Presented by:
Data Miners (Khushi Agarwal, Yang Han, Zofia Stefankovic, Tanya Thomas)
Due Oct 5, 2025

Database Requirement Specification

A detailed specification that includes an overview, entities, relationships, and business rules.

Overview

Our project is a professional examination scheduling database. Individuals looking to develop their career may turn to professional exams, either to demonstrate their expertise in an industry, or as a requirement for a stepping stone in their career (graduate school, mandated exams). Nearly every industry has some form of professional exam that is relevant to that field. Examples of common professional exams include: GRE, PMP, CFA, Salesforce Administrator, and more. Many organizations work together to administer these exams around the world, and each organization serves a unique role in the process. Our database will enable test takers to schedule their exam, and for test centers to offer appointments available for scheduling.

Entities

Test Taker = test_taker_id (PK), first_name, last_name, phone_number, street_address, city, state, country, zip_code

Exam Registration = exam_registration_id (PK), exam_id (FK), test_taker_id (FK), invoice_number, registration_date, amount_paid

Appointment = appointment_id (PK), exam_registration_id (FK), date, time, accommodations, appointment_status

Seat Assignment = seat_id (PK), availability_slot_id (FK), appointment_id (FK), seat_number

Test Center Availability = availability_slot_id (PK), test_center_id (FK), date_of_availabilty, start_time_slot, end_time_slot, seat_capacity

Test Center = test_center_id (PK), test_center_name, street_address, city, state, country, zip_code

Test Center Contract = test_center_contract_id (PK), exam_provider_id (FK), test_center_id (FK)

Exam Provider = exam_provider_id (PK), company_name

Sponsor Contract = sponsor_contract_id (PK), exam_provider_id (FK), exam_sponsor_id (FK)

Exam Sponsor = exam_sponsor_id (PK), company_name

Exam = exam_id (PK), exam_sponsor_id (FK), exam_name, exam_duration, domain

Relationships

General Relational Flow:

- **Flow A:** A <u>test taker</u> pays for an <u>exam registration</u>, which authorizes them to make an <u>appointment</u>. To book an appointment, they must pick from one of many different <u>test center availability slots</u>. When the test taker makes a selection, it is recorded as a <u>seat assignment</u>.
- **Flow B:** An <u>exam sponsor</u> offers <u>exams</u>, and to administer these exams, they sign a <u>sponsor contract</u> with an <u>exam provider</u>, who will indirectly connect them with <u>test centers</u> that can administer the exams. An exam provider signs a <u>test center contract</u> with test centers also.
- The connection between *Flow A* and *Flow B* is test centers, who post test center availability slots for test takers to book.

Bidirectional Relationship Rules and Mandatory-ness Rules:

- **Test Taker & Exam Registration** *(one to many)*
  - A test taker will pay for one or many exam registrations. This may include registrations for the same exam (reattempts), or registrations for different exams.
  - An exam registration is associated with exactly one test taker.
  - Test takers enter the database system during exam registration, therefore there are no test takers with no exam registrations.
- **Exam Registration & Exam** *(one to many)*
  - An exam registration is associated with exactly one exam.
  - An exam may have multiple exam registrations (from multiple test takers, or not-concurrent reattempts from a single test taker).
- **Exam Registration & Appointment** *(one to one)*
  - Each exam registration grants authorization for one appointment, but not all exam registrations have appointments (yet).
  - Every appointment has been granted authorization to be made through one exam registration.
  - An appointment cannot exist without being linked to both an exam registration and a seat assignment.
- **Seat Assignment & Test Center Availability** *(one to many)*
  - A seat assignment is booked for a single test center availability time slot.
  - A test center availability slot can have zero or many seat assignments booked in it.
  - A seat assignment cannot exist without being tied to an availability slot and an appointment.
- **Appointment & Seat Assignment** *(one to one)*
  - Every appointment is tied to exactly one seat assignment.
  - Every seat assignment is associated with exactly one appointment.
- **Test Center Availability & Test Center** *(one to many)*
  - A test center availability slot is specific to exactly one particular test center.
  - A test center can list one more many availability slots for booking.

- **Test Center & Exam Provider** (many to many, converted to one to many via associative entity Test Center Contract)
    - **Test Center & Test Center Contract** *(one to many)*
        - A test center will enter a contract with one provider, or enter different contracts with many providers.
    - **Exam Provider & Test Center Contrac**t *(one to many)*
        - An exam provider will enter a contract with one test center, or enter different contracts with many test centers.
    - A given test center contract is associated with one exam provider and one test center.
- **Exam Sponsor & Exam Provider** (many to many, converted to one to many via associative entity Sponsor Contract)
    - **Exam Sponsor & Sponsor Contract** *(one to many)*
        - An exam sponsor will enter a contract with one provider, or enter different contracts with many providers
    - **Exam Provider and Sponsor contract** *(one to many)*
        - An exam provider will enter a contract with one exam sponsor, or enter different contracts with many exam sponsors.
    - A given sponsor contract is associated with one exam provider and one exam sponsor.
- **Exam Sponsor & Exam** *(one to many)*
    - A given exam is created and owned by exactly one exam sponsor (example: College Board, PMI, CFA Institute).
    - A given exam sponsor can offer multiple exams (example: PMI offers the PMP, CAPM, and PMI-ACP, etc)

