

CIS 344 — Process Documentation

Project: Online Career Coaching Database (coaching_db)

Student: Michael Konadu

Course: CIS 344

Semester: Spring 2026

1) Mini-World Description

This database is an online career coaching service. Clients create their profiles and book sessions with one or more coaches who provide a variety of services, for example, resume review and interview preparation, which may be performed by multiple coaches. The service price for each coach can be overridden. Each session consists of one client, one coach, and one service, and records the session date/time, status, and notes. Each client may either pay for each session individually or purchase packages of multiple sessions; package purchases will track the date purchased, purchase status, and remaining number of sessions. Payment for services will be recorded, including the amount, method, status, and date; the payment may also be associated with either a session or a package purchase. Clients will also have the ability to leave reviews for individual coaches that may link with the specific session; each review will include ratings and comments.

2) How Requirements Were Gathered

Requirements were gathered using:

- Online research + observation of common coaching systems: I reviewed a few common coaching and appointment systems to see what features they usually include, like client/coach profiles, service listings, scheduling (date and time), payments, packages, and reviews.
- Scenario-based analysis (mini-world workflows): I wrote out realistic workflows such as:
 - “Client books a session with a coach for a specific service”
 - “Client pays for a session or pays for a package”
 - “Coach offers multiple services with possible price differences”
 - “Client leaves a review after a session”These workflows were translated into entities, relationships, and business rules.

3) Business Rules

Clients / Coaches

1. Each Client has a unique client_id and must include: first name, last name, email, phone, and created date/time.
2. Each Coach has a unique coach_id and must include: first name, last name, email, phone, bio, hourly rate, and created date/time.
3. Client email must be unique in the Client table.
4. Coach email must be unique in the Coach table.

Services

5. Each Service has a unique service_id and includes: service name (unique), optional description, and base price.
6. A coach can offer many services, and a service can be offered by many coaches (many-to-many).
7. When a coach offers a service, the coach can optionally have a price override for that service.

Sessions

8. Each Session must belong to exactly one Client, exactly one Coach, and exactly one Service.
9. A Client can have many Sessions; a Coach can have many Sessions; a Service can appear in many Sessions.
10. Each Session stores: session date, start time, end time, status, and notes.

Packages / Purchases

11. Each Package includes: package name, number of sessions included, and package price.
12. A Client can purchase multiple Packages over time, and a Package can be purchased by many Clients (many-to-many).
13. Each Client package purchase tracks: purchase date, remaining sessions, and status.

Payments

14. Each Payment must belong to exactly one Client.
15. A Payment can be linked to a Session (optional) and/or a Client_Package purchase

(optional), depending on what the payment is for.

16. Each Payment stores amount, payment date/time, method, and payment status.

Reviews

17. Each Review must belong to exactly one Client and exactly one Coach.

18. A Review may be linked to a specific Session (optional).

19. Each Review stores rating, optional comment, and review date/time.

4) Assumptions

- Clients and coaches both require a phone number in this model (phone is stored as NOT NULL in the current implementation).
- A session always has one service (sessions do not exist without a service).
- Payment records can exist even if they are not tied to a session or package yet (because session_id and client_package_id are nullable).
- Reviews can be general coach reviews or tied to a session (because session_id is nullable in Review).
- Ratings are stored as integers (example range intended: 1–5).
- Primary keys for Client, Service, Session, Package, Client_Package, Payment, Review are AUTO_INCREMENT. Coach is also AUTO_INCREMENT in the final version.

5) Design Decisions

- Session was made a separate entity to store scheduling information (date/time/status/notes) and to connect Client + Coach + Service in one place.
- Coach_Service exists to resolve the Coach ↔ Service many-to-many relationship and to support a relationship attribute (price_override).
- Client_Package exists to resolve the Client ↔ Package many-to-many relationship and to store purchase-specific attributes (purchase_date, remaining_sessions, status).
- Payment is separated to track multiple payment events, payment methods, statuses, and to support payments tied to either sessions or packages.
- Review is separated to store ratings/comments independently and support reviews linked optionally to specific sessions.

6) Keys and Constraints Summary

- Primary Keys (PK):
 - Client(client_id), Coach(coach_id), Service(service_id), Session(session_id), Package(package_id), Client_Package(client_package_id), Payment(payment_id), Review(review_id)
 - Coach_Service has a composite PK: (coach_id, service_id)
- Unique Constraints:
 - Client(email) unique
 - Coach(email) unique
 - Service(service_name) unique
- Foreign Keys (FK) major links:
 - Session = Client, Coach, Service
 - Coach_Service = Coach, Service
 - Client_Package = Client, Package
 - Payment = Client, Session (optional), Client_Package (optional)
 - Review = Client, Coach, Session (optional)

7) Implementation Evidence

- Chen ER Diagram (hand-drawn) showing entities, attributes, relationships, and cardinalities
- UML/EER Diagram in MySQL Workbench with all tables + relationships
- Forward-engineered SQL script (coaching_db.sql) that creates the schema, tables, indexes, and constraints
- Workbench model file (CIS344_coaching_db.mwb)