SAW Education Database Design

Thanh Nam Vu (ID: 104991276)
Swinburne University of Technology
Tutorial time: Thursday 4:30pm
Date of submission: 22 October 2024

Contents

Abstract

This report presents the design and implementation of a relational database for SAW Education, a consultancy specializing in study abroad and immigration services. The project addresses key inefficiencies in the company's previous data management system, which hindered its ability to generate insights and make data-driven decisions. A new database structure was developed using MySQL, with clearly defined entities and relationships represented through an Entity-Relationship Diagram (ERD). The design ensures seamless data interactions across departments, enhancing operational efficiency and maintaining data integrity.

The report covers the use of primary and foreign keys to enforce referential integrity, as well as normalization techniques to prevent redundancy. Several SQL queries are employed to provide business insights, such as tracking event effectiveness, client conversions, and profitability. The improved database design supports the company's strategic decision-making, enabling it to monitor client journeys from event participation to case enrollment. In conclusion, this new database framework lays a robust foundation for SAW Education's operational optimization and future growth through enhanced data analysis capabilities.

I. Introduction

In this report, I present the design of a relational database for SAW Education, a study abroad and immigration consultancy. Throughout my time working for the company, I identified several challenges in the way data was stored and managed. The previous database lacked a well-structured design, leading to inefficiencies in data interactions between entities. As a result, data analysts struggled to retrieve clean and analyze data efficiently, impacting the company's ability to generate insights, identify trends, and make data-driven decisions. This inefficiency prevented SAW Education from fully capitalizing on business opportunities and improving operational outcomes.

To address these issues, I developed a new database structure using MySQL. This process involved designing a schema with clearly defined relationships, ensuring seamless interactions between entities. I also added sample data to test the functionality and validate the design. Using an Entity-Relationship Diagram (ERD), I ensured the relationships between the entities were clear, logical, and free from redundancy or collisions. Lastly, I executed several SQL queries to generate insights from the database, supporting strategic decision-making and helping the company improve profitability. The purpose of this new design is to enable smooth data management across all departments, optimize operations, and provide accurate data to track client journeys from initial engagement at events to becoming paying customers. This report will demonstrate how the new database structure solves the company's previous issues and provides a robust foundation for analysis.

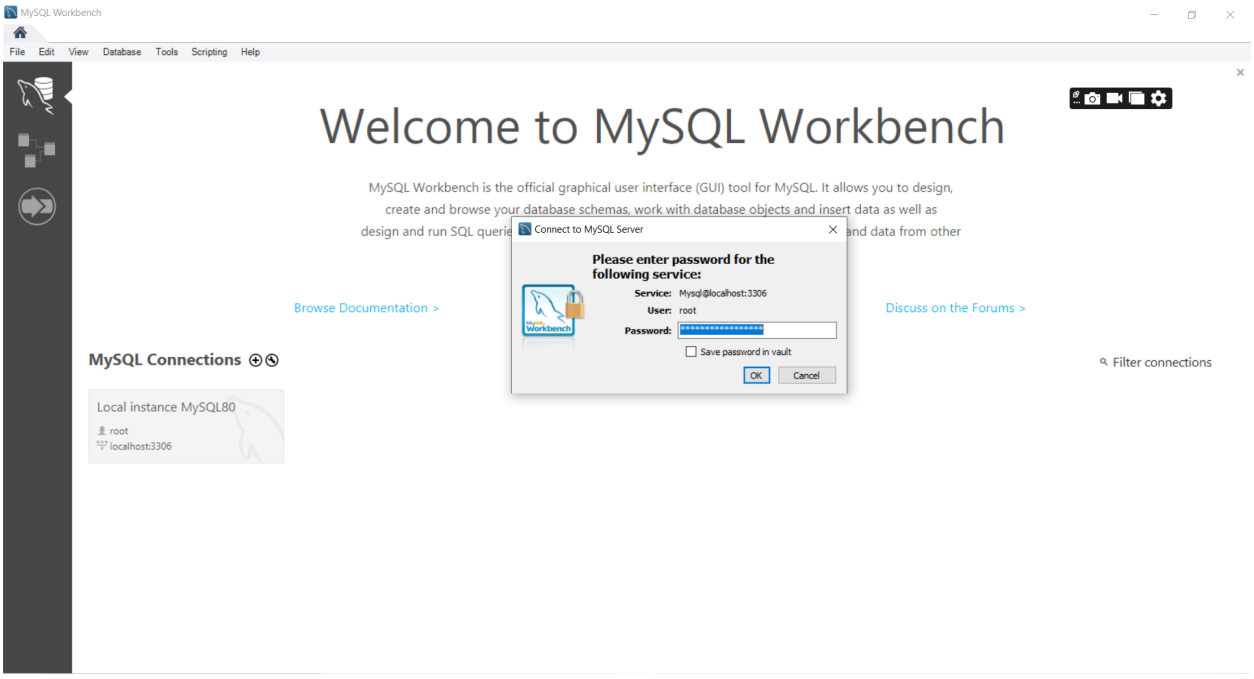

Fig. 1: Local host in MySQL Workbench

II. How the Company Works

SAW Education offers various services to help students with study abroad and immigration processes. The company regularly organizes events such as:

- Trivia Night
- Gradutrack
- VCE
- Welcome Party

These events are critical for generating leads and attracting potential customers. Attendees who engage with the company at these events have the option to book appointments with counselors. If an attendee schedules an appointment, they become a client. However, not every attendee follows this path—some may decide to explore services with competing agencies or choose not to proceed due to a lack of trust or readiness.

Clients who proceed further become cases when they commit to enrolling in a university or immigration process through SAW Education. At this stage, the company generates profits through commission fees from successful university placements or immigration services.

Managing this process requires careful data tracking and analysis at each step: from tracking event attendance to monitoring client engagements and university enrollments. This data must be stored securely and systematically to allow data analysts to retrieve and analyze key metrics. Insights derived from this data help the company evaluate event performance, track client progress, and optimize marketing strategies.

