User Behavior Analytics DBMS Project

By

Shravya Aellanagoji

22MMB0A57

Problem Statement

In this project, our goal is to develop a comprehensive **database management system** to track and analyze user behavior on a SaaS-based web application.

This system is designed to assist product and business teams in understanding how users interact with the application across various touchpoints. The database will store detailed user information including sign-up details, subscription plans, login sessions, and in-app events such as clicks and page visits.

This information will help identify engagement trends, analyze feature usage, and assess the overall performance of different modules. Insights from this data can be used to improve the user experience, optimize the application's layout, and prioritize feature development.

Additionally, the database will manage aspects such as subscription status, user feedback, and session duration, helping teams monitor customer satisfaction and churn risk. By analyzing feedback and usage metrics, decisions regarding pricing, UX improvements, and marketing campaigns can be made more data-driven.

To ensure usability and scalability, the system will support efficient querying and normalization, and be designed with relational integrity, allowing for seamless integration into business intelligence dashboards or product reports.

TABLES

Users

ATTRIBUTE	DATATYPE	CONSTRAINTS
user_id	INT	PRIMARY KEY NOT NULL
name	VARCHAR(50)	NOT NULL
email	VARCHAR(100)	NOT NULL
signup_date	DATE	NOT NULL
plan_type	VARCHAR(20)	NOT NULL

Sessions

ATTRIBUTE	DATATYPE	CONSTRAINTS
session_id	INT	PRIMARY KEY NOT NULL
user_id	INT	FOREIGN KEY NOT NULL
login_time	TIMESTAMP	NOT NULL
logout_time	TIMESTAMP	NOT NULL

Events

ATTRIBUTE	DATATYPE	CONSTRAINTS
event_id	INT	PRIMARY KEY NOT NULL
session_id	INT	FOREIGN KEY NOT NULL
event_type	VARCHAR(20)	NOT NULL
page_name	VARCHAR(50)	NOT NULL

Subscriptions

ATTRIBUTE	DATATYPE	CONSTRAINTS
subscription_id	INT	PRIMARY KEY NOT NULL
user_id	INT	FOREIGN KEY NOT NULL
start_date	DATE	NOT NULL
end_date	DATE	NOT NULL
status	VARCHAR(20)	NOT NULL
payment_mode	VARCHAR(20)	NOT NULL

FeedBack

ATTRIBUTE	DATATYPE	CONSTRAINTS
feedback_id	INT	PRIMARY KEY NOT NULL
user_id	INT	FOREIGN KEY NOT NULL
rating	INT	NOT NULL
comments	VARCHAR(255)	NOT NULL
submitted_on	DATE	NOT NULL

NORMALISATION

1. **STUDENT**

   Primary key: user_id

   Prime attribute: user_id

   Non-Prime attributes: name, email, signup_date, plan_type

   There is no Partial Dependency. Hence, the table is in **2NF**

   There is no Transitive Dependency. Hence, the table is in **3NF**

   All attributes are fully dependent on the candidate key. Hence, the table is in **BCNF**

2. **SESSIONS**

   Primary key: session_id

   Prime attribute: session_id

   Non-Prime attributes: user_id, login_time, logout_time

   There is no Partial Dependency. Hence, the table is in **2NF**

   There is no Transitive Dependency. Hence, the table is in **3NF**

   All attributes are fully dependent on the candidate key. Hence, the table is in **BCNF**

3. **EVENTS**

   Primary key: event_id

   Prime attribute: event_id

   Non-Prime attributes: session_id, event_type, page_name, timestamp

There is no Partial Dependency. Hence, the table is in **2NF**

There is no Transitive Dependency. Hence, the table is in **3NF**

All attributes are fully dependent on the candidate key. Hence, the table is in **BCNF**

4. **SUBSCRIPTIONS**

Primary key: subscription_id

Prime attribute: subscription_id

Non-Prime attributes: user_id, start_date, end_date, status, payment_mode

There is no Partial Dependency. Hence, the table is in **2NF**

There is no Transitive Dependency. Hence, the table is in **3NF**

All attributes are fully dependent on the candidate key. Hence, the table is in **BCNF**

