Presented by:
Data Miners (Khushi Agarwal, Yang Han, Zofia Stefankovic, Tanya Thomas)
Due November 9, 2025

Exam Booker

Database Setup and SQL Statements

Table of Contents

PART 1: Database Specification

Our database is implemented through MySQL because we are most familiar with this framework. At present, we are developing the database locally. To facilitate collaborative development, we have also established a GitHub repo to store the latest SQL files and keep a revision history.

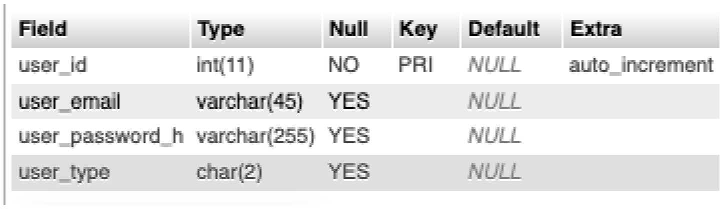
Since we plan to eventually present it in the form of a publicly accessible website, we will eventually download MySQL on a cloud server and run our database on it. The cloud server (AWS Lightsail) is essentially a virtual machine using the Linux system, we will also deploy our HTML and python scripts on it.

PART 2: Table Schemas

View Appendix for updated ERD Diagram and updated schema statements

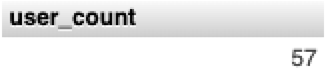
User

DESC user



Field	Type	Null	Key	Default	Extra
user_id	int(11)	NO	PRI	*NULL*	auto_increment
user_email	varchar(45)	YES		*NULL*	
user_password_h	varchar(255)	YES		*NULL*	
user_type	char(2)	YES		*NULL*	

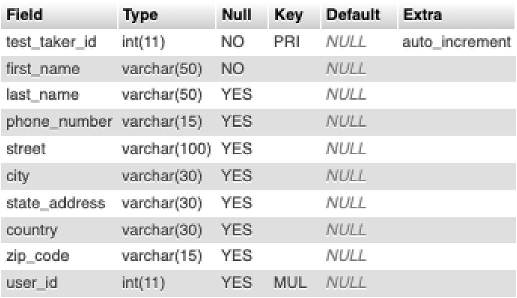
SELECT COUNT(*) AS user_count FROM user;



user_count
57

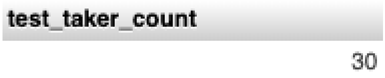
Test Taker

DESC test_taker



Field	Type	Null	Key	Default	Extra
test_taker_id	int(11)	NO	PRI	*NULL*	auto_increment
first_name	varchar(50)	NO		*NULL*	
last_name	varchar(50)	YES		*NULL*	
phone_number	varchar(15)	YES		*NULL*	
street	varchar(100)	YES		*NULL*	
city	varchar(30)	YES		*NULL*	
state_address	varchar(30)	YES		*NULL*	
country	varchar(30)	YES		*NULL*	
zip_code	varchar(15)	YES		*NULL*	
user_id	int(11)	YES	MUL	*NULL*	

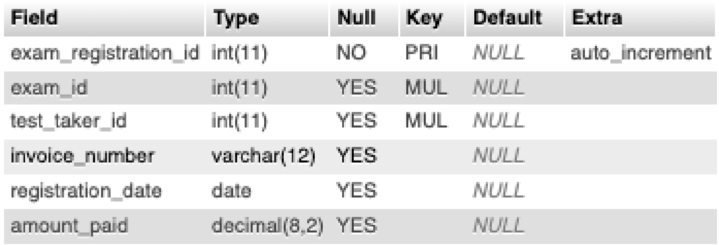
SELECT COUNT(*) AS test_taker_count FROM test_taker;



test_taker_count
30

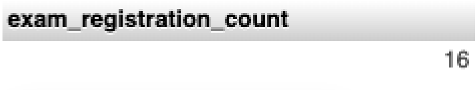
Exam Registration

DESC exam_registration



Field	Type	Null	Key	Default	Extra
exam_registration_id	int(11)	NO	PRI	*NULL*	auto_increment
exam_id	int(11)	YES	MUL	*NULL*	
test_taker_id	int(11)	YES	MUL	*NULL*	
invoice_number	varchar(12)	YES		*NULL*	
registration_date	date	YES		*NULL*	
amount_paid	decimal(8,2)	YES		*NULL*	

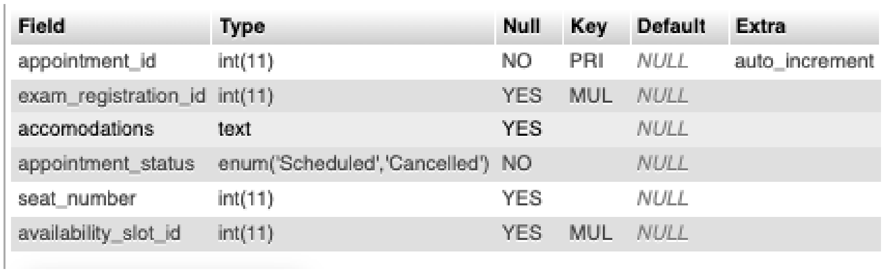
SELECT COUNT(*) AS exam_registration_count FROM exam_registration;



exam_registration_count
16

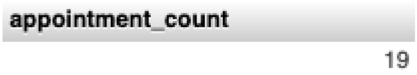
Appointment

DESC appointment



Field	Type	Null	Key	Default	Extra
appointment_id	int(11)	NO	PRI	*NULL*	auto_increment
exam_registration_id	int(11)	YES	MUL	*NULL*	
accomodations	text	YES		*NULL*	
appointment_status	enum('Scheduled','Cancelled')	NO		*NULL*	
seat_number	int(11)	YES		*NULL*	
availability_slot_id	int(11)	YES	MUL	*NULL*	

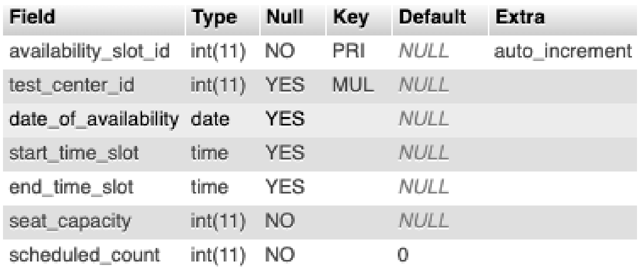
SELECT COUNT(*) AS appointment_count FROM appointment;



appointment_count
19

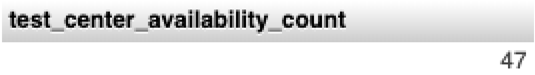
Test Center Availability

DESC test_center_availability



Field	Type	Null	Key	Default	Extra
availability_slot_id	int(11)	NO	PRI	*NULL*	auto_increment
test_center_id	int(11)	YES	MUL	*NULL*	
date_of_availability	date	YES		*NULL*	
start_time_slot	time	YES		*NULL*	
end_time_slot	time	YES		*NULL*	
seat_capacity	int(11)	NO		*NULL*	
scheduled_count	int(11)	NO		0	

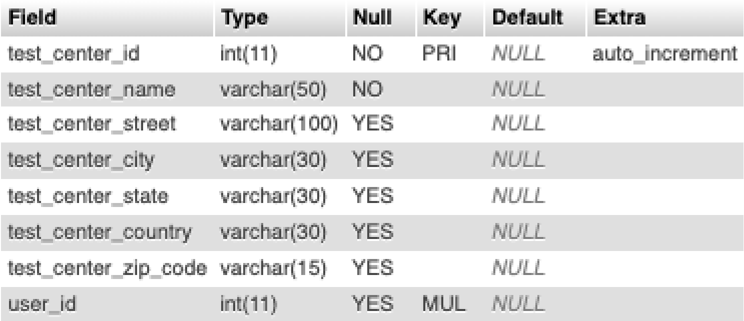
SELECT COUNT(*) AS test_center_availability_count FROM test_center_availability;



test_center_availability_count
47

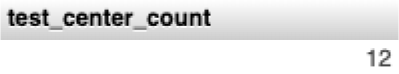
Test Center

DESC test_center



Field	Type	Null	Key	Default	Extra
test_center_id	int(11)	NO	PRI	NULL	auto_increment
test_center_name	varchar(50)	NO		NULL	
test_center_street	varchar(100)	YES		NULL	
test_center_city	varchar(30)	YES		NULL	
test_center_state	varchar(30)	YES		NULL	
test_center_country	varchar(30)	YES		NULL	
test_center_zip_code	varchar(15)	YES		NULL	
user_id	int(11)	YES	MUL	NULL	