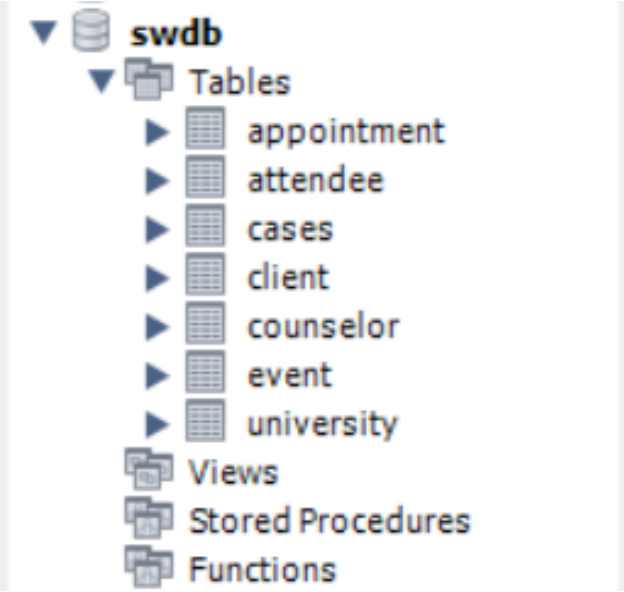

Fig. 2: All 7 Entities

III. Creating a Database and Designing the ERD Diagram

The database consists of seven key entities:

- Event
- Attendee
- Client
- Appointment
- Counselor
- Cases
- University

IV. Designing the ERD (Entity-Relationship Diagram)

The Entity-Relationship Diagram (ERD) for the SAW Education database captures the relationships and structure of the company's data. The diagram provides a detailed view of how the core entities interact with each other to reflect business processes accurately. This ERD ensures that the data flow remains efficient, with minimal redundancy and optimized interactions between the entities. Each entity represents a critical part of the business, including clients, attendees, appointments, cases, universities, events, and counselors.
Let's break down each entity, their attributes, and relationships in detail:

V. Table Design: Primary and Foreign Keys

The first step in the design process was defining primary keys (PK) for each entity to ensure every record was uniquely identifiable. Some examples include:

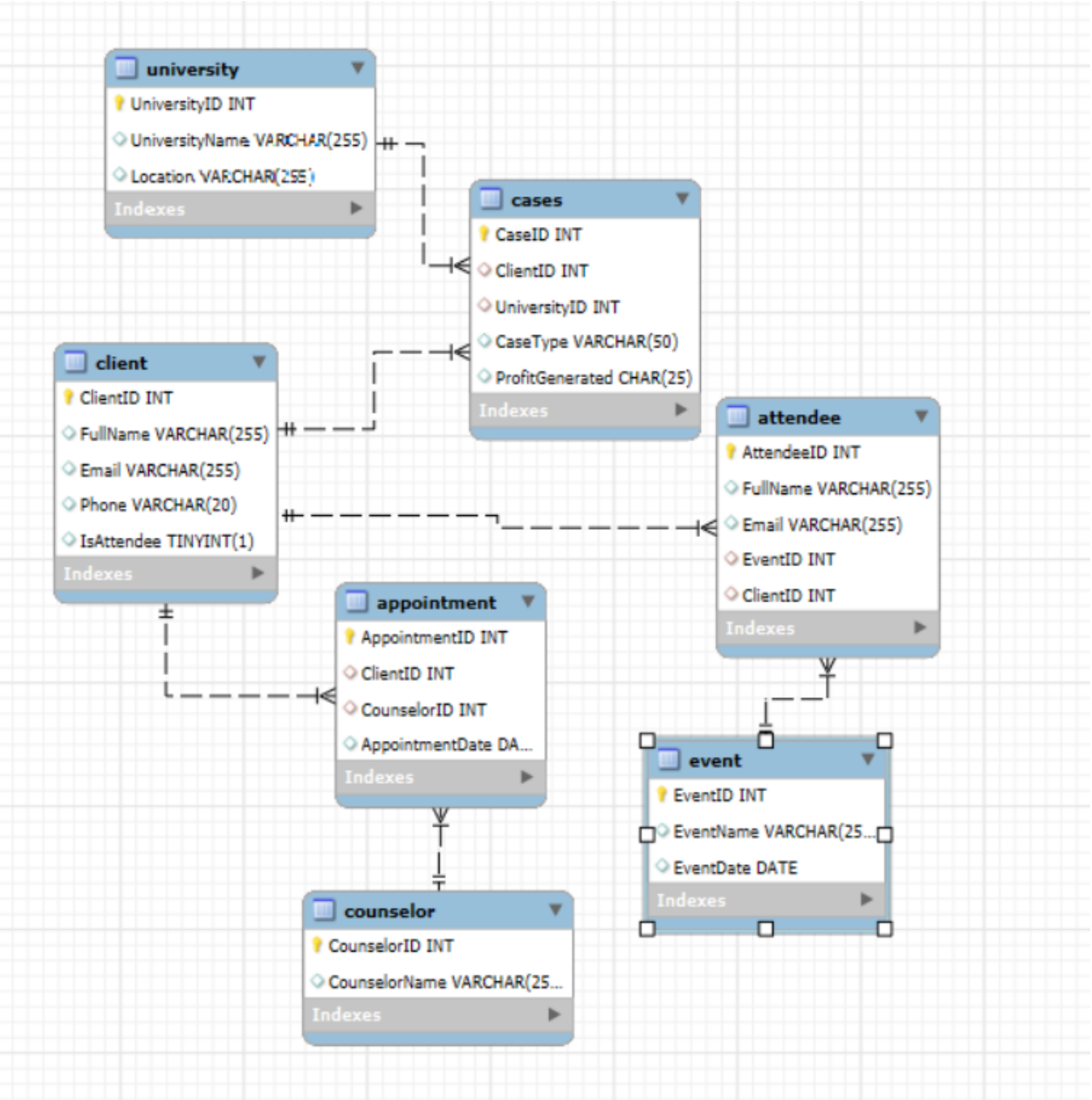

Fig. 3: ERD Diagram

- **Event**: Uses `EventID` as the primary key to uniquely identify each event.
- **Client**: Uses `ClientID` as the primary key to track every individual client.
- **Appointment**: Uses `AppointmentID` to ensure each appointment is distinct.

After defining the primary keys, I established foreign keys (FK) to maintain relationships between tables. These foreign keys enforce referential integrity, ensuring that data is consistent across the database. For example:

- The **Cases** table includes `ClientID` and `UniversityID` as foreign keys, linking each

case to a specific client and university.

- The **Appointment** table contains `ClientID` and `CounselorID` as foreign keys, reflecting the interaction between clients and counselors.

Using foreign keys allows the system to prevent inconsistencies, such as an appointment being recorded without an associated client. The foreign key constraints also make it easier to track relationships between entities and enforce data consistency.

VI. Ensuring Data Integrity and Optimization

The ERD design ensures data integrity through the proper use of primary keys and foreign keys. Foreign keys enforce referential integrity, ensuring that records in related tables are consistent. For example:

- A case cannot exist without an associated client, as enforced by the foreign key `ClientID` in the **Cases** table.
- Similarly, an appointment cannot exist without a corresponding client and counselor.

Indexes were also used on key columns (such as `ClientID` and `UniversityID`) to enhance query performance. This ensures that even as the database grows, queries remain efficient and responsive.

The normalization of the database design—ensuring that data is stored logically and without redundancy—also helps maintain consistency across the system. For instance, universities are stored in a separate **University** table to avoid duplicate entries in the **Cases** table. This not only ensures data consistency but also simplifies updates and management.

