

Database Design and

Implementation

MOD002589

Faculty: Science and Technology

Department: Computing and Technology

Student ID: Erik Spasov SID: 2279588

Your Team: Anish Kumar(2290275), Babatunde Bello(2246953), Chinedu Ugwu(2278973)

Academic Year: 2024/25

Trimester: 1

**Contents**

[1 Requirement Analysis 4](#_Toc184759428)

[1.1 Brief Introduction 4](#_Toc184759429)

[1.2 List of data fields (Entities and their attributes) 5](#_Toc184759430)

[2 Database design (20%) 13](#_Toc184759432)

[2.1 Entity Relationship Modelling 13](#_Toc184759433)

[2.1.1 Initial Entity Relationship Model 13](#_Toc184759434)

[2.1.2 Extended Entity Relationship Model 14](#_Toc184759435)

[Assumptions: 19](#_Toc184759436)

[**Extended Notations** 21](#_Toc184759437)

[2.2 Normalised Model 23](#_Toc184759438)

[**Normalization Process in Summary:** 27](#_Toc184759439)

[**Final Normalized Model (After 3NF):** 28](#_Toc184759440)

[2.3 Database Schema 30](#_Toc184759441)

[**Schema Design Considerations** 34](#_Toc184759442)

[3. Mapping 35](#_Toc184759443)

[5. Database implementation (10%) 41](#_Toc184759444)

[5. SQL Queries (50%) 53](#_Toc184759445)

[5.1 Query 1 53](#_Toc184759446)

[53](#_Toc184759447)

[5.2 Query 2 53](#_Toc184759448)

[5.3 Query 3 54](#_Toc184759449)

[5.4 Query 4 54](#_Toc184759450)

[5.5 Query 5 55](#_Toc184759451)

[5.6 Query 6 56](#_Toc184759452)

[5.7 Query 7 56](#_Toc184759453)

[5.8 Query 8 57](#_Toc184759454)

[5.9 Query 9 58](#_Toc184759455)

[6 Personal Query 10 58](#_Toc184759456)

[6.1. Personal Query 11 59](#_Toc184759457)

[6.2. References 61](#_Toc184759458)

# Requirement Analysis

## 1.1 Brief Introduction

In today’s data-driven landscape, effective database design and implementation are vital for organizations that aim to optimize operational processes and decision-making (Elmasri & Navathe, 2021). The focus of this project is to design and implement a relational database for Together Culture, a community-led company, to enable efficient storage, retrieval, and management of data. The goal is to create a robust, scalable, and flexible database that will support the organization in monitoring member engagement, space usage, and event attendance while laying a foundation for future growth.

Database systems are integral to structuring organizational data, enabling seamless access to insights necessary for improving services and fostering relationships with members (Connolly & Begg, 2019). Through careful planning, the project will develop an Extended Entity Relationship Diagram (EERD) and a normalized data model, which will subsequently guide the database schema design. The database will incorporate multiple entities representing key areas such as Members, Events, Attendance, Spaces, and User Engagement. Each entity and its relationships will be structured to facilitate data integrity, ensure efficient queries, and support Together Culture’s operations.

To ensure comprehensive analysis, this project adheres to the principles of relational database theory, leveraging normalization techniques to reduce redundancy and promote data consistency (Ramakrishnan & Gehrke, 2020). Furthermore, SQL queries will be crafted to answer real-world questions provided by Together Culture, underscoring the database's practicality for the organization. This design approach aligns with best practices in database development and aims to provide Together Culture with a data model that can grow in parallel with its expanding needs (Silberschatz, Korth & Sudarshan, 2019).

The implementation phase will see the creation of actual tables, data entry, and verification of database functionality through SQL queries. This database is intended as a comprehensive solution for data organization and retrieval, aimed at supporting Together Culture's long-term operational goals.

## 1.2 List of data fields (Entities and their attributes)

Together Culture’s objectives are focused on understanding member behaviour, optimizing space and event use, and tailoring services to member preferences. Our analysis identifies essential entities and fields needed to meet these goals while suggesting additional attributes to enhance the database's flexibility.

**Core Entities and Essential Fields**

Based on the documents and questions, we’ve identified the following entities and core fields required to address Together Culture's objectives:

#### **1. User (Entity)**

**Purpose**: Captures the personal and contact details of individuals using the system.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Justification** |
| User\_ID | Unique identifier for each user. | Ensures each user is uniquely identifiable. |
| First\_Name | User’s first name. | Required for personalization and identity. |
| Last\_Name | User’s last name. | Provides a complete identification. |
| Email | User’s email address. | Essential for communication. |
| Phone\_No | User’s contact number. | Allows direct communication. |
| Date\_Of\_Birth | User’s date of birth. | Enables demographic analysis and validation. |
| Gender | User’s gender. | Useful for targeted engagement strategies. |
| User\_Type\_ID | Foreign key to identify the type of user. | Facilitates role-based access and categorization. |
| User\_Log\_ID | Foreign key linking to the User\_Log table. | Tracks user activity such as login sessions, check-ins, and check-outs, ensuring accountability and providing insights into user engagement patterns. |

**Additional Suggested Attributes**:

* Registration\_Date (datetime): Tracks when the user joined the system. *(Justification: Helps analyze user growth over time.)*

#### **2. Event (Entity)**

**Purpose**: Stores information about events hosted by Together Culture.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Justification** |
| Event\_ID | Unique identifier for each event. | Enables event tracking and management. |
| Event\_Name | Title of the event. | Describes the event purpose clearly. |
| Event\_Type | Category of the event (e.g., Workshop, Seminar). | Useful for filtering events by type. |
| Event\_Date | Scheduled date of the event. | Allows chronological planning. |
| Event\_Time | Start time of the event. | Important for scheduling. |
| Event\_Duration | Duration of the event. | Aids in resource allocation. |
| Organiser\_ID | Foreign key referencing the organiser. | Links the event to its organiser. |
| Location | Venue or online link for the event. | Helps participants locate the event. |
| Description | Detailed description of the event. | Provides additional context. |
| Ticket\_Price | Cost of attending the event. | Enables pricing transparency and budgeting for participants. |
| Ticket\_Type | Specifies the type of ticket (e.g., Regular, Standard, Premium). | Useful for differentiating access levels or benefits tied to the ticket. |

**Additional Suggested Attributes**:

* Event\_Status (varchar(50)): Tracks whether the event is Scheduled, Ongoing, or Completed. (Justification: Enhances scheduling and status tracking capabilities.)
* Target\_Audience (varchar(100)): Specifies who the event is for (e.g., youth, professionals). *(Justification: Improves marketing efforts.)*
* Feedback\_Score (decimal(3,2)): Average rating based on attendee feedback. *(Justification: Enables quality improvement.)*

#### **3. Space (Entity)**

**Purpose**: Represents physical or virtual spaces available for events and bookings.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Justification** |
| Space\_ID | Unique identifier for each space. | Allows differentiation between spaces. |
| Space\_Name | Name of the space. | Useful for user-friendly identification. |
| Space\_Capacity | Maximum number of people the space can hold. | Crucial for event planning. |
| Area | Total area of the space (e.g., in square meters). | Provides information for assessing space utility. |
| Location | Address or virtual link of the space. | Helps identify the space for bookings. |

