**CS571 - DATABASE MANAGEMENT SYSTEMS**

**PROJECT – 01**

**TEAM 01**

**BY**

**SATHYAPAL REDDY AGAMAMIDI (360987)**

**MADHAV SAI (351499)**

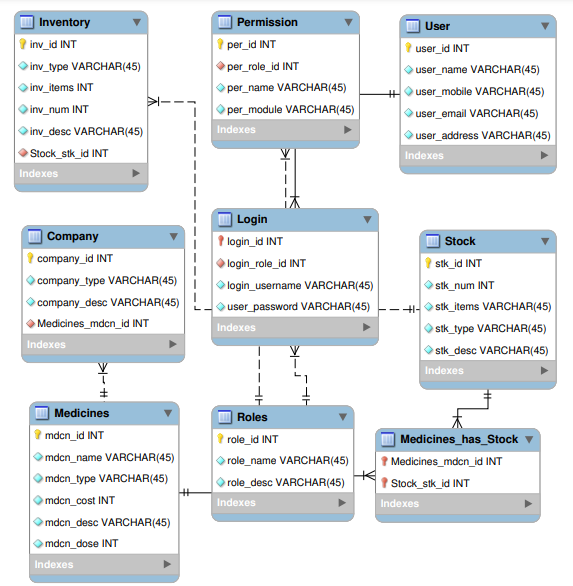
**INTRODUCTION**

This report outlines a proposed design to merge the Hospital Management Subsystem and the Pharmacy Management Subsystem into a unified database system. This integrated system encompasses both the Drug@FDA and FAERS subsystems, offering a holistic platform for overseeing hospital and pharmacy functions. The report provides details on the necessary tables, relationships, and queries essential for the system's implementation.

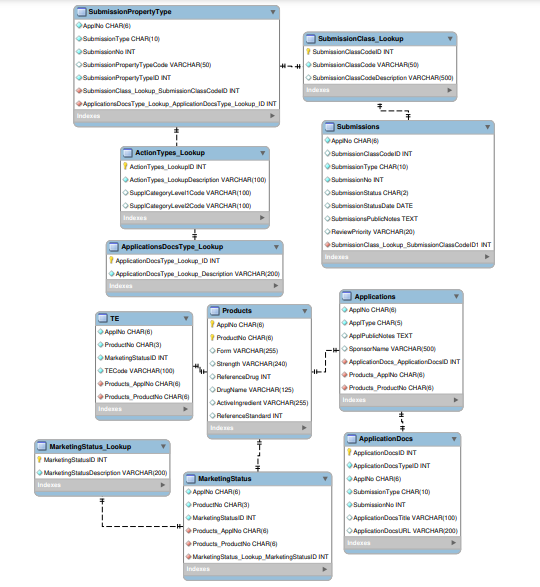
* 1. **Hospital Management System:** The below ER Diagram represents the different types of tables present in the Hospital Management Sub-System and the relations between them.

****

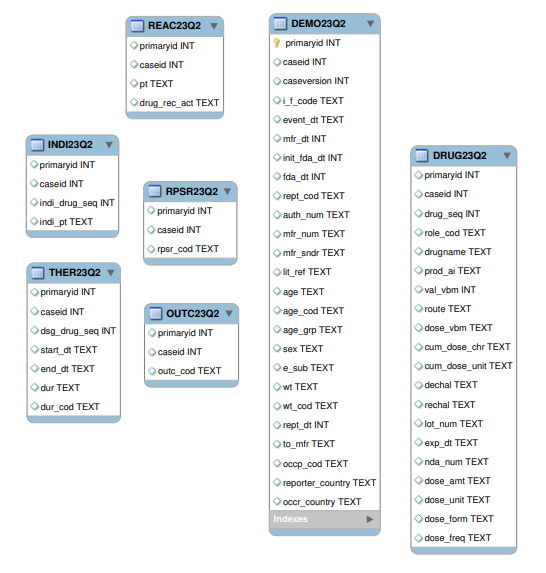
* 1. **Pharmacy Management System:** The below ER Diagram represents the different types of tables present in the Hospital Management Sub-System and the relations between them.