VII. Entities and Their Attributes

University

- **Primary Key**: `UniversityID` (INT) – uniquely identifies each university.
- **Attributes**:
  - `UniversityName` (VARCHAR)
  - `Location` (VARCHAR)
- **Purpose**: Captures information about universities that clients are applying to, providing insight into popular study destinations.

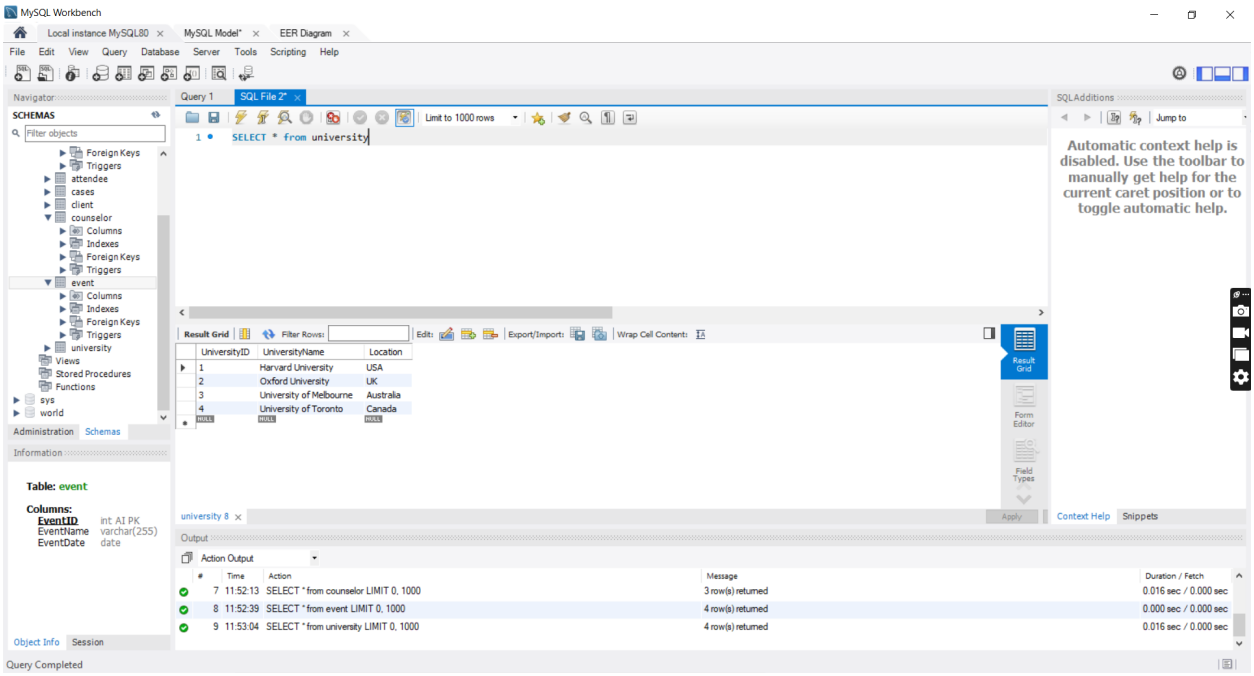

Fig. 4: University table

Cases

- **Primary Key**: `CaseID` (INT) – uniquely identifies each case.
- **Foreign Keys**:
  - `ClientID` – links each case to a specific client.
  - `UniversityID` – links the case to the university the client has selected.
- **Attributes**:
  - `CaseType` (VARCHAR) – the type of service (study abroad or immigration)
  - `ProfitGenerated` (CHAR) – records the commission earned for each case.
- **Purpose**: Tracks individual client cases as they progress and generates revenue data based on successful placements.

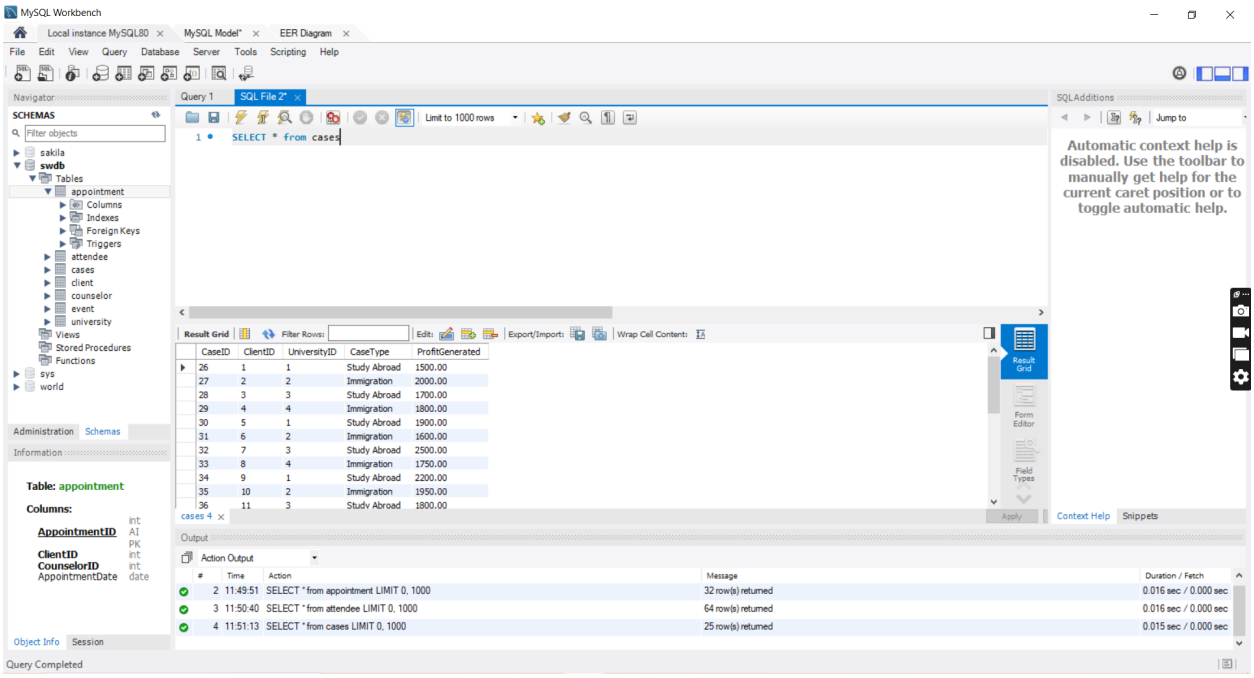

Fig. 5: Cases table

Client

- **Primary Key**: `ClientID` (INT) – uniquely identifies each client.
- **Attributes**:
  - `FullName` (VARCHAR)
  - `Email` (VARCHAR)
  - `Phone` (VARCHAR)
  - `IsAttendee` (TINYINT) – records whether the client started as an event attendee.
- **Purpose**: Manages client information and tracks whether they started as attendees. This table is critical in tracing the client journey from the first point of contact to becoming a paying case.

