Mental Health Disorders and Prescription and Appointment Automation with SQL

by Anna Ryan, Camille Dupuy, Casey Benzing, Zachariah Rasheed
Northeastern University, Boston, MA, USA

Abstract

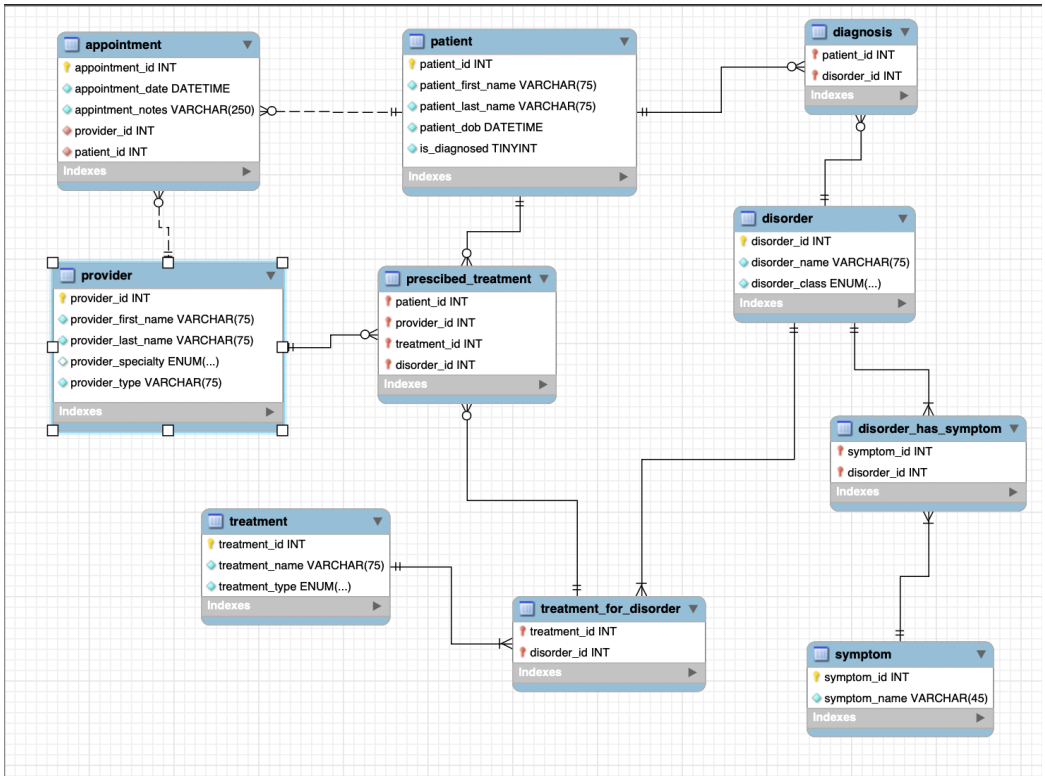
Our project works to create a database application to automate and provide a solution to screening for mental health disorders and future appointment scheduling. Our application creates appointments and adds screening results into appointment notes to prepare providers before appointments.

Introduction

Mental health disorders can be incredibly difficult for licensed professionals to diagnose due to a variety of challenges, and we wanted to create a solution to this. Our goal was that the application could be utilized by professionals in the mental health specialty to organize appointments with patients and have the patients screen for a variety of mental health disorders before meeting, to aid in the diagnosis of very complex disorders. Additionally, by creating appointments automatically, this is conducive for the flow of the practice's workplace and removes human error. This provides mental health practices with a network of records that actively are implemented to inform the provider's diagnosis and prescription of treatment.

Database Design

(see below)



The first and most important entity of any medical practice is the patient being cared for and such, this is the first entity in our database. Our **patient** entity contains an ID, full name, date of birth, and their diagnosis status. Next, we created the **provider** entity to encapsulate information about the mental health professional providing care to patients. This contains an ID, full name, their specialty, and the type of professional they are. This works to provide a comprehensive understanding for filtering and for patients when receiving care. Then, we have the **disorder** table which has an ID, the name of the disorder, and the class of the disorder. By categorizing disorders, the database can sort easily into subsections for querying and automating. **Treatment** is an entity that serves as a home for an ID, the name of the treatment, and the treatment type. Again, by classifying into a type, narrowing down a diagnosis and organizing the database is much easier. Additionally, the **symptom** table holds an ID and the name of the symptom. Our first joined entity is the diagnosis which relates to the patient and disorder entities by their ID primary keys in two one to many relationships, where a diagnosis can be made for many to no patients and many to no disorders. Next, we created the **disorder_has_symptoms** joined entity to have symptom ID and disorder ID as foreign keys, where one disorder can have many symptoms and a symptom can be present in many disorders. Also, the joined entity **treatment_for_disorder** has treatment ID and disorder ID for foreign keys where there are many treatments for one disorder, one treatment for a prescribed_treatment (more on that next), and many treatments for many treatments. The **prescribed_treatment** table has patient ID, provider ID, treatment ID and disorder ID, where there can be many to no prescribed treatments

