

DT TRACKER Database Design

By: Caroline Newcom

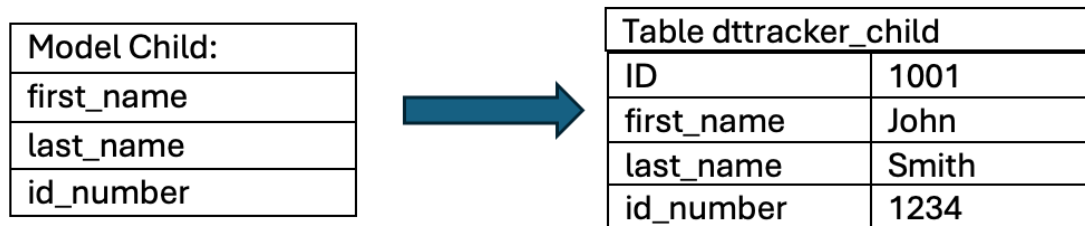
Overview of Database Technology

I will be using PostgreSQL as the database for my web application. I chose this relational database because the web application that I am building involves different relationship types that work together for the DT Tracker web application to function properly. All classes are related as each user has many children on their caseload. Each child has multiple sessions that will be connected to them. For this reason, my web application requires a database that can properly manage the connected data. I chose to use the PostgreSQL database because it supports Python and JSON, both of which will be used within my project. PostgreSQL is known for its ability to analyze data well. A key function of my application is to analyze the data from each session and return information to display attendance percentages. Having a database that can support this function is crucial for the success of my web application. Finally, I chose PostgreSQL as my database because of its atomicity. PostgreSQL ensures that each transaction the database encounters either fully succeeds or fully fails. This adds an extra layer of protection to block partial or invalid data which in return will aid with the accuracy of the information within the database.

I will use the Django framework to interact with the PostgreSQL database. I will create models to define the information needed to go into the database. I will make the migrations that will build the tables in the database. Views will be used to retrieve and display information from the database. I will configure URL routing to ensure that every endpoint is mapped to the matching view.