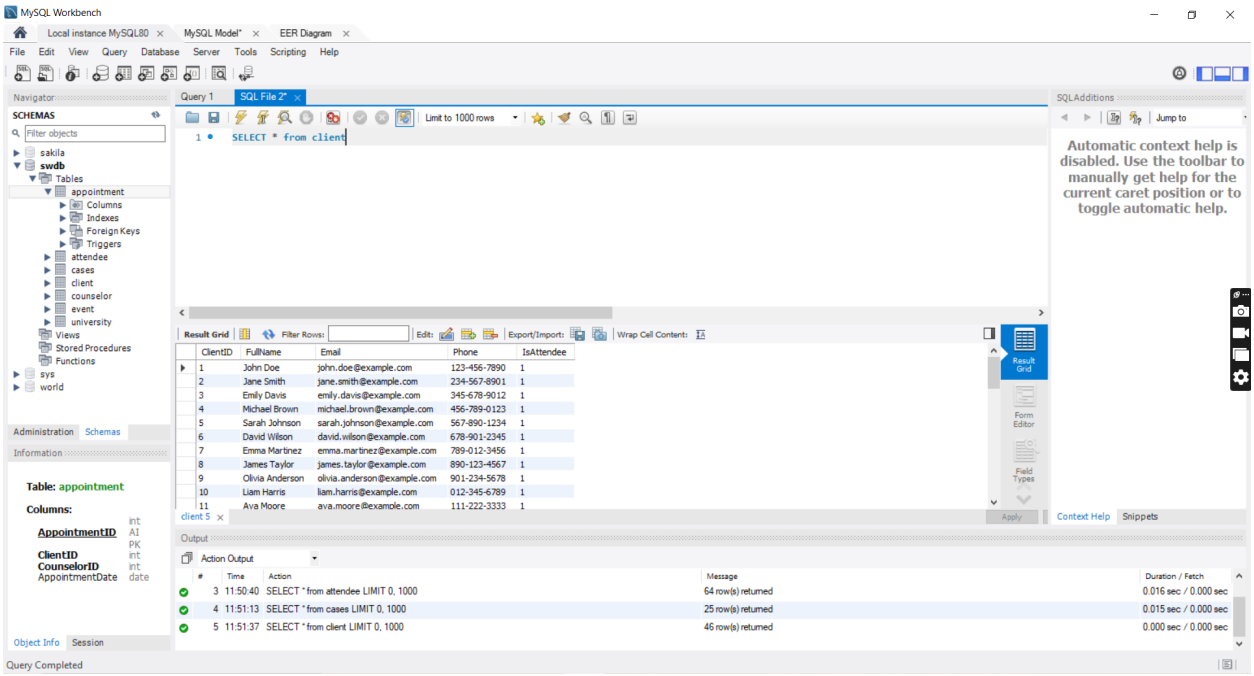

Fig. 6: Client table

Appointment

- **Primary Key**: `AppointmentID` (INT) – uniquely identifies each appointment.
- **Foreign Keys**:
  - `ClientID` – links the appointment to a client.
  - `CounselorID` – identifies the counselor involved in the appointment.
- **Attributes**:
  - `AppointmentDate` (DATE)
- **Purpose**: Stores appointment details between clients and counselors. It helps manage and distribute counselor workloads effectively.

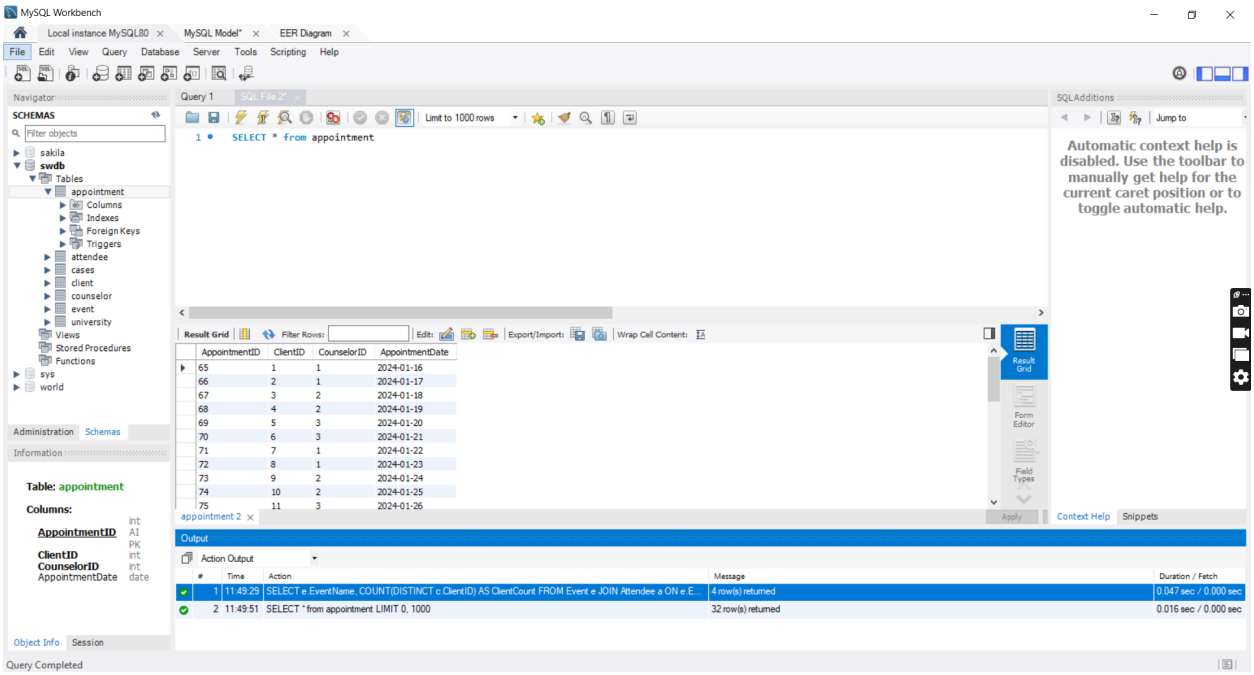

Fig. 7: Appointment table

Counselor

- **Primary Key**: `CounselorID` (INT) – uniquely identifies each counselor.
- **Attributes**:
  - `CounselorName` (VARCHAR)
- **Purpose**: Captures counselor details and helps manage appointments. This entity allows the company to assess each counselor's workload and performance.

