CIS 344 Final Report

Project: Online Career Coaching Database (coaching_db)
Student: Michael Konadu
Course: CIS 344
Semester: Spring 2026
Date: 2/23/2026

1. Project Scope

This project creates a virtual career coaching business using a database built in MySQL Workbench, which has been implemented in MySQL. The database will support the basic business functions of the coaching platform: creating and maintaining client and coach profiles, defining services offered by coaches, scheduling sessions, providing session packages, tracking payments made by clients, and storing reviews for coaches. The features of this application have been designed at a realistic small-business level to illustrate the correct use of entities, keys, constraints, and relationships.When a client books a session with a coach for a given service, the client may pay for just that session or purchase services through a package, which allows them to book multiple coaching sessions at once. The application will track all payments, and clients can leave coach reviews with an optional link to the associated session.

2. Database Design Summary

The final design includes nine tables:

- Client: stores client identity and contact information.
- Coach: stores coach identity, contact information, bio, and hourly rate.
- Service: stores available service types and base prices.
- Session: stores scheduled coaching sessions, tied to one client, one coach, and one service.
- Package: stores package offerings (bundle pricing and sessions included).
- Client_Package: records package purchases per client and tracks remaining sessions and status.

- Coach_Service: connects coaches to the services they offer and supports coach-specific pricing.
- Payment: stores payment records, including amount, method, status, and date; optionally links to a session or a package purchase.
- Review: stores client reviews for coaches; optionally links to a session.

Key constraints and design features include:

- Primary keys on all entity tables; the Coach_Service table uses a composite primary key (coach_id, service_id).
- Foreign keys enforce relationships (for example, Session references Client, Coach, and Service).
- Unique constraints ensure uniqueness for key identifiers such as Client.email, Coach.email, and Service.service_name.
- Optional relationships were modeled using nullable foreign keys where appropriate (for example, Payment may reference a Session or a Client_Package purchase).

3. Implementation Details

The database was created using MySQL Workbench EER modeling tools and then implemented using Forward Engineering. The deliverables produced include:

- UML/EER diagram export from MySQL Workbench
- Chen-style ER diagram (hand-drawn)
- SQL script generated from Forward Engineering (coaching_db.sql)
- Workbench model file (.mwb)

The final SQL script creates the schema, tables, indexes, and foreign key constraints, and the schema was successfully generated without errors during execution.

4. Challenges Encountered and How They Were Resolved

1. Modeling many-to-many relationships correctly

    a. Challenge: A coach can offer many services, and a service can be offered by many coaches. Also, a client can purchase many packages over time, and a package can be purchased by many clients.
    b. Solution: I created the Coach_Service and Client_Package tables to resolve these many-to-many relationships and store relationship-specific attributes, such as price_override for Coach_Service and purchase details for Client_Package.
2. Keeping the model, SQL script, and diagram exports consistent
    a. Challenge: Small schema changes (such as enabling AUTO_INCREMENT or correcting a column name) can cause the exported diagram and the generated SQL script to become inconsistent if files are not regenerated.
    b. Solution: After finalizing the schema, I re-exported the UML/EER diagram and regenerated the SQL script so that the Workbench model, diagram image, and SQL file matched.
3. Recording payments for different purchase types
    a. Challenge: Payments can apply to either an individual session or a package purchase.
    b. Solution: I designed the Payment table with a required client_id and optional references to session_id and client_package_id. This supports both cases while maintaining referential integrity.

5. Outcomes and Results

The project successfully met the required objectives:

- A complete relational schema was designed with appropriate entities, attributes, and constraints.
- Relationships were enforced using foreign keys, including bridge tables for many-to-many relationships.
- Unique constraints were implemented for important business identifiers (emails and service name).
- The schema was implemented using MySQL Workbench Forward Engineering and successfully created in MySQL.
- All required deliverables were produced: Chen ER diagram, UML/EER diagram, SQL script, and Workbench model file.

Overall, the final database is normalized, organized, and aligned with the mini-world requirements for an online career coaching service.

6. Conclusion

The purpose of this project is to illustrate how to develop a database design from its inception (using the mini-world the database represents), through the development of both conceptual and logical Entity Relationship models, and then to implement the designed schema in the MySQL database management system. The final database design supports scheduling of services, service offerings, packages, payments, and reviews, and it is defined to maintain data integrity through the use of keys and constraints. Future enhancements to the database could include availability calendars, messaging, discount codes, and administrative reporting.