5. **FEEDBACK**

Primary key: feedback_id

Prime attribute: feedback_id

Non-Prime attributes: user_id, rating, comments, submitted_on

There is no Partial Dependency. Hence, the table is in **2NF**

There is no Transitive Dependency. Hence, the table is in **3NF**

All attributes are fully dependent on the candidate key. Hence, the table is in **BCNF**

SQL OPERATIONS

Creation of Tables

1. **Users**

```
CREATE TABLE users (

  user_id INT PRIMARY KEY,

  name VARCHAR(50),

  email VARCHAR(100),

  signup_date DATE,

  plan_type VARCHAR(20) -- 'free' or 'premium'

);
```

2. **Sessions**

```
CREATE TABLE sessions (
  session_id INT PRIMARY KEY,
  user_id INT,
  login_time TIMESTAMP,
  logout_time TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

3. **Events**

```
CREATE TABLE events (
  event_id INT PRIMARY KEY,
  session_id INT,
  event_type VARCHAR(20), -- click, view
  page_name VARCHAR(50),
  timestamp TIMESTAMP,
  FOREIGN KEY (session_id) REFERENCES sessions(session_id)
);
```

4. **Subscriptions**

```
CREATE TABLE subscriptions (
  subscription_id INT PRIMARY KEY,
  user_id INT,
  start_date DATE,
  end_date DATE,
  status VARCHAR(20), -- active, expired
  payment_mode VARCHAR(20),
  FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

5. **FeedBack**

```
CREATE TABLE feedback (
  feedback_id INT PRIMARY KEY,
  user_id INT,
  rating INT,
  comments VARCHAR(255),
  submitted_on DATE,
  FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

Insertion of values

1. **Users**

```
INSERT INTO users VALUES

(1, 'Shravya', 'shravya@mail.com', '2024-01-10', 'free'),

(2, 'Ravi', 'ravi@mail.com', '2024-02-15', 'premium'),

(3, 'Anjali', 'anjali@mail.com', '2024-03-05', 'free'),

(4, 'Kiran', 'kiran@mail.com', '2024-03-12', 'premium'),
(5, 'Meena', 'meena@mail.com', '2024-04-01', 'free');
```

2. **Sessions**

```
INSERT INTO sessions VALUES
(101, 1, '2024-05-01 09:00:00', '2024-05-01 09:45:00'),
(102, 2, '2024-05-01 10:00:00', '2024-05-01 10:30:00'),
(103, 3, '2024-05-02 08:30:00', '2024-05-02 09:00:00'),
(104, 1, '2024-05-03 07:00:00', '2024-05-03 07:20:00'),
(105, 4, '2024-05-04 11:00:00', '2024-05-04 11:50:00');
```

3. **Events**

```
INSERT INTO events VALUES
(1001, 101, 'click', 'dashboard', '2024-05-01 09:05:00'),
(1002, 101, 'view', 'settings', '2024-05-01 09:10:00'),
(1003, 102, 'click', 'profile', '2024-05-01 10:05:00'),
(1004, 104, 'view', 'dashboard', '2024-05-03 07:10:00'),
(1005, 105, 'click', 'pricing', '2024-05-04 11:15:00');
```

4. **Subscriptions**

```
INSERT INTO subscriptions VALUES
(201, 2, '2024-02-15', '2025-02-14', 'active', 'credit_card'),
(202, 4, '2024-03-12', '2025-03-11', 'active', 'upi');
```

5. **FeedBack**

```
INSERT INTO feedback VALUES
(301, 1, 4, 'Good UI, needs more features.', '2024-05-01'),
(302, 2, 5, 'Very useful app!', '2024-05-02'),
(303, 3, 3, 'Average experience.', '2024-05-03');
```

SQL Queries for Analytics

1. **Total sessions by each user**

```
SELECT u.name, COUNT(s.session_id) AS session_count
FROM users u
JOIN sessions s ON u.user_id = s.user_id
GROUP BY u.name;
```

2. **Daily active users**