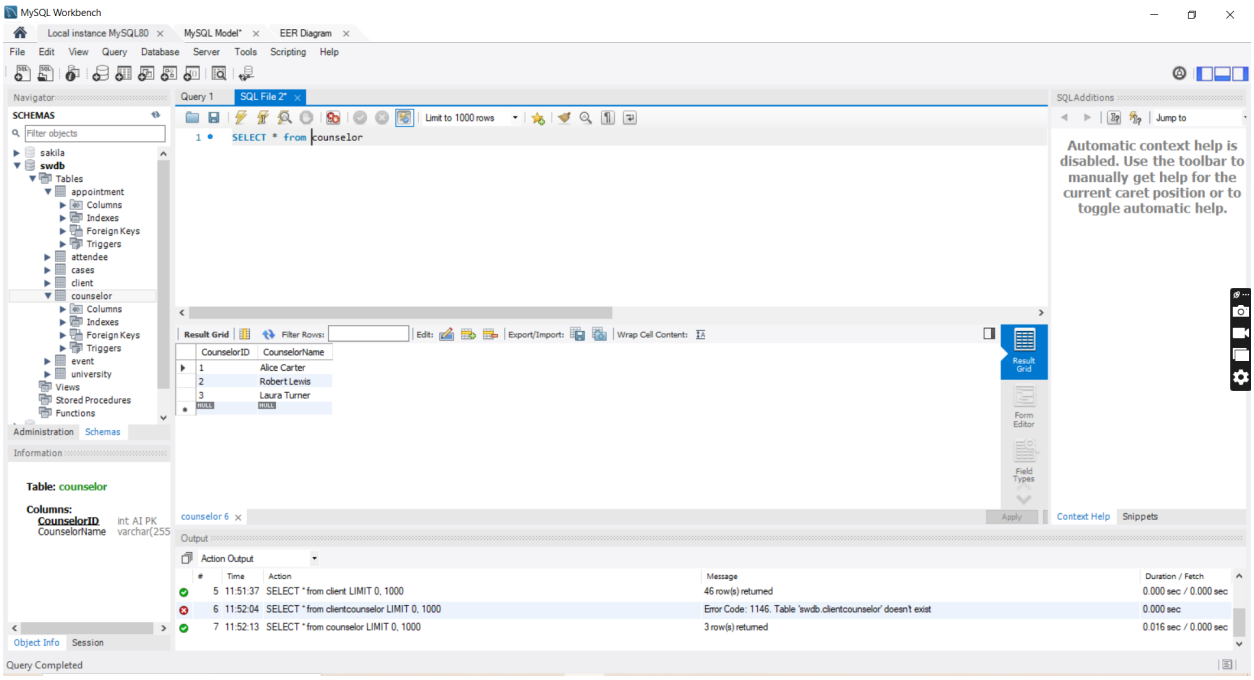

Fig. 8: Counselor table

Event

- **Primary Key**: `EventID` (INT) – uniquely identifies each event.
- **Attributes**:
  - `EventName` (VARCHAR)
  - `EventDate` (DATE)
- **Purpose**: Tracks company events held to attract potential customers. Each event's performance can be evaluated based on how many attendees convert into clients.

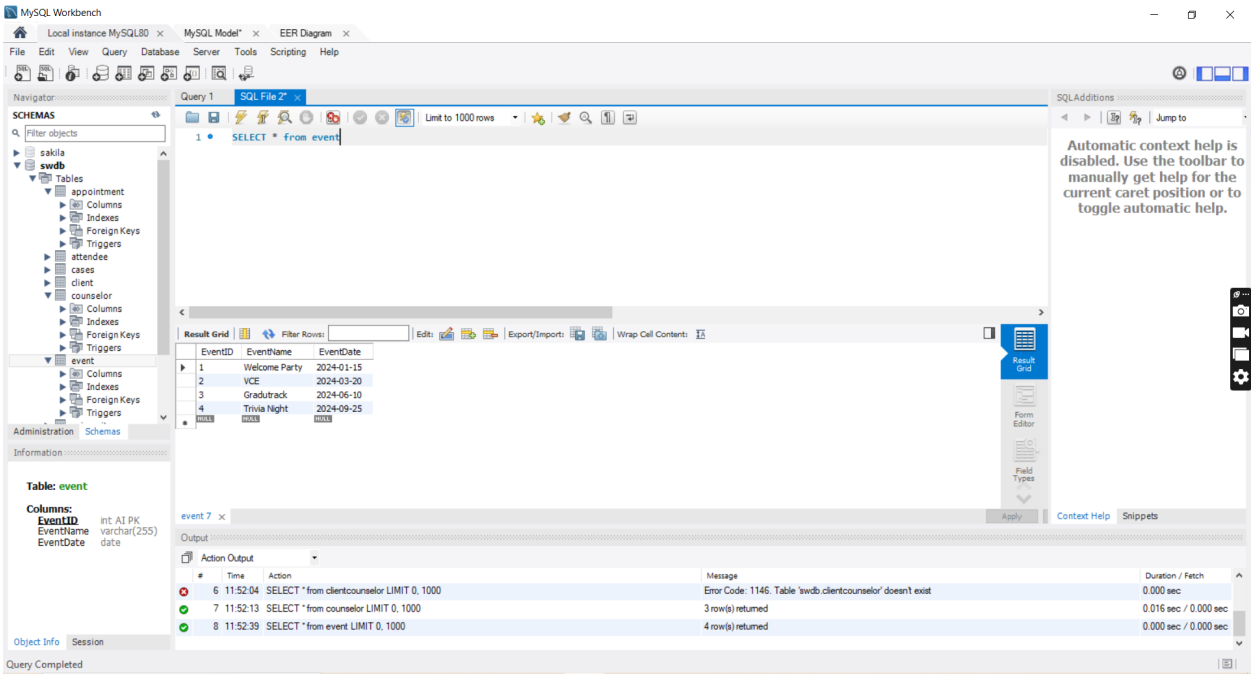

Fig. 9: Event table

Attendee

- **Primary Key**: `AttendeeID` (INT) – uniquely identifies each event attendee.
- **Foreign Keys**:
  - `EventID` – links each attendee to the event they attended.
  - `ClientID` – links the attendee to the corresponding client, if they transition to become one.
- **Attributes**:
  - `FullName` (VARCHAR)
  - `Email` (VARCHAR)
- **Purpose**: Records attendees and tracks whether they convert into paying clients. This information is critical for assessing the effectiveness of events.