**Additional Suggested Attributes**:

* Equipment\_Available (varchar(255)): List of equipment provided (e.g., projector, microphone). *(Justification: Assists users in selecting appropriate spaces.)*

#### **4. User\_Type (Entity)**

**Purpose**: Defines various roles or categories of users in the system.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Justification** |
| **User\_Type\_ID** | Unique identifier for each user type. | Differentiates and tracks user roles across the system. |
| **Type\_Name** | Name of the user type (e.g., Admin, Guest). | Simplifies identification of roles for functionality and reporting. |
| **Description** | Details about the user type’s permissions. | Explains the purpose and capabilities of each role, ensuring clarity. |
| **Created\_Date** | Timestamp of when the user type was created. | Tracks when a role was added to the system for historical records. |
| **Permission\_Level** | Numeric or descriptive level indicating the access rights or privileges associated with the user type. | Defines hierarchical or specific access permissions, enabling role-based access control within the system. |
| **IsActive** | Boolean attribute (e.g., True or False) indicating whether the user type is currently active. | Facilitates deactivation of roles that are no longer in use, ensuring the system operates efficiently and securely. |
| **Organisation\_ID** | Links the user type to a specific organization. | Supports multi-organization functionality if applicable in the future. |

**Additional Suggested Attributes**:

* Default\_Status (varchar(50)): Initial status assigned to users of this type. *(Justification: Helps standardize user onboarding.)*

#### **5. Booking (Entity)**

**Purpose**: Records event and space bookings by users, replacing the concept of attendance tracking.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Justification** |
| Booking\_ID | Unique identifier for each booking. | Ensures each booking is uniquely identifiable. |
| User\_ID | Foreign key referencing the user. | Links the booking to a specific user. |
| Event\_ID | Foreign key referencing the booked event (nullable). | Tracks event-related bookings. |
| Space\_ID | Foreign key referencing the booked space (nullable). | Tracks space-related bookings. |
| Booking\_Date | Date and time when the booking was made. | Provides historical tracking of bookings. |
| Status | Current status of the booking (e.g., Pending, Confirmed, Cancelled). | Helps manage the booking lifecycle. |

**Additional Suggested Attributes**:

* Special\_Requests (text): Captures user-specific requests (e.g., accessibility requirements, equipment setup).  
  *(Justification: Enhances user satisfaction by addressing individual needs.)*
* Payment\_Status (varchar(50)): Indicates whether the booking payment has been completed.  
  *(Justification: Tracks financial aspects of bookings for better management.)*

#### **6. Tag (Entity)**

**Purpose**: Represents labels or categories assigned to users, events, or spaces to aid in classification and personalization.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Justification** |
| Tag\_ID | Unique identifier for each tag. | Ensures each tag is distinct. |
| Tag\_Name | Name of the tag (e.g., "Youth", "Networking"). | Helps categorize and filter content. |
| Tag\_Description | Description of the tag. | Provides context for usage. |
| Created\_Date | Date and time the tag was created. | Useful for tracking changes over time. |

### 

### **7.Intrest (Entity)**

**Purpose**: Captures the specific interests of users for personalized engagement and recommendations.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Justification** |
| Interest\_ID | Unique identifier for each interest. | Ensures each interest is uniquely tracked. |
| Interest\_Name | Name of the interest (e.g., Art, Technology). | Categorizes user preferences. |

#### **8. Digital\_Engagement (Entity)**

**Purpose**: Captures and tracks user interactions with digital content or events to analyze engagement patterns and improve virtual engagement strategies.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Justification** |
| **Engagement\_ID** | Unique identifier for each engagement record. | Ensures unique tracking of each digital interaction. |
| **Engagement\_Type** | Type of engagement (e.g., Event Check-In, Online Survey). | Categorizes engagements for reporting and analysis. |
| **User\_ID** | Foreign key referencing the user. | Links the engagement to the specific user. |
| **Status** | Status of the engagement (e.g., Completed, Pending). | Tracks the progress or completion of the interaction. |
| **Engagement\_Date** | Date and time of the engagement. | Provides a timestamp for chronological tracking. |
| **Engagement\_Time** | Time of the engagement. | Allows detailed scheduling and analysis of engagement activity based on time. |

**Additional Suggested Attributes**:

* **Engagement\_Purpose (varchar(100))**: Provides additional context for the engagement (e.g., "Pre-event survey"). (Justification: Adds clarity and allows detailed reporting on user activities.)
* **Feedback\_Rating (decimal(3,2))**: Captures a rating (if applicable) for feedback-related engagements. (**Justification**: Enables analysis of satisfaction for improvement purposes.)

#### **9. Organisation (Entity)**

**Purpose**: Captures information about organizations involved in the Together Culture platform, enabling collaboration, reporting, and communication.

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Description** | **Justification** |
| Organisation\_ID | Unique identifier for each organization. | Ensures each organization is uniquely identifiable. |
| Organisation\_Name | Name of the organization. | Helps identify organizations for collaboration and reporting purposes. |
| Department | Department within the organization. | Provides detailed context for specific collaborations or roles. |
| Contact\_No | Contact number for the organization. | Facilitates communication with the organization. |
| Email | Email address for the organization. | Essential for correspondence and record-keeping. |
| Location | Address of the organization. | Useful for tracking geographical distribution of collaborators or participants. |

#### **10. User\_Log (Entity)**

**Purpose**: The User\_Log entity tracks and records user activity within the system, capturing key details about session engagement and interactions.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Justification** |
| User\_Log\_ID | Unique identifier for each user log entry. | Ensures every log entry is uniquely identifiable. |
| Check\_In | Date and time the user checked into the system. | Tracks when users start their sessions or activities. |
| Check\_Out | Date and time the user checked out of the system. | Tracks session or activity end time. |
| Type | Type of log activity (e.g., Login, Event Check-In). | Helps classify and analyze various user interactions. |
| Duration | Duration of the user's activity in the system. | Provides insights into session lengths and user engagement. |

**Additional Suggested Attributes**:

* **Activity\_Details (text)**: Captures additional context or specifics of the activity, such as event attended or space booked. (Justification: Improves the depth of activity tracking, enabling better analytics.)

#### **11. Organiser (Entity)**

**Purpose**: Stores details of individuals or organizations responsible for organizing events.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Justification** |
| Organiser\_ID | Unique identifier for each organiser. | Ensures each organiser is uniquely identifiable. |
| Organiser\_Name | Name of the organiser (individual or organization). | Helps identify organisers easily and associate them with events. |
| Contact | Contact information for the organiser. | Facilitates communication and coordination with organisers. |
| Organiser\_Type | Type of organiser (e.g., Corporate, Non-Profit). | Provides categorization for analytical and operational purposes. |

**Additional Suggested Attributes**:

* **Address (varchar(255))**: Stores the organiser's address.  
  **Justification**: Useful for correspondence or physical meetings.
* **Website (varchar(100))**: Provides a URL for the organiser's website.  
  **Justification**: Enables participants or admin users to find more information about the organiser.

# Database design (20%)

## 2.1 Entity Relationship Modelling

This section provides two models: an initial Entity Relationship Diagram (ERD) as a preliminary draft, followed by the Extended Entity Relationship Diagram (EERD) which represents the finalized model, implementing extended notations for enhanced functionality and data clarity.