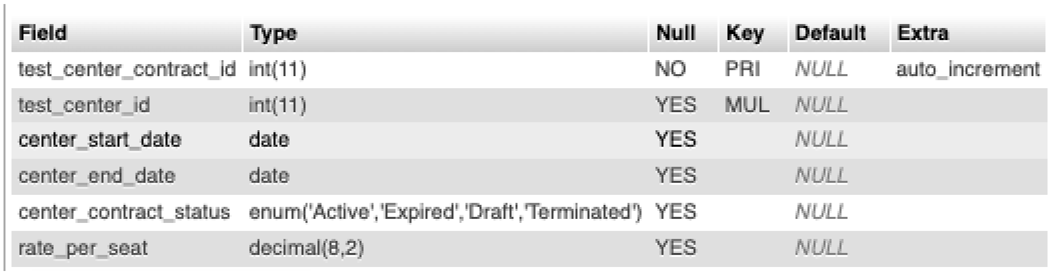
SELECT COUNT(*) AS test_center_count FROM test_center;



test_center_count
12

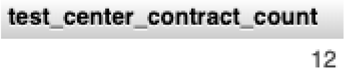
Test Center Contract

DESC test_center_contract



Field	Type	Null	Key	Default	Extra
test_center_contract_id	int(11)	NO	PRI	NULL	auto_increment
test_center_id	int(11)	YES	MUL	NULL	
center_start_date	date	YES		NULL	
center_end_date	date	YES		NULL	
center_contract_status	enum('Active','Expired','Draft','Terminated')	YES		NULL	
rate_per_seat	decimal(8,2)	YES		NULL	

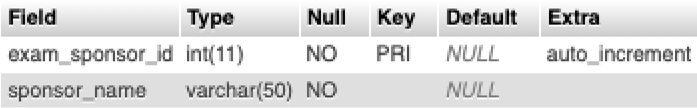
SELECT COUNT(*) AS test_center_contract_count FROM test_center_contract;



test_center_contract_count
12

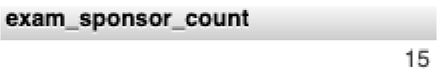
Exam Sponsor

DESC exam_sponsor



Field	Type	Null	Key	Default	Extra
exam_sponsor_id	int(11)	NO	PRI	NULL	auto_increment
sponsor_name	varchar(50)	NO		NULL	

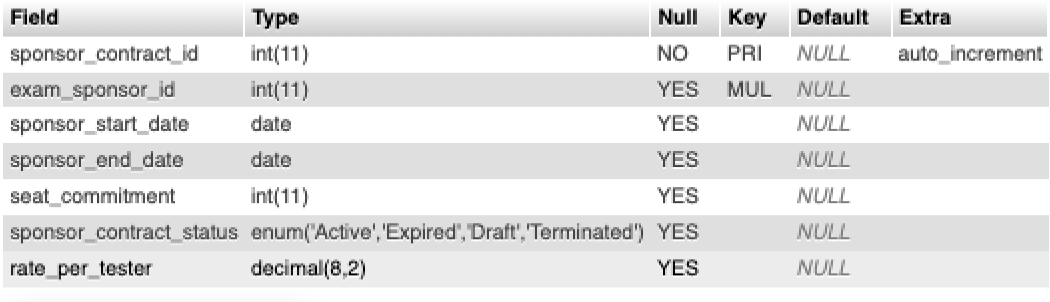
SELECT COUNT(*) AS exam_sponsor_count FROM exam_sponsor;



exam_sponsor_count
15

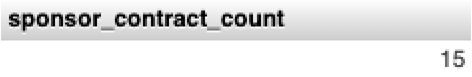
Sponsor Contract

DESC sponsor_contract



Field	Type	Null	Key	Default	Extra
sponsor_contract_id	int(11)	NO	PRI	*NULL*	auto_increment
exam_sponsor_id	int(11)	YES	MUL	*NULL*	
sponsor_start_date	date	YES		*NULL*	
sponsor_end_date	date	YES		*NULL*	
seat_commitment	int(11)	YES		*NULL*	
sponsor_contract_status	enum('Active','Expired','Draft','Terminated')	YES		*NULL*	
rate_per_tester	decimal(8,2)	YES		*NULL*	

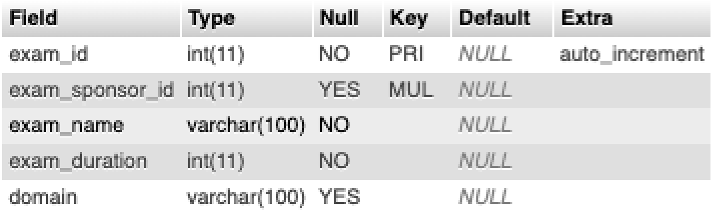
SELECT COUNT(*) AS sponsor_contract_count FROM sponsor_contract;



sponsor_contract_count
15

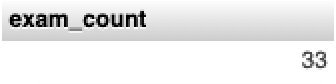
Exam

DESC exam



Field	Type	Null	Key	Default	Extra
exam_id	int(11)	NO	PRI	*NULL*	auto_increment
exam_sponsor_id	int(11)	YES	MUL	*NULL*	
exam_name	varchar(100)	NO		*NULL*	
exam_duration	int(11)	NO		*NULL*	
domain	varchar(100)	YES		*NULL*	

SELECT COUNT(*) AS exam_count FROM exam;



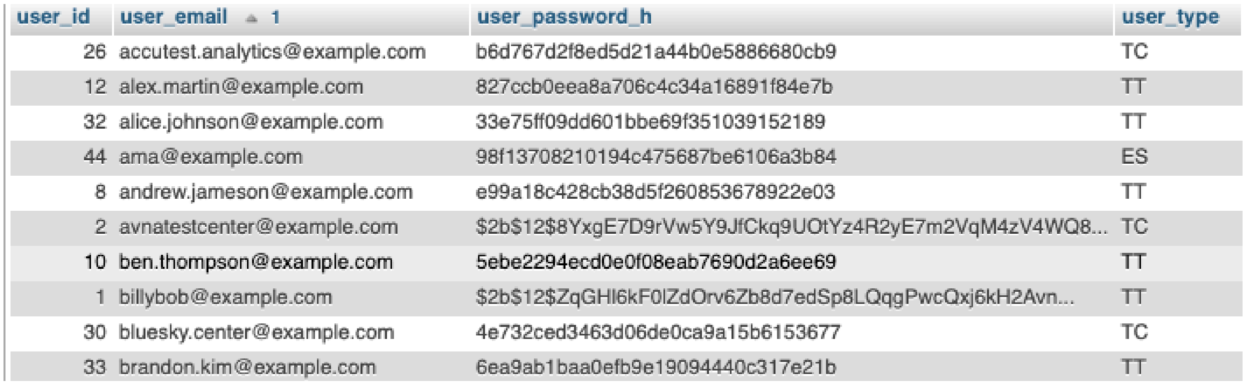
exam_count
33

PART 3: Sample Data

User

SELECT * FROM user;

showing 10 out of 57, sorted by email to demonstrate different user types

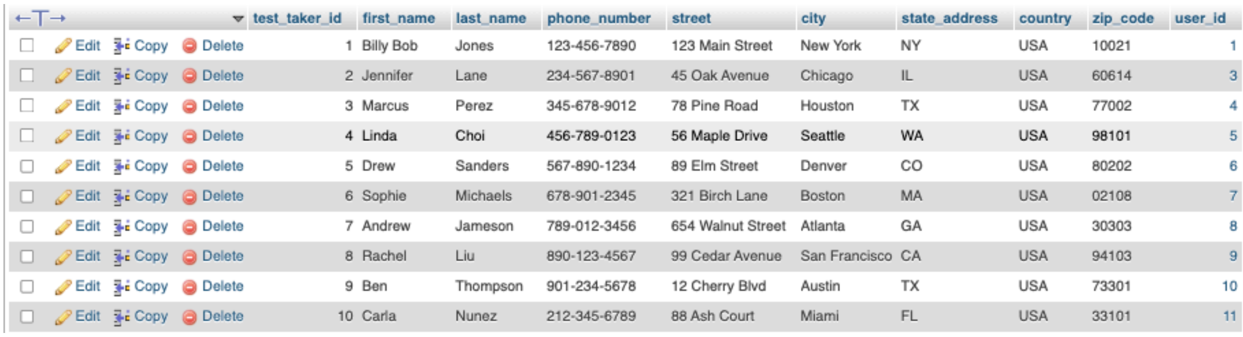


user_id	user_email ▲ 1	user_password_h	user_type
26	accutest.analytics@example.com	b6d767d2f8ed5d21a44b0e5886680cb9	TC
12	alex.martin@example.com	827ccb0eea8a706c4c34a16891f84e7b	TT
32	alice.johnson@example.com	33e75ff09dd601bbe69f351039152189	TT
44	ama@example.com	98f13708210194c475687be6106a3b84	ES
8	andrew.jameson@example.com	e99a18c428cb38d5f260853678922e03	TT
2	avnatestcenter@example.com	$2b$12$8YxgE7D9rVw5Y9JfCkq9UOtYz4R2yE7m2VqM4zV4WQ8...	TC
10	ben.thompson@example.com	5ebe2294ecd0e0f08eab7690d2a6ee69	TT
1	billybob@example.com	$2b$12$ZqGHl6kF0lZdOrv6Zb8d7edSp8LQqgPwcQxj6kH2Avn...	TT
30	bluesky.center@example.com	4e732ced3463d06de0ca9a15b6153677	TC
33	brandon.kim@example.com	6ea9ab1baa0efb9e19094440c317e21b	TT

