**Database design and integration for pharmacogenomic testing application**

Medical terms used in the task to better understand:

**Pharmacogenetic (PGx)** – The study of how a person’s individual DNA affects his or her response to medications.

**Gene** – Genes are made up of sequences of DNA, that contain instructions for making a specific protein. It is a fundamental unit of heredity that’s passed from parents.

**Genotype** – Genotype is the combination of two alleles for a specific gene, where an individual inherits one allele from each parent. If both alleles are the same, the genotype is considered homozygous, while if they are different, it is heterozygous.

**Phenotype** – The observable characteristics of an organism. For example: if a person has the genotype “Bb” (one allele for brown eyes “B” and one for blue eyes “b”), their phenotype would be brown eyes because brown is the dominant allele.

**Objective:**

The goal is to design and implement a database that matches patient genotypes with dosing recommendations for psychiatric drugs and demonstrate how this database can be integrated with the testing application.

1. **Database design and implementation**
2. **Tables**

**1. Therapeutic Area Table**

* **Table Name**: Therapeutic\_Area
* **Description**: This table stores information about different therapeutic areas (e.g., oncology, cardiology, psychiatry).

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | INT | Primary key, NOT NULL | Unique identifier for the therapeutic area. |
| TherapeuticAreaName | VARCHAR (50) | NOT NULL | Name of the therapeutic area |

**2. Drug Table**

* **Table Name**: Drug
* **Description**: This table stores information about drugs that belong to various therapeutic areas.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | INT | PRIMARY KEY, NOT NULL | Unique identifier for the drug. |
| DrugName | VARCHAR (50) | NOT NULL | Name of the drug. |
| TherapeuticAreaId | INT | FOREIGN KEY | Links the drug to a specific therapeutic area. |

**3. Gene Table**

* **Table Name**: Gene
* **Description**: This table stores genetic information and links the gene to a specific drug.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | INT | PRIMARY KEY, NOT NULL | Unique identifier for the gene. |
| GeneName | VARCHAR (50) | NOT NULL | Name of the gene. |
|  |  |  |  |

**4. Genotype Table**

* **Table Name**: Genotype
* **Description**: This table stores genotype information for each gene.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | INT | PRIMARY KEY, NOT NULL | Unique identifier for the genotype. |
| Allele | VARCHAR (50) | NOT NULL | Allele information for the genotype. |
| GeneID | INT | UNIQUE, FOREIGN KEY, NOT NULL | Links the genotype to a specific gene. |

**Mandatory gene requirement:** Each gene must have either a genotype or a phenotype. This is enforced by the uniqueness. Constraint on the gene\_id in both Genotype and Phenotype tables.

**5. Phenotype Table**

* **Table Name**: Phenotype
* **Description**: This table stores phenotype information for each gene.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | INT | PRIMARY KEY, NOT NULL | Unique identifier for the phenotype. |
| Phenotype | VARCHAR (50) | NOT NULL | Phenotype information for the gene. |
| GeneID | INT | UNIQUE, FOREIGN KEY, NOT NULL | Links the phenotype to a specific gene. |

**6. Dosing Recommendation Table**

* **Table Name**: Dosing\_Recommendation
* **Description**: This table stores dosing recommendations based on drug, gene, genotype, and phenotype information.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| Id | INT | PRIMARY KEY, NOT NULL | Unique identifier for the dosing recommendation. |
| Recommendation | VARCHAR (50) |  | Text description of the dosing recommendation. |
| DrugID | INT | FOREIGN KEY | Links the recommendation to a specific drug. |
| GeneID | INT | FOREIGN KEY | Links the recommendation to a specific gene. |
| GenotypeID | INT | FOREIGN KEY,  NULL | Links the recommendation to a specific genotype(optional). |
| PhenotypeID | INT | FOREIGN KEY, NULL | Links the recommendation to a specific drug(optional). |
| Source | VARCHAR (50) |  | Source of the recommendation (CPIC, FDA, DPWG) |

**Optional genotype or phenotype:** Every gene must have either a genotype or a phenotype, the optionality is represented in the Dosing Recommendation table, where either genotype or phenotype information will be stored not both.

1. **Key Relationships**
2. **Therapeutic Area and Drug**:

* One therapeutic area can have multiple drugs, but one drug is only applicable to one therapeutic area.

1. **Gene to Genotype (1:1)**:

Each gene can be associated with one genotype (2 alleles or 1 allele), and each genotype must associate with one gene.