### **2.1.1 Initial Entity Relationship Model**

In the initial Entity Relationship Diagram (ERD), we create a simplified model to outline the fundamental entities and their relationships based on the initial requirement analysis. This model focuses on capturing essential data points and their interactions to meet the identified objectives.

A screenshot of a computer

Description automatically generated

Fig.2.1.1: Initial Entity Relational Diagram

### **2.1.2 Extended Entity Relationship Model**

The Extended Entity Relationship Diagram (EERD) for TogetherCulture represents an enhanced data model designed to capture the comprehensive relationships, attributes, and extended functionalities required for the Community Resource Management System (CRM). This model incorporates extended notations such as and aggregation and composition to ensure optimal data integrity and functional scalability.

A diagram of a computer

Description automatically generated

Fig.2.1.2: Extended Entity Relational Diagram

#### **EERD Diagram Explanation**

The diagram includes key entities such as **User**, **User\_Type**, **Event**, **Space**, **Booking**, **Organization**, **Digital\_Engagement**, **User\_Interest**, **Interest**, **Tag**, **Event\_Tag**, **User\_Log**, and **Organizer** with specific relationships such as aggregation, composition, and multi-valued attributes.

Below is a breakdown of the key components of the **EERD** structure:

1. **User**
   * **Attributes**: User\_ID (PK), First\_Name, Last\_Name, Email, Gender, Date\_Of\_Birth, Phone\_No, User\_Type\_ID (FK), User\_Log\_ID (FK)
   * **Relationships**:
     + Each User is associated with a User\_Type (defines the type of user, such as admin or regular member).
     + Each User has an associated User\_Log, which tracks their check-in and check-out times.
   * **Multi-Valued Attribute**:
     + User\_Interest: A user can have multiple interests (e.g., Sports, Technology), which is captured through the User\_Interest entity linking users to specific interests.
2. **User\_Type**
   * **Attributes**: User\_Type\_ID (PK), Type\_Name, Description, Created\_Date, Permission\_Level, IsActive, Organisation\_ID (FK)
   * **Relationships**:
     + User\_Type is linked to the Organisation entity (each user type belongs to a specific organization).
3. **User\_Log**
   * **Attributes**: User\_Log\_ID (PK), Check\_In, Check\_Out, Type, Duration
   * **Relationships**:
     + A User\_Log is associated with a single User and tracks their engagement times (check-in and check-out).
4. **Booking**
   * **Attributes**: Booking\_ID (PK), User\_ID (FK), Event\_ID (FK), Booking\_Date, Space\_ID (FK), Booking\_Status
   * **Relationships**:
     + A Booking represents the booking of a specific Event by a User for a particular Space.
     + **Aggregation**: The Booking entity aggregates User, Event, and Space. It tracks additional attributes like Booking\_Date and Booking\_Status to provide detailed insights about the booking.
5. **Event**
   * **Attributes**: Event\_ID (PK), Event\_Name, Event\_Type, Event\_Date, Event\_Time, Event\_Duration, Event\_Description, Location, Organiser\_ID (FK), Ticket\_Price, Ticket\_Type
   * **Relationships**:
     + Events are organized by an Organiser.
     + **Aggregation**: The Event\_Tag entity links events to specific tags of interest.
     + **Multi-Valued Attribute**: Events can have multiple tags, such as "Networking" or "Workshop", represented by the Event\_Tag table.
6. **Organiser**
   * **Attributes**: Organiser\_ID (PK), Organiser\_Name, Contact, Type
   * **Relationships**:
     + An event is associated with an organiser, responsible for the event's management.
7. **Organisation**
   * **Attributes**: Organisation\_ID (PK), Organisation\_Name, Department, Location, Contact\_No, Email
   * **Relationships**:
     + An Organisation is linked to User\_Type, where each user type corresponds to an organization.
8. **Space**
   * **Attributes**: Space\_ID (PK), Name, Capacity, Area, Location
   * **Relationships**:
     + A Space is associated with multiple Bookings. Each booking links a user to an event and space.
     + The relationship between Space and Booking is one-to-many, meaning each space can be booked multiple times but a booking pertains to one specific space.
9. **Interest**
   * **Attributes**: Interest\_ID (PK), Interest\_Name
   * **Relationships**:
     + Users are linked to their interests through the User\_Interest table, allowing for many-to-many relationships between Users and Interests.
10. **Tag**
    * **Attributes**: Tag\_ID (PK), Tag\_Name, Tag\_Description, Created\_Date
    * **Relationships**:
      + Tags are assigned to events via the Event\_Tag entity, establishing a many-to-many relationship between events and tags.
11. **Event\_Tag**
    * **Attributes**: Event\_Tag\_ID (PK), Event\_ID (FK), Tag\_ID (FK), Date\_Assign
    * **Relationships**:
      + This table captures the many-to-many relationship between events and tags. It shows which tags are assigned to which events, allowing for flexible categorization of events.
12. **Digital Engagement**
    * **Attributes**: Engagement\_ID (PK), User\_ID (FK), Engagement\_Type, Status, Engagement\_Date, Engagement\_Time
    * **Relationships**:
      + The Digital Engagement table records a user's engagement activities, capturing whether they are paused or completed.
      + **Composition**: The engagement activity is directly tied to the User entity, showing that a digital engagement cannot exist without a corresponding user.

### Assumptions:

1. **Users and Membership**:
   * Users can either be fully registered members or non-members. Non-members may still interact with events but cannot book spaces or access restricted features.
   * A user's membership status (active or inactive) is tracked in the system, influencing what they can access.
2. **Event Types**:
   * Events can be categorized into various types (e.g., Networking, Workshop, Webinar, etc.), and these types help define the event's purpose.
   * Events have specific dates and times, but users may be able to book or interact with events even after the date has passed.
3. **Booking Process**:
   * A user can make multiple bookings for different events, and each booking is associated with a specific space for an event.
   * A booking can exist even if the associated event or space details change (e.g., rescheduled events or updated space information).
4. **Event Tags**:
   * Each event can have multiple tags (e.g., "Technology", "Music", "Networking") to help categorize and filter events.
   * Tags are contextually dependent on the event and cannot exist independently of the event they are linked to.
5. **Engagement Activities**:
   * Digital engagement (e.g., comments, likes, and shares) is tied to a specific user and is considered a composition of that user entity.
   * Engagement activities are deleted when the associated user is deleted from the system.
6. **Spaces**:
   * Spaces are physical or virtual locations that can be booked for events, but not all events require a space (e.g., online webinars may not have a physical space).
   * A space can host multiple events, and its availability is linked directly to event scheduling.
7. **Interests and Personalization**:
   * Users can have multiple interests, which can inform personalized recommendations for events and activities.
   * Interests are tracked separately from user profiles and allow for the establishment of a many-to-many relationship between users and their interests.
8. **Data Integrity**:
   * Data integrity is paramount, ensuring that when an event is deleted, all associated bookings, tags, and engagements are appropriately deleted or handled.
   * There are no orphan records. If a referenced entity (like an event or user) is deleted, all dependent entities (like tags or engagement activities) must also be deleted or appropriately handled.