Test Taker

SELECT * FROM test_taker;

showing 10 out of 30

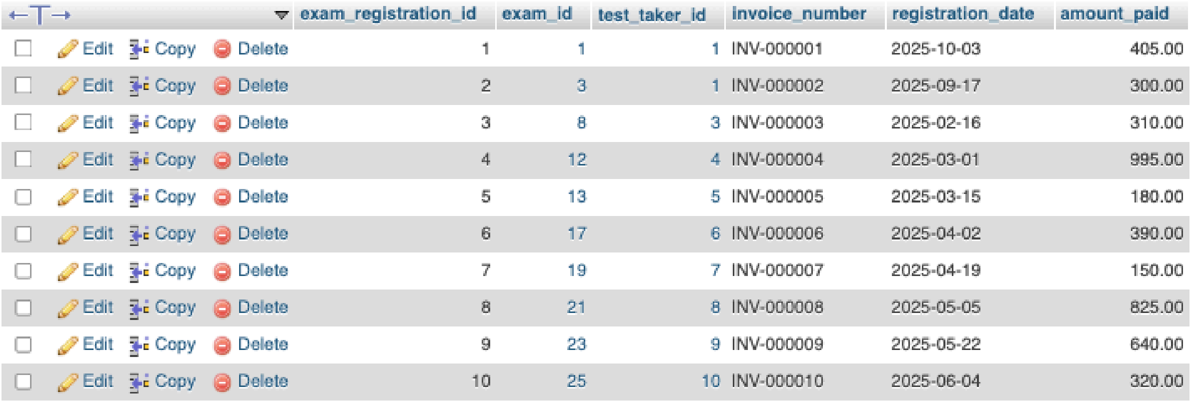


←T→	test_taker_id	first_name	last_name	phone_number	street	city	state_address	country	zip_code	user_id
☐ Edit Copy Delete	1	Billy Bob	Jones	123-456-7890	123 Main Street	New York	NY	USA	10021	1
☐ Edit Copy Delete	2	Jennifer	Lane	234-567-8901	45 Oak Avenue	Chicago	IL	USA	60614	3
☐ Edit Copy Delete	3	Marcus	Perez	345-678-9012	78 Pine Road	Houston	TX	USA	77002	4
☐ Edit Copy Delete	4	Linda	Choi	456-789-0123	56 Maple Drive	Seattle	WA	USA	98101	5
☐ Edit Copy Delete	5	Drew	Sanders	567-890-1234	89 Elm Street	Denver	CO	USA	80202	6
☐ Edit Copy Delete	6	Sophie	Michaels	678-901-2345	321 Birch Lane	Boston	MA	USA	02108	7
☐ Edit Copy Delete	7	Andrew	Jameson	789-012-3456	654 Walnut Street	Atlanta	GA	USA	30303	8
☐ Edit Copy Delete	8	Rachel	Liu	890-123-4567	99 Cedar Avenue	San Francisco	CA	USA	94103	9
☐ Edit Copy Delete	9	Ben	Thompson	901-234-5678	12 Cherry Blvd	Austin	TX	USA	73301	10
☐ Edit Copy Delete	10	Carla	Nunez	212-345-6789	88 Ash Court	Miami	FL	USA	33101	11

Exam Registration

SELECT * FROM exam_registration;

showing 10 out of 16

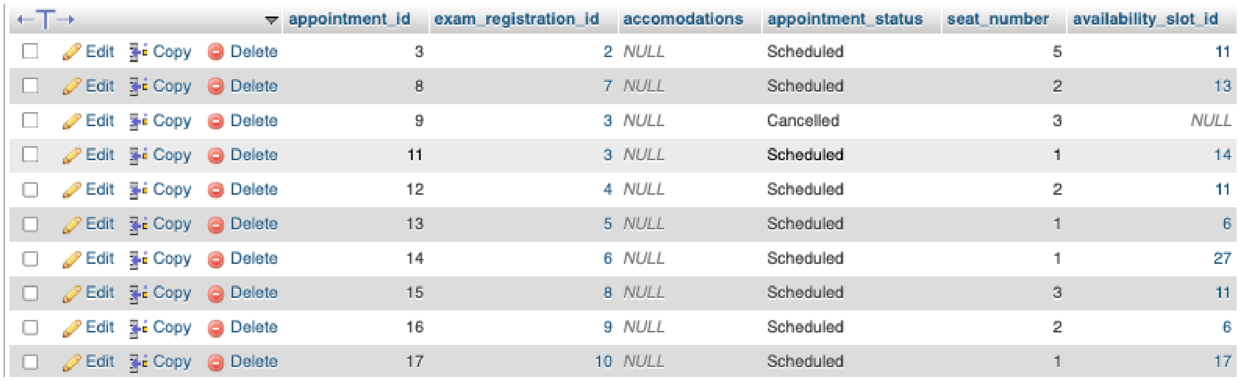


←T→	exam_registration_id	exam_id	test_taker_id	invoice_number	registration_date	amount_paid
☐ Edit Copy Delete	1	1	1	INV-000001	2025-10-03	405.00
☐ Edit Copy Delete	2	3	1	INV-000002	2025-09-17	300.00
☐ Edit Copy Delete	3	8	3	INV-000003	2025-02-16	310.00
☐ Edit Copy Delete	4	12	4	INV-000004	2025-03-01	995.00
☐ Edit Copy Delete	5	13	5	INV-000005	2025-03-15	180.00
☐ Edit Copy Delete	6	17	6	INV-000006	2025-04-02	390.00
☐ Edit Copy Delete	7	19	7	INV-000007	2025-04-19	150.00
☐ Edit Copy Delete	8	21	8	INV-000008	2025-05-05	825.00
☐ Edit Copy Delete	9	23	9	INV-000009	2025-05-22	640.00
☐ Edit Copy Delete	10	25	10	INV-000010	2025-06-04	320.00

Appointment

SELECT * FROM appointment;

showing 10 out of 19. Also demonstrates how an exam can be cancelled and therefore not have an availability slot associated with it anymore

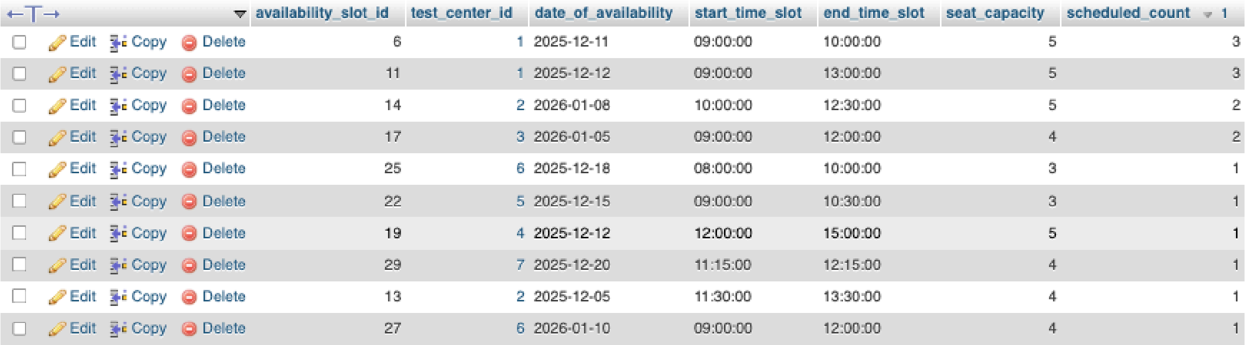


←T→	appointment_id	exam_registration_id	accomodations	appointment_status	seat_number	availability_slot_id
□ Edit Copy Delete	3	2	NULL	Scheduled	5	11
□ Edit Copy Delete	8	7	NULL	Scheduled	2	13
□ Edit Copy Delete	9	3	NULL	Cancelled	3	NULL
□ Edit Copy Delete	11	3	NULL	Scheduled	1	14
□ Edit Copy Delete	12	4	NULL	Scheduled	2	11
□ Edit Copy Delete	13	5	NULL	Scheduled	1	6
□ Edit Copy Delete	14	6	NULL	Scheduled	1	27
□ Edit Copy Delete	15	8	NULL	Scheduled	3	11
□ Edit Copy Delete	16	9	NULL	Scheduled	2	6
□ Edit Copy Delete	17	10	NULL	Scheduled	1	17

