JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Special Education Student Management

Prepared by

Bogdan Tesileanu
11/22/2020

Table of Contents

1. INTRODUCTION 3

1.1. SCOPE AND PURPOSE OF DOCUMENT 3

1.2. PROJECT OBJECTIVE 3

2. SYSTEM REQUIREMENTS 4

2.1 HARDWARE REQUIREMENTS 4

2.2 SOFTWARE REQUIREMENTS 4

2.3 FUNCTIONAL REQUIREMENTS 4

2.4 DATABASE REQUIREMENTS 4

3. DATABASE DESIGN DESCRIPTION 5

3.1 DESIGN RATIONALE 5

3.2 E/R MODEL 5

3.2.1 Entities 5

3.2.2 Relationships 7

3.2.3 E/R Diagram 8

3.3 RELATIONAL MODEL 9

3.3.1 Data Dictionary 9

3.3.2 Integrity Rules 10

3.3.3 Operational Rules 10

3.3.4 Operations 10

3.4 SECURITY 11

3.5 DATABASE BACKUP AND RECOVERY 11

3.6 USING DATABASE DESIGN OR CASE TOOL 11

3.7 OTHER POSSIBLE E/R RELATIONSHIPS 12

4. IMPLEMENTATION DESCRIPTION 13

4.1 DATA DICTIONARY 13

4.2 ADVANCED FEATURES 17

4.3 QUERIES 19

4.3.1 Current Students 19

4.3.2 Names of Physicians who are affiliated with Memorial Hospital 19

4.3.3 All teachers who have a classroom with a capacity of 15 students 19

4.3.4 All medications that need to be taken with food 19

4.3.5 List of how many physicians work for each hospital 20

4.3.6 Intervention outcome when parents were present. 20

4.3.7 List students with Attention Deficit Disorder 20

4.3.8 List all students whose parent last name is Girland 21

5. CRUD MATRIX 22

5.1 LIST OF ENTITY TYPES 22

5.2 LIST OF FUNCTIONS 22

6. CONCLUDING REMARKS 23

APPENDICES 24

REFERENCES 41

1. Introduction

1.1. Scope and Purpose of Document

Working in the field of exceptional student education can be difficult but rewarding. One of the greater challenges is accurate record keeping, which is crucial in tracking student progress, both academically and behaviorally. Creating a database that takes into account the student's diagnosis, medication, contacts, and any other medical information in addition to the regular course data could increase the data available on the student, thereby allowing teachers and therapists to make more informed decisions.

The first part of this document (namely "Part 2") specifies the system requirements, such as hardware and software, as well as the RDBMS used. Part 3 will describe the database that was implemented. Attributes and relationships will be discussed, as well as any relationships between them. The logic behind choosing each of the tables and attributes will be discussed, as well as the possible users and operations they may perform.

1.2. Project Objective

This Special Education Student Management Database is designed using MySQL Workbench, a relational database management system in order to aid teachers, therapists, and doctors keep accurate records on their students/clients/patients. It will allow teachers to stay up to date with any new medication or diagnoses the students may have received, and it will allow the therapist to maintain a constant eye on the student's performance in the classroom. Using this database will create a unified summary of the student that can be accessed by all the important players in the student's life, in order to better serve the student in succeeding in both at school and at home.

2. System Requirements

2.1 Hardware Requirements

- Processor: Minimum 1 GHz; Recommended 2GHz or more.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more.
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above

2.2 Software Requirements

- OS: Windows 8.1 64 Bit, Windows 8 64 Bit, Windows 7 64 Bit Service Pack 1, Windows Vista 64 Bit Service Pack 2
- OS: MacOS Catalina (10.11.2) or later
- MySQL Workbench (Version: 8.0.22) or later

2.3 Functional Requirements

The database will be used by three main users: school staff, therapists, and doctors. School staff, mostly teachers, will be able to create/update/delete student records, class grades, class schedules, and parent contacts. On the other hand, therapists will be able to alter any kind psychological diagnosis, as well as any interventions performed. Finally, doctors will be able to add/modify/delete their own medical diagnoses, in addition to any medications prescribed to the student. All three users will have access to view all the student information.

2.4 Database Requirements

- MySQL Workbench (Version: 8.0.22) or later
- MySQL Community Server (Version 8.0.22) or later

3. Database Design Description

3.1 Design Rationale

Several factors were considered when building the ER design. The goal of the database is to have a student-centered design that can quickly updated when necessary. Thus, it was decided that multiple foreign keys would exist in the student table, referencing a student's medication, support team, diagnosis, and intervention. Indexes to these other tables were also created in order to increase the efficiency.

The support team is the main relationship between student and the several people that take part in that student's education. Each support team is made up of a primary parent, a therapist, a teacher, and a physician. Each support team can be made up of only one of each, and some teams may not even need all of them, depending on the student's needs.

Each teacher has one classroom only, but a classroom may be used by more than one teacher, or none at all. Each student will have one diagnosis, and each student will have one diagnostic id. Because the diagnosis table includes the diagnosis date, it will be individualized for each student, thus making it a 1:1 relationship.

Each medication can have a schedule that describes how many times it is taken per day, whether it is taken with food or not, and whether a child has to fast before taking the medication. Because these conditions can be generalized, it can apply to several medications, thus making it a many to one relationship. Additionally, some medication may only have to be taken as necessary, so not all require a medication schedule.

Finally, the intervention record describes any interventions performed with the student, and each one has a specific type: whether it occurred with the parent, teacher, or physician present. There is only a specified number of intervention types, thus each intervention record can only have one intervention type. The intervention type has numbers 1-7 as their id, one for each combination of teacher, parent, and physician. The intervention type has the only id that is not auto-incremented, because the number of records will not change.

3.2 E/R Model

3.2.1 Entities

Student: The student entity contains the student it, name, birthday, support team, diagnosis, intervention, and medication. A student must have all of the above to be accepted as a student into the program (and into the database). The student id auto-increments so as to avoid repeats and maintain uniqueness.

Medication: The entity contains the medication id, name, amount, and schedule of when and how it should be taken, which is a foreign key to the medication_schedule entity. The medication id auto-increments so that any medication that is added gets its own id. The medication name is the name of the drug (aspirin, acetaminophen, etc.). The amount is a varchar so that specific instructions can be written such as: 50 mg, 1 pill, 2 tablets.

Medication_schedule: The medication schedule contains its id, how many times it should be taken, whether or not it should be taken with food, and whether or not it should be taken while fasting. The last 3 attributes should be designated with a 1 for "Yes" and a 0 for "No". A medication may not necessarily have a schedule, as it may be a supplement or taken as needed.

Diagnosis: The diagnosis contains its id, which is auto-incremented so that multiple diagnoses can be added as needed. The ICD_10 is a 10 character code that the medical community uses for each diagnosis. The diagnosis name is the name of the ICD_10 diagnosis. The diagnosis_date is the date on which the student was diagnosed with that specific disease.

Intervention_record: The intervention_record entity documents the number and types of interventions that a student has had. An intervention is an ad-hoc meeting to address a time-critical behavioral concern. The outcome of the intervention is documented, as is the time. The type describes who attended the intervention, described below, and is a foreign key.

Intervention_type: The intervention type has 7 options, each with its own type id. It describes who attended an intervention. A 1 for attended and a 0 for not attended. The type is the primary key and is linked to the foreign key in the intervention_record.

Support_team: The support team is the backbone of the support a student received while enrolled in the program. It is made up of 4 players, only one of which is mandatory, the parent. Each of the players, parent, physician, therapist, and teacher is a foreign key to their respective tables. Since support teams can be added and updated frequently, an auto-incrementor has been placed on the primary key.

Primary_parent: The primary parent is the parent that should be contacted in case of emergency. Each parent has an auto-incrementing id, name, phone number, and optional address for correspondence.

Physician contact: The physician responsible for the student's medication and diagnosis. The physician's name, cell phone number, and hospital affiliation (optional) are all attributes.

Primary_therapist: Similarly, each student has a therapist on their support team. Each therapist has an auto-incrementing id, name, cell phone number, and if they wish to provide them, an office number and certificate number. The certificate number is a 9 character code that is issued on their certificate once they earn their degree as a therapist. It can be used for verification purposes.

Teacher: Each student will also have a teacher on their support team. This will be the teacher that teaches them all their subject and who they spend most of their day with. The teacher's name, cell phone number, and classroom id will be entered. The classroom id references which classroom they teach in.

Classroom: Each classroom at the school will have a classroom id, a classroom phone or extension up to 10 characters, and classroom number posted on the outside of the door so that visitors can identify it, and a maximum capacity. Maximum capacity is an integer value that can be updated for COVID concerns.

3.2.2 Relationships

Student – Diagnosis: This is a 1:1 relationship. Each student has one diagnosis and each diagnosis belongs to one student only. This is caused by the date field in the diagnosis. If the field did not exist, one diagnosis may apply to multiple students.

Student – Medication: This is a many to many relationship. Each student can take multiple medications, and one medication can be taken by multiple students. Because of the nature of the program the students are enrolled in, the medications of students are likely to repeat. This is an identifying relationship has an intermediary entity Student_has_medication had to be created, but would not be had it been a 1:M relationship.

Medication – Medication_schedule: This is a 1:M relationship. Each medication can only have one schedule, but the same schedule can apply to multiple medications. The characteristics of the method of intake for each will not change among patients.

Student – Intervention_record: This is a 1:M relationship. Each student can have one intervention record, but the same intervention record can apply to multiple students. Usually, behavioral problems occur between two students, thus an intervention will be had with both.

Intervention_record – Intervention_type: Similarly, each intervention record can only have one type (out of 7), but one of the seven types can belong to multiple interventions. The intervention_type describes the people present during the intervention, and this will slightly differ for each intervention, but will eventually repeat.

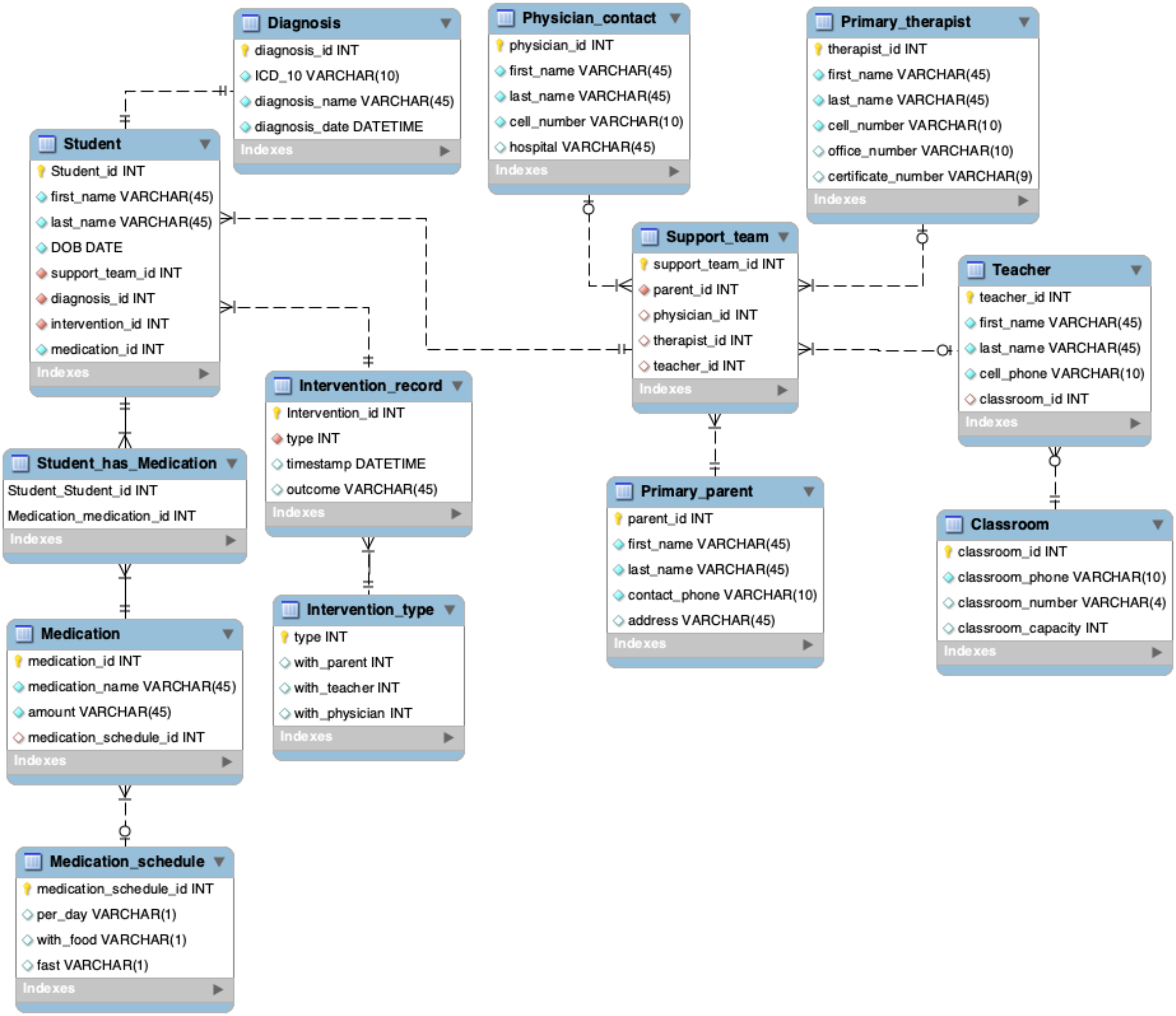
Student – Support_team: This is a 1:M relationship because each student will have one support team, but the same support team can support multiple students. In the case of siblings, the support team will more than likely involve the same parent, physician, therapist, and teacher, and thus some may repeat.