### **Extended Notations**

#### 1. **Aggregation**

* **Booking**: The aggregation of User, Event, and Space allows the **Booking** entity to hold attributes such as Booking\_Date and Booking\_Status. This is more efficient as it avoids redundant relationships and allows these attributes to be associated with the entire booking process rather than with individual entities.
  + **Example**: The relationship between a User and an Event is simplified by focusing on the **Booking** entity, which records that a user has booked a space for a particular event. The aggregation also ensures the event and space relationships are grouped, meaning they can be easily accessed or updated as one unit.

#### 2. **Composition**

* **Digital Engagement**: This relationship shows a **Composition** between the User and Engagement activities. The **Digital Engagement** entity represents engagement details (such as posts, comments, or interactions), and the deletion of a user record would automatically delete their associated engagement activities.
  + **Example**: A user’s comment on an event or a blog post within your system would be tightly coupled to that specific user. If the user account is deleted, the comment or interaction would also be deleted, preventing orphan records.
* **Event\_Tag**: This relationship implies that tags do not exist independently but are tied directly to events. The **Tag** entity is part of the **Event\_Tag** composition, reinforcing the idea that tags must always belong to an event.
  + **Example**: You cannot have a tag "Music" unless it is associated with an event, as the tag’s existence is contextually dependent on the event. Deleting the event will cascade to delete the associated tags.

#### 3. **Multi-Valued Attributes**

* **User\_Interest**: This represents a multi-valued attribute, allowing users to have multiple interests. Instead of storing all user interests as a single, delimited field within a user record, the **User\_Interest** table normalizes this into a separate entity.
  + **Example**: A user can have multiple interests, such as Music, Technology, and Sports. The **User\_Interest** table would establish the many-to-many relationship between users and interests, allowing the system to query users by interest or list all users with a particular interest.
* **Event\_Tag**: Each event can be associated with multiple tags. This is represented as a multi-valued attribute, where an event can be tagged as both Networking and Workshop. The system can query and filter events based on their tags.
  + **Example**: When creating an event, the system can allow event organizers to assign multiple tags (e.g., Networking, Technology, Startup). This flexibility ensures that events are classified in ways that improve searchability and categorization.

## 2.2 Normalised Model

The normalization process involved restructuring the database to eliminate anomalies and redundancy while ensuring data integrity by adhering to the principles of database normalization up to the Third Normal Form (3NF). Here's an explanation of the steps taken:

#### **First Normal Form (1NF):**

A table is in **First Normal Form (1NF)** if it meets the following conditions:

1. **Atomicity**: Each column contains atomic (indivisible) values, meaning there are no repeating groups or arrays in a single column.
2. **Uniqueness**: Each record (row) is uniquely identified by a primary key.

**Process for Achieving 1NF:**

* **Atomic Columns**: We ensured that each column contains a single value (no sets or lists within a field). For example, instead of storing multiple phone numbers in a single field, we store each phone number in a separate row.
* **Primary Keys**: Each table contains a primary key (e.g., User\_ID, Event\_ID, Booking\_ID, Space\_ID) that uniquely identifies each record.

For instance:

* In the **User** table, each user has a unique User\_ID.
* In the **Booking** table, each booking is identified by a unique Booking\_ID.

**1NF Example:**

* **Before 1NF**: A column called Phone\_Numbers might store multiple values like "1234567890, 9876543210".
* **After 1NF**: The phone numbers would be split into separate rows, with one phone number per row.

Before 1NF:

|  |  |  |
| --- | --- | --- |
| **User\_ID** | **User\_Name** | **Phone\_Numbers** |
| 101 | Alice | 1234567890, 9876543210 |
| 102 | Bob | 2345678901 |

After 1NF:

|  |  |  |
| --- | --- | --- |
| **User\_ID** | **User\_Name** | **Phone\_Number** |
| 101 | Alice | 1234567890 |
| 101 | Alice | 9876543210 |
| 102 | Bob | 2345678901 |

#### **Second Normal Form (2NF):**

A table is in **Second Normal Form (2NF)** if:

1. It is in **1NF**.
2. All non-key attributes are **fully dependent** on the primary key (no partial dependencies).

**Process for Achieving 2NF:**

* **Removing Partial Dependencies**: If a table has a composite primary key (a primary key consisting of multiple attributes), we remove any non-key attributes that depend on only part of the composite primary key.

For example, in a table like Booking where the composite key might involve both User\_ID and Event\_ID, any attributes like Booking\_Date or Booking\_Status depend on both, not just one part of the key.

* We split the **Booking** table into separate tables to eliminate partial dependencies:
  + A table for Booking will hold only the Booking\_ID, User\_ID, Event\_ID, and Booking\_Status.
  + A separate table for Event will hold event details (Event\_Name, Event\_Date, Event\_Time).

Before 2NF:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Booking\_ID** | **User\_ID** | **Event\_ID** | **Event\_Name** | **Event\_Date** | **Event\_Time** | **Booking\_Status** |
| 1 | 101 | 201 | Tech Conference | 2024-12-15 | 10:00 AM | Confirmed |
| 2 | 102 | 202 | Art Expo | 2024-12-17 | 2:00 PM | Pending |

Here, Event\_Name, Event\_Date, and Event\_Time depend only on Event\_ID, not on the composite key (Booking\_ID + User\_ID).

After 2NF:  
Booking Table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Booking\_ID** | **User\_ID** | **Event\_ID** | **Booking\_Status** |
| 1 | 101 | 201 | Confirmed |
| 2 | 102 | 202 | Pending |

Event Table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Event\_ID** | **Event\_Name** | **Event\_Date** | **Event\_Time** |
| 201 | Tech Conference | 2024-12-15 | 10:00 AM |
| 202 | Art Expo | 2024-12-17 | 2:00 PM |

#### **Third Normal Form (3NF):**

A table is in **Third Normal Form (3NF)** if:

1. It is in **2NF**.
2. It has no **transitive dependencies**, meaning non-key attributes do not depend on other non-key attributes.

**Process for Achieving 3NF:**

* **Removing Transitive Dependencies**: In this step, we ensure that non-key attributes depend only on the primary key and not on other non-key attributes.

For example:

* In the User table, if an attribute User\_Type\_Name (which describes the type of user) exists, it should not be stored directly in the User table. Instead, it should be in the User\_Type table, where User\_Type\_ID is the primary key. The User table then only stores the User\_Type\_ID as a foreign key, linking to the User\_Type table.

Before 3NF:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Event\_ID** | **Event\_Name** | **Event\_Date** | **Event\_Time** | **Organiser\_ID** | **Organiser\_Name** |
| 201 | Tech Conference | 2024-12-15 | 10:00 AM | 301 | TechOrg |
| 202 | Art Expo | 2024-12-17 | 2:00 PM | 302 | ArtWorld |

Here, Organiser\_Name depends on Organiser\_ID, creating a transitive dependency.

After 3NF:

Move Organiser\_Name to separate table

Event Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Event\_ID** | **Event\_Name** | **Event\_Date** | **Event\_Time** | **Organiser\_ID** | **Event\_ID** |
| 201 | Tech Conference | 2024-12-15 | 10:00 AM | 301 | 201 |
| 202 | Art Expo | 2024-12-17 | 2:00 PM | 302 | 202 |

Organiser Table:

|  |  |
| --- | --- |
| **Organiser\_ID** | **Organiser\_Name** |
| 301 | TechOrg |
| 302 | ArtWorld |

### **Normalization Process in Summary:**

**1NF in your database:**

* All tables in the schema are converted to 1NF by ensuring that each field contains atomic values. For example:
  + The User table stores one Phone\_No per row, rather than multiple phone numbers in a single field.
  + The Booking table has a single Booking\_Date, ensuring no repetition of data.