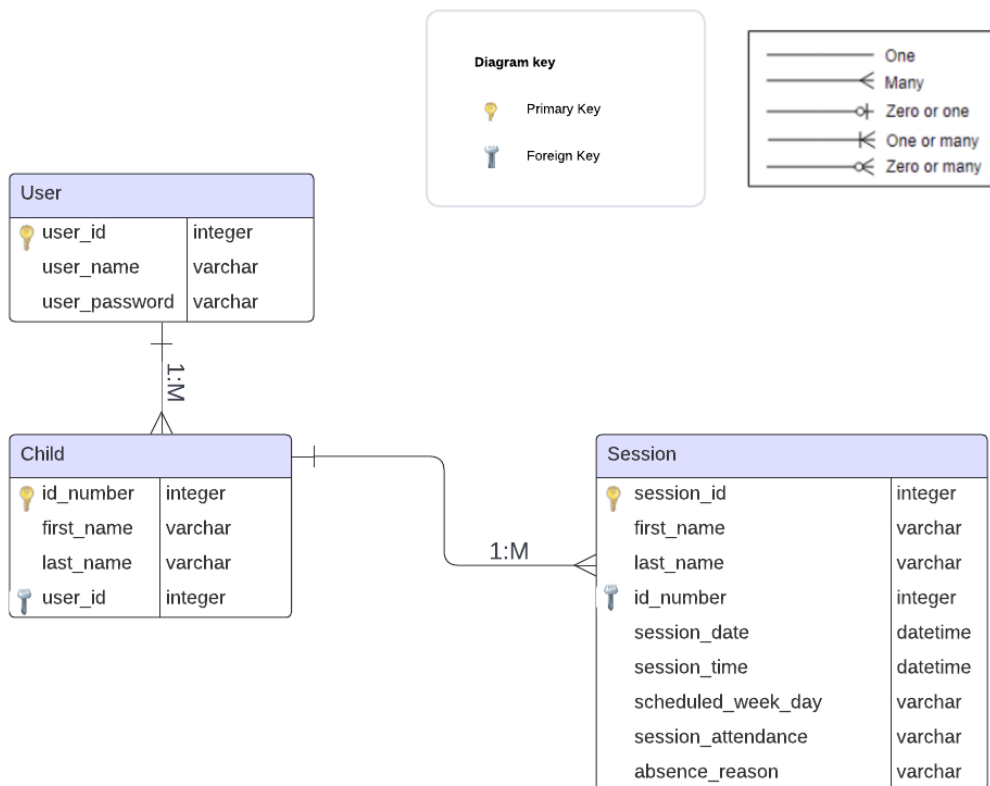
Data Structures

As previously mentioned, there are relationships among the classes within my web application. These classes will be created as models using the Django Framework. The Framework will communicate with the PostgreSQL database to build the tables for each class as shown in the example below.



Entity-Relationship Diagram

I have created an Entity-Relationship Diagram that is displayed below. This diagram shows the connections between the User, Child, and Session classes.



Entity-Relationship Diagram Explained

As shown on the diagram, the user Class has a one-to-many relationship with the Child class. The user (being the Developmental Therapist utilizing the web application) will have many children on their caseload. I will use views within the Django framework to give the user the ability to add and delete children on their caseload, add and delete sessions on their schedule, add attendance information, and read attendance percentage data. The Entity-Relationship Diagram also shows that the Child and Session classes have a one-to-many relationship. One child will have many sessions connected to it. The primary key for the User class will be user_id. User_id will serve as the foreign key within the Child class. The primary key within the Child class is the id_number which is also the foreign key for the Session class. The primary key for the Session class is session_id. Although these tables are relationally connected, I have tried my best to limit repeated attributes to avoid data duplication. My goal is for this to allow me to only have to modify data in one place within the database.

User Accounts

I will create a user account for myself that enables read and write access to the database. Ideally, this web application would only be used in congruence with an Early Intervention clinic. The username and password would be given to each user by their employer rather than the user creating their own account. The code to create a user and grant access is below.

To Create a User within the PostgreSQL Database:

```
CREATE USER cnewcom WITH PASSWORD 'maryville';
```

To Grant Read and Write Access to the User within the PostgreSQL Database:

```
GRANT ALL PRIVILEGES ON DATABASE dttrackerdb TO cnewcom;
```

Connecting Django Framework to the Database:

To connect the database to the Django Framework, I would go into the 'settings.py' file within Django and update the information under the databases section.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'dttracker',
        'USER': 'cnewcom',
        'PASSWORD': 'maryville',
        'HOST': 'localhost',
        'PORT': '',
    }
}
```

How Data Structures will be Used

User

Purpose:

The purpose of the User class is for the Developmental Therapist using the application to have a username and password that they can use to login to the web application and gain access to the content.

Implementation:

Every login request will check username and password information for authentication purposes. This is accomplished by comparing the login information to the stored records of usernames and passwords of preexisting users.

Interaction:

Upon entering the web application URL into a search engine, the web application will first display a login page. The user will be prompted to enter their username and password. If the user enters the correct login information, they will be granted access to the web application and may begin viewing content. If the user enters incorrect information, they will be denied access and prompted to try entering the information again.

Child

Purpose:

The purpose of the Child class is to help the user better manage the children on their caseload. The Child class also works alongside the Session class to collect attendance data, which is the focus of the web application.

Implementation:

The 'Caseload' page of the web application will utilize the database data pertaining to the Child class. Once this tab has been selected, all Child data will be rendered. On this page, each child's name, ID number, and a 'delete' button will be displayed within individual containers in a grid format. If a user wishes to delete a child from their caseload, they will select the 'delete' button under that child's name. Users may select the 'Add Child' button to be redirected to another page that allows them to add a new child, including child information, to the web application/database.

Interaction:

Once a user logs into their account, they will be immediately redirected to the 'Caseload' page. Users may scroll up and down on the page to view each child's name and ID number. To delete a child from their caseload, the user will click on the 'delete' button displayed within the container listing the child's information. If a user would like to add a new child to their caseload, they would click on the 'Add Child' button in the top right corner. After pressing the button, the user will be redirected to the 'Add New Child' page. The user may then enter the child's first name, last name, and ID number. Once user clicks submit, that information is sent directly to the database and the new child's information will be displayed on the 'Caseload' page. To avoid any partial data being submitted, the web application will only allow the user to submit the information once all fields have been filled in. If a user decides to cancel before submitting, they may press the 'Cancel' button before being redirected back to the 'Caseload' page.

Session

Purpose:

The purpose of the Session class is to allow the user (the Developmental Therapist utilizing the web application) to be able to manage their session schedule, enter attendance data for each session, and use the session data to view attendance percentages. This class is crucial to my web application as the primary purpose of the web application is to help Developmental Therapists track attendance percentages more easily.

Implementation:

The 'Schedule' page of the web application will utilize the Session class data. Upon viewing the 'Schedule' page, users will see their weekly therapy session schedule. This information will also be displayed in containers separated by days of the week. The layout will be in a grid format. Each schedule block will contain the child's name, ID number, session time, and a 'delete' button. This information will be rendered from the Session class. If a user wishes to delete a specific session from their schedule, the user will click on the 'delete' button listed within the container that lists that session's information. Users may select the 'Add Session' button to be redirected to another page that allows them to add a new session to their schedule.

The 'Session' page of the web application will also utilize the Session class data. The 'Session' page will first display a search bar that prompts the user to search for a child by entering their name or ID number. Once entered, the information submitted will be reviewed against database information to find the matching child. If the information entered does not match a preexisting child

in the database, an error message will be displayed, and the user will be prompted to reenter the information. If the information entered is valid, the user will be redirected to the 'Session Log' page. From there, the user will enter the session date and select the correct attendance outcome. Once user clicks submit, that information is sent directly to the database. To avoid any partial data being submitted, the web application will only allow the user to submit the information once all fields have been filled in. To prevent any duplicated data, the database will compare the session information being submitted to previous information stored. If that specific child already has session information entered for that specific date, an error message will be displayed to the user detailing that they have already submitted that service log.

The 'Attendance' page of the web application uses the information from the Session class as well. This page allows the user to view attendance reports. When the user first lands on the 'Attendance' page, the page will display an option to either search for a child by typing in their name or ID number or selecting the box that allows the user to view the total caseload attendance report. The user will then select the month they would like the attendance report for before clicking the 'Submit' button. The user will be redirected to the 'Attendance Report' page to view the attendance percentage for the child specified or total caseload. A graph will also display the information requested. The database will produce this information by finding the number of attended sessions and missed sessions for the specific month and averaging those numbers together to produce the overall attendance percentage.

Interaction:

Once a user logs into their account, they will be immediately redirected to the 'Caseload' page. The user may then select the 'Schedule' page to view their weekly developmental therapy session schedule. Users may scroll up and down the page to see the days and times they are scheduled to meet with each child. To delete a session from their schedule, a user may click on the 'delete' button underneath that specific session. A user may then click on the 'Add Session' button in the top right corner to add a new session to their schedule. Once redirected to the 'Add New Session' page, a form will be displayed that will prompt the user to enter the child's first name, last name, ID number, day session will be held on, and time of session. Once the user clicks submit, that information will then be sent to the database as well as displayed on the 'Schedule' page.