```
SELECT login_time::date AS day, COUNT(DISTINCT user_id) AS dau
FROM sessions
GROUP BY day
ORDER BY day;
```

3. **Active vs Free plan users**

```
SELECT plan_type, COUNT(*) AS total_users
FROM users
GROUP BY plan_type;
```

4. **Average session duration**

```
SELECT AVG(TIMESTAMPDIFF(MINUTE, login_time, logout_time))
AS avg_session_minutes
FROM sessions;
```

5. **Average rating from feedback**

```
SELECT AVG(rating) AS avg_rating FROM feedback;
```

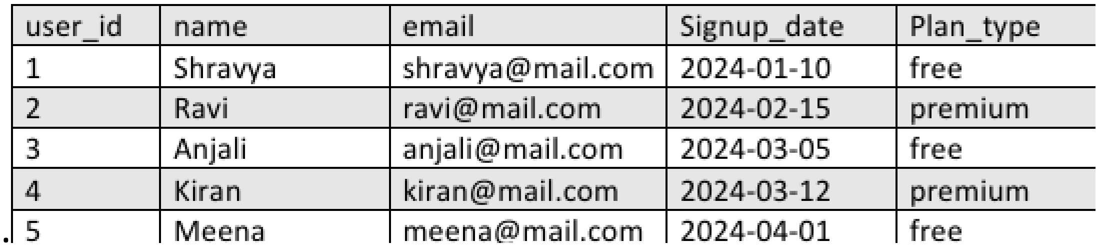
6. **Top used features**

