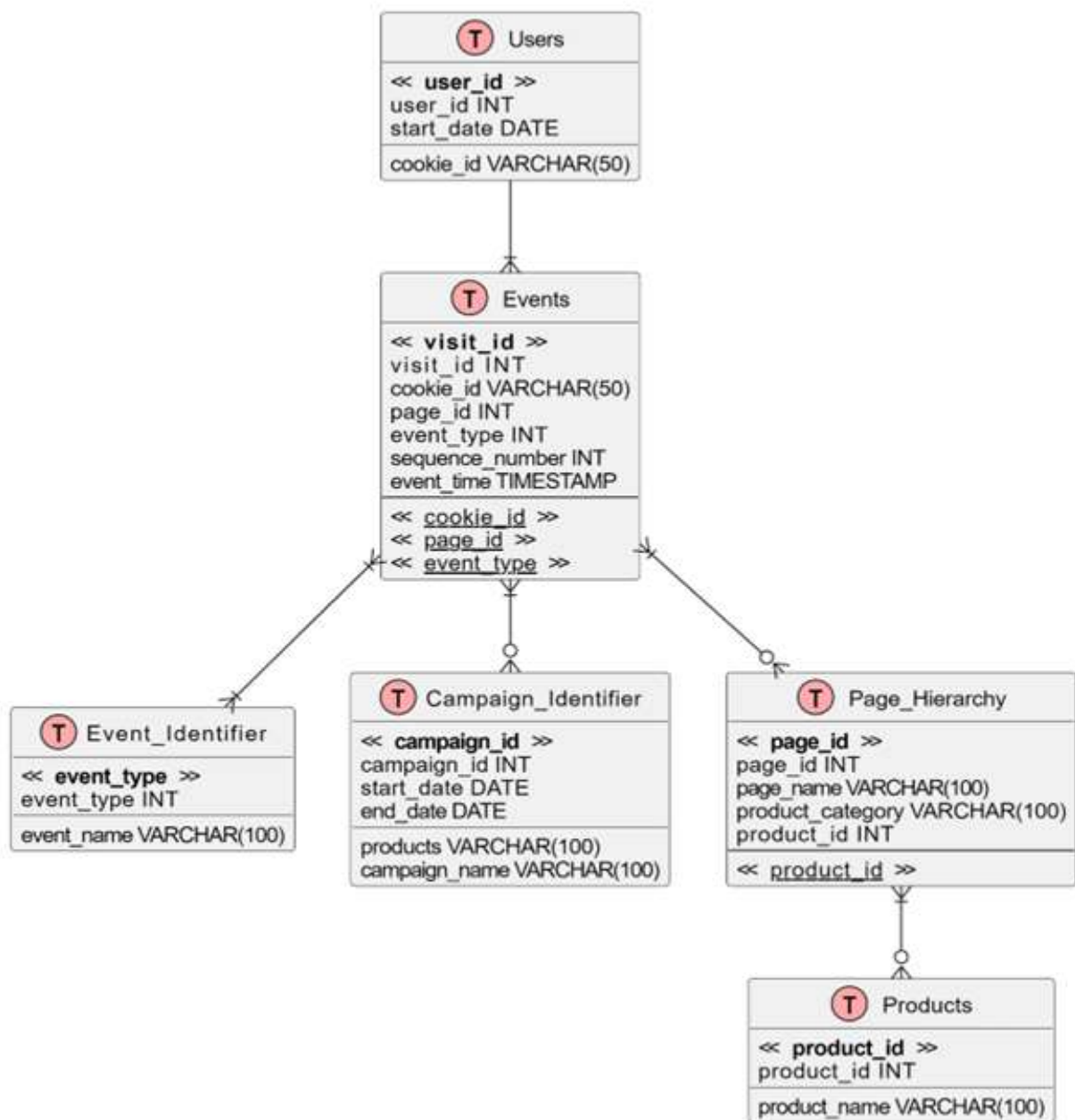
The data provided by SeaFoodKart consists of several CSV files which capture various aspects of the business operations and customer interactions. The CSV files include:

1. **users.csv**: Contains information about customers visiting the SeaFoodKart website.
  - o `user_id`: Unique identifier for each customer.
  - o `cookie_id`: Unique identifier associated with the customer's browser.
  - o `join_date`: Date the user joined the platform.
2. **events.csv**: Captures the different events that occur on the website.
  - o `visit_id`: Unique identifier for each visit.
  - o `cookie_id`: Unique identifier associated with the customer's browser.
  - o `event_time`: Timestamp when the event occurred.
  - o `event_type`: Identifier for the type of event.
  - o `sequence_number`: Order of the event in the visit.
  - o `page_id`: Identifier for the page on which the event occurred.
  - o `campaign_id`: Identifier for the campaign related to the event.
3. **event\_identifier.csv**: Maps event types to their descriptions.
  - o `event_type`: Identifier for the type of event.
  - o `event_name`: Description of the event type.
4. **page\_heirarchy.csv**: Provides information about the different pages on the website.
  - o `page_id`: Identifier for the page.
  - o `page_name`: Name of the page.
  - o `product_id`: Identifier for the product on the page.
  - o `product_category`: Category of the product.
5. **campaign\_identifier.csv**: Contains information about marketing campaigns.
  - o `campaign_id`: Identifier for the campaign.
  - o `products`: Products associated with the campaign.
  - o `campaign_name`: Name of the campaign.
  - o `start_date`: Start date of the campaign.
  - o `end_date`: End date of the campaign.

## Relation Between Data

- **Users and Events**: The `cookie_id` field links the `users` and `events` tables, allowing us to associate events with specific users.

- **Events and Event Identifier:** The `event_type` field in `events` links to the `event_identifier` table, providing descriptions for the events.
- **Events and Page Hierarchy:** The `page_id` field in `events` links to the `page_heirarchy` table, allowing us to understand the page on which an event occurred.
- **Events and Campaign Identifier:** The `campaign_id` field in `events` links to the `campaign_identifier` table, providing context about marketing campaigns related to events.



## What We Found from the Data

## 1. User Metrics

- **Unique Users:** We found that there are a total of [unique users] unique users visiting the website.

```
SELECT COUNT(DISTINCT user_id) AS unique_users
FROM Users;
```

- **Average Number of Visits per User:** On average, users visit the website [average visits] times.

```
SELECT AVG(visit_count) AS avg_visits_per_user
FROM (
    SELECT u.user_id, COUNT(e.visit_id) AS visit_count
    FROM Events e
    JOIN Users u ON e.cookie_id = u.cookie_id
    GROUP BY u.user_id
) AS user_visits;
```

## 2. Visit Analysis

- **Distribution of Visits Over Time:** We analyzed the number of visits per day, identifying trends and patterns in user activity.

```
SELECT CAST(event_time AS DATE) AS visit_date, COUNT(DISTINCT
visit_id) AS visit_count
FROM Events
GROUP BY CAST(event_time AS DATE)
ORDER BY visit_date;
```

- **Peak Visit Times and Days:** We found the hours of the day when visits peak, helping to understand user behavior.

```
SELECT DATEPART(HOUR, event_time) AS visit_hour, COUNT(DISTINCT
visit_id) AS visit_count
FROM Events
GROUP BY DATEPART(HOUR, event_time)
ORDER BY visit_count DESC;
```

## 3. Event Analysis

- **Count Different Event Types:** We identified the frequency of different events such as page views, purchases, etc.

```
SELECT e.event_name, COUNT(ev.event_type) AS event_count
```

```
FROM Events ev
JOIN Event_Identifier e ON ev.event_type = e.event_type
GROUP BY e.event_name;
```

- **Analyze the Sequence of Events:** We looked into the sequence of events during visits to understand user interactions.

```
SELECT visit_id, sequence_number, event_name
FROM Events ev
JOIN Event_Identifier e ON ev.event_type = e.event_type
ORDER BY visit_id, sequence_number;
```

## 4. Page Performance

- **Most and Least Visited Pages:** We determined which pages are the most and least popular among users.

```
SELECT p.page_name, COUNT(e.event_type) AS visit_count
FROM Events e
JOIN page_heirarchy p ON e.page_id = p.page_id
GROUP BY p.page_name
ORDER BY visit_count DESC;
```

- **Time Spent on Each Page:** We calculated the average time spent by users on each page.

```
SELECT p.page_name, AVG(DATEDIFF(SECOND, e1.event_time,
e2.event_time)) AS avg_time_spent
FROM Events e1
JOIN Events e2 ON e1.visit_id = e2.visit_id AND e1.sequence_number =
e2.sequence_number - 1
JOIN page_heirarchy p ON e1.page_id = p.page_id
GROUP BY p.page_name;
```