1. **Gene to Phenotype (1:1)**:
   1. Each gene can be associated with one phenotype, and each phenotype must be associated with one gene.
2. **Genotype to Dosing Recommendations (1: N)**:
   1. Each genotype can have multiple dosing recommendations. Each dosing recommendation refers to only one genotype (or NULL if based on phenotype)
3. **Phenotype to Dosing Recommendations (1: N)**:
   1. Each phenotype can have multiple dosing recommendations. Each dosing recommendation refers to only one phenotype (or NULL if based on genotype)
4. **Drug to Dosing Recommendations (1: N)**:
   1. Each drug can have multiple dosing recommendations based on different genotype or phenotype.
5. **Gene to Dosing Recommendations**:
   1. **Indirect via Genotype (1: N)**

* Each gene can have one genotype, and each genotype can have multiple dosing recommendations. Therefore, through the genotype, each gene can be indirectly related to multiple dosing recommendations.
  1. **Indirect via Phenotype (1: N)**
* Each gene can have one phenotype, and each phenotype can have multiple dosing recommendations. Therefore, through the phenotype, each gene can be indirectly related to multiple dosing recommendations.
  1. **Direct relationship (1: N)**
* Each gene can have multiple dosing recommendations, but each recommendation can be linked to one gene.

**Schema Design:**

CREATE TABLE Drug (

ID INT PRIMARY KEY AUTO\_INCREMENT,

DrugName VARCHAR(50)

);

CREATE TABLE Gene (

ID INT PRIMARY KEY AUTO\_INCREMENT,

GeneName VARCHAR(50)

);

CREATE TABLE Genotype (

ID INT PRIMARY KEY AUTO\_INCREMENT,

GeneID INT UNIQUE,

Allele1 VARCHAR(50),

Allele2 VARCHAR(50),

FOREIGN KEY (GeneID) REFERENCES Gene(ID)

);

CREATE TABLE Phenotype (

ID INT PRIMARY KEY AUTO\_INCREMENT,

GeneID INT UNIQUE,

Phenotype VARCHAR(255),

FOREIGN KEY (GeneID) REFERENCES Gene(ID)

);

CREATE TABLE Dosing\_Recommendations (

ID INT PRIMARY KEY AUTO\_INCREMENT,

DrugID INT,

GenotypeID INT NULL,

GeneID INT,

PhenotypeID INT NULL,

Recommendation VARCHAR(255),

Source VARCHAR(50),

FOREIGN KEY (DrugID) REFERENCES Drug(ID),

FOREIGN KEY (GeneID) REFERENCES Gene(ID),

FOREIGN KEY (GenotypeID) REFERENCES Genotype(ID),

FOREIGN KEY (PhenotypeID) REFERENCES Phenotype(ID)

);

**Database Design:**

A diagram of a computer

Description automatically generated with medium confidence

1. **Integrating with testing application**

* **The Pharmacogenetic Test:** The pharmacogenetic test is conducted on a sample (e.g. saliva, blood) using the test device. The results include genetic information like genes and their genotype.
* **Extract Genotype information and query the database:** The test application on the test device extracts the genotype information from the test result and the application constructs a query to search for dosing recommendations based on the genotype.

# Query the database

query = """

SELECT

d.DrugName,

g.GeneName,

gr.Allele1,

gr.Allele2,

dr.Recommendation,

dr.Source

FROM

Drug d

JOIN

Dosing\_Recommendations dr ON d.ID = dr.DrugID

JOIN

Genotype gr ON dr.GenotypeID = gr.ID

JOIN

Gene g ON dr.GeneID = g.ID

WHERE g.Allele1 = %s AND g.Allele2 = %s

"""

* **API Request and Server Response:** The application sends an API request to a server to query the database. The API server receives the request, queries the database, and returns the results.
* **Display Results:** The test device application processes the API response from the server and displays the dosing recommendations to the user.

**1. Development Environment**

**Tools:**

**Node.js:** JavaScript runtime for server-side development.

**Code Editor:** Visual Studio Code

**Database Management Tool:** MySQL Workbench

**2. Backend Development**

**Technologies:**

**Express.js:** Web framework for Node.js to create the server and API endpoints.

**mysql2:** Node.js library to interact with MySQL database.

**Libraries/Packages:**

**express:** For handling HTTP requests and responses.

**mysql2:** For connecting to and querying MySQL databases.

**3. Frontend Development**

**Technologies:**

**HTML/CSS/JavaScript:** For creating the user interface.

**Libraries/Packages:**

**fetch:** For making HTTP requests from the front end to the API.

**5. API Development**

**Design:**

**Endpoints:** Define RESTful endpoints

**Request Handling:** Process incoming requests and query the database.

**Response Formatting:** Return data in JSON format.

**Implementation:**

**1. Create API Routes:**

- Set up routes for querying dosing recommendations.

- Handle errors and validation.

Github: <https://github.com/mjotangi/Database-design-to-recommend-drug-dose-based-on-gene/tree/main> It has an ER diagram and example MySQL queries to create a database and retrieve data.