Test Center Availability

SELECT * FROM test_center_availability;

showing 10 out of 47, sorted by scheduled_count to demonstrate scheduling status

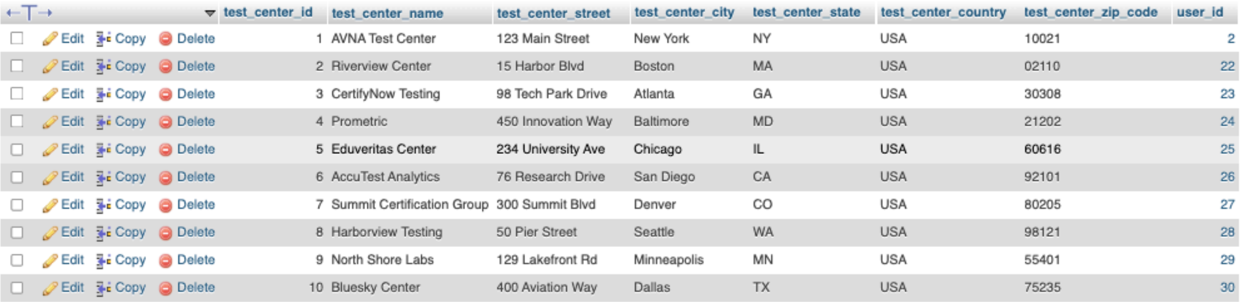


←T→	availability_slot_id	test_center_id	date_of_availability	start_time_slot	end_time_slot	seat_capacity	scheduled_count 1
□ Edit Copy Delete	6	1	2025-12-11	09:00:00	10:00:00	5	3
□ Edit Copy Delete	11	1	2025-12-12	09:00:00	13:00:00	5	3
□ Edit Copy Delete	14	2	2026-01-08	10:00:00	12:30:00	5	2
□ Edit Copy Delete	17	3	2026-01-05	09:00:00	12:00:00	4	2
□ Edit Copy Delete	25	6	2025-12-18	08:00:00	10:00:00	3	1
□ Edit Copy Delete	22	5	2025-12-15	09:00:00	10:30:00	3	1
□ Edit Copy Delete	19	4	2025-12-12	12:00:00	15:00:00	5	1
□ Edit Copy Delete	29	7	2025-12-20	11:15:00	12:15:00	4	1
□ Edit Copy Delete	13	2	2025-12-05	11:30:00	13:30:00	4	1
□ Edit Copy Delete	27	6	2026-01-10	09:00:00	12:00:00	4	1

Test Center

SELECT * FROM test_center;

showing 10 out of 12

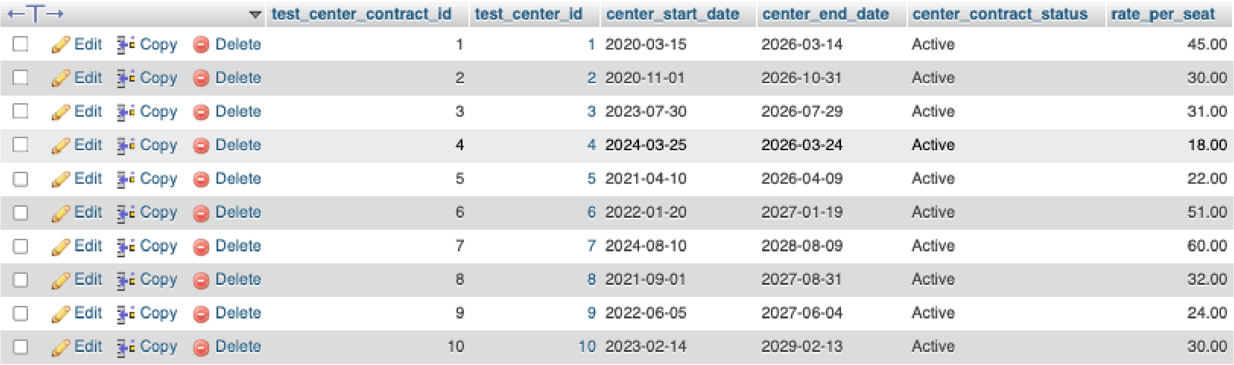


←T→	test_center_id	test_center_name	test_center_street	test_center_city	test_center_state	test_center_country	test_center_zip_code	user_id
□ Edit Copy Delete	1	AVNA Test Center	123 Main Street	New York	NY	USA	10021	2
□ Edit Copy Delete	2	Riverview Center	15 Harbor Blvd	Boston	MA	USA	02110	22
□ Edit Copy Delete	3	CertifyNow Testing	98 Tech Park Drive	Atlanta	GA	USA	30308	23
□ Edit Copy Delete	4	Prometric	450 Innovation Way	Baltimore	MD	USA	21202	24
□ Edit Copy Delete	5	Eduveritas Center	234 University Ave	Chicago	IL	USA	60616	25
□ Edit Copy Delete	6	AccuTest Analytics	76 Research Drive	San Diego	CA	USA	92101	26
□ Edit Copy Delete	7	Summit Certification Group	300 Summit Blvd	Denver	CO	USA	80205	27
□ Edit Copy Delete	8	Harborview Testing	50 Pier Street	Seattle	WA	USA	98121	28
□ Edit Copy Delete	9	North Shore Labs	129 Lakefront Rd	Minneapolis	MN	USA	55401	29
□ Edit Copy Delete	10	Bluesky Center	400 Aviation Way	Dallas	TX	USA	75235	30

Test Center Contract

SELECT * FROM test_center_contract;

showing 10 out of 12

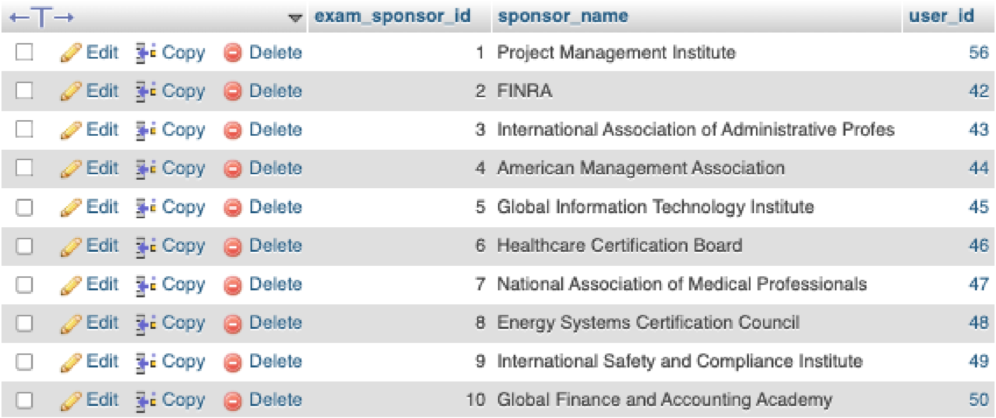


←T→	test_center_contract_id	test_center_id	center_start_date	center_end_date	center_contract_status	rate_per_seat
☐ Edit Copy Delete	1	1	2020-03-15	2026-03-14	Active	45.00
☐ Edit Copy Delete	2	2	2020-11-01	2026-10-31	Active	30.00
☐ Edit Copy Delete	3	3	2023-07-30	2026-07-29	Active	31.00
☐ Edit Copy Delete	4	4	2024-03-25	2026-03-24	Active	18.00
☐ Edit Copy Delete	5	5	2021-04-10	2026-04-09	Active	22.00
☐ Edit Copy Delete	6	6	2022-01-20	2027-01-19	Active	51.00
☐ Edit Copy Delete	7	7	2024-08-10	2028-08-09	Active	60.00
☐ Edit Copy Delete	8	8	2021-09-01	2027-08-31	Active	32.00
☐ Edit Copy Delete	9	9	2022-06-05	2027-06-04	Active	24.00
☐ Edit Copy Delete	10	10	2023-02-14	2029-02-13	Active	30.00

Exam Sponsor

SELECT * FROM exam_sponsor;