## 5. Product Performance

- **Most and Least Popular Products:** We identified which products are viewed the most and the least.

```
SELECT p.product_id, p.product_category, COUNT(e.event_type) AS
view_count
FROM Events e
JOIN page_heirarchy p ON e.page_id = p.page_id
WHERE e.event_type = 1 -- Use the correct integer value for
'page_view'
GROUP BY p.product_id, p.product_category
ORDER BY view_count DESC;
```

- **Conversion Rates from Product Page Views to Purchases:** We calculated conversion rates for each product.

```
SELECT p.product_id, p.page_name,
       COALESCE(COUNT(CASE WHEN e.event_type = '2' THEN 1 END) * 1.0
/ NULLIF(COUNT(CASE WHEN e.event_type = '1' THEN 1 END), 0), 0) AS
conversion_rate
FROM Events e
JOIN page_hierarchy p ON e.page_id = p.page_id
GROUP BY p.product_id, p.page_name;
```

## 6. Campaign Analysis

- **Campaign Performance:** We analyzed the effectiveness of different marketing campaigns.

```
SELECT c.campaign_name,
       COUNT(DISTINCT e.visit_id) AS visit_count,
       COUNT(CASE WHEN e.event_type = 4 THEN 1 END) AS click_count,
       COUNT(CASE WHEN e.event_type = 3 THEN 1 END) AS purchase_count
FROM Events e
JOIN Campaign_Identifier c
  ON e.event_time BETWEEN c.start_date AND c.end_date
GROUP BY c.campaign_name;
```

- **Visit-level Campaign Analysis:** We looked at campaign performance at the level of individual visits.

```
SELECT e.visit_id, u.user_id, MIN(e.event_time) AS visit_start_time,
       COUNT(CASE WHEN e.event_type = 1 THEN 1 END) AS page_views,
       COUNT(CASE WHEN e.event_type = 2 THEN 1 END) AS cart_adds,
       MAX(CASE WHEN e.event_type = 3 THEN 1 ELSE 0 END) AS
purchase_flag,
       c.campaign_name,
       COUNT(CASE WHEN e.event_type = 4 THEN 1 END) AS
impression_count,
       COUNT(CASE WHEN e.event_type = 5 THEN 1 END) AS click_count,
       STUFF(
         (SELECT ', ' + p.page_name
          FROM Events e2
          JOIN page_hierarchy p ON e2.page_id = p.page_id
          WHERE e2.visit_id = e.visit_id AND e2.event_type = 2
          FOR XML PATH(''), TYPE).value('.', 'NVARCHAR(MAX)'), 1,
2, ''
       ) AS cart_products
FROM Events e
JOIN Users u ON e.cookie_id = u.cookie_id
LEFT JOIN Campaign_Identifier c
  ON e.event_time BETWEEN c.start_date AND c.end_date
GROUP BY e.visit_id, u.user_id, c.campaign_name;
```

## Problems Identified

1. **Data Inconsistencies:** There were issues with data type mismatches, especially with event types being stored as integers rather than strings.
2. **Missing or Incomplete Data:** Some records had missing values, particularly in the campaign and event data.
3. **Complex Query Requirements:** Some analyses required complex queries to aggregate and join data from multiple tables.

## How We Solved the Problems

1. **Standardized Data Types:** Ensured consistency in data types by properly mapping event types and other identifiers to their respective descriptions.
2. **Data Cleaning:** Handled missing values and ensured that all necessary data was present for analysis.
3. **Optimized Queries:** Used best practices to write efficient and clear queries, making use of subqueries, joins, and aggregation functions as needed.

## Suggestions for Increasing Sales

1. **Targeted Marketing Campaigns:** Use insights from campaign performance analysis to focus on the most effective campaigns.
2. **Personalized Recommendations:** Leverage user behavior data to offer personalized product recommendations.
3. **Optimized User Experience:** Improve the design and functionality of the most visited pages to enhance user engagement.
4. **Enhanced Product Listings:** Highlight popular products and provide detailed information to improve conversion rates.
5. **Peak Time Promotions:** Run special promotions during peak visit times to maximize sales.