Users may go to the 'Sessions' page to enter their attendance information each week. Users will be prompted to enter either the child's name or ID number to add session information for each specific child. If the child information is valid, the user will be redirected to the 'Session Log' page. A form will be displayed that prompts the user to enter the date of the session. Users will then add attendance outcome by marking the box for session completed, excused absence, or unexcused absence. Upon hitting the 'Submit' button, the attendance information would be submitted to the database.

By selecting the 'Attendance' page, users will determine if they would like to view the attendance information of one specific child or their total caseload. To view a specific child's attendance

percentage information, the user would enter either the child's name or ID number into the search bar. To view all their caseload's attendance information, the user would click on the box beside 'Select Total Caseload Attendance Report'. Next, the user will enter the month for which they would like to view. The user will then be redirected to the 'Attendance Report' page. This page will list the average attendance percentage for that month as well as display a line graph that shows the attendance results of each session for that month.

Stretch Feature

The stretch feature that I would add would be the ability for the DT Tracker web application to record information regarding the reasons for missed sessions, determine any patterns within the reasons given for missed sessions, and display those patterns using a chart for the user (Developmental Therapist) to view. The Session class will have an attribute called 'absence_reason'. This will be used to store those reasons within the PostgreSQL database under the Session class. On the user's side, the user will be able to go into the web application, input the reason why certain sessions did not occur, and view a pie chart to display the most common reasons for missed sessions each month.

There will be no need for change in the type of database that I will be using for this project as this feature would still require a relational database. The Child class is still connected to the Session class through a one-to-many relationship. One child will have many sessions connected to it. The feature of an absence pattern report being added to the web application would allow a user to view the common reason behind each child's missed sessions. The user can search by inputting a child's name or ID number to view this information.

How Stretch Feature will be Included and Used within the Session Data Structure

Session

Purpose:

The purpose of the Session class is to allow the user (the Developmental Therapist utilizing the web application) to be able to manage their session schedule, enter attendance data for each session, use the session data to view attendance percentages, and use session data to calculate and view an absence report. This class is crucial to my web application as the primary purpose of the web application is to help Developmental Therapists track attendance percentages more easily. This class will also be key for my stretch feature as the information needed to create these absence report charts will be stored within the Session class.

Implementation:

The 'Session' page of the web application will also utilize the Session class data. The 'Session' page will first display a search bar that prompts the user to search for a child by entering their name or ID number. Once entered, the information submitted will be reviewed against database information to find the matching child. If the information entered does not match a preexisting child in the database, an error message will be displayed, and the user will be prompted to reenter the information. If the information entered is valid, the user will be redirected to the 'Session Log' page. From there, the user will enter the session date and select the correct attendance outcome. If the session did not occur, the user would select the drop-down arrow beside the 'reason for missed session' header. A list of potential reasons for a missed session would be displayed. The user would select the correct reason for the missed visit. Once user clicks submit, that information is sent directly to the database. To avoid any partial data being submitted, the web application will only allow the user to submit the information once all fields have been filled in. To prevent any duplicated data, the database will compare the session information being submitted to previous information stored. If that specific child already has session information entered for that specific date, an error message will be displayed to the user detailing that they have already submitted that service log.

The 'Attendance' page of the web application uses the information from the Session class as well. This page allows the user to view attendance reports and will allow users to view the 'absence pattern report' stretch feature. When the user first lands on the 'Attendance' page, the page will display an option to either search for a child by typing in their name or ID number or selecting the box that allows the user to view the total caseload report. The user will then select the month they would like the report for before clicking the 'Submit' button. Lastly, two boxes will be displayed at the bottom of the page. By selecting the first option, the users will be able to view the attendance percentage report for a specific child or their entire caseload. By selecting the second option, the user will be able to view the absence pattern report that details the most common reasons for missed sessions for a certain child or the total caseload.

Interaction:

Once a user logs into their account, they will be immediately redirected to the 'Caseload' page. Users may go to the 'Sessions' page to enter their attendance information each week. Users will be prompted to enter either the child's name or ID number to add session information for each specific child. If the child information is valid, the user will be redirected to the 'Session Log' page. A form will be displayed that prompts the user to enter the date of the session. Users will then add attendance outcome by marking the box for session completed, excused absence, or unexcused absence. If the session was missed, the user would then click the drop-down button to view a list of reasons for the missed session before clicking the correct reason for their missed session. Upon hitting the 'Submit' button, the attendance information would be submitted to the database.

By selecting the 'Attendance' page, users will determine if they would like to view the attendance information of one specific child or their total caseload. To view a specific child's attendance information, the user would enter either the child's name or ID number into the search bar. To view all their caseload's attendance information, the user would click on the box beside 'Select Total Caseload Attendance Report'. Next, the user will enter the month for which they would like to view. Finally, the user will select to view either the attendance percentage report or the absence pattern report page.

If the user clicks on the box beside the first option, the user will then be redirected to the 'Attendance Report' page. This page will list the average attendance percentage for that month as well as display a line graph that shows the attendance results of each session for that month. If the user clicks the box beside the second option, the user will then be redirected to the 'Absence Pattern Report' that will display the most common reasons for missed sessions using a pie chart.