```
SELECT  page_name,
   COUNT(*) AS count
FROM events
```

```
GROUP BY page_name
ORDER BY count DESC;
```

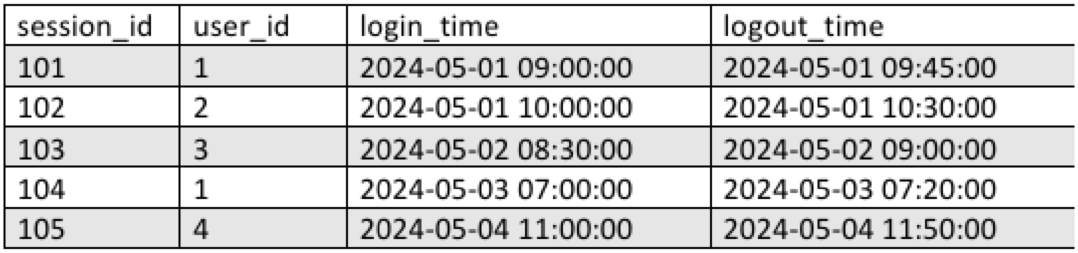
OUTPUTS

1. User table



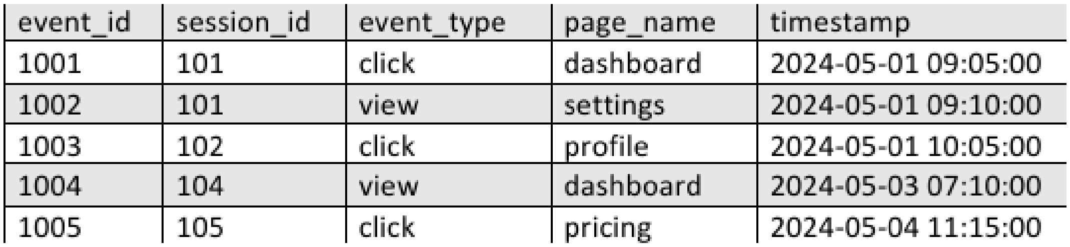
user_id	name	email	Signup_date	Plan_type
1	Shravya	shravya@mail.com	2024-01-10	free
2	Ravi	ravi@mail.com	2024-02-15	premium
3	Anjali	anjali@mail.com	2024-03-05	free
4	Kiran	kiran@mail.com	2024-03-12	premium
5	Meena	meena@mail.com	2024-04-01	free

2. Sessions table



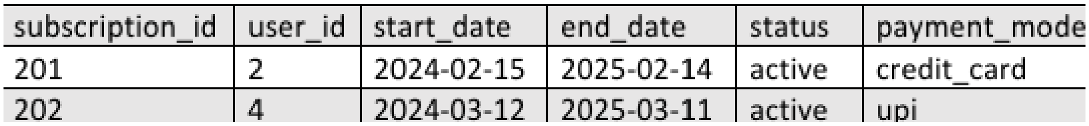
session_id	user_id	login_time	logout_time
101	1	2024-05-01 09:00:00	2024-05-01 09:45:00
102	2	2024-05-01 10:00:00	2024-05-01 10:30:00
103	3	2024-05-02 08:30:00	2024-05-02 09:00:00
104	1	2024-05-03 07:00:00	2024-05-03 07:20:00
105	4	2024-05-04 11:00:00	2024-05-04 11:50:00

3. Events table



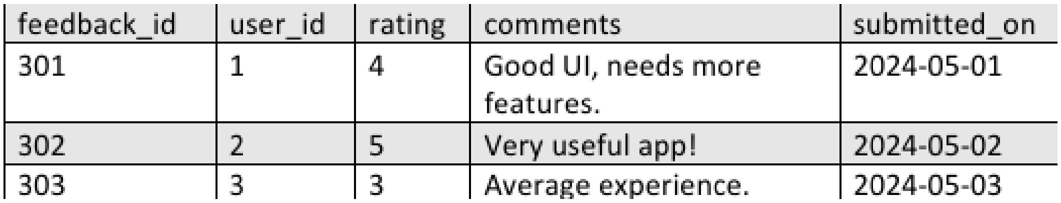
event_id	session_id	event_type	page_name	timestamp
1001	101	click	dashboard	2024-05-01 09:05:00
1002	101	view	settings	2024-05-01 09:10:00
1003	102	click	profile	2024-05-01 10:05:00