**2NF in your database:**

* After achieving 1NF, we identified and removed partial dependencies. For example:
  + The Booking table, initially with a composite key (User\_ID, Event\_ID), has been split to ensure each non-key attribute (like Booking\_Status) depends fully on the primary key (Booking\_ID), not just part of it.
  + The Event table holds event-related information, while the User table holds user details, avoiding any dependency between non-key attributes.

**3NF in your database:**

* We removed transitive dependencies, ensuring that non-key attributes depend only on the primary key. For example:
  + The User\_Type\_Name attribute is removed from the User table and placed in the User\_Type table, with User\_Type\_ID as the foreign key. This ensures that User table entries are not dependent on non-key attributes like User\_Type\_Name.

### **Final Normalized Model (After 3NF):**

* **User** table: Contains only user-related information like User\_ID, First\_Name, Last\_Name, and User\_Type\_ID as a foreign key to the User\_Type table.
* **Booking** table: Stores booking information with a unique Booking\_ID, and foreign keys linking to User, Event, and Space tables.
* **Event** table: Contains details about events such as Event\_Name, Event\_Date, Event\_Time, and Organiser\_ID as a foreign key linking to the Organiser table.
* **User\_Type** table: Contains User\_Type\_ID, Type\_Name, and other user type-specific details.
* **Organisation** table: Stores organization-related data like Organisation\_Name and Contact\_No.

**Final Normalized Table Model**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Attributes** | **Primary Key (PK)** | **Foreign Key (FK)** |
| **User** | User\_ID, First\_Name, Last\_Name, Email, Gender, Date\_Of\_Birth, Phone\_No, User\_Type\_ID | User\_ID | User\_Type\_ID |
| **User\_Type** | User\_Type\_ID, Type\_Name, Description, Created\_Date, Organisation\_ID | User\_Type\_ID | Organisation\_ID |
| **Organisation** | Organisation\_ID, Organisation\_Name, Contact\_No, Email, Location | Organisation\_ID | None |
| **Interest** | Interest\_ID, Interest\_Name | Interest\_ID | None |
| **User\_Interest** | User\_ID, Interest\_ID, Rank, Date\_Added | User\_ID, Interest\_ID | User\_ID, Interest\_ID |
| **Event** | Event\_ID, Event\_Name, Event\_Type, Event\_Date, Event\_Time, Event\_Duration, Description, Organiser\_ID, Location, Ticket\_Price, Ticket\_Type | Event\_ID | Organiser\_ID |
| **Organiser** | Organiser\_ID, Organiser\_Name, Contact, Organiser\_Type | Organiser\_ID | None |
| **Event\_Tag** | Event\_ID, Tag\_ID | Event\_ID, Tag\_ID | Event\_ID, Tag\_ID |
| **Tag** | Tag\_ID, Tag\_Name, Tag\_Description, Created\_Date | Tag\_ID | None |
| **Space** | Space\_ID, Space\_Name, Space\_Capacity, Area, Location | Space\_ID | None |
| **Booking** | Booking\_ID, User\_ID, Event\_ID, Space\_ID, Booking\_Date, Booking\_Status | Booking\_ID | User\_ID, Event\_ID, Space\_ID |
| **Digital\_Engagement** | Engagement\_ID, Engagement\_Type, Status, Engagement\_Date | Engagement\_ID | None |

## 2.3 Database Schema

This section describes the database schema developed for the project, highlighting the tables, attributes, primary keys, foreign keys, and their respective data types. Each table is aligned with the project’s requirements, ensuring a robust and consistent implementation.

**Tables and Attributes**:

#### 1. **User Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| User\_ID | PK | INT(11) | Unique identifier for users. |
| First\_Name | - | VARCHAR(50) | User's first name. |
| Last\_Name | - | VARCHAR(50) | User's last name. |
| Email | - | VARCHAR(100) | User's email address. |
| Gender | - | VARCHAR(50) | User's gender. |
| Date\_Of\_Birth | - | DATE | User's date of birth. |
| Phone\_No | - | INT(15) | User's phone number. |
| User\_Type\_ID | FK | INT(11) | Links user to their type. |
| User\_Log\_ID | FK | INT(11) | Links user to their log. |

2. User\_Type Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| User\_Type\_ID | PK | INT(11) | Unique identifier for each user type. |
| Type\_Name | - | VARCHAR(50) | Descriptive name of the user type. |
| Description | - | TEXT | Detailed description of the user type. |
| Created\_Date | - | DATETIME | Timestamp when the user type was created. |
| Permission\_Level | - | INT | Access level associated with the user type. |
| IsActive | - | BOOLEAN | Indicates if the user type is active. |
| Organisation\_ID | FK | INT(11) | Links user type to an organization. |

3. Event Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| Event\_ID | PK | INT(11) | Unique identifier for events. |
| Event\_Name | - | VARCHAR(100) | Name of the event. |
| Event\_Type | - | VARCHAR(50) | Type of event. |
| Event\_Date | - | DATE | Date of the event. |
| Event\_Time | - | TIME | Time of the event. |
| Event\_Duration | - | TIME | Duration of the event. |
| Event\_Description | - | TEXT | Detailed description of the event. |
| Location | - | VARCHAR(150) | Location of the event. |
| Organiser\_ID | FK | INT(11) | Links event to an organiser. |
| Ticket\_Price | - | VARCHAR(16) | Price of tickets. |
| Ticket\_Type | - | VARCHAR(50) | Type of ticket. |

4. Organizer Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| Organiser\_ID | PK | INT(11) | Unique identifier for organisers. |
| Organiser\_Name | - | VARCHAR(50) | Name of the organiser. |
| Contact | - | INT(15) | Contact number of the organiser. |
| Type | - | VARCHAR(50) | Type of organiser (e.g., individual, group). |

5. Booking Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| Booking\_ID | PK | INT(11) | Unique identifier for bookings. |
| User\_ID | FK | INT(11) | Links booking to a user. |
| Event\_ID | FK | INT(11) | Links booking to an event. |
| Booking\_Date | - | DATE | Date of the booking. |
| Space\_ID | FK | INT(11) | Links booking to a space. |
| Booking\_Status | - | ENUM("Active","Cancelled") | Status of the booking. |

6. Space Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| Space\_ID | PK | INT(11) | Unique identifier for spaces. |
| Name | - | VARCHAR(50) | Name of the space. |
| Capacity | - | INT(15) | Capacity of the space. |
| Area | - | VARCHAR(50) | Area covered by the space. |
| Location | - | VARCHAR(50) | Location of the space. |

7. User\_Interest Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| User\_Interest\_ID | PK | INT(11) | Unique identifier for user interests. |
| User\_ID | FK | INT(11) | Links interest to a user. |
| Interest\_ID | FK | INT(11) | Links user interest to an interest category. |
| Rank | - | INT(15) | Rank or priority of the interest for the user. |
| Date\_Added | - | DATE | Date the interest was added. |

8. Interest Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| Interest\_ID | PK | INT(11) | Unique identifier for interests. |
| Interest\_Name |  | VARCHAR(50) | Name of the interest. |