Support_team – Primary_parent: This is a required 1:M relationship as a parent (or authorized adult) must be part of a student's team. Each support team will have one parent, but again, in the case of siblings, one parent can be part of multiple support teams.

Support_team – Primary_therapist, physician_contact, teacher: These are all 0,1:M relationships. While a support team must have a parent, it may have a physician, teacher, or therapist if necessary. Furthermore, each of these players will belong to multiple support teams as all of them, especially teachers, will share some of the students.

Teacher – Classroom: This is a 1:0,M relationship. Each teacher has only one classroom to use and teach in. However, a classroom may be shared by many teachers, or it may be a brand new classroom that does not have nay teachers assigned to it yet.

3.2.3 E/R Diagram



3.3 Relational Model

3.3.1 Data Dictionary

Name	Description	Type	Size	Constraint	Null	Valid Values
Student_id	Student identification	INT		Primary Key	No	All positive integers
first_name	Student first name	VARC	45	-	No	Letters A-Z
last_name	Student last name	VARC	45	-	No	Letters A-Z
DOB	Student date of birth	DATE		-	No	YYYY-MM-DD
medication_id	Student's medication	INT		Foreign Key	No	All positive integers
support_team_id	Student's support team	INT		Foreign Key	No	All positive integers
Diagnosis_id	Student medical diagnosis	INT		Foreign Key	No	All positive integers
intervention_id	Intervention information	INT		Foreign Key	No	All positive Integers
medication_id	Medication identification	INT		Primary Key	No	All positive Integers
medication_name	Name of drug/medication	VARC	45	-	No	Letters/Numbers
amount	Medication dose	VARC	45	-	No	Letters/Numbers
medication_schedule_id	Identification of med sched	INT		Foreign Key	Yes	All positive integers
medication_schedule_id	Identification of med sched	INT		Primary Key	No	All positive integers
per_day	Times taken per day	INT			Yes	All positive integers
with_food	Meds taken with/out food	INT			Yes	1 – yes, 0 - no
fast	Student must fast or not	INT			Yes	1 – yes, 0 - no
diagnosis_id	Diagnosis unique identifier	INT		Primary Key	No	All positive Integers
ICD_10	Medical reference code	VARC	10		No	Letters/Numbers
diagnosis_name	Name of diagnosis	VARC	45		No	Letters/Numbers
diagnosis_date	Date of original diagnosis	DATE			Yes	Datetime timestamp
intervention_id	Intervention identifier	INT		Primary Key	No	All positive integers
type	Type of intervention	INT		Foreign Key	No	Description of inter.
timestamp	Date/time of intervention	DATE			Yes	Datetime timestamp
outcome	Outcome of intervention	VARC	45		Yes	Letters/Numbers
Type	Type of intervention	INT		Primary Key	No	All positive integers
with_parent	Parent involved in intervent	INT			Yes	1 – yes, 0 - no
with_teacher	teacher involved in interven	INT			Yes	1 – yes, 0 - no
with_physician	physician involved in interve	INT			Yes	1 – yes, 0 - no
contact_phone	Primary phone number	VARC	45		No	Numbers/dashes
address	Primary address of person	VARC	45		Yes	Letters/Numbers
hospital	Hospital affiliation	VARC	45		Yes	Letters/Number
cell_number	Cell phone number	VARC	10		No	Numbers
office_number	Office phone number	VARC	10		Yes	Numbers
certificate_number	Number of certification	VARC	9		Yes	Letters/Numbers
cell_phone	Teacher cell phone number	VARC	10		No	Numbers
classroom_id	Classroom identification #	INT		Primary Key	No	Numbers
classroom_phone	Classroom direct phone num	VARC	10		No	Numbers
classroom_number	Classroom door number	VARC	4		Yes	Numbers/Letters
classroom_capacity	Maximum capacity of room	INT			Yes	Number

3.3.2 Integrity Rules

In order to avoid any issues with the primary keys, all but one are auto_incrementing. The one that is not auto-incrementing is a table for reference that is not to be updated. The student table references the support team, diagnosis, intervention, and medication tables through their foreign keys. The intervention record table mandatory attribute “type” references the intervention type table through a foreign key. The support_team table is purely created through the use of primary and foreign keys, each referencing the appropriate player in the support system. Finally, the teacher table classroom_id references the classroom table to access data about the classroom.

All mandatory fields not filled in will throw an error upon populating the data alerting the user that there is missing data. For instance, a parent’s address is not mandatory, but their contact phone is, and thus if a parent is to be entered into the database, they are required to provide a valid phone number. Additionally, all phone numbers have a limit of 10 digits, thus they cannot be out of country phones. Lastly, the certificate number for a therapist is limited to 9 characters, and thus only the specific certificate required by the program is accepted in to the database.

3.3.3 Operational Rules

Several operations have embedded constraints. For instance, the system will not allow a user to create or update the intervention record of a student if they did not specify who was present at the intervention (type). Likewise, when enrolling in the program and being entered into a database, a parent or legal guardian must be put down in the support team table.

Because of the nature of the database, and the multitude of people that are entered, duplicate first and last names are allowed. Similarly, so are phone numbers and addresses. This is solved by creating unique primary keys for all persons in the tables in order to differentiate them.

Lastly, the system will not allow a child to be entered into out system if that child does not have a diagnosis. This is to prevent a multitude of problems, such as not knowing how to address behavior concerns. All students must have an up to date diagnosis, which means testing must be done prior to entering the program and being entered into the database.

3.3.4 Operations

Generally, enrolling a student into the program will require an insert for the student and all the subsequent fields. That means the student’s medication, support (parent at least), diagnosis, and intervention record will need to be created.

If a teacher is added to the program, then all the student’s records will need to be updated. Additionally, a new classroom may need to be inserted for her, or at least her classroom field will need to be updated with an existing classroom.

Any creation of a new teacher, physician, or therapist will automatically cause an update to the student table. This may happen if any one of these players passes away or if the student changes doctors or therapist.

If a student's symptoms get worse, several things will happen. An intervention will need to be held and thus the student's intervention record will be inserted with new data. Second, the student's medication will need to be updated to compensate for the change in behavior, as well as the student's medication amount.

3.4 Security

One of the way this system can be made more secure is through authentication. Each user, whether its teacher, therapist, or doctor, can have their own login information that will allow them to update the database accordingly. Access control can also be used to make sure users can only access and modify their own areas of expertise while being able to only view the rest. Users should also be unable to change any records that are not their own, even if their roles are the same (i.e. teachers should not be able to change a student's grade in a different class than their own). Lastly, encryption can be used to increase the application security. This can prevent common threats, especially SQL injections that may can unauthorized access to the database. This is crucial because of the nature of the database, which contains sensitive medical information and should be HIPPA compliant.

3.5 Database Backup and Recovery

A backup plan is crucial when dealing with data, especially when data is sensitive. To facilitate a proper database, several measures will be taken to ensure that no data is lost. First, the OS and RDBMS software should be backed up and any patches and updates performed. Database backups should have both logical and physical backups, in the cloud and also preferably on an external drive. Backlogs and automatic backups should be instituted in order to maintain a recent version of the database, and monitoring of the theses backups should be set up so that any errors are brought to the attention of the DBA.

3.6 Using Database Design or CASE Tool

A CASE tool can be of great help at every step of the design of the database. They are split up into 3 categories: upper, lower, and integrated. Upper CASE tools focus on concepts more than design, while lower CASE tools focus on the design and testing. The best type of CASE tool are integrated, as they do the job of both upper and lower. A well know tool is Oracle Designer, which is a package to help with database design. It has the ability to help with collecting and analyzing data, designing a data model, feasibility analysis, requirements definition, implementing the database, and prototyping.

3.7 Other Possible E/R Relationships

Several other relationships were considered when designing the ERD. The support_team table could be taken out, and each therapist, physician, parent, and teacher could be added to the student table. However, this would unnecessarily increase the side of the student table and may increase redundancy.

Second, an allergies table could be added to account for students' allergies. This could have a trigger that checks their medication before anything is put into the system to ensure that any new medication given to the student is not part of their allergies.

Third, multiple parents could be added. One could be a primary contact, the other could be a secondary contact. Additionally authorized family members could also be added so that they are allowed the same permissions as the parents, such as pick up and drop off, and to change any field in the parent table.

Lastly, the student – intervention record relationship could be change to a M:N relationship. A student may have multiple interventions that would be need to be tracked, and this would create a good way to look back to see who was involved and what the outcome was depending the type of intervention.