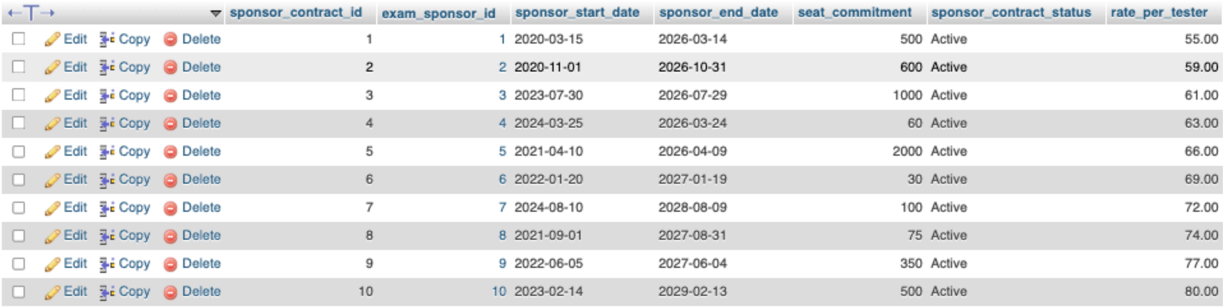
showing 10 out of 15



←T→	exam_sponsor_id	sponsor_name	user_id
☐ Edit Copy Delete	1	Project Management Institute	56
☐ Edit Copy Delete	2	FINRA	42
☐ Edit Copy Delete	3	International Association of Administrative Profes	43
☐ Edit Copy Delete	4	American Management Association	44
☐ Edit Copy Delete	5	Global Information Technology Institute	45
☐ Edit Copy Delete	6	Healthcare Certification Board	46
☐ Edit Copy Delete	7	National Association of Medical Professionals	47
☐ Edit Copy Delete	8	Energy Systems Certification Council	48
☐ Edit Copy Delete	9	International Safety and Compliance Institute	49
☐ Edit Copy Delete	10	Global Finance and Accounting Academy	50

Sponsor Contract

SELECT * FROM sponsor_contract

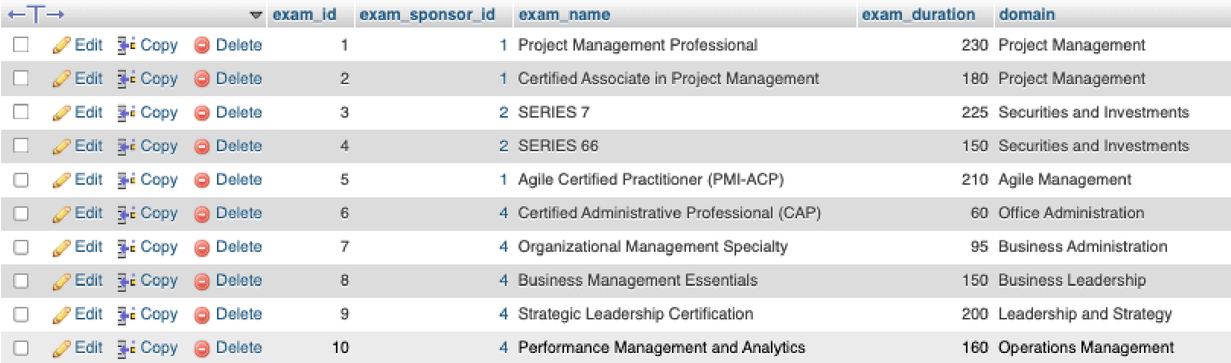


←T→	sponsor_contract_id	exam_sponsor_id	sponsor_start_date	sponsor_end_date	seat_commitment	sponsor_contract_status	rate_per_tester
☐ Edit Copy Delete	1	1	2020-03-15	2026-03-14	500	Active	55.00
☐ Edit Copy Delete	2	2	2020-11-01	2026-10-31	600	Active	59.00
☐ Edit Copy Delete	3	3	2023-07-30	2026-07-29	1000	Active	61.00
☐ Edit Copy Delete	4	4	2024-03-25	2026-03-24	60	Active	63.00
☐ Edit Copy Delete	5	5	2021-04-10	2026-04-09	2000	Active	66.00
☐ Edit Copy Delete	6	6	2022-01-20	2027-01-19	30	Active	69.00
☐ Edit Copy Delete	7	7	2024-08-10	2028-08-09	100	Active	72.00
☐ Edit Copy Delete	8	8	2021-09-01	2027-08-31	75	Active	74.00
☐ Edit Copy Delete	9	9	2022-06-05	2027-06-04	350	Active	77.00
☐ Edit Copy Delete	10	10	2023-02-14	2029-02-13	500	Active	80.00

Exam

SELECT * FROM exam;

showing 10 out of 33



exam_id	exam_sponsor_id	exam_name	exam_duration	domain
1	1	Project Management Professional	230	Project Management
2	1	Certified Associate in Project Management	180	Project Management
3	2	SERIES 7	225	Securities and Investments
4	2	SERIES 66	150	Securities and Investments
5	1	Agile Certified Practitioner (PMI-ACP)	210	Agile Management
6	4	Certified Administrative Professional (CAP)	60	Office Administration
7	4	Organizational Management Specialty	95	Business Administration
8	4	Business Management Essentials	150	Business Leadership
9	4	Strategic Leadership Certification	200	Leadership and Strategy
10	4	Performance Management and Analytics	160	Operations Management

PART 4: SQL Commands

Create Table SQL

User

```
CREATE TABLE user (
   user_id int(11) NOT NULL AUTO_INCREMENT,
   user_email varchar(45),
   user_password_h varchar(255),
   user_type char(2)  CHECK (user_type in ('TC','TT','ES')),
   PRIMARY KEY (user_id)
);
```

Test Taker

```
CREATE TABLE test_taker (
       test_taker_id int(11) NOT NULL AUTO_INCREMENT,
       first_name varchar(50) NOT NULL,
       last_name varchar(50) ,
       phone_number varchar(15) ,
       street varchar(100) ,
       city varchar(30) ,
       state_address varchar(30) ,
       country varchar(30) ,
       zip_code varchar(15) ,
       user_id int(11) ,
   PRIMARY KEY (test_taker_id),
   FOREIGN KEY (user_id) REFERENCES user (user_id)
);
```

Exam Registration

```
CREATE TABLE exam_registration (
       exam_registration_id int(11) NOT NULL AUTO_INCREMENT,
       exam_id int(11),
       test_taker_id int(11),
       invoice_number varchar(12),
       registration_date date,
       amount_paid decimal(8,2),
   PRIMARY KEY (exam_registration_id),
   FOREIGN KEY (exam_id) REFERENCES exam (exam_id),
   FOREIGN KEY (test_taker_id) REFERENCES test_taker (test_taker_id)
);
```

Appointment

```
CREATE TABLE appointment (
        appointment_id int(11) NOT NULL AUTO_INCREMENT,
        exam_registration_id int(11),
        accomodations text,
        appointment_status enum('Scheduled','Cancelled') NOT NULL,
        seat_number int(11),
        availability_slot_id int(11),
   PRIMARY KEY (appointment_id),
   FOREIGN KEY (exam_registration_id) REFERENCES exam_registration
(exam_registration_id),
   FOREIGN KEY (availability_slot_id) REFERENCES test_center_availability
(availability_slot_id)
);
```

Test Center Availability

```
CREATE TABLE test_center_availability (
        availability_slot_id int(11) NOT NULL AUTO_INCREMENT,
        test_center_id int(11) ,
        date_of_availability date ,
        start_time_slot time ,
        end_time_slot time ,
        seat_capacity int(11) NOT NULL CHECK (seat_capacity >= 0),
        scheduled_count int(11) NOT NULL DEFAULT 0
   PRIMARY KEY (availability_slot_id),
   FOREIGN KEY (test_center_id) REFERENCES test_center (test_center_id)
);
```

Test Center

```
CREATE TABLE test_center (
        test_center_id int(11) NOT NULL AUTO_INCREMENT,
        test_center_name varchar(50) NOT NULL,
        test_center_street varchar(100),
        test_center_city varchar(30),
        test_center_state varchar(30),
        test_center_country varchar(30),
        test_center_zip_code varchar(15),
        user_id int(11) ,
   PRIMARY KEY (test_center_id),
   FOREIGN KEY (user_id) REFERENCES user (user_id)
);
```

Test Center Contract

```
CREATE TABLE test_center_contract (
       test_center_contract_id int(11) NOT NULL AUTO_INCREMENT,
       test_center_id int(11) ,
       center_start_date date ,
       center_end_date date ,
       center_contract_status enum('Active','Expired','Draft','Terminated') ,
       rate_per_seat decimal(8,2),
   PRIMARY KEY (test_center_contract_id),
   FOREIGN KEY (test_center_id) REFERENCES test_center (test_center_id)
);
```

Exam Sponsor

```
CREATE TABLE exam_sponsor (
       exam_sponsor_id int(11) NOT NULL AUTO_INCREMENT,
       sponsor_name varchar(50) NOT NULL,
       user_id int(11) NOT NULL,
   PRIMARY KEY (exam_sponsor_id),
   FOREIGN KEY (user_id) REFERENCES user (user_id)
);
```

Sponsor Contract

```
CREATE TABLE sponsor_contract (
       sponsor_contract_id int(11) NOT NULL AUTO_INCREMENT,
       exam_sponsor_id int(11),
       sponsor_start_date date,
       sponsor_end_date date,
       seat_commitment int(11),
       sponsor_contract_status enum('Active','Expired','Draft','Terminated'),
       rate_per_tester decimal(8,2),
   PRIMARY KEY (sponsor_contract_id),
   FOREIGN KEY (exam_sponsor_id) REFERENCES exam_sponsor (exam_sponsor_id)
);
```

Exam

```
CREATE TABLE exam (
       exam_id int(11) NOT NULL AUTO_INCREMENT,
       exam_sponsor_id int(11),
       exam_name varchar(100) NOT NULL,
       exam_duration int(11) NOT NULL CHECK (exam_duration > 0),
       domain varchar(100),
   PRIMARY KEY (exam_id),
   FOREIGN KEY (exam_sponsor_id) REFERENCES exam_sponsor (exam_sponsor_id)
);
```

Non-Advanced SQL

We have three different types of users that use our website in a logged in state. We have listed each user type and a bullet for each of the capabilities on the website for said user. Additionally, we have tagged each SQL as either “retrieve data” , “add data”, “update data”, or “delete data”.