9. Event\_Tag

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| Event\_Tag\_ID | PK | INT(11) | Unique identifier for event tags. |
| Event\_ID | FK | INT(11) | Links the tag to an event. |
| Tag\_ID | FK | INT(11) | Links the tag to a tag entity. |
| Date\_Assign | - | DATE | Date the tag was assigned to the event. |

10. Tag

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| Tag\_ID | PK | INT(11) | Unique identifier for tags. |
| Tag\_Name | - | VARCHAR(50) | Name of the tag. |
| Tag\_Description | - | VARCHAR(50) | Description of the tag. |
| Created\_Date | - | DATE | Date the tag was created. |

11. Organization

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| Organisation\_ID | PK | INT(11) | Unique identifier for organisations. |
| Organisation\_Name | - | VARCHAR(50) | Name of the organisation. |
| Department | - | VARCHAR(50) | Organisation's department. |
| Location | - | VARCHAR(150) | Location of the organisation. |
| Contact\_No | - | VARCHAR(15) | Contact number of the organisation. |
| Email | - | VARCHAR(50) | Email address of the organisation. |

12. Digital\_Engagement

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| Engagement\_ID | PK | INT(11) | Unique identifier for each digital engagement. |
| User\_ID | FK | INT(11) | Links engagement to a user. |
| Engagement\_Type |  | VARCHAR(50) | Type of engagement (e.g., webinar, workshop). |
| Status |  | ENUM("Paused","Completed") | Status of engagement. |
| Engagement\_Date |  | DATE | Date of the engagement. |
| Engagement\_Time |  | TIME | Time of the engagement. |

13. User\_Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Key** | **Data Type** | **Description** |
| User\_Log\_ID | PK | INT(11) | Unique identifier for user log entries. |
| Check\_In |  | DATETIME | Timestamp for when the user checked in. |
| Check\_Out |  | DATETIME | Timestamp for when the user checked out. |
| Type |  | VARCHAR(50) | Type of activity logged. |
| Duration |  | INT | Duration of the activity in minutes. |

### **Schema Design Considerations**

1. **Primary Keys**: All tables include a Primary Key to ensure entity integrity.
2. **Foreign Keys**: Foreign Keys are used to establish relationships between tables, maintaining referential integrity.
3. **Data Types**: Attributes are assigned appropriate data types based on their purpose (e.g., int for identifiers, varchar for textual data, date/datetime for temporal data).
4. **Consistency**: The schema is implemented consistently across all tables to facilitate easy integration and querying.

# 3. Mapping

In this section, we map each question from Together Culture’s provided list to the relevant entities in the Extended Entity Relationship Diagram (EERD). This mapping ensures that the database design supports the required analysis and reporting functionalities. The table below presents each question, the related entity or entities, and a rationale, where applicable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Question Category** | **Question** | **Mapped Entities** | **Rationale** |
| **General Usage Patterns** | What are the overall patterns of members’ use of the space over time? | Booking, User, Event, Space | Booking records provide dates and times of visits, linked to Members and Spaces, allowing for trend analysis over time. |
| Which days of the week see the highest member attendance? | Booking, User | Booking\_Date in Booking enables date-based queries, providing weekday trend insights. |
| What times of day are the busiest? | Booking, User | Booking contains timestamps, which allow for time-based filtering and analysis. |
| How do usage patterns vary between weekdays and weekends? | Booking, User | Filtering Booking\_Date by weekdays and weekends offers insights into daily usage patterns. |
| Are there any seasonal variations in space usage? | Booking, User | Queries on Booking\_Date across seasons enable seasonal trend identification. |
| **Individual Member Usage** | When are individual members visiting the space most frequently? | Booking, User | Booking data, filtered by User\_ID, supports member-specific usage frequency analysis. |
| Can we track if a particular member's attendance has changed over time? | Booking, User | User-specific Attendance entries provide historical attendance data for trend analysis over time. |
| Has a member's usage increased significantly, indicating higher engagement? | Booking, User | User and Booking linked by User\_ID allow for comparison of attendance trends with engagement levels, helping track engagement increases. |
| Has a member been absent for an extended period, suggesting possible attrition risk? | Booking, User | Lack of recent records in Attendance for a given User\_ID can indicate inactivity. |
| Can we generate alerts for unusual changes in individual member attendance patterns? | Booking, User | Analyzing Booking records over time supports alerts for deviations in attendance patterns. |
| **Event Participation and Interests** | What types of events are members most interested in (e.g., wellbeing, citizenship, making activities)? | Events, User, Booking | Event\_Type in Events can be analyzed against Booking for insights into popular event types among members. |
| Can we use a system of tags to track areas of interest for members? | User\_Interest, Tag, Event\_Tag | Interests in User and Event\_Type tags in Events facilitate interest tracking for personalized engagement. |
| How many events has each member attended within a specific period? | Booking, User Events | Filtering Booking by User\_ID and date range enables specific member event attendance counts. |
| Which events have the highest attendance rates? | Events, Booking | Booking linked to Event allows for attendance aggregation per event, highlighting popular events. |
| Can we identify trends in event participation among different member segments? | User\_Type, Booking, Events | Member segmentation via Interests in Members and analysis of Attendance by event type support insights into segmented event trends. |
| How do event interests correlate with members' overall engagement and space usage? | User\_Interest, Booking, Event | Track the correlation between event attendance and space usage |
| **Audience Segmentation and Personalization** | Can we segment potential members based on their interests and interactions before joining? | User\_Interest, User | Segmenting based on Interests in Members and supports audience personalization. |
| How effective are our current acquisition strategies based on engagement data? | Digital\_Engagement, User | Tracking and analyzing member acquisition helps evaluate engagement strategies. |
| What is the average volume of digital engagements before converting to membership? | Digital\_Engagement, User | Calculate the average engagement before a user becomes a member |
| Can we track and respond to potential members' interests more effectively? | Digital\_Engagement, User\_Interest | Engagement types and interests allow tailored follow-up and content curation for potential members. |
| How can we personalize communication with members based on their segmented interests? | User\_Interest, User | Interests field in Members supports personalized communication based on segmented interests. |
| **Space Utilization and Capacity Planning** | What is the average capacity utilization of the space at different times? | Booking, Space | Aggregating UsageCount in Spaces and filtering by Booking\_Date and time enables average utilization calculation. |
| How can we optimize workspace allocation to ensure a dynamic and integrated community? | Space, Booking | Utilization data by space allows management to adjust layouts or allocate space as needed to foster integration. |
| Are there specific areas within the space that are underutilized? | Space, Booking | UsageCount for each space provides data on underutilized areas. |
| How can we avoid having segregated areas and promote mixing of different disciplines? | Space, Booking | Analyzing space usage by member interests helps identify opportunities to promote mixed-use spaces. |
| How can we back up our capacity estimates with data for future planning? | Space, Booking | Usage trends from Attendance data support data-backed space capacity planning. |
| **Reporting and Data Integration** | Can we generate detailed reports on student member activity for organizational members like ARU? | User, Booking, Organisation | Reports and data on User and Booking for Members associated with ARU enables member activity reporting. |
| Can we automate data entry processes and report generation to reduce manual efforts? | Booking, Digital\_Engagement | Automating Reports generation from Booking and Digital\_Engagement data reduces manual data handling. |
| How easily can we pull reports from the CRM that provide the necessary insights? | User, Booking, Digital\_Engagement | Reports entity allows for queries combining Booking and Digital\_Engagement data for customized insights. |
| Are there any data integrity issues we need to address during integration? | User, Booking, Digital\_Engagement | Ensuring foreign key constraints and field validations in all entities promotes data integrity and reduces errors. |
| **User Journey and Acquisition** | How can we better track potential members' interactions before they join? | User, Digital\_Engagement | UserEngagement provides detailed data on interactions with potential members before joining. |
| What tools can help us gather useful data on digital engagements? | Digital\_Engagement | Expanding Digital\_EngagementType to include tools and platforms enables detailed tracking of engagement sources. |
| What additional insights can tools like Mouseflow and Hotjar provide to improve user engagement? | Digital\_Engagement | Additional data fields in UserEngagement can be designed to include metrics from tracking tools, offering insights into digital engagement improvements |