Business Rules
Rules listed in this section only include rules that have not already been mentioned previously

- **Test Taker & Exam**
    - *The relationship between test taker and exam is not direct and is instead defined by a shared exam registration.*
    - **Test Taker & Exam Registration**
        - A test taker cannot have multiple registrations for a single exam type at the same time, but can have different registrations for different exams simultaneously.
- **Exam Registration & Appointment**
    - Since an appointment is associated with exactly one registration, the appointment will therefore be associated with exactly one exam.
- **Appointments & Test Center Availability**
    - *To bridge the gap between test center availability and appointments, there is a middle ground entity called seat assignment.*
    - **Seat Assignment & Test Center Availability**
        - Test center availability is specified in duration-blocks (specified by start time slot and end time slot). Seat capacity determines how many seat assignments a test center availability slot can accommodate.
            - Example: A test center posts eight 1-hour availability slots every day, Monday through Friday, with each slot capable of booking 5 test

takers. This means the test center can administer two-hundred exams per week.

- In order for an appointment to be booked, duration of the exam must be the exact duration of the availability slot. Therefore, test centers can choose whether to offer many short-duration exams, or fewer long-duration exams, or a mixture of both.
  - Example: A test taker looking to book a 1-hour exam, is permitted to schedule their exam in any test center offering 1-hour availability slots. A test taker looking to book a 4-hour exam must find a different test center that has 4-hour availability slots instead.
- **Appointment & Seat Assignment**
  - Appointment Status:
    - When an exam is scheduled for the first time, the appointment status will read "scheduled", and the seat assignment it is associated with will be blocked from the scheduling portal.
    - When an exam is rescheduled, the appointment status will read "rescheduled". The original seat assignment will be released back to the scheduling portal, and the new seat assignment will be blocked instead.
    - When an exam is cancelled, the appointment status will read "cancelled" and the seat assignment will be released back to the scheduling portal.
  - Test takers can schedule their exam any time in the future and for the sake of this project there is no capped authorization period or scheduling deadline, thus those elements are not represented in this database.
- **Test Center & Exam Sponsor**
  - *There is no direct link between the test center and exam sponsor. These two entities instead work through a middle man, the Exam Provider, who will establish contracts on each side to facilitate scheduling services, connecting a test center and an exam sponsor.*
  - A test center cannot operate without a valid test center contract and an exam sponsor cannot deliver any exams without a valid sponsor contract.
  - **Test Center Contract**
    - Test centers use the test center contract to connect to a wider variety of exam sponsors.
    - Exam providers use the test center contract to widen their pool of locations to support exam sponsors.
  - **Sponsor Contract**
    - Exam sponsors use the sponsor contract to connect to a wider variety of test centers.
    - An exam provider uses the sponsor contract to increase revenue sources.
- **Exam Sponsor & Exam**
  - Each exam must define a duration which will be used for scheduling purposes.

Entity-Relationship Diagram (ERD):

A high-level diagram showing all entities and relationships.

See appendix for a copy of the ERD

Schema Statements:

Convert your ERD into a set of relational tables with appropriate primary and foreign keys.

See appendix for relational tables in table format

test_taker (<u>test_taker_id</u>, first_name, last_name, phone_number, street_address, city, state, country, zip_code)

exam_registration (<u>exam_registration_id</u>, @exam_id, @test_taker_id, invoice_number, registration_date, amount_paid)

appointment (<u>appointment_id</u>, @exam_registration_id, date, time, accommodations, appointment_status)

seat_assignment (<u>seat_id</u>, @availability_slot_id, @appointment_id, seat_number)

test_center_availability (<u>availability_slot_id</u>, @test_center_id, date_of_availability, start_time_slot, end_time_slot, seat_capacity)

test_center (<u>test_center_id</u>, test_center_name, street_address, city, state, country, zip_code)

test_center_contract (<u>test_center_contract_id</u>, @exam_provider_id, @test_center_id)

exam_provider (<u>exam_provider_id</u>, company_name)