We designed our database to prevent duplication of data, and doing so means that at times many entities must be traversed. We designed it this way to ensure data consistency between entities. For example, the address for a test taker’s appointment can be found, not in the appointment details, but rather in the test center details for the test center they are scheduled for. A database admin must traverse through appointment > test center availability > test center to locate a test taker’s appointment location. A change in a test center’s address would then automatically reflect in a test taker’s appointment, since the appointment address is actually the test center address. Since many of our queries require traversal of multiple tables, we created Views that makes it easy for a database administrator to view the relationship between our entities. Our website will also query our views as a security measure.

Views

- **Registered Test Takers** - shows all test takers that have an exam registration, and whether or not they have an appointment scheduled or cancelled for that exam registration.

```
CREATE VIEW registered_test_takers AS
SELECT
       r.test_taker_id,r.exam_registration_id,
       e.exam_name, e.exam_duration, e.exam_id, e.exam_sponsor_id,
       a.appointment_status
FROM exam e
INNER JOIN exam_registration r
ON e.exam_id = r.exam_id
LEFT JOIN appointment a
ON r.exam_registration_id = a.exam_registration_id;
```

- **Scheduled Test Takers** - shows all test takers that have scheduled appointments for their exam registrations. The inner join ensures only exam registrations with associated scheduled (or cancelled) appointments are visible.

```
CREATE VIEW scheduled_test_takers AS
SELECT
       e.exam_id, e.exam_sponsor_id, e.exam_duration, e.exam_name
       t.test_taker_id, t.state_address, t.country,
       r.exam_registration_id,
       a.appointment_id,a.accomodations,a.appointment_status, a.seat_number,
       av.availability_slot_id, av.date_of_availability, av.start_time_slot,
       av.end_time_slot, av.seat_capacity,av.scheduled_count,
        tc.test_center_id, tc.test_center_name, tc.test_center_street,
        tc.test_center_state, tc.test_center_city,tc.test_center_country,
        tc.test_center_zip_code
FROM test_taker t
INNER JOIN exam_registration r
ON t.test_taker_id = r.test_taker_id
INNER JOIN exam e
ON r.exam_id = e.exam_id
INNER JOIN appointment a
ON r.exam_registration_id = a.exam_registration_id
INNER JOIN test_center_availability av
ON a.availability_slot_id = av.availability_slot_id
INNER JOIN test_center tc
ON av.test_center_id = tc.test_center_id;
```

- **Test Centers with Availability -** shows all test centers that have posted availability. The inner join ensures only test centers with availability are shown. Slot duration is also calculated here for ease of reference.

  ```
  CREATE VIEW test_centers_with_availability AS
  SELECT
          tc.test_center_id, tc.test_center_name, tc.test_center_street,
          tc.test_center_city, tc.test_center_state, tc.test_center_country,
          tc.test_center_zip_code,
          av.availability_slot_id, av.date_of_availability, av.start_time_slot,
          av.end_time_slot, TIMESTAMPDIFF(MINUTE, start_time_slot, end_time_slot)
          AS "slot_duration", av.seat_capacity, av.scheduled_count
  FROM test_center tc
  INNER JOIN test_center_availability av
  ON tc.test_center_id = av.test_center_id;
  ```

Public

- Public users to our site can view the list of sponsors we have partnered with
  State: Logged out
  Query:

  ```
          SELECT sponsor_name
          FROM exam_sponsor;
  ```

Test Taker

For queries that assume a logged in state, we use sample Test Taker user_id = 1 and test_taker_id = 1

- ### Can create an account

  State: Logged in

  Tags: Add data

  Query:

  A password hash will be generated via Python when creating an account. After a password has been created, the following query will insert the new user into the database. The user_type = TT is defined by the fact the test taker clicked to create a test taker account.

  ```
  INSERT INTO user(user_email, user_password_h, user_type)
  VALUES
  ("billybob@example.com",
  "$2b$12$ZqGHl6kF0lZdOrv6Zb8d7edSp8LQqgPwcQxj6kH2AvnDJR7rF3vG", "TT");
  ```

- ### Can log in

  State: Logged out

  Tags: Retrieve data

  Query:

  Retrieves the correct (hashed) password for the entered email and evaluates whether the user should be logged in

  ```
  SELECT user_password_h
  FROM user
  WHERE user_email = "billybob@example.com";
  ```

- ### Can register for an exam

  State: Logged in

  Tags: Retrieve data, Add data

  Query:

  First select an exam sponsor from the drop-down list of available sponsors to test with. Drop down list is generated by the following query:

  ```
  SELECT sponsor_name FROM exam_sponsor;
  ```

  After selecting a sponsor, select an exam (filtered out by previously selected sponsor). Drop down list of exams to choose is generated by the following query:

  ```
  SELECT exam_name
  FROM exam
  WHERE exam_sponsor_id = (
     SELECT exam_sponsor_id
     FROM exam_sponsor
     WHERE sponsor_name = "Project Management Institute"
  );
  ```

After selecting an exam, and successfully filling out the payment information, exam registration details will be inserted into the DB.
To extract the exam_id from exam entity for insert:

```
SELECT exam_id
FROM exam
WHERE exam_name = "Project Management Professional";
```

The above query returns exam_id = 1. An incrementing invoice number is generated, and registration date is filled in with today's date, and price is filled in.

```
INSERT INTO exam_registration(exam_id, test_taker_id, invoice_number,
registration_date, amount_paid)
VALUES (1,1,"INV-000001",CURDATE(),405);
```

- Can view their exam registrations
  State: Logged in
  Tags: Retrieve data
  Query:

  We have a view "registered_test_takers" (click here to view). The below query filters down the view to just the records associated with the logged in test taker (test_taker_id). The view does not include sponsor_name, so to extract the sponsor name, we do a join of the view and exam_sponsor tables.

```
SELECT s.sponsor_name, reg_view.exam_name
FROM registered_test_takers reg_view
INNER JOIN exam_sponsor s
ON reg_view.exam_sponsor_id = s.exam_sponsor_id
WHERE reg_view.test_taker_id = 1;
```

- Can view test center availability for an exam they have already registered for
  State: Logged in
  Tags: Retrieve data
  Query:

  To get to this page a test taker must first select the exam they would like to see availability for. We have a view "test_centers_with_availability" (click here to view) which shows all test centers that have posted availability. This view has a calculated "slot duration", which will be compared against the test taker's registered exam duration to ensure the test taker does not see availability for slots that are shorter than their exam. We assume in the below example, the test taker has selected exam_id = 1. The below query filters out availability slots that have been filled as well. The view includes:

  From test center: full test center address and name
  From test center availability: date and time of exam appointment slot

```
SELECT * FROM test_centers_with_availability tca_view
WHERE tca_view.slot_duration > (
   SELECT reg_view.exam_duration
   FROM registered_test_takers reg_view
   WHERE reg_view.test_taker_id = 1
   AND reg_view.exam_registration_id = 1
)
AND (tca_view.seat_capacity - tca_view.scheduled_count) > 0;
```

When a test taker conducts a search by either city, state, country, or zipcode, the following lines are included at the end of the above query to search for any matches in any of those fields.

```
AND tca.test_center_city = "New York"
OR tca.test_center_state = "New York"
OR tca.test_center_country = "New York"
OR tca.test_center_zip_code = "New York";
```

- Can schedule an exam they have already registered for
  State: Logged in
  Tags: Add data
  Query:

  After viewing all availability and selecting a preferred availability slot (in this example they have selected availability_slot_id = 4), the appointment is scheduled (inserted into the database). Since the test taker had to click into the exam registration they were looking to schedule to get to the view-availability page, we know the exam_registration_id = 1. Scheduling an appointment also triggers a change to schedule_count and increments it by 1, taking up one of the capacity slots for that availability. Click here to view the related trigger.

```
INSERT INTO appointment (exam_registration_id,
appointment_status,seat_number, availability_slot_id)
VALUES
(1, 'Scheduled', 1, 4);
```

- Can view their scheduled appointments
  State: Logged in
  Tags: Retrieve data
  Query:

  We have a view “scheduled_test_takers” (click here to view) which is filtered down in the below query by the specific test taker ID, and also omits cancelled exam appointments.