4. Implementation Description

4.1 Data Dictionary

Classroom

Field	Type	Null	Key	Default	Extra
classroom_id	int	NO	PRI	NULL	auto_increment
classroom_phone	varchar(10)	NO		NULL	
classroom_number	varchar(4)	YES		NULL	
classroom_capacity	int	YES		NULL	

Diagnosis

Field	Type	Null	Key	Default	Extra
diagnosis_id	int	NO	PRI	NULL	
ICD_10	varchar(10)	NO		NULL	
diagnosis_name	varchar(45)	NO		NULL	
diagnosis_date	datetime	NO		NULL	

Intervention_record

Field	Type	Null	Key	Default	Extra
Intervention_id	int	NO	PRI	NULL	auto_increment
type	int	NO		NULL	
timestamp	datetime	YES		NULL	
outcome	varchar(45)	YES		NULL	

Intervention_Type

Field	Type	Null	Key	Default	Extra
type	int	NO	PRI	NULL	
with_parent	int	YES		NULL	
with_teacher	int	YES		NULL	
with_physician	int	YES		NULL	

Medication

Field	Type	Null	Key	Default	Extra
medication_id	int	NO	PRI	NULL	auto_increment
medication_name	varchar(45)	NO		NULL	
amount	varchar(45)	NO		NULL	
medication_schedule_id	int	YES		NULL	

Medication_schedule

Field	Type	Null	Key	Default	Extra
medication_schedule_id	int	NO	PRI	NULL	auto_increment
per_day	varchar(45)	YES		NULL	
with_food	varchar(45)	YES		NULL	
fast	varchar(45)	YES		NULL	

Physician_contact

Field	Type	Null	Key	Default	Extra
physician_id	int	NO	PRI	NULL	auto_increment
first_name	varchar(45)	NO		NULL	
last_name	varchar(45)	NO		NULL	
cell_number	varchar(10)	NO		NULL	
hospital	varchar(45)	YES		NULL	

Primary_parent

Field	Type	Null	Key	Default	Extra
parent_id	int	NO	PRI	NULL	auto_increment
first_name	varchar(45)	NO		NULL	
last_name	varchar(45)	NO		NULL	
contact_phone	varchar(10)	NO		NULL	
address	varchar(45)	YES		NULL	

Primary_therapist

Field	Type	Null	Key	Default	Extra
therapist_id	int	NO	PRI	NULL	auto_increment
first_name	varchar(45)	NO		NULL	

last_name	varchar(45)	NO		NULL	
cell_number	varchar(10)	NO		NULL	
office_number	varchar(10)	YES		NULL	
certificate_number	varchar(9)	YES		NULL	

Student

Field	Type	Null	Key	Default	Extra
Student_id	int	NO	PRI	NULL	auto_increment
first_name	varchar(45)	NO		NULL	
last_name	varchar(45)	NO		NULL	
DOB	date	NO		NULL	
medication_id	int	NO	MUL	NULL	
support_team_id	int	NO	MUL	NULL	
diagnosis_id	int	NO	MUL	NULL	
Intervention_id	int	NO	MUL	NULL	

Student_has_Medication

Field	Type	Null	Key	Default	Extra
Student_Student_id	int	NO	PRI	NULL	
Medication_medication_id	int	NO	PRI	NULL	

Support_team

Field	Type	Null	Key	Default	Extra
support_team_id	int	NO	PRI	NULL	auto_increment
parent_id	int	YES	MUL	NULL	
physician_id	int	YES	MUL	NULL	
therapist_id	int	YES	MUL	NULL	

teacher_id	int	YES	MUL	NULL	

Teacher

Field	Type	Null	Key	Default	Extra
teacher_id	int	NO	PRI	NULL	auto_increment
first_name	varchar(45)	NO		NULL	
last_name	varchar(45)	NO		NULL	
cell_phone	varchar(10)	NO		NULL	
classroom_id	int	YES	MUL	NULL	

4.2 Advanced Features

I have added two stored procedure to be used with the database. One is 'student_birthday' which fetches a list of students and their birthdays, while the other is called "classroom_min_capacity() which takes an int parameter, and returns a list of all the classroom that can accommodate that number of students.

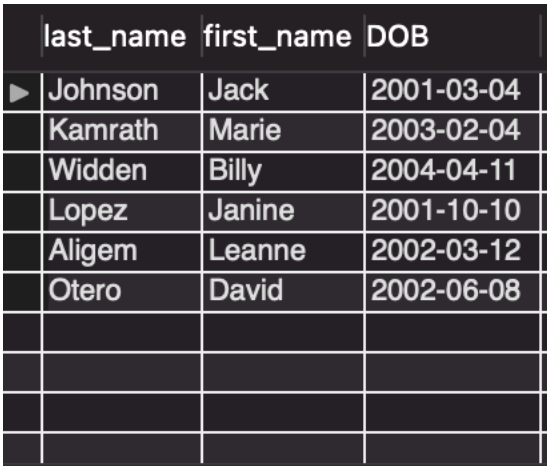
Student_birthday()

```
USE `SESMS`;
DROP procedure IF EXISTS `student_birthday`;

DELIMITER $$
USE `SESMS`$$
CREATE PROCEDURE `student_birthday` ()
BEGIN
        SELECT last_name, first_name, DOB FROM SESMS.Student;
END$$

DELIMITER ;

CALL student_birthday();
```



last_name	first_name	DOB
Johnson	Jack	2001-03-04
Kamrath	Marie	2003-02-04
Widden	Billy	2004-04-11
Lopez	Janine	2001-10-10
Aligem	Leanne	2002-03-12
Otero	David	2002-06-08

Classroom_min_capacity()

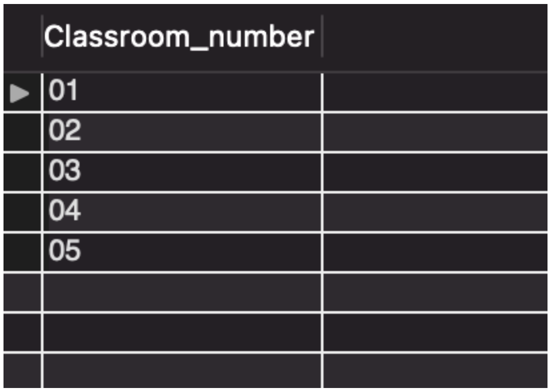
```
USE `SESMS`;
DROP procedure IF EXISTS `Classroom_min_capacity`;

DELIMITER $$
USE `SESMS`$$
CREATE PROCEDURE `Classroom_min_capacity` (min_cap INT)
```

```
BEGIN
        SELECT Classroom_number FROM SESMS.Classroom WHERE
Classroom_capacity > min_cap;
END$$

DELIMITER ;

CALL Classroom_min_capacity(11);
```

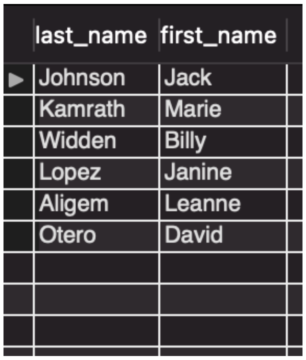


Classroom_number
01
02
03
04
05

4.3 Queries

4.3.1 Current Students

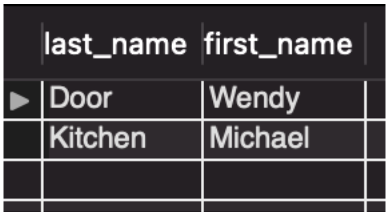
```
SELECT last_name, first_name FROM SESMS.Student;
```



last_name	first_name
Johnson	Jack
Kamrath	Marie
Widden	Billy
Lopez	Janine
Aligem	Leanne
Otero	David

4.3.2 Names of Physicians who are affiliated with Memorial Hospital

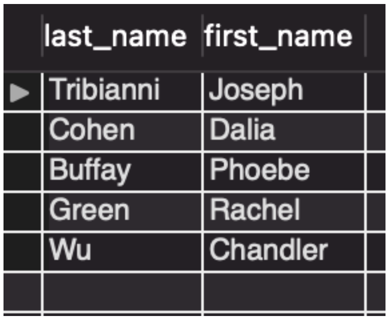
```
SELECT last_name, first_name FROM SESMS.Physician_contact WHERE
hospital = 'miramar' ORDER BY last_name
```



last_name	first_name
Door	Wendy
Kitchen	Michael

4.3.3 All teachers who have a classroom with a capacity of 15 students

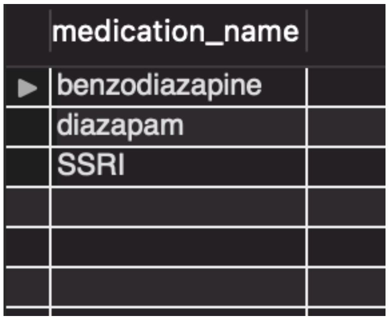
```
SELECT Teacher.last_name, Teacher.first_name FROM SESMS.Teacher JOIN
SESMS.Classroom USING (classroom_id) WHERE classroom_capacity = 15;
```



last_name	first_name
Tribianni	Joseph
Cohen	Dalia
Buffay	Phoebe
Green	Rachel
Wu	Chandler

4.3.4 All medications that need to be taken with food

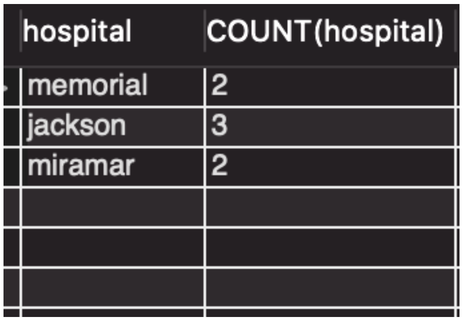
SELECT Medication.medication_name FROM SESMS.Medication JOIN SESMS.Medication_schedule USING (medication_schedule_id) WHERE with_food = 1;



medication_name
benzodiazapine
diazapam
SSRI

4.3.5 List of how many physicians work for each hospital

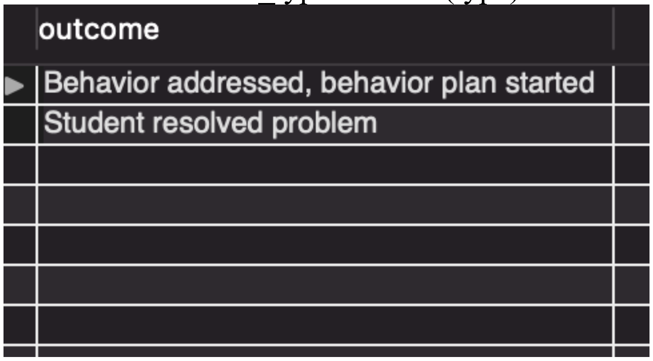
SELECT hospital, COUNT(hospital) FROM SESMS.Physician_contact GROUP BY hospital



hospital	COUNT(hospital)
memorial	2
jackson	3
miramar	2

4.3.6 Intervention outcome when parents were present.

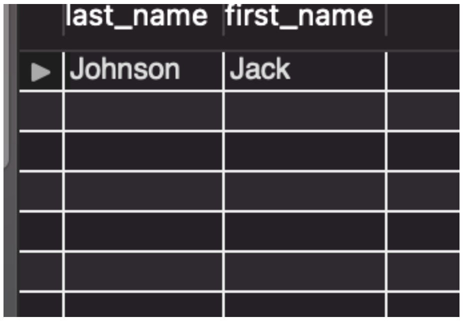
SELECT outcome FROM SESMS.Intervention_record JOIN SESMS.Intevention_type USING (type) WHERE with_parent = 1;



outcome
Behavior addressed, behavior plan started
Student resolved problem

4.3.7 List students with Attention Deficit Disorder

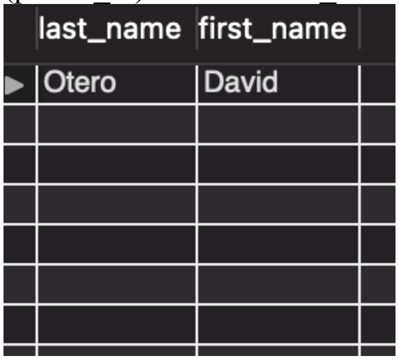
SELECT last_name, first_name FROM SESMS.Student JOIN SESMS.Diagnosis USING (diagnosis_id) WHERE diagnosis_name = 'ADD';



last_name	first_name
Johnson	Jack

4.3.8 List all students whose parent last name is Girland

SELECT last_name, first_name FROM SESMS.Student JOIN SESMS.Support_team USING (support_team_id) WHERE parent_id = (SELECT parent_id FROM SESMS.Support_team JOIN SESMS.Primary_parent USING (parent_id) WHERE last_name = 'Girland')



last_name	first_name
Otero	David

5. CRUD Matrix

5.1 List of Entity Types

E1: Student
E2: Medication
E3: Medication_schedule
E4: Intervention_record
E5: Intervention_type
E6: Diagnosis
E7: Support_team
E8: Physician_contact
E9: Primary_therapist
E10: Teacher
E11: Classroom
E12: Primary_Parent

5.2 List of Functions

F1: Insert/update/delete/retrieve a student
F2: Insert/update/delete/retrieve a medication
F3: Retrieve a medication schedule
F4: Insert/update/delete/retrieve a physician
F5: Insert/update/delete/retrieve a therapist
F6: Insert/update/delete/retrieve a teacher
F7: Insert/update/delete/retrieve a classroom
F8: Insert/update/delete/retrieve a diagnosis
F9: Insert/update/retrieve an intervention_record
F10: Insert/update/delete/retrieve a support team
F11: Insert/update/delete/retrieve a parent

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
F1	CRUD	CRUD		CRUD		CRUD	CRUD					
F2		CRUD	CRUD									
F3			R									
F4							U	CRUD				
F5							U		CRUD			
F6							U			CRUD		
F7										U	CRUD	
F8	CRUD											
F9				CRUD	R							
F10	U						CRUD					
F11							U					CRUD

6. Concluding Remarks

Several lessons were learned during the entire process of the database project. First and foremost, the importance of a well thought out design was highlighted. I have had to run through several iterations of different design schemas because I found that once I started populating and running queries, I would get several error messages. Secondly, the original designs did not allow the intent of the application to come forward. The role of the database is to manage students enrolled in a special education program. The relationships between entities had to be reorganized until a proper profile of each student could be created.