# Database implementation (10%)

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

THE TABLES WITH PREFILLED INFORMATION BELOW:

Booking table with prefilled information:

A screenshot of a computer

Description automatically generated

Digital Engagement table with prefilled information

A screenshot of a computer

Description automatically generated

Event table with prefilled information

A screenshot of a computer

Description automatically generated

Event Tag table with prefilled information

A screenshot of a computer

Description automatically generated

Interest table with prefilled information

A screenshot of a computer

Description automatically generated

Organisation table with prefilled information

A screenshot of a computer

Description automatically generated

Organiser table with prefilled information

A screenshot of a computer

Description automatically generated

Tag table with prefilled information

A screenshot of a computer

Description automatically generated

Space table with prefilled information

A screenshot of a computer

Description automatically generated

User table with prefilled information

A screenshot of a computer

Description automatically generated

User Intrest table with prefilled information

A screenshot of a computer

Description automatically generated

User Type table with prefilled information

A screenshot of a computer

Description automatically generated

User Log table with prefilled information

A screenshot of a check out

Description automatically generated

# 5. SQL Queries (50%)

## 5.1 Query 1

1. 5.1.2 Which days of the week see the highest member attendance?

### 

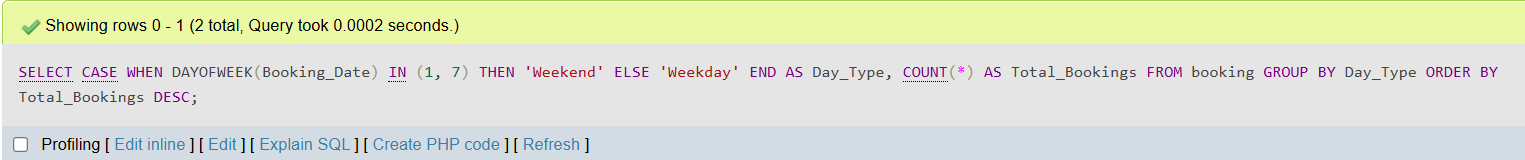
A screenshot of a computer

Description automatically generated

This query will return as output the highest member attendance on each day of the week in descending order, by converting the booking date into name of the day, counting the total number of bookings for each day, grouping that data by day of the week and sorting the data in descending order.

## 5.2 Query 2

1. How do usage patterns vary between weekdays and weekends?

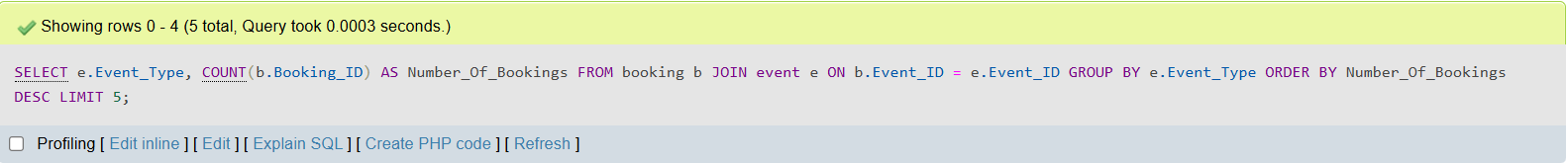
 A close-up of a website

Description automatically generated

With this query we can see the usage patters by extract the days of the week where if 1 is Sunday and 7 Saturday it will meaning it’s a weekend, else its weekdays, count and group the total number of bookings in weekends and weekdays and display result in descending order.

## 5.3 Query 3

1. What types of events are members most interested in (e.g., wellbeing, citizenship, making activities)?

 A screenshot of a computer

Description automatically generated

With this query we can find out in which events members are most intrested in by getting information from Event\_Type, then count the number of booking for each event type, connecting booking to event to get event details, group it by event type in descending order aswell as add limit of 5 to make a top 5 list.

## 5.4 Query 4

1. How many events has each member attended within a specific period?

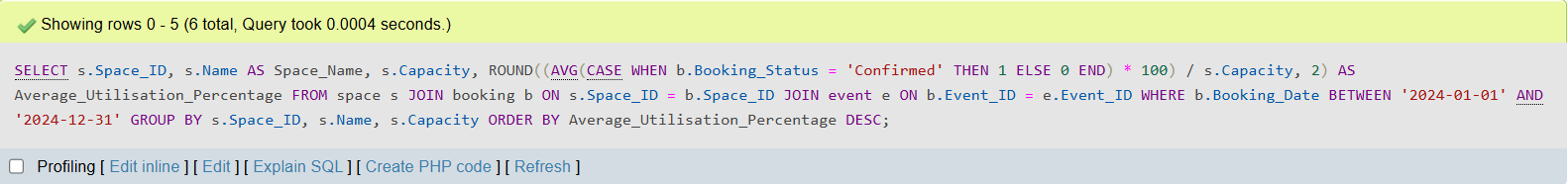


A screenshot of a computer

Description automatically generated

To find how many events each member attended in a specific period, we can do that by first getting the user id and names, counting how many events each user has attended, put a specific date range , filter out only “Confirmed” bookings, group the result by user to get count per user in descending order.

## 5.5 Query 5

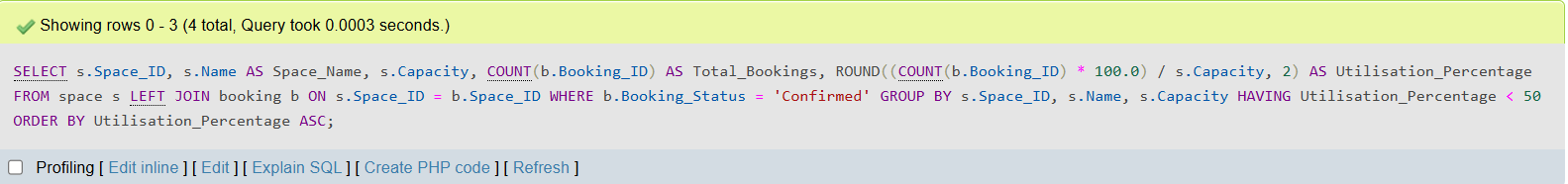
1. What is the average capacity utilisation of the space at different times?  A screenshot of a graph

   Description automatically generated

In order to find the average capacity utilisation we need to calculates the average number of attendees per event for each space and compares it to the space's maximum capacity, to do that we first need to take the space id, name and total capacity, then calculate average number of confirmed bookings, dividing it by total space capacity and converting he result into percentage to the second decimal place, adjusting the desired date range and sorting the result in descending order.