for a patient, many to no prescribed by a provider, and many or no prescribed treatments for many disorders through the treatment_for_disorder joined table. Finally, the **appointment** entity is made for scheduling and automation purposes, containing the appointment ID, the date, notes, and foreign keys of provider ID and patient ID. This is to inform both sides of users and organize for appointments. Many to no appointments can be made for a patient, and similarly, many to no appointments can be scheduled with a specific provider.

Data Sources and Methods

Since we made a database application, all data loaded in is of our own creation that could be replicated on a larger scale with real data. Our data was created with functionality in mind, to try and test cases that did not challenge the application, and records that would be flagged by checks or edge cases. We assumed there was an even distribution of disorders, provider specialties,

User Cases

Question: What disorders might a patient with the symptom: “Loss of Interest” have?

Query:

```
select
    disorder_name
from disorder dis
join disorder_has_symptom dhs on dis.disorder_id = dhs.disorder_id
join symptom s on dhs.symptom_id = s.symptom_id
where s.symptom_name = "Loss of Interest";
```

Output:

	disorder_name
▶	Major Depressive Disorder
	Post-Traumatic Stress Disorder

Question: What class of disorders tends to cause the symptom "Change in sleep"?

Query:

```
select
    disorder_class, count(*) as num_disorders
from disorder dis
join disorder_has_symptom dhs on dis.disorder_id = dhs.disorder_id
join symptom s on dhs.symptom_id = s.symptom_id
where s.symptom_name = "Change in sleep"
group by disorder_class;
```

Output:

	disorder_class	num_disorders
▶	Mood	2

Question: If someone had generalized anxiety disorder, what provider might be useful to them?

Query:

```
select
    provider_first_name,
    provider_last_name,
    provider_specialty,
    provider_type
from disorder dis join provider prv on dis.disorder_class = prv.provider_specialty
where dis.disorder_name = "Generalized Anxiety Disorder";
```

Output:

	provider_first_name	provider_last_name	provider_specialty	provider_type
▶	Singed	Revik	Anxiety	Therapist

Question: What treatments are available for someone with MDD (Major Depressive Disorder)?

Query:

```
select
    treatment_name,
    treatment_type
from disorder dis
join treatment_for_disorder trdis on dis.disorder_id = trdis.disorder_id
join treatment tr on tr.treatment_id = trdis.treatment_id
where dis.disorder_name = "Major Depressive Disorder";
```

Output:

	treatment_name	treatment_type
▶	SSRI	Pharmaceutical
	Cognitive Behavioral Therapy	Therapy

Question: Is there an association between disorder class and treatment type?

Query:

```
select
    disorder_class,
    treatment_type,
    count(*) as "associations"
from disorder dis
join treatment_for_disorder trdis on dis.disorder_id = trdis.disorder_id
join treatment tr on tr.treatment_id = trdis.treatment_id
group by disorder_class, treatment_type;
```

Output:

	disorder_class	treatment_type	associations
▶	Mood	Pharmaceutical	2
	Anxiety	Therapy	3
	Mood	Therapy	1
	Trauma	Therapy	2

Conclusions

Our project was successful in many ways in terms of our original ambitions to create a solution and provide an aide to mental healthcare providers. By using a stored procedure and a trigger,

our group was able to automate the scheduling of appointments and the flagging of possible disorders through a screening process. This solidified our application as tangibly useful for professionals to organize their practice and gain more insight on patients before seeing them to diagnose. However, a limitation of our database is the lack of handling of a feature such as prescription records by providers. There is no automation for this process, which is something that could be expanded upon in future work.

Author Contributions

Zachariah Rasheed: Led the effort of making stored procedures and triggers functional for our database. Also contributed to creating the database via create and insert statements.

Casey Benzing: Worked on database design, queries, and the creation of the presentation. Also worked on creating tables.

Camille Dupuy: Headed up the presentation slides, designed the database and wrote queries along with editing.

Anna Ryan: Created tables and data to insert into the database. Contributed to stored procedures and wrote the report for the project.