The strengths I've found in this application is the simplicity of running queries on the data. Because the foreign keys are contained within the main tables (student, support team), the logic for the queries is much more straightforward. However, one of the drawbacks or weaknesses of the application is that it does not have much efficiency built in. Earlier versions included triggers for the student table, which allowed the foreign keys to be found much faster, but there were not enough students to warrant the use of them.

In the future, an allergies table would be useful to add to the database so that any kind of interactions with the medication can be caught by the system. For instance, if a student is allergic to amoxicillin, the database would alert if such a medication is added to a student's profile. Furthermore, additional information about behavior in the classroom could be added for each student. For instance, a daily count of the amount of breaks the student received, or the amount of time a student worked quietly or showed an expected behavior could be recorded in order to show progress over time.

Finally, I learned a lot from this project about how to use an RDBMS and that there are a plethora of resources on the internet for any kind of problem one may face. Once I learned how to use the database, the queries were fairly straightforward and I enjoyed the amount of back-end work the program does, in comparison to other languages such as java or python.

Appendices

Appendix A - DDL, INSERT, SELECT Statements

DDL

```
DROP SCHEMA IF EXISTS `SESMS2`;
CREATE SCHEMA IF NOT EXISTS `SESMS2` DEFAULT CHARACTER SET
utf8mb4 COLLATE utf8mb4_0900_ai_ci ;
USE `SESMS2` ;

-- -----------------------------------------------------
-- Table `SESMS2`.`Classroom`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Classroom` (
  `classroom_id` INT NOT NULL AUTO_INCREMENT,
  `classroom_phone` VARCHAR(10) NOT NULL,
  `classroom_number` VARCHAR(4) NULL DEFAULT NULL,
  `classroom_capacity` INT NULL DEFAULT NULL,
  PRIMARY KEY (`classroom_id`))
ENGINE = InnoDB
AUTO_INCREMENT = 11
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `SESMS2`.`Diagnosis`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Diagnosis` (
  `diagnosis_id` INT NOT NULL,
  `ICD_10` VARCHAR(10) NOT NULL,
  `diagnosis_name` VARCHAR(45) NOT NULL,
  `diagnosis_date` DATETIME NOT NULL,
  PRIMARY KEY (`diagnosis_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `SESMS2`.`Intervention_type`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Intervention_type` (
  `type` INT NOT NULL,
  `with_parent` INT NULL DEFAULT NULL,
  `with_teacher` INT NULL DEFAULT NULL,
  `with_physician` INT NULL DEFAULT NULL,
  PRIMARY KEY (`type`))
ENGINE = InnoDB
AUTO_INCREMENT = 16
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `SESMS2`.`Intervention_record`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Intervention_record` (
  `Intervention_id` INT NOT NULL AUTO_INCREMENT,
  `type` INT NOT NULL,
  `timestamp` DATETIME NULL DEFAULT NULL,
  `outcome` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`Intervention_id`),
  INDEX `type_idx` (`type` ASC) VISIBLE,
  CONSTRAINT `type`
    FOREIGN KEY (`type`)
    REFERENCES `SESMS2`.`Intervention_type` (`type`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 7
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `SESMS2`.`Medication_schedule`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Medication_schedule` (
  `medication_schedule_id` INT NOT NULL AUTO_INCREMENT,
  `per_day` VARCHAR(1) NULL DEFAULT NULL,
  `with_food` VARCHAR(1) NULL DEFAULT NULL,
  `fast` VARCHAR(1) NULL DEFAULT NULL,
  PRIMARY KEY (`medication_schedule_id`))
ENGINE = InnoDB
AUTO_INCREMENT = 10
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;
```

```
-- -----------------------------------------------------
-- Table `SESMS2`.`Medication`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Medication` (
  `medication_id` INT NOT NULL AUTO_INCREMENT,
  `medication_name` VARCHAR(45) NOT NULL,
  `amount` VARCHAR(45) NOT NULL,
  `medication_schedule_id` INT NULL DEFAULT NULL,
  PRIMARY KEY (`medication_id`),
  CONSTRAINT `medication_schedule_id`
    FOREIGN KEY (`medication_schedule_id`)
    REFERENCES `SESMS2`.`Medication_schedule` (`medication_schedule_id`))
ENGINE = InnoDB
AUTO_INCREMENT = 6
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `SESMS2`.`Physician_contact`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Physician_contact` (
  `physician_id` INT NOT NULL AUTO_INCREMENT,
  `first_name` VARCHAR(45) NOT NULL,
  `last_name` VARCHAR(45) NOT NULL,
  `cell_number` VARCHAR(10) NOT NULL,
  `hospital` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`physician_id`))
ENGINE = InnoDB
AUTO_INCREMENT = 8
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `SESMS2`.`Primary_parent`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Primary_parent` (
  `parent_id` INT NOT NULL AUTO_INCREMENT,
  `first_name` VARCHAR(45) NOT NULL,
  `last_name` VARCHAR(45) NOT NULL,
  `contact_phone` VARCHAR(10) NOT NULL,
  `address` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`parent_id`))
ENGINE = InnoDB
AUTO_INCREMENT = 9
```

```
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `SESMS2`.`Primary_therapist`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Primary_therapist` (
  `therapist_id` INT NOT NULL AUTO_INCREMENT,
  `first_name` VARCHAR(45) NOT NULL,
  `last_name` VARCHAR(45) NOT NULL,
  `cell_number` VARCHAR(10) NOT NULL,
  `office_number` VARCHAR(10) NULL DEFAULT NULL,
  `certificate_number` VARCHAR(9) NULL DEFAULT NULL,
  PRIMARY KEY (`therapist_id`))
ENGINE = InnoDB
AUTO_INCREMENT = 10
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `SESMS2`.`Teacher`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Teacher` (
  `teacher_id` INT NOT NULL AUTO_INCREMENT,
  `first_name` VARCHAR(45) NOT NULL,
  `last_name` VARCHAR(45) NOT NULL,
  `cell_phone` VARCHAR(10) NOT NULL,
  `classroom_id` INT NULL DEFAULT NULL,
  PRIMARY KEY (`teacher_id`),
  INDEX `classroom_id_idx` (`classroom_id` ASC) VISIBLE,
  CONSTRAINT `classroom_id`
    FOREIGN KEY (`classroom_id`)
    REFERENCES `SESMS2`.`Classroom` (`classroom_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 10
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `SESMS2`.`Support_team`
-- -----------------------------------------------------
```

```
CREATE TABLE IF NOT EXISTS `SESMS2`.`Support_team` (
  `support_team_id` INT NOT NULL AUTO_INCREMENT,
  `parent_id` INT NOT NULL NOT NULL,
  `physician_id` INT NULL NOT NULL,
  `therapist_id` INT NULL NOT NULL,
  `teacher_id` INT NULL NOT NULL,
  PRIMARY KEY (`support_team_id`),
  INDEX `parent_support_idx` (`parent_id` ASC) VISIBLE,
  INDEX `physician_support_idx` (`physician_id` ASC) VISIBLE,
  INDEX `therapist_support_idx` (`therapist_id` ASC) VISIBLE,
  INDEX `teacher_idx` (`teacher_id` ASC) VISIBLE,
  CONSTRAINT `parent_support`
    FOREIGN KEY (`parent_id`)
    REFERENCES `SESMS2`.`Primary_parent` (`parent_id`),
  CONSTRAINT `physician_support`
    FOREIGN KEY (`physician_id`)
    REFERENCES `SESMS2`.`Physician_contact` (`physician_id`),
  CONSTRAINT `teacher`
    FOREIGN KEY (`teacher_id`)
    REFERENCES `SESMS2`.`Teacher` (`teacher_id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `therapist_support`
    FOREIGN KEY (`therapist_id`)
    REFERENCES `SESMS2`.`Primary_therapist` (`therapist_id`))
ENGINE = InnoDB
AUTO_INCREMENT = 9
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `SESMS2`.`Student`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Student` (
  `Student_id` INT NOT NULL AUTO_INCREMENT,
  `first_name` VARCHAR(45) NOT NULL,
  `last_name` VARCHAR(45) NOT NULL,
  `DOB` DATE NOT NULL,
  `support_team_id` INT NOT NULL,
  `diagnosis_id` INT NOT NULL,
  `intervention_id` INT NOT NULL,
  `medication_id` INT NOT NULL,
  PRIMARY KEY (`Student_id`),
  INDEX `idx_id` (`Student_id` ASC) VISIBLE,
  INDEX `support_idx` (`support_team_id` ASC) VISIBLE,
  INDEX `diagnosis_idx` (`diagnosis_id` ASC) VISIBLE,
  INDEX `fk_Student_Intervention_record1_idx` (`intervention_id` ASC) VISIBLE,
  CONSTRAINT `intervention`
    FOREIGN KEY (`intervention_id`)
    REFERENCES `SESMS2`.`Intervention_record` (`Intervention_id`),
  CONSTRAINT `support`
    FOREIGN KEY (`support_team_id`)
    REFERENCES `SESMS2`.`Support_team` (`support_team_id`),
  CONSTRAINT `diagnosis`
    FOREIGN KEY (`diagnosis_id`)
    REFERENCES `SESMS2`.`Diagnosis` (`diagnosis_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 14
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `SESMS2`.`Student_has_Medication`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `SESMS2`.`Student_has_Medication` (
  `Student_Student_id` INT NOT NULL,
  `Medication_medication_id` INT NOT NULL,
  PRIMARY KEY (`Student_Student_id`, `Medication_medication_id`),
  INDEX `fk_Student_has_Medication_Medication1_idx` (`Medication_medication_id`
ASC) VISIBLE,
  INDEX `fk_Student_has_Medication_Student1_idx` (`Student_Student_id` ASC)
VISIBLE,
  CONSTRAINT `fk_Student_has_Medication_Student1`
    FOREIGN KEY (`Student_Student_id`)
    REFERENCES `SESMS2`.`Student` (`Student_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Student_has_Medication_Medication1`
    FOREIGN KEY (`Medication_medication_id`)
    REFERENCES `SESMS2`.`Medication` (`medication_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

USE `SESMS2` ;
```

```
-- -----------------------------------------------------
-- procedure Classroom_min_capacity
-- -----------------------------------------------------

DELIMITER $$
USE `SESMS2`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE
`Classroom_min_capacity`(min_cap INT)
BEGIN
	SELECT Classroom_number FROM SESMS2.Classroom WHERE
Classroom_capacity > min_cap;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure student_birthday
-- -----------------------------------------------------

DELIMITER $$
USE `SESMS2`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `student_birthday`()
BEGIN
	SELECT last_name, first_name, DOB FROM SESMS2.Student;
END$$

DELIMITER ;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

INSERT STATEMENTS