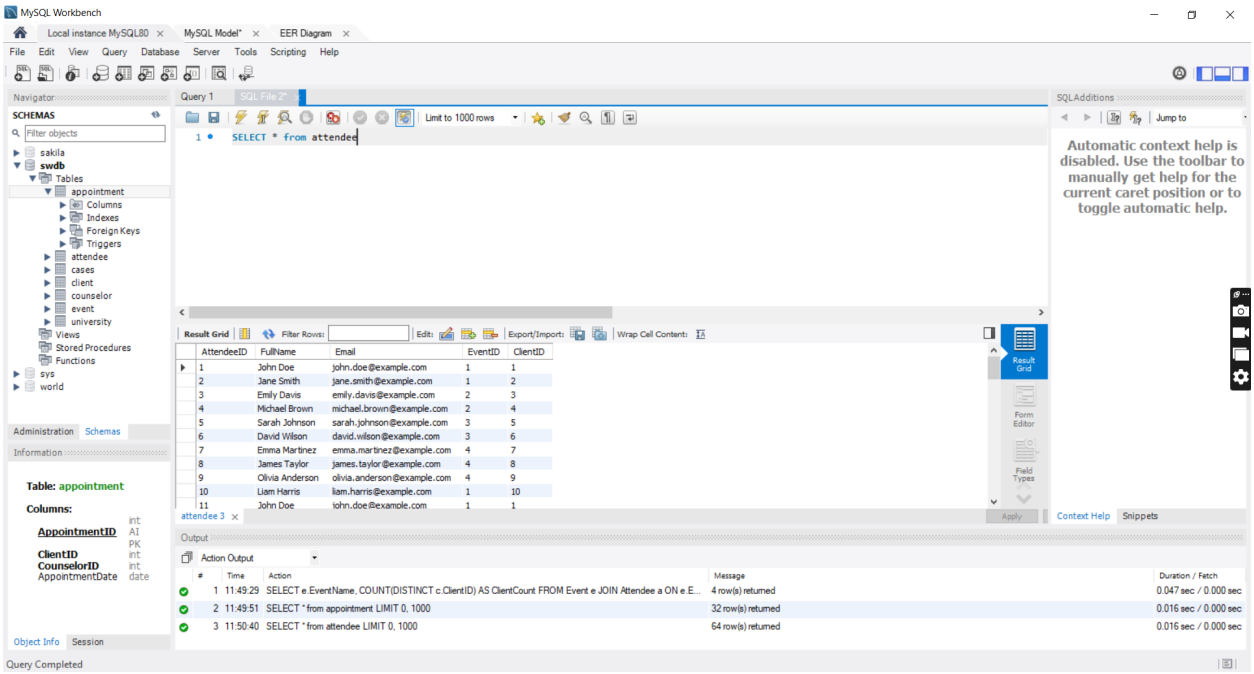

Fig. 10: Attendee table

VIII. Entity Relationships and Cardinalities

The relationships between entities in the SAW Education database were carefully modeled to reflect real-world interactions. Below is a breakdown of the key relationships, including their cardinalities:

- **Event – Attendee (One-to-Many)**: Each event can attract multiple attendees, but each attendee is associated with only one event. This helps the company track attendance trends and assess which events generate the most engagement.
- **Attendee – Client (One-to-One / One-to-Many)**: Not all attendees become clients, but those who do must book an appointment with a counselor. This relationship ensures that only engaged attendees are promoted to the client status.
- **Client – Appointment (One-to-Many)**: A client can have multiple appointments with different counselors. The foreign keys in the `Appointment` table ensure that each appointment is correctly linked to both the client and the counselor.
- **Client – Cases (One-to-One)**: Each client who completes the enrollment process becomes a case, and each case corresponds to a single client. This relationship ensures data clarity and prevents duplication.
- **Cases – University (Many-to-One)**: Multiple cases can be linked to the same university, reflecting that several clients may choose the same institution for their studies. This relationship allows the company to analyze trends in university selections.

The carefully designed cardinalities ensure data integrity, prevent redundancy, and allow the company to track the entire customer journey seamlessly—from event attendance to becoming a paying case.

IX. Writing SQL Queries for Business Insights

The purpose of this database goes beyond data storage—it enables the company to generate insights and make informed business decisions. Here are some key SQL queries and their business value:

1) **Which event brings the most clients?**

```
SELECT e.EventName, COUNT(DISTINCT c.ClientID) AS ClientCount
FROM Event e
JOIN Attendee a ON e.EventID = a.EventID
JOIN Client c ON a.AttendeeID = c.ClientID
GROUP BY e.EventName
ORDER BY ClientCount DESC;
```

Insight: This query identifies which event attracts the most clients, allowing the company to focus on high-performing events and allocate resources more effectively.

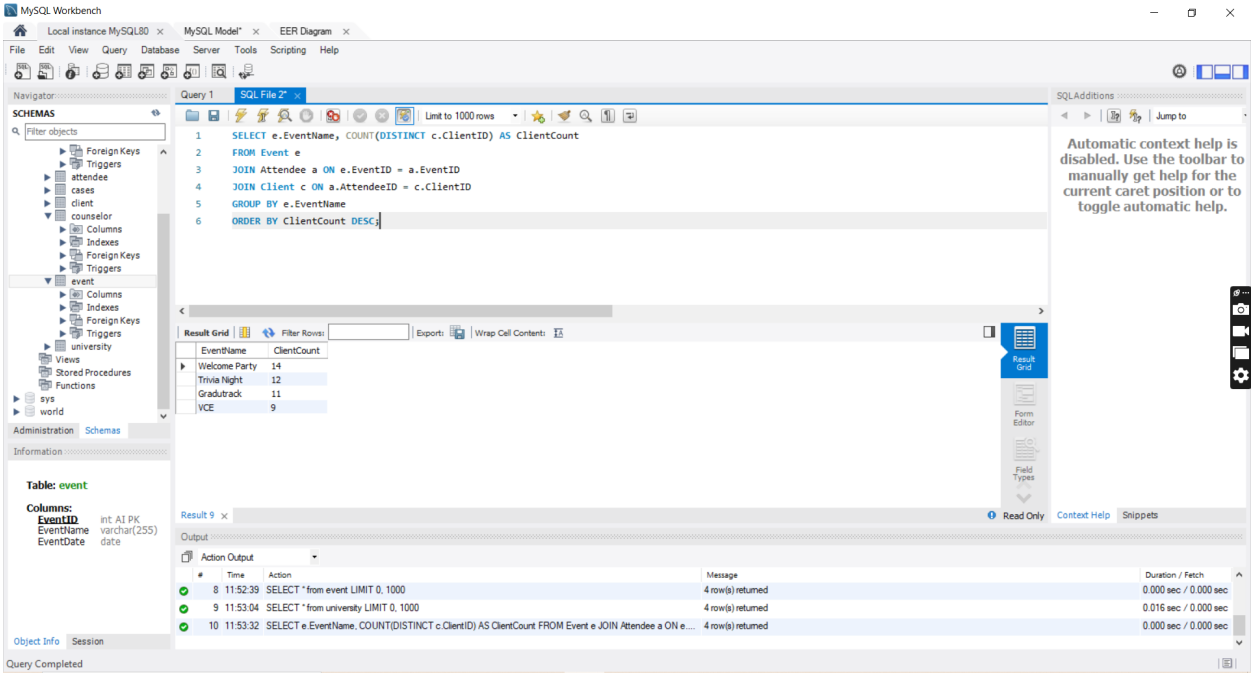

Fig. 11: SQL code illustration

2) **How many clients from each event proceed to cases?**

```
SELECT e.EventName, COUNT(DISTINCT cs.CaseID) AS CaseCount
```

```
FROM Event e
JOIN Attendee a ON e.EventID = a.EventID
JOIN Client c ON a.AttendeeID = c.ClientID
JOIN Cases cs ON c.ClientID = cs.ClientID
GROUP BY e.EventName
ORDER BY CaseCount DESC;
```

Insight: This query helps the company assess which events convert the most clients into cases, guiding future event strategies.