1004	104	view	dashboard	2024-05-03 07:10:00
1005	105	click	pricing	2024-05-04 11:15:00

4. Subscriptions table



subscription_id	user_id	start_date	end_date	status	payment_mode
201	2	2024-02-15	2025-02-14	active	credit_card
202	4	2024-03-12	2025-03-11	active	upi

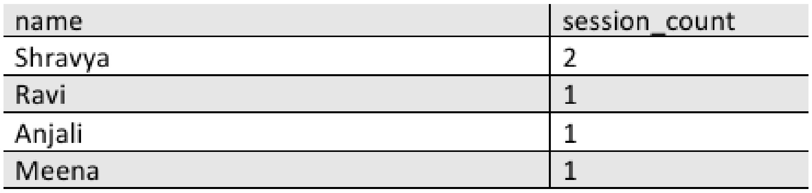
5. Feedback table



feedback_id	user_id	rating	comments	submitted_on
301	1	4	Good UI, needs more features.	2024-05-01
302	2	5	Very useful app!	2024-05-02
303	3	3	Average experience.	2024-05-03

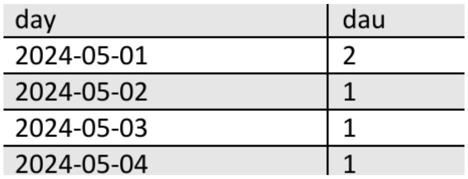
OUTPUTS FOR SQL QUERIES

- **Total sessions by each user**



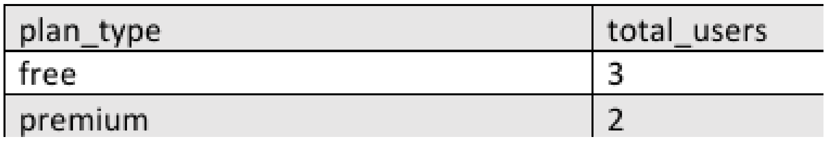
name	session_count
Shravya	2
Ravi	1
Anjali	1
Meena	1

- **Daily active users**



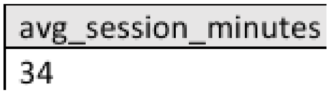
day	dau
2024-05-01	2
2024-05-02	1
2024-05-03	1
2024-05-04	1

- **Active vs Free Plan Users**



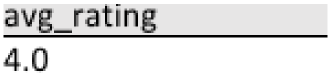
plan_type	total_users
free	3
premium	2

- **Average Session Duration (in Minutes)**



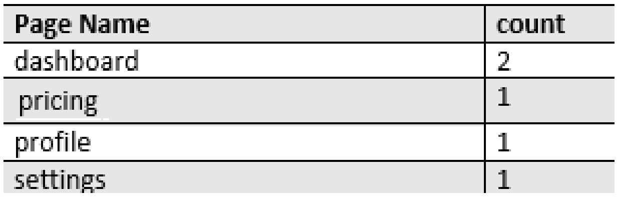