## 5.6 Query 6

1. Are there specific areas within the space that are underutilised?

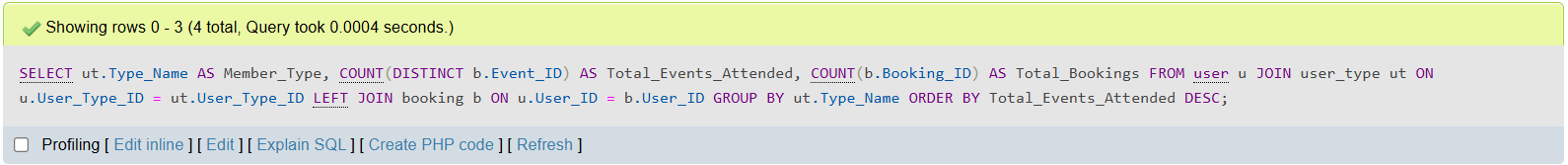
 A screenshot of a number

Description automatically generated

To find out specific spaces that are underutilised we compare the number of bookings for each space to its total capacity, to do that we select space id,name and capacity count the total bookings for that space calculate it convert it into percentages and filter out utilisation percentages threshold(currently 50) to filter out spaces with less than 50% utilisation then order result by ascending order lowest on top.

## 5.7 Query 7

1. Can we identify trends in event participation among different member segments?

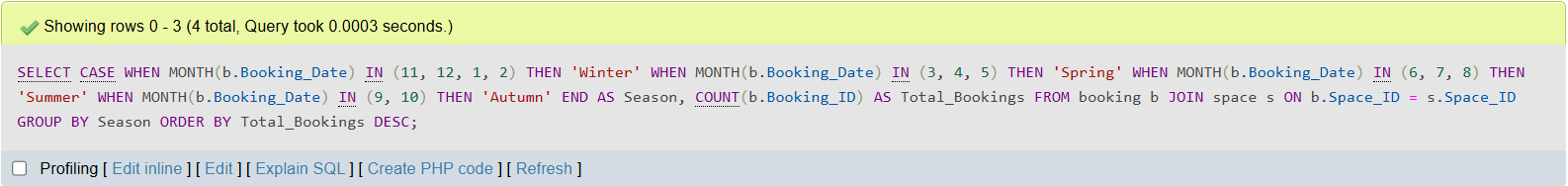
 A screenshot of a web page

Description automatically generated

To identify trends in different member types/segments we need to get information from user type to identify different members, user to provide user details to link to user type and booking containing information about events that are attended by users, then count the total bookings and group result in descending order.

## 5.8 Query 8

1. Are there any seasonal variations in space usage?

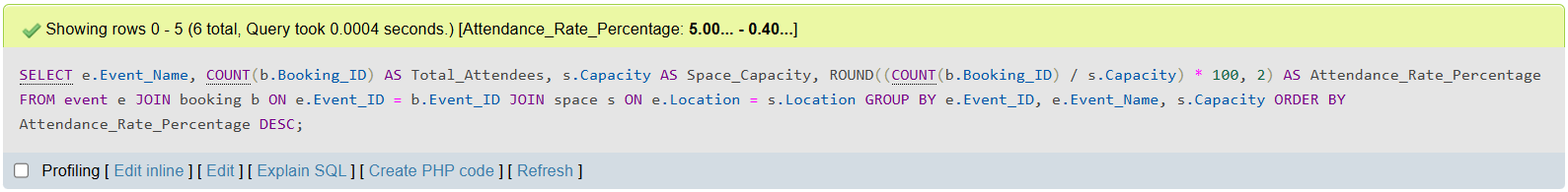
 A screenshot of a computer

Description automatically generated

To find out seasonal trends we first need to determine which months are in what season,aswell as total bookings made in that month and total bookings for that season then we can list that as a final result in descending order.

## 5.9 Query 9

1. Which events have the highest attendance rates?

 A screenshot of a computer

Description automatically generated

To identify events with the highest attendance rate, we have to calculate the attendance percentage for each event by comparing the number of attendees total bookings to the capacity of event convert into percentage and order by in descending order.

## 6 Personal Query 10

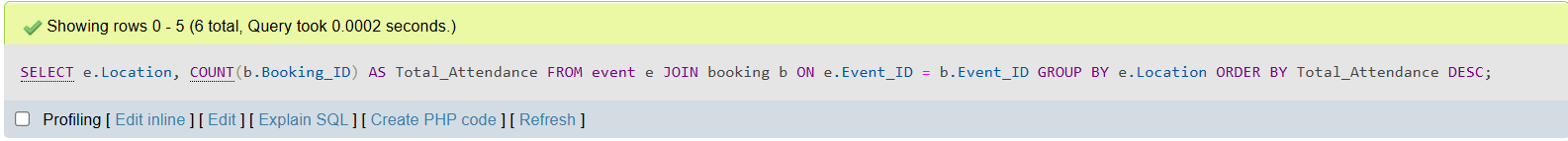
1. For what purpose will this query be used?

This query will provide a general overview of how space/city location affects event attendance by calculating the total attendance for each location.

2.Query in natural language:

Does event location/city impact event attendance and whats the average attendance per location/city?

3.SQL Code and output:

 A screenshot of a computer

Description automatically generated

4.Explain the output of the data (was this what was predicted?)

This query will provide a breakdown of the total number of attendees for each location, by taking a location counting total attendance from event in that location printing out result in descending order

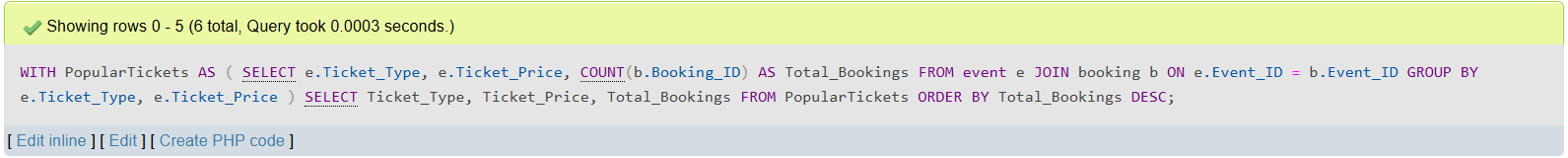
## 6.1. Personal Query 11

1.For what purpose will this query be used?

This query will identify the ticket type and price combination with the highest number of bookings.

2.Query in natural language:

What are the most booked ticket types along with ticket prices?

3.SQL Code and output: A screen shot of a ticket

Description automatically generated

4.Explain the output of the data (was this what was predicted?)

Produces a report of the most booked ticket type, price and bookings, by taking ticket type and ticket price counting the bookings of each ticket type and printing out result by descending order.

# 6.2. References

* Connolly, T. & Begg, C., 2019. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 6th ed. Harlow: Pearson Education.
* Elmasri, R. & Navathe, S.B., 2021. *Fundamentals of Database Systems*. 7th ed. New York: Pearson.
* Ramakrishnan, R. & Gehrke, J., 2020. *Database Management Systems*. 3rd ed. New York: McGraw-Hill Education.
* Silberschatz, A., Korth, H.F. & Sudarshan, S., 2019. *Database System Concepts*. 7th ed. New York: McGraw-Hill Education.