```
INSERT INTO `SESMS2`.`Intervention_type` (`type`, `with_parent`, `with_teacher`,
`with_physician`) VALUES ('1', '0', '0', '1');
INSERT INTO `SESMS2`.`Intervention_type` (`type`, `with_parent`, `with_teacher`,
`with_physician`) VALUES ('2', '0', '1', '0');
INSERT INTO `SESMS2`.`Intervention_type` (`type`, `with_parent`, `with_teacher`,
`with_physician`) VALUES ('3', '1', '0', '0');
INSERT INTO `SESMS2`.`Intervention_type` (`type`, `with_parent`, `with_teacher`,
`with_physician`) VALUES ('4', '0', '1', '1');
INSERT INTO `SESMS2`.`Intervention_type` (`type`, `with_parent`, `with_teacher`,
`with_physician`) VALUES ('5', '1', '0', '1');
INSERT INTO `SESMS2`.`Intervention_type` (`type`, `with_parent`, `with_teacher`,
`with_physician`) VALUES ('6', '1', '1', '0');
```

```
INSERT INTO `SESMS2`.`Intervention_type` (`type`, `with_parent`, `with_teacher`,
`with_physician`) VALUES ('7', '1', '1', '1');

INSERT INTO `SESMS2`.`Intervention_record` (`type`, `timestamp`, `outcome`)
VALUES ('4', '2012-03-04 16:00:00', 'Behavior addressed, behavior plan started');
INSERT INTO `SESMS2`.`Intervention_record` (`type`, `timestamp`, `outcome`)
VALUES ('6', '2013-05-02 15:03:00', 'Student sent home');
INSERT INTO `SESMS2`.`Intervention_record` (`type`, `timestamp`, `outcome`)
VALUES ('3', '2014-07-01 14:00:00', 'Student sent to nurse');
INSERT INTO `SESMS2`.`Intervention_record` (`type`, `timestamp`, `outcome`)
VALUES ('2', '2013-05-06 12:03:13', 'Student resolved problem');
INSERT INTO `SESMS2`.`Intervention_record` (`type`, `timestamp`, `outcome`)
VALUES ('6', '2019-11-11 12:11:10', 'Parents notified');
INSERT INTO `SESMS2`.`Intervention_record` (`type`, `timestamp`, `outcome`)
VALUES ('1', '2013-12-01 20:13:13', 'Student sent home');

INSERT INTO `SESMS2`.`Medication_schedule` (`per_day`, `with_food`, `fast`)
VALUES ('1', '1', '1');
INSERT INTO `SESMS2`.`Medication_schedule` (`per_day`, `with_food`, `fast`)
VALUES ('1', '1', '0');
INSERT INTO `SESMS2`.`Medication_schedule` (`per_day`, `with_food`, `fast`)
VALUES ('1', '0', '1');
INSERT INTO `SESMS2`.`Medication_schedule` (`per_day`, `with_food`, `fast`)
VALUES ('1', '0', '0');
INSERT INTO `SESMS2`.`Medication_schedule` (`per_day`, `with_food`, `fast`)
VALUES ('2', '1', '1');
INSERT INTO `SESMS2`.`Medication_schedule` (`per_day`, `with_food`, `fast`)
VALUES ('2', '0', '1');
INSERT INTO `SESMS2`.`Medication_schedule` (`per_day`, `with_food`, `fast`)
VALUES ('2', '1', '0');
INSERT INTO `SESMS2`.`Medication_schedule` (`per_day`, `with_food`, `fast`)
VALUES ('2', '0', '0');
INSERT INTO `SESMS2`.`Medication_schedule` (`per_day`, `with_food`, `fast`)
VALUES ('3', '1', '1');

INSERT INTO `SESMS2`.`Medication` (`medication_name`, `amount`,
`medication_schedule_id`) VALUES ('propolol', '1', '3');
INSERT INTO `SESMS2`.`Medication` (`medication_name`, `amount`,
`medication_schedule_id`) VALUES ('benzodiazapine', '3', '1');
INSERT INTO `SESMS2`.`Medication` (`medication_name`, `amount`,
`medication_schedule_id`) VALUES ('aspirin', '1', '4');
INSERT INTO `SESMS2`.`Medication` (`medication_name`, `amount`,
`medication_schedule_id`) VALUES ('diazapam', '3', '2');
INSERT INTO `SESMS2`.`Medication` (`medication_name`, `amount`,
`medication_schedule_id`) VALUES ('SSRI', '2', '5');
```

```
INSERT INTO `SESMS2`.`Diagnosis` (`diagnosis_id`, `ICD_10`, `diagnosis_name`,
`diagnosis_date`) VALUES ('1', '1239F8FJO', 'ADD', '2010-05-06');
INSERT INTO `SESMS2`.`Diagnosis` (`diagnosis_id`, `ICD_10`, `diagnosis_name`,
`diagnosis_date`) VALUES ('2', 'O86490P931', 'ADHD', '2005-01-22');
INSERT INTO `SESMS2`.`Diagnosis` (`diagnosis_id`, `ICD_10`, `diagnosis_name`,
`diagnosis_date`) VALUES ('3', '048H38FP09', 'Dyslexia', '2009-05-04');
INSERT INTO `SESMS2`.`Diagnosis` (`diagnosis_id`, `ICD_10`, `diagnosis_name`,
`diagnosis_date`) VALUES ('4', '8492BFHS7U', 'Auditory Processing DIsorder', '2006-
07-01');
INSERT INTO `SESMS2`.`Diagnosis` (`diagnosis_id`, `ICD_10`, `diagnosis_name`,
`diagnosis_date`) VALUES ('5', '19IO0OP0K9', 'Autism', '2001-04-15');
INSERT INTO `SESMS2`.`Diagnosis` (`diagnosis_id`, `ICD_10`, `diagnosis_name`,
`diagnosis_date`) VALUES ('6', '8GU90LP309', 'Asperger\'s', '2006-09-19');
INSERT INTO `SESMS2`.`Diagnosis` (`diagnosis_id`, `ICD_10`, `diagnosis_name`,
`diagnosis_date`) VALUES ('7', '6GHB89A78E', 'Dyscalculia', '2007-05-03');
INSERT INTO `SESMS2`.`Diagnosis` (`diagnosis_id`, `ICD_10`, `diagnosis_name`,
`diagnosis_date`) VALUES ('8', '1239F8FJO', 'ADD', '2006-03-09');
INSERT INTO `SESMS2`.`Diagnosis` (`diagnosis_id`, `ICD_10`, `diagnosis_name`,
`diagnosis_date`) VALUES ('9', '19IO0OP0K9', 'Autism', '2006-05-04');

INSERT INTO `SESMS2`.`Classroom` (`classroom_phone`, `classroom_number`,
`classroom_capacity`) VALUES ('101', '01', '15');
INSERT INTO `SESMS2`.`Classroom` (`classroom_phone`, `classroom_number`,
`classroom_capacity`) VALUES ('102', '02', '15');
INSERT INTO `SESMS2`.`Classroom` (`classroom_phone`, `classroom_number`,
`classroom_capacity`) VALUES ('103', '03', '15');
INSERT INTO `SESMS2`.`Classroom` (`classroom_phone`, `classroom_number`,
`classroom_capacity`) VALUES ('104', '04', '15');
INSERT INTO `SESMS2`.`Classroom` (`classroom_phone`, `classroom_number`,
`classroom_capacity`) VALUES ('105', '05', '15');
INSERT INTO `SESMS2`.`Classroom` (`classroom_phone`, `classroom_number`,
`classroom_capacity`) VALUES ('301', '06', '10');
INSERT INTO `SESMS2`.`Classroom` (`classroom_phone`, `classroom_number`,
`classroom_capacity`) VALUES ('302', '07', '10');
INSERT INTO `SESMS2`.`Classroom` (`classroom_phone`, `classroom_number`,
`classroom_capacity`) VALUES ('303', '08', '10');
INSERT INTO `SESMS2`.`Classroom` (`classroom_phone`, `classroom_number`,
`classroom_capacity`) VALUES ('304', '09', '10');
INSERT INTO `SESMS2`.`Classroom` (`classroom_phone`, `classroom_number`,
`classroom_capacity`) VALUES ('305', '10', '10');

INSERT INTO `SESMS2`.`Teacher` (`first_name`, `last_name`, `cell_phone`,
`classroom_id`) VALUES ('Dalia', 'Cohen', '9547393910', '2');
INSERT INTO `SESMS2`.`Teacher` (`first_name`, `last_name`, `cell_phone`,
`classroom_id`) VALUES ('Lauren', 'Losada', '7589036784', '6');
```

```
INSERT INTO `SESMS2`.`Teacher` (`first_name`, `last_name`, `cell_phone`,
`classroom_id`) VALUES ('Martin', 'Bing', '9746290564', '7');
INSERT INTO `SESMS2`.`Teacher` (`first_name`, `last_name`, `cell_phone`,
`classroom_id`) VALUES ('Monica', 'Geller', '7489036153', '9');
INSERT INTO `SESMS2`.`Teacher` (`first_name`, `last_name`, `cell_phone`,
`classroom_id`) VALUES ('Ross', 'Geller', '9038757123', '10');
INSERT INTO `SESMS2`.`Teacher` (`first_name`, `last_name`, `cell_phone`,
`classroom_id`) VALUES ('Joseph', 'Tribianni', '1234567890', '1');
INSERT INTO `SESMS2`.`Teacher` (`first_name`, `last_name`, `cell_phone`,
`classroom_id`) VALUES ('Phoebe', 'Buffay', '0987654321', '3');
INSERT INTO `SESMS2`.`Teacher` (`first_name`, `last_name`, `cell_phone`,
`classroom_id`) VALUES ('Rachel', 'Green', '5647382910', '4');
INSERT INTO `SESMS2`.`Teacher` (`first_name`, `last_name`, `cell_phone`,
`classroom_id`) VALUES ('Chandler', 'Wu', '0192837465', '5');

INSERT INTO `SESMS2`.`Primary_therapist` (`first_name`, `last_name`, `cell_number`,
`office_number`, `certificate_number`) VALUES ('Shmuel', 'Blue', '7483920156',
'7483019364', '7342927');
INSERT INTO `SESMS2`.`Primary_therapist` (`first_name`, `last_name`, `cell_number`,
`office_number`, `certificate_number`) VALUES ('Noah', 'Green', '0192837465',
'6574839201', '7495234');
INSERT INTO `SESMS2`.`Primary_therapist` (`first_name`, `last_name`, `cell_number`,
`office_number`, `certificate_number`) VALUES ('Ethan', 'Black', '7593750009',
'4778833442', '5858302');
INSERT INTO `SESMS2`.`Primary_therapist` (`first_name`, `last_name`, `cell_number`,
`office_number`, `certificate_number`) VALUES ('Gabriel', 'Pink', '4134098101',
'1548923252', '2354350');
INSERT INTO `SESMS2`.`Primary_therapist` (`first_name`, `last_name`, `cell_number`,
`office_number`, `certificate_number`) VALUES ('Levi', 'Purple', '0123497854',
'9243850430', '4239057');
INSERT INTO `SESMS2`.`Primary_therapist` (`first_name`, `last_name`, `cell_number`,
`office_number`, `certificate_number`) VALUES ('Isaac', 'White', '0295347234',
'2928520354', '2098523');
INSERT INTO `SESMS2`.`Primary_therapist` (`first_name`, `last_name`, `cell_number`,
`office_number`, `certificate_number`) VALUES ('Brody', 'Orange', '1092834712',
'1290847321', '1290438');
INSERT INTO `SESMS2`.`Primary_therapist` (`first_name`, `last_name`, `cell_number`,
`office_number`, `certificate_number`) VALUES ('Kayla', 'Hudson', '0129347821',
'2053948435', '1902384');
INSERT INTO `SESMS2`.`Primary_therapist` (`first_name`, `last_name`, `cell_number`,
`office_number`, `certificate_number`) VALUES ('Rob', 'Roy', '1923843921',
'1230948731', '1294380');