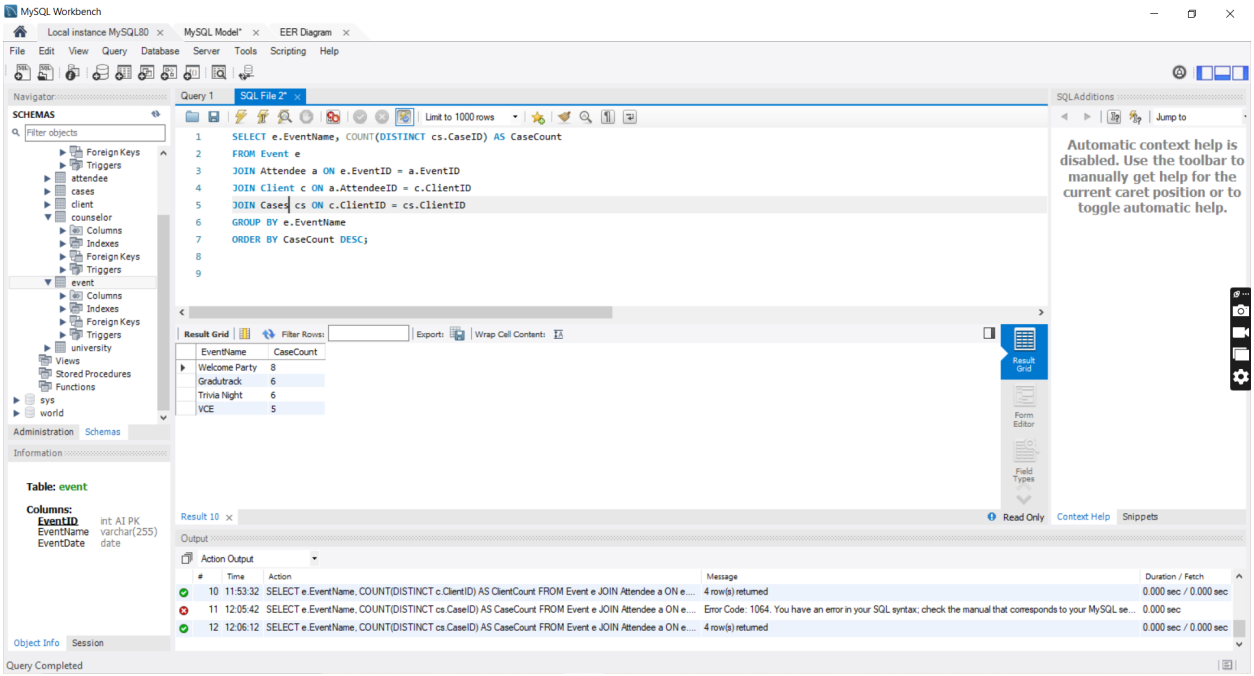

Fig. 12: SQL code illustration

3) **Total profit generated per event:**

```
SELECT e.EventName, SUM(cs.ProfitGenerated) AS TotalProfit
FROM Event e
JOIN Attendee a ON e.EventID = a.EventID
JOIN Client c ON a.AttendeeID = c.ClientID
JOIN Cases cs ON c.ClientID = cs.ClientID
GROUP BY e.EventName;
```

Insight: By identifying the most profitable events, the company can prioritize those that yield the highest returns.

4) **Number of appointments handled by each counselor:**

```
SELECT co.CounselorName, COUNT(ap.AppointmentID) AS AppointmentCount
FROM Counselor co
```

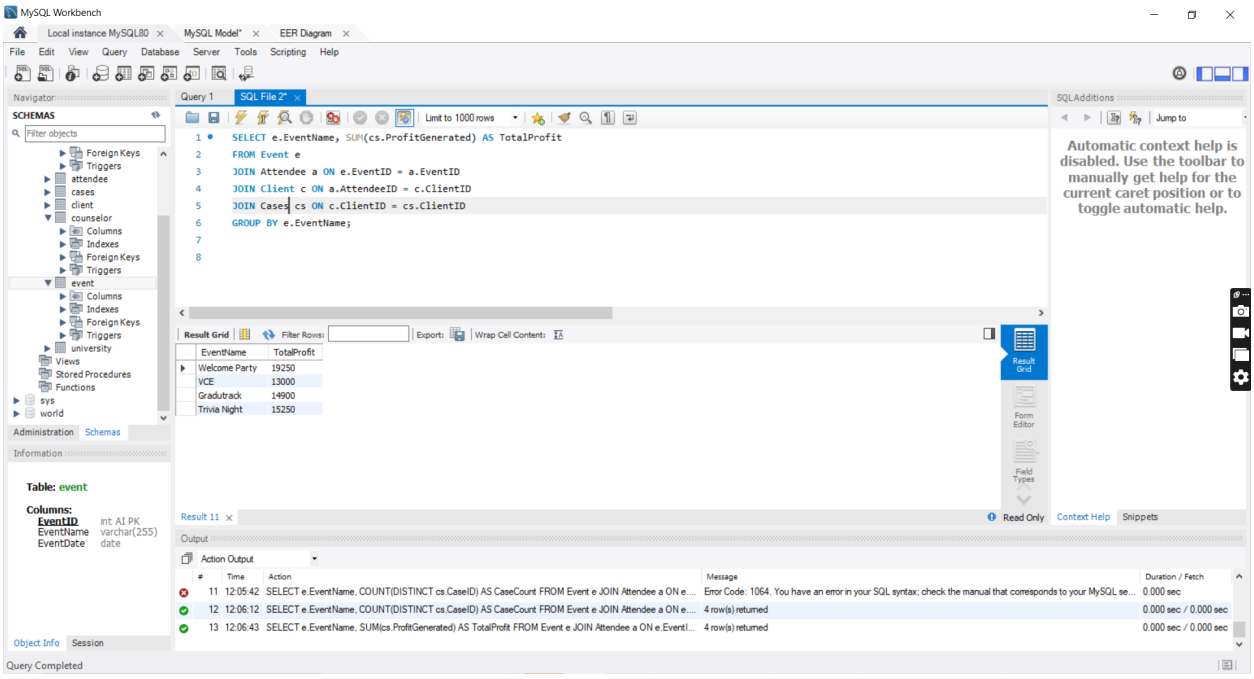

Fig. 13: SQL code illustration

```
JOIN Appointment ap ON co.CounselorID = ap.CounselorID
GROUP BY co.CounselorName
ORDER BY AppointmentCount DESC;
```

Insight: This query ensures fair workload distribution among counselors and identifies top performers.