sponsor_contract (<u>sponsor_contract_id</u>, @exam_provider_id, @exam_sponsor_id)
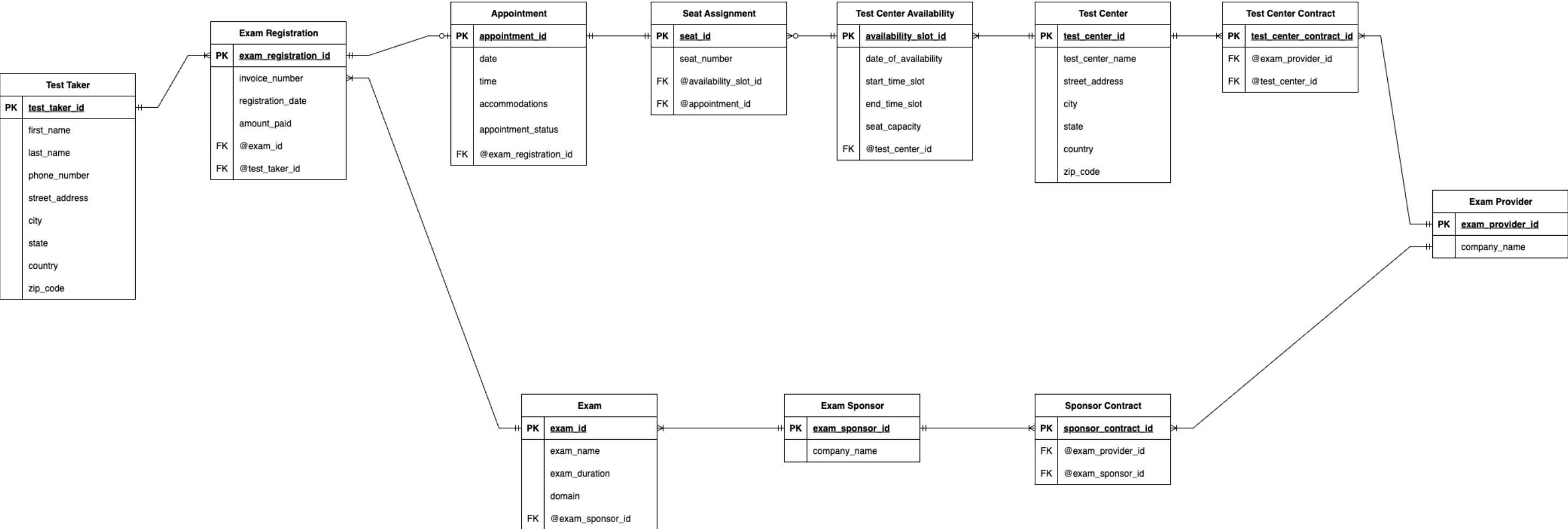
exam_sponsor (<u>exam_sponsor_id</u>, company_name)

exam (<u>exam_id</u>, @exam_sponsor_id, exam_name, exam_duration, domain)

Assumptions and justifications:

1. We removed test center availability duration, as it could be calculated by start/end time. We also removed appointment_duration as that can be determined from the exam_duration and ultimately appointment_duration would end up being a lookup field from the exam entity.

2. It is possible that in the "real world" test centers list hours of operation and then simply implement an algorithm to allow for dynamic, variable-length appointments to be booked within the hours of operation. For this assignment we opted instead to require test centers to create test center availability slots that can accommodate predetermined exam durations, up to a specified capacity per slot. We felt we could ensure a clear connection between a test center's availability and the appointments that are booked by implementing it this way.

3. The relationship between registration and appointment is one to one. While yes, test takers can schedule for multiple appointments in the real world, we made the assumption that if a test taker misses the exam appointment they have to pay (register) for a new exam. Therefore, every registration would have just 1 appointment associated with it. To further this 1 to 1 relationship, instead of generating multiple exam appointments for cancelled or rescheduled exams, we opted instead to simply update the appointment's status and have the status be a trigger to block/release a seat assignment.

4. While primary keys could have been created by concatenating attributes, we opted to assign a unique ID attribute to each entity, as each entity could reasonably serve specific reporting purposes, and modern databases tend to automatically generate primary keys for entities anyways as they are more stable and a database restructure would not impact an attribute as important as a primary key.

5. Some attributes have repeating names (ex: city is found in both test_center and test_taker), but we intentionally did not alter this, as most reporting systems and SQL itself will declare the entity the attribute came from. To increase the legibility of our SQL statements, we opted to omit the entity name from the attribute. (ex: exam_sponsor.company_name **vs.** exam_sponsor.sponsor_company_name)

Professional Examination Scheduling Database

Professional Exam Scheduling Database

test_taker	test_taker_id	first_name	last_name	phone_number	street_address	city	state	country	zip_code

exam_registration	exam_registration_id	@exam_id	@test_taker_id	invoice_number	registration_date	amount_paid

appointment	appointment_id	@exam_registration_id	date	time	accommodations	appointment_status

seat_assignment	seat_id	@availability_slot_id	@appointment_id	seat_number

test_center_availability	availability_slot_id	@test_center_id	date_of_availability	start_time_slot	end_time_slot	seat_capacity

test_center	test_center_id	test_center_name	street_address	city	state	country	zip_code

test_center_contract	test_center_contract_id	@exam_provider_id	@test_center_id

exam_provider	exam_provider_id	company_name

sponsor_contract	sponsor_contract_id	@exam_provider_id	@exam_sponsor_id

exam_sponsor	exam_sponsor_id	company_name

exam	exam_id	@exam_sponsor_id	exam_name	exam_duration	domain

Legend	
Entity Name	
Primary Key	
@Foreign Key	
Attribute	