INSERT INTO `SESMS2`.`Primary_parent` (`parent_id`, `first_name`, `last_name`,
`contact_phone`, `address`) VALUES ('1', 'Akiva', 'One', '1344920483', '23 Fairfax Ave,
Hollywood, FL');
```

```
INSERT INTO `SESMS2`.`Primary_parent` (`parent_id`, `first_name`, `last_name`,
`contact_phone`, `address`) VALUES ('2', 'Jessica', 'Two', '2059437829', '67 Purple Lane,
Green, WY');
INSERT INTO `SESMS2`.`Primary_parent` (`parent_id`, `first_name`, `last_name`,
`contact_phone`, `address`) VALUES ('3', 'Norah', 'Three', '0239854723', '1 Main Lane,
Springfield, AR');
INSERT INTO `SESMS2`.`Primary_parent` (`parent_id`, `first_name`, `last_name`,
`contact_phone`, `address`) VALUES ('4', 'Nathalie', 'Fours', '2390547300', '100 Left
Lane, Pines, NC');
INSERT INTO `SESMS2`.`Primary_parent` (`parent_id`, `first_name`, `last_name`,
`contact_phone`, `address`) VALUES ('5', 'Kayla', 'Girland', '9023857239', '200 Right
Lane, Sunrise, HI');
INSERT INTO `SESMS2`.`Primary_parent` (`parent_id`, `first_name`, `last_name`,
`contact_phone`, `address`) VALUES ('6', 'Ben', 'River', '5023485234', '76 Alton Road,
Miami, FL');
INSERT INTO `SESMS2`.`Primary_parent` (`parent_id`, `first_name`, `last_name`,
`contact_phone`, `address`) VALUES ('7', 'Michael', 'Jordan', '0234985723', '2 Bottom St,
San Diego, CA');
INSERT INTO `SESMS2`.`Primary_parent` (`parent_id`, `first_name`, `last_name`,
`contact_phone`, `address`) VALUES ('8', 'Sarah', 'Parker', '9825349083', '4 Up St, Doral,
CA');
```

```
INSERT INTO `SESMS2`.`Physician_contact` (`first_name`, `last_name`,
`cell_number`, `hospital`) VALUES ('John', 'Living', '4932788312', 'memorial');
INSERT INTO `SESMS2`.`Physician_contact` (`first_name`, `last_name`,
`cell_number`, `hospital`) VALUES ('Steve', 'Bath', '1092478012', 'jackson');
INSERT INTO `SESMS2`.`Physician_contact` (`first_name`, `last_name`,
`cell_number`, `hospital`) VALUES ('Bill', 'Base', '1029487239', 'memorial');
INSERT INTO `SESMS2`.`Physician_contact` (`first_name`, `last_name`,
`cell_number`, `hospital`) VALUES ('Leanne', 'Bed', '1290417234', 'jackson');
INSERT INTO `SESMS2`.`Physician_contact` (`first_name`, `last_name`,
`cell_number`, `hospital`) VALUES ('Michael', 'Kitchen', '1901284370', 'miramar');
INSERT INTO `SESMS2`.`Physician_contact` (`first_name`, `last_name`,
`cell_number`, `hospital`) VALUES ('Martin', 'Closet', '9078540232', 'jackson');
INSERT INTO `SESMS2`.`Physician_contact` (`first_name`, `last_name`,
`cell_number`, `hospital`) VALUES ('Wendy', 'Door', '9108471929', 'miramar');
```

```
INSERT INTO `SESMS2`.`Support_team` (`parent_id`, `physician_id`, `therapist_id`,
`teacher_id`) VALUES ('1', '1', '3', '4');
INSERT INTO `SESMS2`.`Support_team` (`parent_id`, `physician_id`, `therapist_id`,
`teacher_id`) VALUES ('2', '1', '2', '1');
INSERT INTO `SESMS2`.`Support_team` (`parent_id`, `physician_id`, `therapist_id`,
`teacher_id`) VALUES ('3', '1', '1', '2');
INSERT INTO `SESMS2`.`Support_team` (`parent_id`, `physician_id`, `therapist_id`,
`teacher_id`) VALUES ('4', '3', '5', '3');
```

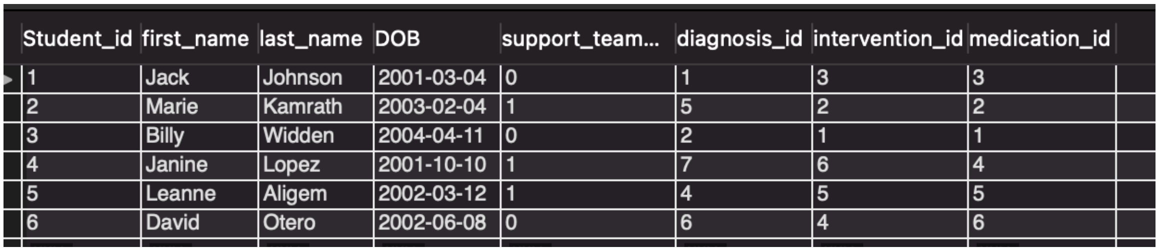
```
INSERT INTO `SESMS2`.`Support_team` (`parent_id`, `physician_id`, `therapist_id`, `teacher_id`) VALUES ('5', '3', '3', '5');
INSERT INTO `SESMS2`.`Support_team` (`parent_id`, `physician_id`, `therapist_id`, `teacher_id`) VALUES ('6', '3', '3', '1');
INSERT INTO `SESMS2`.`Support_team` (`parent_id`, `physician_id`, `therapist_id`, `teacher_id`) VALUES ('7', '2', '2', '3');
INSERT INTO `SESMS2`.`Support_team` (`parent_id`, `physician_id`, `therapist_id`, `teacher_id`) VALUES ('8', '4', '1', '3');

INSERT INTO `SESMS2`.`Student` (`first_name`, `last_name`, `DOB`, `medication_id`, `support_team_id`, `diagnosis_id`, `Intervention_id`) VALUES ('Jack', 'Johnson', '2001-03-04', '3', '0', '1', '3');
INSERT INTO `SESMS2`.`Student` (`first_name`, `last_name`, `DOB`, `medication_id`, `support_team_id`, `diagnosis_id`, `Intervention_id`) VALUES ('Marie', 'Kamrath', '2003-02-04', '2', '1', '5', '2');
INSERT INTO `SESMS2`.`Student` (`first_name`, `last_name`, `DOB`, `medication_id`, `support_team_id`, `diagnosis_id`, `Intervention_id`) VALUES ('Billy', 'Widden', '2004-04-11', '1', '0', '2', '1');
INSERT INTO `SESMS2`.`Student` (`first_name`, `last_name`, `DOB`, `medication_id`, `support_team_id`, `diagnosis_id`, `Intervention_id`) VALUES ('Janine', 'Lopez', '2001-10-10', '4', '1', '7', '6');
INSERT INTO `SESMS2`.`Student` (`first_name`, `last_name`, `DOB`, `medication_id`, `support_team_id`, `diagnosis_id`, `Intervention_id`) VALUES ('Leanne', 'Aligem', '2002-03-12', '5', '1', '4', '5');
INSERT INTO `SESMS2`.`Student` (`first_name`, `last_name`, `DOB`, `medication_id`, `support_team_id`, `diagnosis_id`, `Intervention_id`) VALUES ('David', 'Otero', '2002-06-08', '6', '0', '6', '4');

INSERT INTO `SESMS2`.`Student_has_Medication` (`Student_Student_id`, `Medication_medication_id`) VALUES ('1', '3');
INSERT INTO `SESMS2`.`Student_has_Medication` (`Student_Student_id`, `Medication_medication_id`) VALUES ('2', '2');
INSERT INTO `SESMS2`.`Student_has_Medication` (`Student_Student_id`, `Medication_medication_id`) VALUES ('3', '1');
INSERT INTO `SESMS2`.`Student_has_Medication` (`Student_Student_id`, `Medication_medication_id`) VALUES ('4', '4');
INSERT INTO `SESMS2`.`Student_has_Medication` (`Student_Student_id`, `Medication_medication_id`) VALUES ('5', '5');
INSERT INTO `SESMS2`.`Student_has_Medication` (`Student_Student_id`, `Medication_medication_id`) VALUES ('6', '6');
```

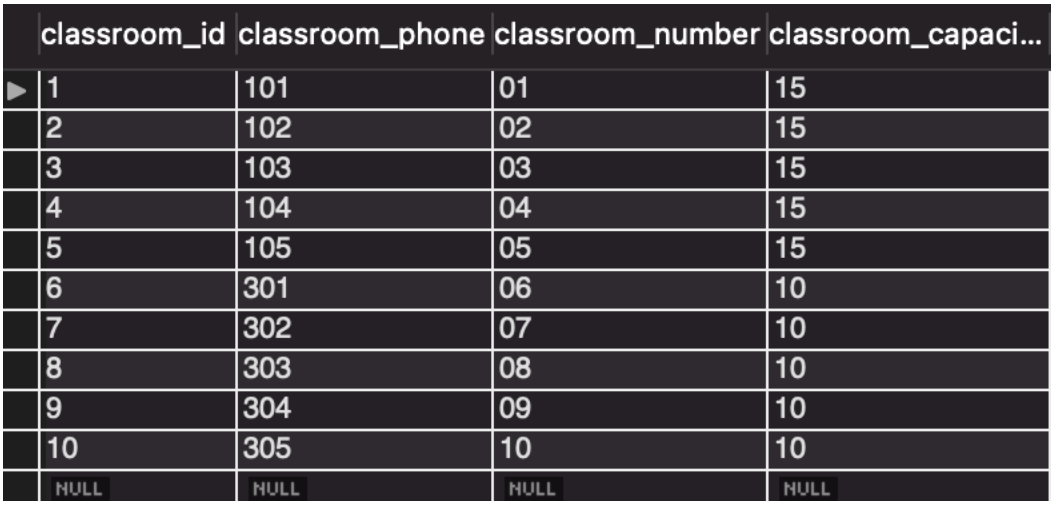
SELECT STATEMENTS