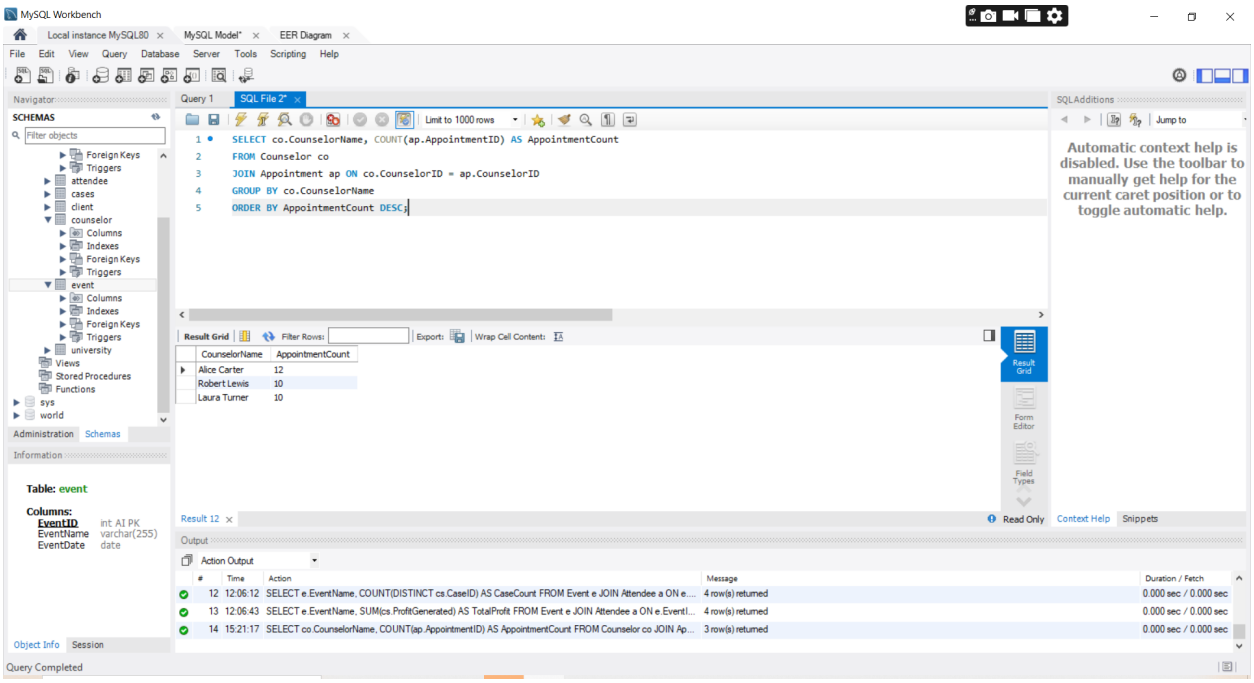

Fig. 14: SQL code illustration

5) **Which university was chosen the most?**

```
SELECT u.UniversityName, COUNT(DISTINCT CaseID) AS CasesCount
FROM University u
JOIN Cases cs ON u.UniversityID = cs.UniversityID
GROUP BY u.UniversityName
ORDER BY CasesCount DESC;
```

Insight: Knowing the most popular universities helps the company target key partners and refine marketing efforts.

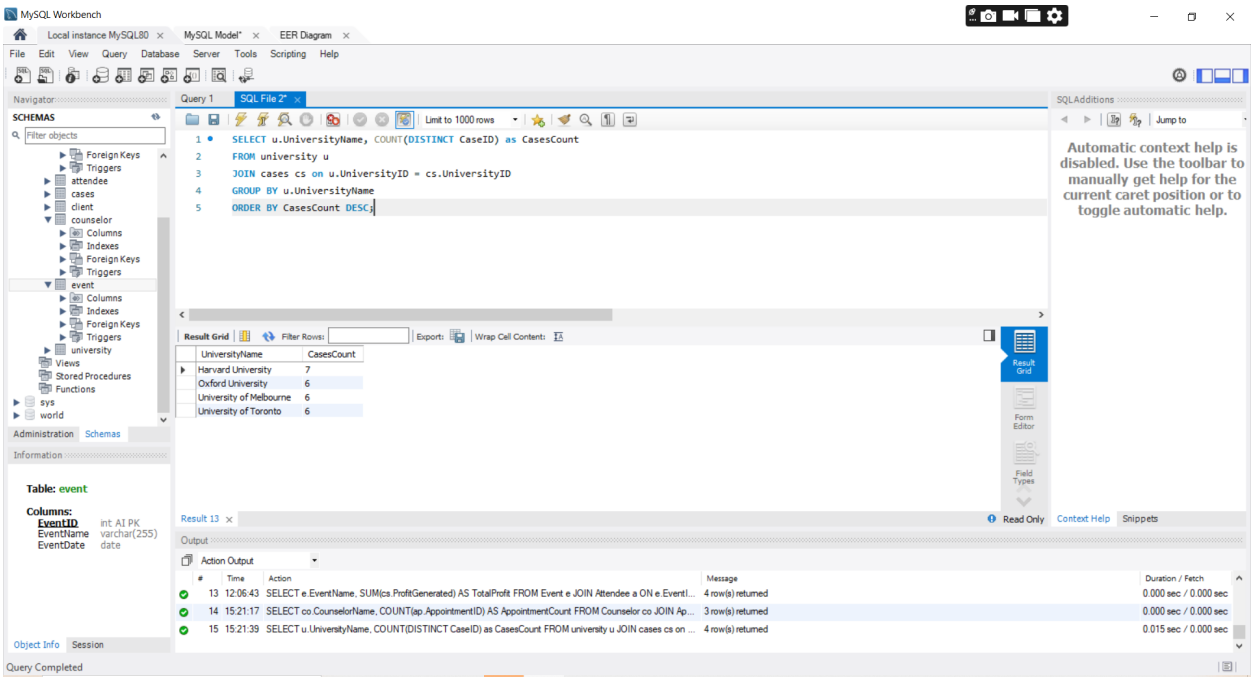

Fig. 15: SQL code illustration

X. Conclusion

The SAW Education database was designed to address the company's previous data management challenges and optimize operations. By defining primary keys, foreign keys, and clear entity relationships, the design ensures data integrity and consistency. The use of SQL queries provides valuable business insights, helping the company identify trends, improve event strategies, and maximize profitability.

In summary, this database offers a robust foundation for data-driven decision-making. With improved data access and analysis, the company is better equipped to make strategic decisions that enhance operational efficiency and profitability.