****

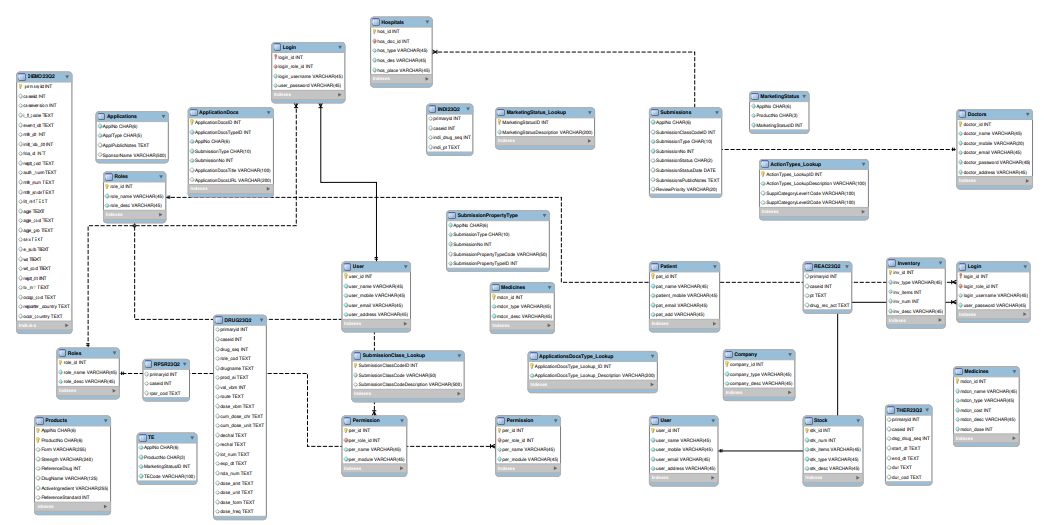
* 1. **Drugs@FDA:** The below ER Diagram represents the different types of tables present in the Hospital Management Sub-System and the relations between them.

****

* 1. **FAERS:** The quarterly data files, which are available in ASCII or SGML formats, include:
* demographic and administrative information and the initial report image ID number (if available);
* drug information from the case reports;
* reaction information from the reports;
* patient outcome information from the reports;
* information on the source of the reports;



**2 Merged ER diagram for entire Bradley Database Management System:**



**3. Converting the E-R diagram into a SQL CREATE TABLE statements and present in the report.**

**3.1 Hospital Management System:**

3.1.1 Doctors Table:

CREATE TABLE `Doctors` (

`doctor\_id` int NOT NULL,

`doctor\_name` varchar(45) NOT NULL,

`doctor\_mobile` varchar(20) NOT NULL,

`doctor\_email` varchar(45) NOT NULL,

`doctor\_password` varchar(45) NOT NULL,

`doctor\_address` varchar(45) NOT NULL,

PRIMARY KEY (`doctor\_id`));

3.1.2 Hospitals Table:

CREATE TABLE `Hospitals` (

`hos\_id` int NOT NULL,

`hos\_doc\_id` int NOT NULL,

`hos\_type` varchar(45) NOT NULL,

`hos\_des` varchar(200) NOT NULL,

`hos\_place` varchar(45) NOT NULL,

PRIMARY KEY (`hos\_id`),

KEY `hos\_doc\_id\_idx` (`hos\_doc\_id`),

CONSTRAINT `hos\_doc\_id` FOREIGN KEY (`hos\_doc\_id`) REFERENCES `Doctors` (`doctor\_id`));