```
SELECT * FROM Student
```



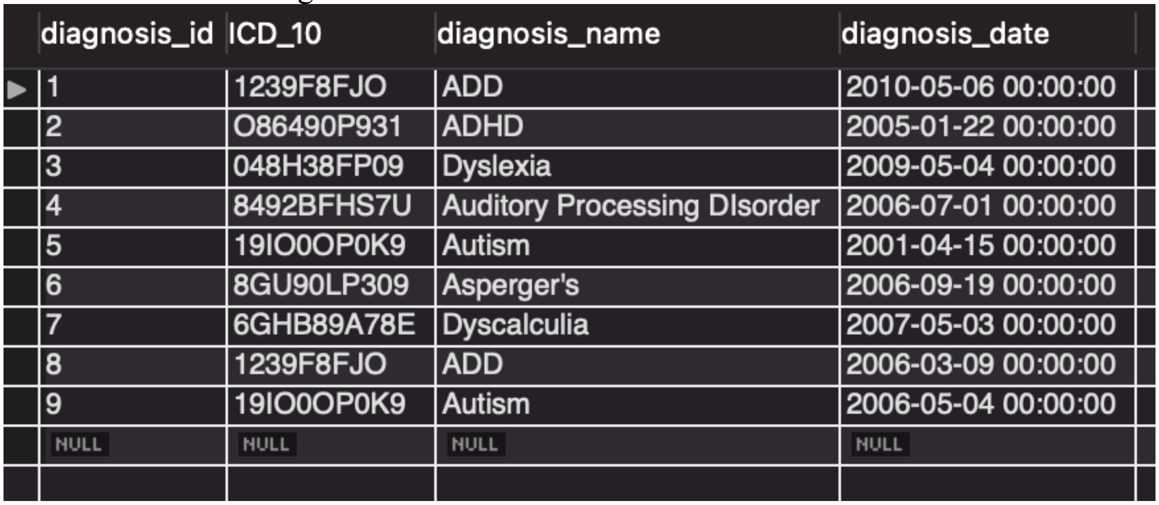
Student_id	first_name	last_name	DOB	support_team...	diagnosis_id	intervention_id	medication_id
1	Jack	Johnson	2001-03-04	0	1	3	3
2	Marie	Kamrath	2003-02-04	1	5	2	2
3	Billy	Widden	2004-04-11	0	2	1	1
4	Janine	Lopez	2001-10-10	1	7	6	4
5	Leanne	Aligem	2002-03-12	1	4	5	5
6	David	Otero	2002-06-08	0	6	4	6

SELECT * FROM Classroom



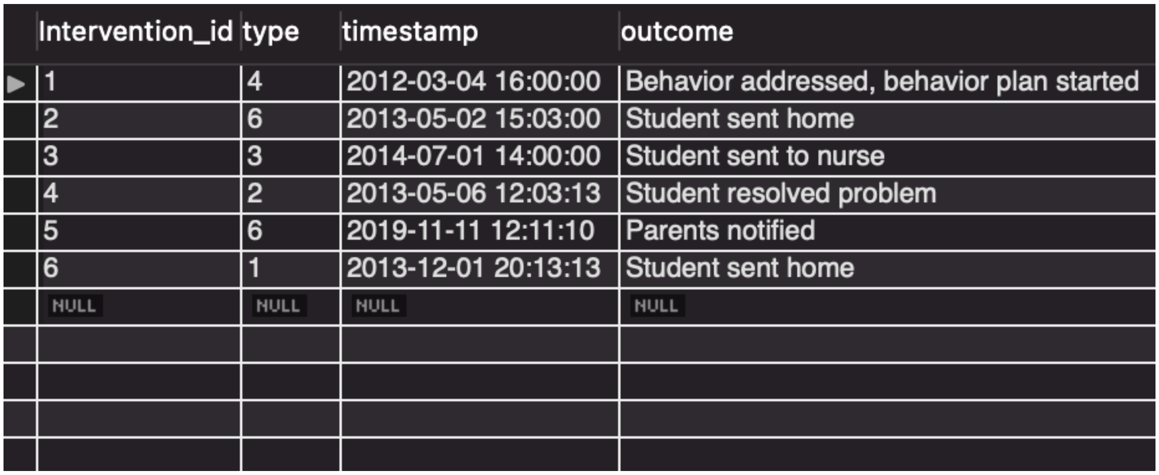
classroom_id	classroom_phone	classroom_number	classroom_capaci...
1	101	01	15
2	102	02	15
3	103	03	15
4	104	04	15
5	105	05	15
6	301	06	10
7	302	07	10
8	303	08	10
9	304	09	10
10	305	10	10
NULL	NULL	NULL	NULL

SELECT * FROM Diagnosis



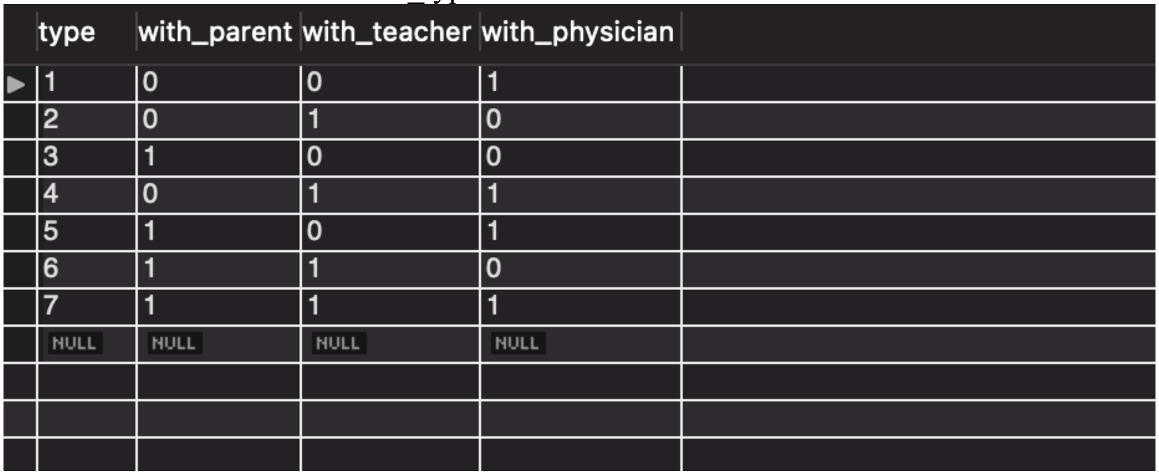
diagnosis_id	ICD_10	diagnosis_name	diagnosis_date
1	1239F8FJO	ADD	2010-05-06 00:00:00
2	O86490P931	ADHD	2005-01-22 00:00:00
3	048H38FP09	Dyslexia	2009-05-04 00:00:00
4	8492BFHS7U	Auditory Processing DIsorder	2006-07-01 00:00:00
5	19IO0OP0K9	Autism	2001-04-15 00:00:00
6	8GU90LP309	Asperger's	2006-09-19 00:00:00
7	6GHB89A78E	Dyscalculia	2007-05-03 00:00:00
8	1239F8FJO	ADD	2006-03-09 00:00:00
9	19IO0OP0K9	Autism	2006-05-04 00:00:00
NULL	NULL	NULL	NULL

SELECT * FROM Intervention_record



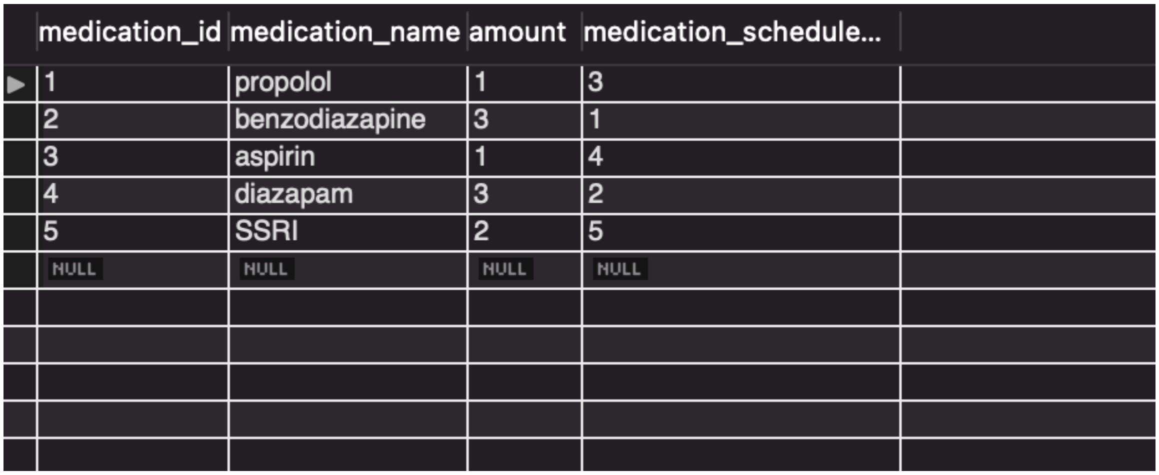
Intervention_id	type	timestamp	outcome
1	4	2012-03-04 16:00:00	Behavior addressed, behavior plan started
2	6	2013-05-02 15:03:00	Student sent home
3	3	2014-07-01 14:00:00	Student sent to nurse
4	2	2013-05-06 12:03:13	Student resolved problem
5	6	2019-11-11 12:11:10	Parents notified
6	1	2013-12-01 20:13:13	Student sent home
NULL	NULL	NULL	NULL

SELECT * FROM Intervention_type



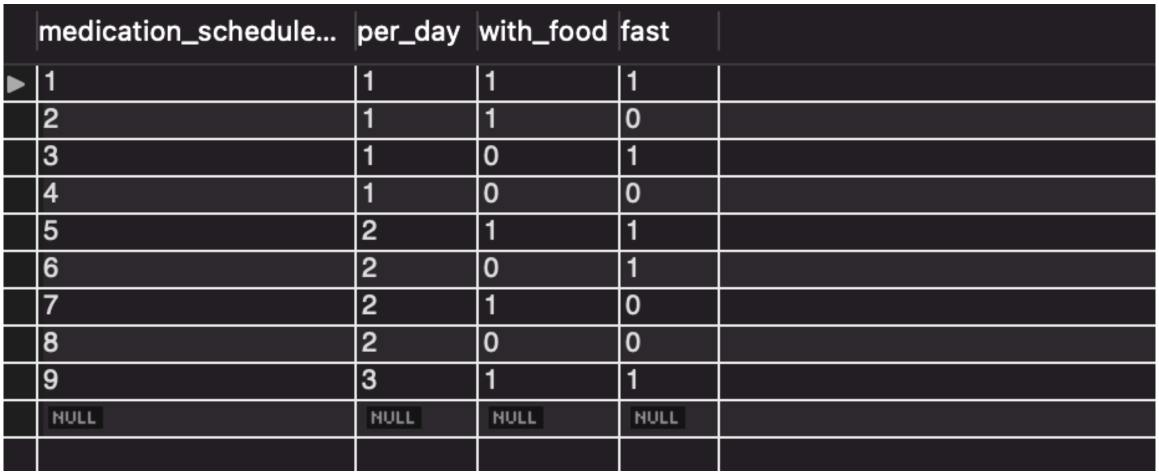
type	with_parent	with_teacher	with_physician
1	0	0	1
2	0	1	0
3	1	0	0
4	0	1	1
5	1	0	1
6	1	1	0
7	1	1	1
NULL	NULL	NULL	NULL

SELECT * FROM Medication



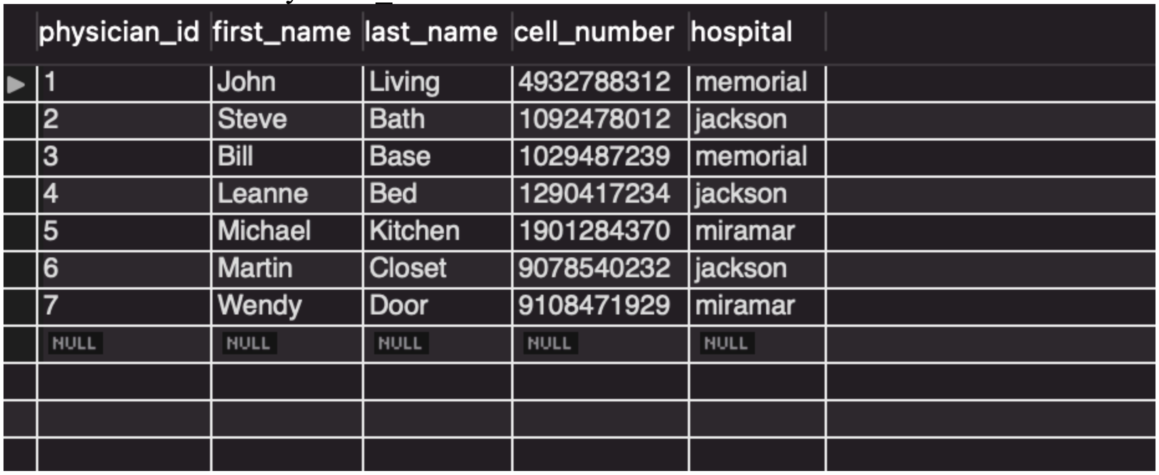
medication_id	medication_name	amount	medication_schedule...
1	propolol	1	3
2	benzodiazapine	3	1
3	aspirin	1	4
4	diazapam	3	2
5	SSRI	2	5
NULL	NULL	NULL	NULL

SELECT * FROM Medication_schedule



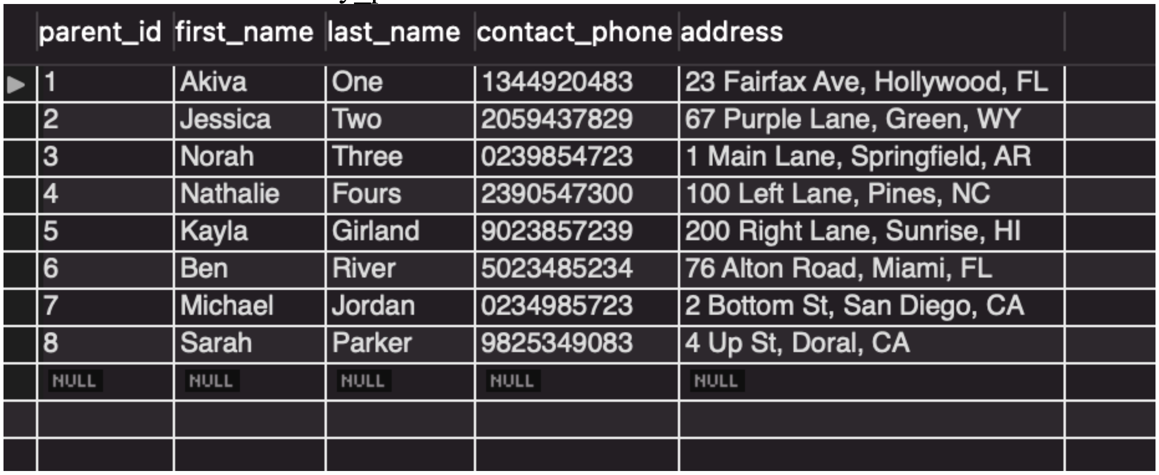
medication_schedule...	per_day	with_food	fast
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	2	1	1
6	2	0	1
7	2	1	0
8	2	0	0
9	3	1	1
NULL	NULL	NULL	NULL

SELECT * FROM Physician_contact



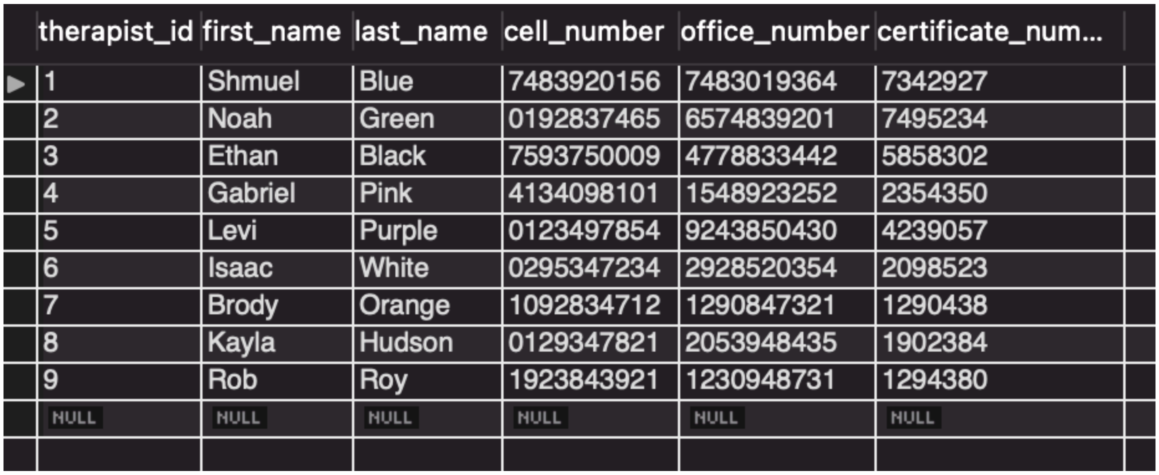
physician_id	first_name	last_name	cell_number	hospital
1	John	Living	4932788312	memorial
2	Steve	Bath	1092478012	jackson
3	Bill	Base	1029487239	memorial
4	Leanne	Bed	1290417234	jackson
5	Michael	Kitchen	1901284370	miramar
6	Martin	Closet	9078540232	jackson
7	Wendy	Door	9108471929	miramar
NULL	NULL	NULL	NULL	NULL

SELECT * FROM Primary_parent



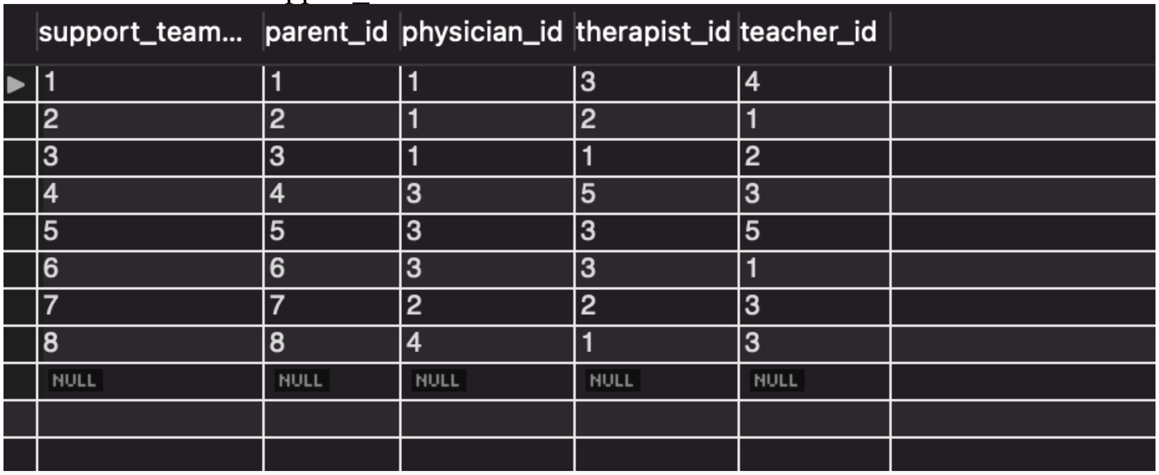
parent_id	first_name	last_name	contact_phone	address
1	Akiva	One	1344920483	23 Fairfax Ave, Hollywood, FL
2	Jessica	Two	2059437829	67 Purple Lane, Green, WY