  The view includes:

  From exam: exam name, exam duration,
  From test center: complete test center address

From test center availability: date and time of exam
From appointment: accommodations and seat number

```
SELECT * FROM scheduled_test_takers
WHERE appointment_status = "Scheduled"
AND test_taker_id = 1;
```

- Can cancel their appointment:
  State: Logged in
  Tags: Update data
  Query:

  When a test taker cancels their appointment, it does not decouple their appointment from the test center availability slot; this way we know historically what exam appointments they had been registered for. Instead the cancellation simply changes the appointment from “Scheduled” to “Cancelled” and we have a trigger that updates the test_center_availability slot’s scheduled count accordingly to “release the availability” back to the public. Click here to view the related trigger.

  ```
  UPDATE appointment
  SET appointment_status = 'Cancelled'
  WHERE exam_registration_id = 1;
  ```

- Can reschedule their appointment:
  State: Logged in
  Tags: Add data, Update data
  Query:

  Rescheduling to the test taker looks like they are changing the appointment details, but in our database it is actually an appointment cancellation followed immediately by an insert of a new record. This triggers the release_availability trigger as well as the acquire_availability trigger.

  ```
  UPDATE appointment
  SET appointment_status = 'Cancelled'
  WHERE exam_registration_id = 1;

  INSERT INTO appointment (exam_registration_id,
  appointment_status,seat_number, availability_slot_id)
  VALUES
  (1, 'Scheduled', 1, 4);
  ```

Test Center

For queries that assume a logged in state, we use sample Test Center user_id = 2 and test_center_id = 1.

- Can create an account

  State: Logged out

  Tags: Add data, Update data

  Query:

  A password hash will be generated via Python when creating an account. After a password has been created, the following query will insert the new user into the database. The user_type = TC is defined by the fact the test center admin clicked to create a test center account.

  ```
  INSERT INTO user(user_email, user_password_h, user_type)
  VALUES
  ("avnatestcenter@example.com",
  "a$2b$12$8YxgE7D9rVw5Y9JfCkq9UOtYz4R2yE7m2VqM4zV4WQ8U1lHq9zZJq", "TC")
  ```

  On the back end, invisible to the user, will then query the recently created user account and update the existing test_center record with the newly created user_id foreign key.

  ```
  UPDATE test_center
  SET user_id = (SELECT user_id FROM user WHERE user_email =
  'avnatestcenter@example.com' AND user_type = 'TC');
  ```

- Can log in

  State: Logged out

  Tags: Retrieve data

  Query:

  Retrieves the correct (hashed) password for the entered email and evaluates whether the user should be logged in

  ```
  SELECT user_password_h
  FROM user
  WHERE user_email = "avnatestcenter@example.com";
  ```

- Can view posted availabilities

  State: Logged in

  Tags: Retrieve data

  Query:

  We have a view "test_centers_with_availability" (click here to view), so we can simply query on the view to show that test center's availability, filtering down by test_center_id. The following information is available:

  From test center availability: date of availability, start time, end time, duration (in minutes), total capacity, capacity left (which can calculate seats left).

  ```
  SELECT
          date_of_availability,
          start_time_slot,
          end_time_slot,
  ```

```
    CONCAT(slot_duration, " minutes") AS duration,
    seat_capacity, (seat_capacity - scheduled_count) AS seats_left
    FROM test_centers_with_availability
WHERE test_center_id = 1;
```

- Can add new availabilities

  State: Logged In
  Tags: Add data
  Query:

  Test centers can upload a CSV that will be parsed and upload the availability slots into the database. The field scheduled_count defaults to 0 during insert, so it is not needed during the insert.

  ```
  INSERT INTO test_center_availability(test_center_id,
  date_of_availability, start_time_slot, end_time_slot, seat_capacity)
  VALUES
  (1, '2026-01-03', '16:00:00', '17:30:00', 2);
  ```

- Can delete availabilities

  State: Logged in
  Tags: Delete data
  Query:

  Test centers can opt to remove individual availability slots if their test center must close on a day that was previously scheduled to be available. Availability slots with no scheduled appointments will simply delete the slot record from the database. Availability with scheduled appointments will release a **before delete** trigger that cancels all of the associated appointments and decouples the appointment from the availability. Below is the query for deleting the availability. The related trigger details can be found here.

  ```
  DELETE FROM test_center_availability
  WHERE availability_slot_id = 49;
  ```

- Can view contract details with Exam Booker

  State: Logged in
  Tags: Retrieve data
  Query:

  Test centers can monitor the status of their contract with Exam Booker. Contract details are added to the database on the back end by an Exam Booker admin. This will show the start/end date of the contract, as well as the status of the contract and the rate per seat.

  ```
  SELECT * FROM test_center_contract WHERE test_center_id = 1;
  ```

Exam Sponsor

For queries that assume a logged in state, we use sample Exam Sponsor user_id = 56 and exam_sponsor_id = 1.

- ### Can create an account

  State: Logged out

  Tags: Add data, Update data

  Query:

  A password hash will be generated via Python when creating an account. After a password has been created, the following query will insert the new user into the database. The user_type = ES is defined by the fact the exam sponsor admin clicked to create an exam sponsor account.

  ```
  INSERT INTO user(user_email, user_password_h, user_type)
  VALUES
  ("pmi@example.com", "5f4dcc3b5aa765d61d8327deb882cf99", "ES")
  ```

  On the back end, invisible to the user, will then query the recently created user account and update the existing exam_sponsor record with the newly created user_id foreign key.

  ```
  UPDATE exam_sponsor
  SET user_id = (SELECT user_id FROM user WHERE user_email = 'pmi@example.com'
  AND user_type = 'ES');
  ```

- ### Can log in

  State: Logged out

  Tags: Retrieve data

  Query:

  Retrieves the correct (hashed) password for the entered email and evaluates whether the user should be logged in

  ```
  SELECT user_password_h
  FROM user
  WHERE user_email = "pmi@example.com";
  ```

- ### Can view posted exam types

  State: Logged in

  Tags: Retrieve data

  Query:

  Test centers can see all of the information about the exams that they offer, as well as a total count of how many have registered, and how many test takers have successfully scheduled thus far. This omits all appointments that were made but then subsequently cancelled.

  ```
  SELECT e.*, exam_regs.reg_count, exam_schedule.schedule_count
  FROM exam e
  LEFT JOIN (
     SELECT exam_id, COUNT(*) AS reg_count
  ```

```
        FROM registered_test_takers r
        WHERE exam_sponsor_id = 1
        GROUP BY exam_id) exam_regs
    ON e.exam_id = exam_regs.exam_id
    LEFT JOIN (
        SELECT exam_id, COUNT(*) AS schedule_count
        FROM scheduled_test_takers s
        WHERE exam_sponsor_id = 1 AND appointment_status = 'Scheduled'
        GROUP BY exam_id) exam_schedule
    ON e.exam_id = exam_schedule.exam_id
    WHERE exam_sponsor_id = 1;
```

- Can add a new exam to register for and schedule
  State: Logged in
  Tags: Add data
  Query:

  Exam sponsors can choose to host new exams that will be administered by Exam Booker's partnered test centers.

```
INSERT INTO exam(exam_sponsor_id, exam_name, exam_duration, domain)
VALUES
(1, "Risk Management Professional", 150, "Risk Management");
```

- Can view contract details with Exam Booker
  State: Logged in
  Tags: Retrieve data
  Query:

  Exam Sponsors can monitor the status of their contract with Exam Booker. This will show the start/end date of the contract, the status of the contract, the rate per test taker, and the minimum number of test takers that the sponsor has committed to scheduling with Exam Booker.

```
SELECT * FROM sponsor_contract WHERE exam_sponsor_id = 1;
```

Advanced SQL

Triggers

Acquire Availability - AFTER INSERT- updates the scheduled count on the test center availability slot that a test taker has just scheduled for, by incrementing it by one.

```
CREATE TRIGGER acquire_availability AFTER INSERT ON appointment
FOR EACH ROW BEGIN
   UPDATE test_center_availability av
   SET av.scheduled_count = av.scheduled_count + 1
   WHERE av.availability_slot_id = NEW.availability_slot_id;
END
```

Cancel Appointment on Delete - BEFORE DELETE - before deleting a test center availability slot, all test takers that are scheduled for that slot have their exam appointments marked as “Cancelled” and their appointments are decoupled from the availability slot.

```
CREATE TRIGGER cancel_appointments_on_slot_delete BEFORE DELETE ON
test_center_availability
FOR EACH ROW BEGIN
   SET @deleting_availability := 1;
   UPDATE appointment ap
   SET ap.appointment_status = 'Cancelled'
   WHERE ap.availability_slot_id = OLD.availability_slot_id
       AND ap.appointment_status = 'Scheduled';
   SET @deleting_availability := NULL;
END
```