avg_session_minutes
34

- **Average rating from feedback**



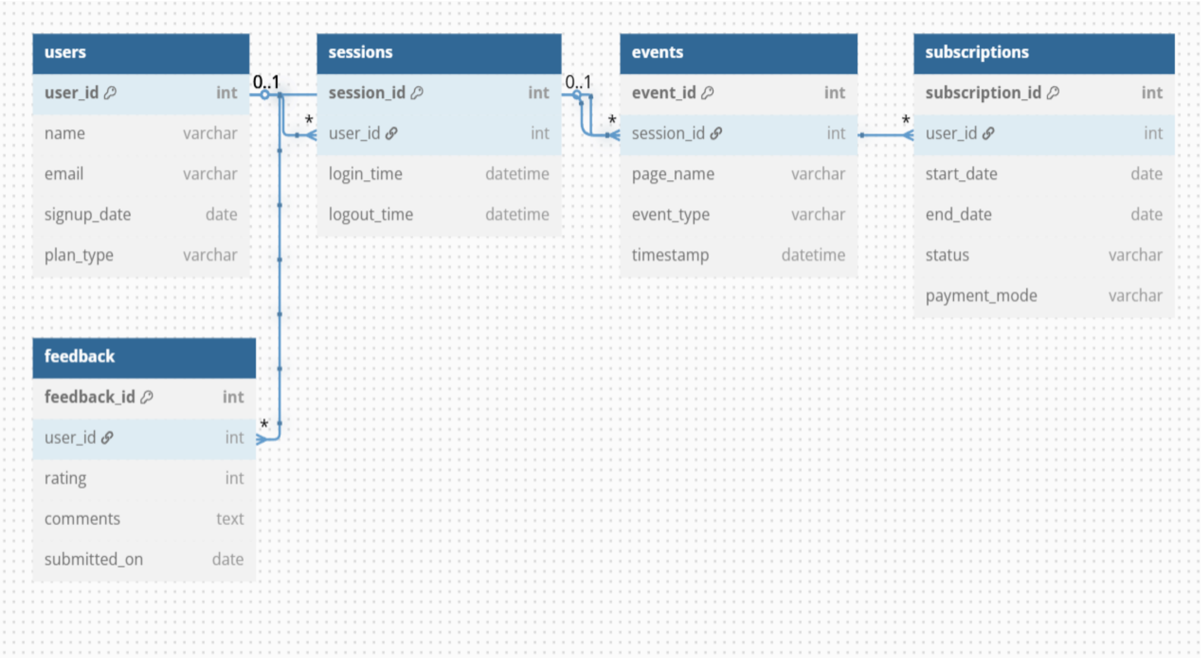
avg_rating
4.0

- **Top used features**



Page Name	count
dashboard	2
pricing	1
profile	1
settings	1

RELATIONAL SCHEMA WITH NORMALISED TABLES



users	
user_id	int
name	varchar
email	varchar
signup_date	date
plan_type	varchar

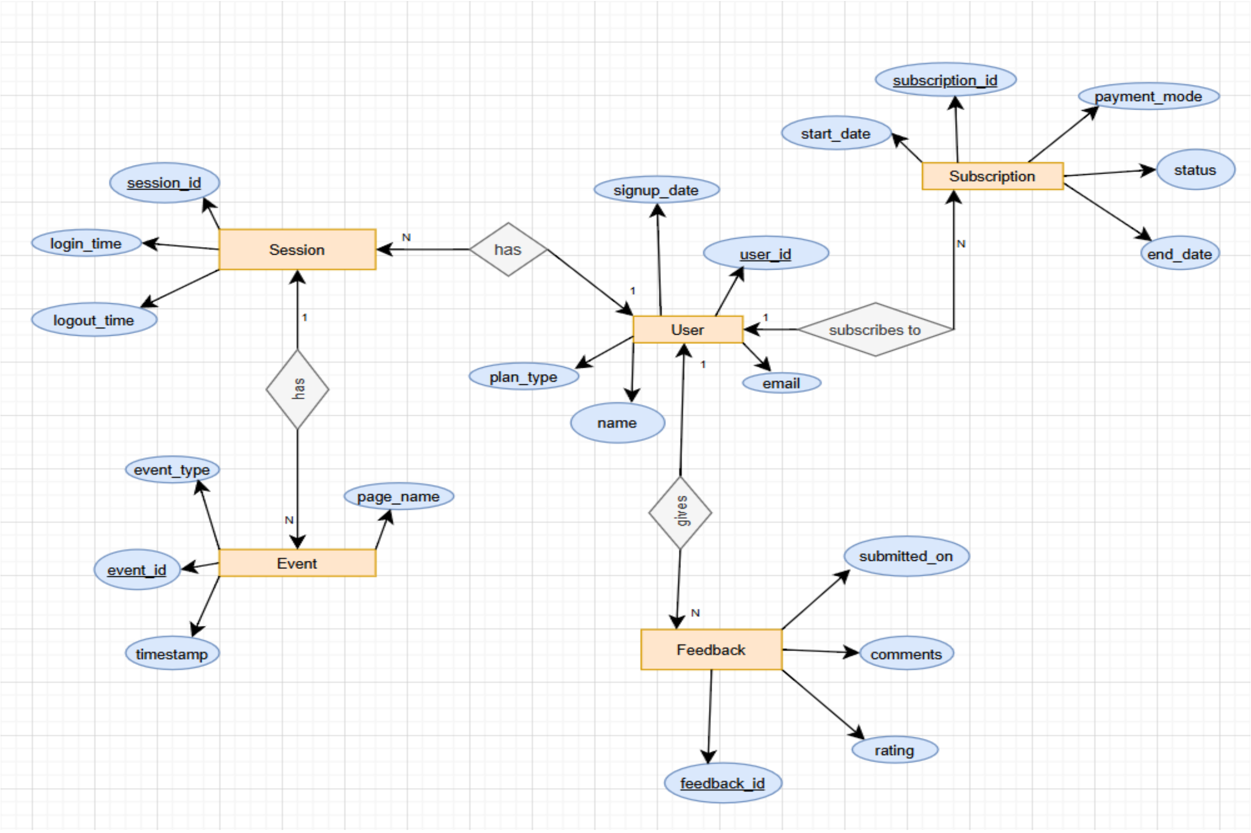
sessions	
session_id	int
user_id	int
login_time	datetime
logout_time	datetime

events	
event_id	int
session_id	int
page_name	varchar
event_type	varchar
timestamp	datetime

subscriptions	
subscription_id	int
user_id	int
start_date	date
end_date	date
status	varchar
payment_mode	varchar

feedback	
feedback_id	int
user_id	int
rating	int
comments	text
submitted_on	date

ER DIAGRAM



subscription_id
payment_mode
start_date
Subscription
status
end_date
session_id
signup_date
login_time
Session
N
has
user_id
1
logout_time
1
User
1
subscribes to
N
plan_type
email
name
1
has
event_type
page_name
N
gives
event_id
Event
submitted_on
N
timestamp
Feedback
comments
rating
feedback_id