3	Norah	Three	0239854723	1 Main Lane, Springfield, AR
4	Nathalie	Fours	2390547300	100 Left Lane, Pines, NC
5	Kayla	Girland	9023857239	200 Right Lane, Sunrise, HI
6	Ben	River	5023485234	76 Alton Road, Miami, FL
7	Michael	Jordan	0234985723	2 Bottom St, San Diego, CA
8	Sarah	Parker	9825349083	4 Up St, Doral, CA
NULL	NULL	NULL	NULL	NULL

SELECT * FROM Primary_therapist



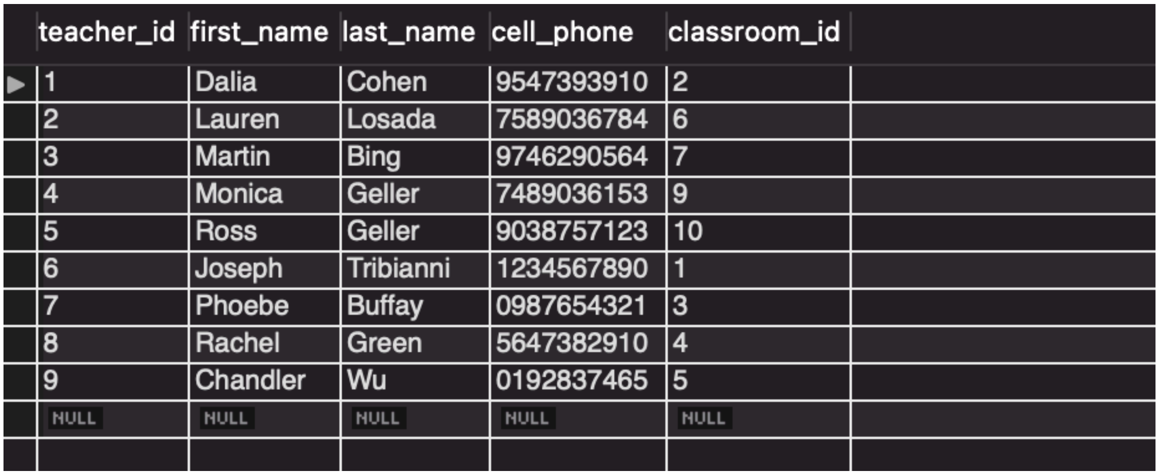
therapist_id	first_name	last_name	cell_number	office_number	certificate_num...
1	Shmuel	Blue	7483920156	7483019364	7342927
2	Noah	Green	0192837465	6574839201	7495234
3	Ethan	Black	7593750009	4778833442	5858302
4	Gabriel	Pink	4134098101	1548923252	2354350
5	Levi	Purple	0123497854	9243850430	4239057
6	Isaac	White	0295347234	2928520354	2098523
7	Brody	Orange	1092834712	1290847321	1290438
8	Kayla	Hudson	0129347821	2053948435	1902384
9	Rob	Roy	1923843921	1230948731	1294380
NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM Support_team



support_team...	parent_id	physician_id	therapist_id	teacher_id
1	1	1	3	4
2	2	1	2	1
3	3	1	1	2
4	4	3	5	3
5	5	3	3	5
6	6	3	3	1
7	7	2	2	3
8	8	4	1	3
NULL	NULL	NULL	NULL	NULL

SELECT * FROM Teacher



teacher_id	first_name	last_name	cell_phone	classroom_id
1	Dalia	Cohen	9547393910	2
2	Lauren	Losada	7589036784	6
3	Martin	Bing	9746290564	7
4	Monica	Geller	7489036153	9
5	Ross	Geller	9038757123	10
6	Joseph	Tribianni	1234567890	1
7	Phoebe	Buffay	0987654321	3
8	Rachel	Green	5647382910	4
9	Chandler	Wu	0192837465	5
NULL	NULL	NULL	NULL	NULL

Appendix B - Data Dictionary Index

Column Name	Table Name
address	Primary_parent
amount	Medication
cell_number	Primary_therapist
cell_phone	Teacher
certificate_number	Primary_therapist
classroom_capacity	Classroom
classroom_id	Classroom
classroom_number	Classroom
classroom_phone	Classroom
contact_phone	Primary_parent
diagnosis_date	Diagnosis
diagnosis_id	Diagnosis
diagnosis_id	Student
diagnosis_name	Diagnosis
DOB	Student
fast	Medication_schedule
first_name	Student, Physician_contact, Primary_therpist, Teacher, Primary_parent
hospital	Physician_contact
ICD_10	Diagnosis
intervention_id	Student
intervention_id	Intervention_record
last_name	Student, Physician_contact, Primary_therpist, Teacher, Primary_parent
medication_id	Student
medication_id	Medication
medication_name	Medication
medication_schedule_id	Medication
medication_schedule_id	Medication_schedule
office_number	Primary_therapist
outcome	Intervention_record
per_day	Medication_schedule

Student_id	Student
support_team_id	Support_team
timestamp	Intervention_record
type	Intervention_type
type	Intervention_record
with_food	Medication_schedule
with_parent	Intervention_type
with_physician	Intervention_type
with_teacher	Intervention_type

References

https://www.w3resource.com/mysql/mysql-procedure.php

https://www.tutorialspoint.com/sql/sql-using-joins.htm

https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-stored-procedures.html