Release Availability - AFTER UPDATE - After a test taker cancels their appointment, it reduces the scheduled count on the related test center availability slot by one.

```
CREATE TRIGGER release_availability AFTER UPDATE ON appointment
FOR EACH ROW BEGIN
   IF COALESCE(@deleting_availability, 0) = 0
       AND OLD.appointment_status = 'Scheduled'
       AND NEW.appointment_status = 'Cancelled'
   THEN
       UPDATE test_center_availability av
       SET av.scheduled_count = av.scheduled_count - 1
       WHERE av.availability_slot_id = NEW.availability_slot_id;
   END IF;
END
```

Check Statements

User Type Validation - making sure only the 3 acceptable user types are entered into the DB

```
CREATE TABLE user (
   user_id int(11) NOT NULL AUTO_INCREMENT,
   user_email varchar(45),
   user_password_h varchar(255),
   user_type char(2) CHECK (user_type in ('TC','TT','ES')),
   PRIMARY KEY (user_id)
)
```

Exam Duration Validation - making sure exam durations are positive numbers only

```
CREATE TABLE exam (
       exam_id int(11) NOT NULL AUTO_INCREMENT,
       exam_sponsor_id int(11),
       exam_name varchar(100) NOT NULL,
       exam_duration int(11) NOT NULL CHECK (exam_duration > 0),
       domain varchar(100),
   PRIMARY KEY (exam_id),
   FOREIGN KEY (exam_sponsor_id) REFERENCES exam_sponsor (exam_sponsor_id)
)
```

Capacity and Scheduled Validation - making sure capacity and scheduled count are positive numbers, and also making sure that scheduled count never exceeds capacity

```
CREATE TABLE test_center_availability (
       availability_slot_id int(11) NOT NULL AUTO_INCREMENT,
       test_center_id int(11) ,
       date_of_availability date ,
       start_time_slot time ,
       end_time_slot time ,
       seat_capacity int(11) NOT NULL
             CHECK (seat_capacity >= 0),
       scheduled_count int(11) NOT NULL DEFAULT 0
             CHECK (scheduled_count >= 0 and scheduled_count <= seat_capacity),
   PRIMARY KEY (availability_slot_id),
   FOREIGN KEY (test_center_id) REFERENCES test_center (test_center_id)
)
```

Appendix:

We removed the “Exam Provider” entity, because our fictitious company, Exam Booker, itself is an exam provider, and would therefore not have an entity called exam provider. We also removed the seat_assignment entity when we realized it was causing an extra unnecessary entity traversal. We also added the User entity to decouple the exam registration process from the account creation process.

Below are the updated schema statements after making the above changes:

user(user_id, user_email, user_password_h, user_type)

test_taker (test_taker_id, first_name, last_name, phone_number, street, city, state_address,country, zip_code, @user_id)

exam_registration (exam_registration_id, @exam_id, @test_taker_id, invoice_number, registration_date, amount_paid)

appointment (appointment_id, @exam_registration_id, accomodations, appointment_status, seat_number, @availability_slot_id)

test_center_availability (availability_slot_id, @test_center_id, date_of_availability, start_time_slot, end_time_slot, seat_capacity, scheduled_count)

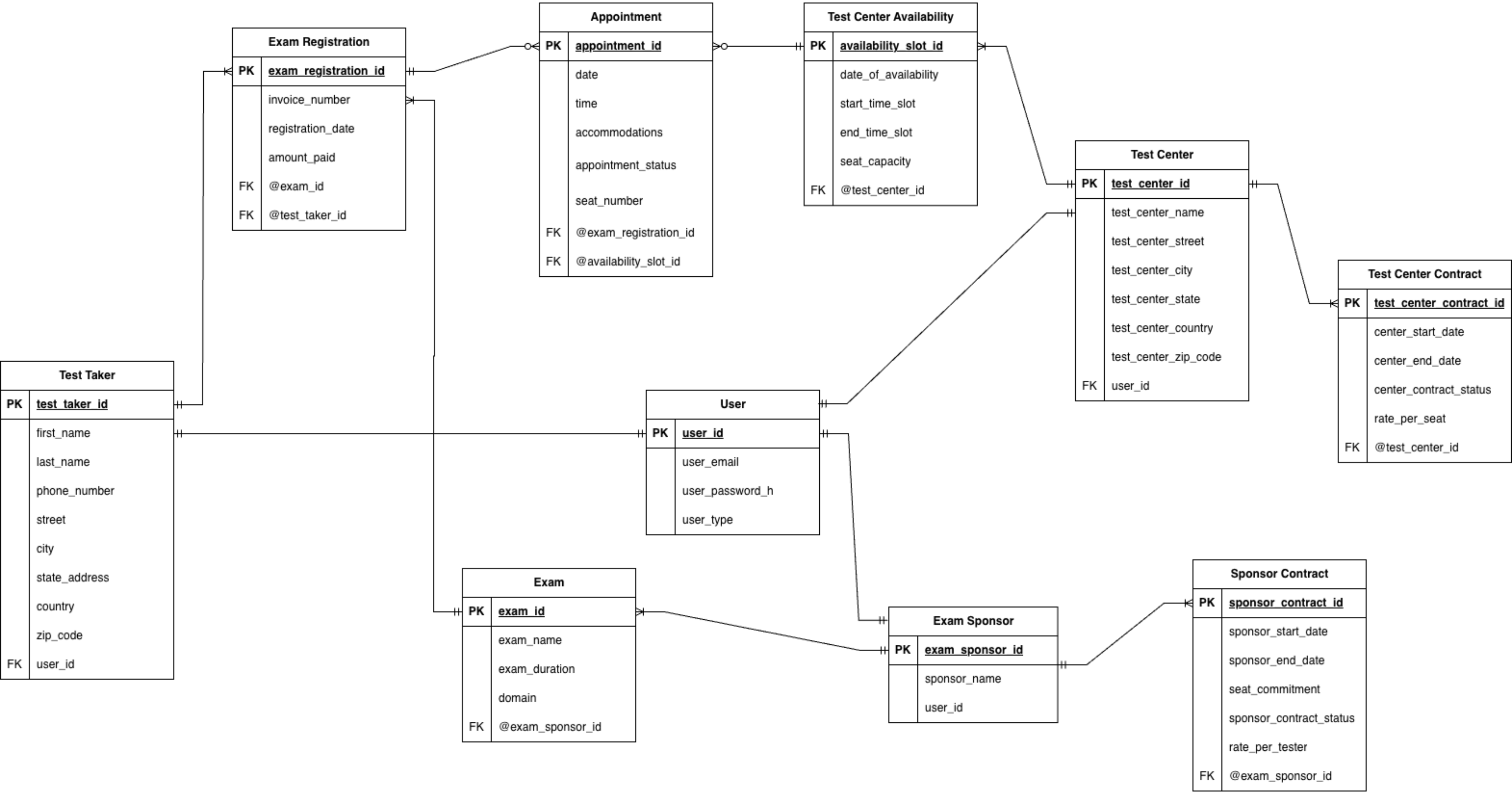
test_center (test_center_id, test_center_name, test_center_street, test_center_city, test_center_state, test_center_country, test_center_zip_code, @user_id)

test_center_contract (test_center_contract_id, @test_center_id, center_start_date, center_end_date, center_contract_status, rate_per_seat)

exam_sponsor (exam_sponsor_id, sponsor_name)

sponsor_contract (sponsor_contract_id, @exam_sponsor_id, sponsor_start_date, sponsor_end_date, seat_commitment, sponsor_contract_status, rate_per_tester)

exam (exam_id, @exam_sponsor_id, exam_name, exam_duration, domain)



	Exam Registration
PK	exam_registration_id
	invoice_number
	registration_date
	amount_paid
FK	@exam_id
FK	@test_taker_id

	Appointment
PK	appointment_id
	date
	time
	accommodations
	appointment_status
	seat_number
FK	@exam_registration_id
FK	@availability_slot_id

	Test Center Availability
PK	availability_slot_id
	date_of_availability
	start_time_slot
	end_time_slot
	seat_capacity
FK	@test_center_id

	Test Center
PK	test_center_id
	test_center_name
	test_center_street
	test_center_city
	test_center_state
	test_center_country
	test_center_zip_code
FK	user_id

	Test Center Contract
PK	test_center_contract_id
	center_start_date
	center_end_date
	center_contract_status
	rate_per_seat
FK	@test_center_id

	Test Taker
PK	test_taker_id
	first_name
	last_name
	phone_number
	street
	city
	state_address
	country
	zip_code
FK	user_id

	User
PK	user_id
	user_email
	user_password_h
	user_type

	Exam
PK	exam_id
	exam_name
	exam_duration
	domain
FK	@exam_sponsor_id

	Exam Sponsor
PK	exam_sponsor_id
	sponsor_name
	user_id

	Sponsor Contract
PK	sponsor_contract_id
	sponsor_start_date
	sponsor_end_date
	seat_commitment
	sponsor_contract_status
	rate_per_tester
FK	@exam_sponsor_id