3.1.3 Login Table:

CREATE TABLE `Login` (

`login\_id` int NOT NULL,

`login\_role\_id` int NOT NULL,

`login\_username` varchar(45) NOT NULL,

`user\_password` varchar(45) NOT NULL,

PRIMARY KEY (`login\_id`),

KEY `login\_role\_id\_idx` (`login\_role\_id`),

CONSTRAINT `login\_id` FOREIGN KEY (`login\_id`) REFERENCES `User` (`user\_id`),

CONSTRAINT `login\_role\_id` FOREIGN KEY (`login\_role\_id`) REFERENCES `Roles` (`role\_id`));

3.1.4 Medicines Table:

CREATE TABLE `Medicines` (

`mdcn\_id` int NOT NULL,

`mdcn\_type` varchar(45) NOT NULL,

`mdcn\_desc` varchar(45) NOT NULL,

PRIMARY KEY (`mdcn\_id`));

3.1.5 Patient Table:

CREATE TABLE `Patient` (

`pat\_id` int NOT NULL,

`pat\_name` varchar(45) NOT NULL,

`patient\_mobile` varchar(45) NOT NULL,

`pat\_email` varchar(45) NOT NULL,

`pat\_add` varchar(45) NOT NULL,

PRIMARY KEY (`pat\_id`));

3.1.6 Permission Table:

CREATE TABLE `Permission` (

`per\_id` int NOT NULL,

`per\_role\_id` int NOT NULL,

`per\_name` varchar(45) NOT NULL,

`per\_module` varchar(45) NOT NULL,

PRIMARY KEY (`per\_id`),

KEY `per\_role\_id\_idx` (`per\_role\_id`),

CONSTRAINT `per\_role\_id` FOREIGN KEY (`per\_role\_id`) REFERENCES `Roles` (`role\_id`));

3.1.7 Roles Table:

CREATE TABLE `Roles` (

`role\_id` int NOT NULL,

`role\_name` varchar(45) NOT NULL,

`role\_desc` varchar(200) NOT NULL,

PRIMARY KEY (`role\_id`));

3.1.8 User Table:

CREATE TABLE `User` (

`user\_id` int NOT NULL,

`user\_name` varchar(45) NOT NULL,

`user\_mobile` varchar(45) NOT NULL,

`user\_email` varchar(45) NOT NULL,

`user\_address` varchar(45) NOT NULL,

PRIMARY KEY (`user\_id`));

**3.2 Pharmacy Management System**

3.2.1 Company

CREATE TABLE `Company` (

`company\_id` int NOT NULL,

`company\_type` varchar(45) NOT NULL,

`company\_desc` varchar(200) NOT NULL,

PRIMARY KEY (`company\_id`));

3.2.2 Inventory

CREATE TABLE `Inventory` (

`inv\_id` int NOT NULL,

`inv\_type` varchar(45) NOT NULL,

`inv\_items` varchar(200) NOT NULL,

`inv\_num` int NOT NULL,

`inv\_desc` varchar(200) NOT NULL,

PRIMARY KEY (`inv\_id`));

3.2.3 Login

CREATE TABLE `Login` (

`login\_id` int NOT NULL,

`login\_role\_id` int NOT NULL,

`login\_username` varchar(45) NOT NULL,

`user\_password` varchar(45) NOT NULL,

PRIMARY KEY (`login\_id`),

KEY `login\_role\_id` (`login\_role\_id`),

CONSTRAINT `Login\_ibfk\_1` FOREIGN KEY (`login\_id`) REFERENCES `User` (`user\_id`),

CONSTRAINT `Login\_ibfk\_2` FOREIGN KEY (`login\_role\_id`) REFERENCES `Roles` (`role\_id`));

3.2.4 Medicines

CREATE TABLE `Medicines` (

`mdcn\_id` int NOT NULL,

`mdcn\_name` varchar(45) NOT NULL,

`mdcn\_type` varchar(45) NOT NULL,

`mdcn\_cost` int NOT NULL,

`mdcn\_desc` varchar(45) NOT NULL,

`mdcn\_dose` varchar(50) NOT NULL,

PRIMARY KEY (`mdcn\_id`));

3.2.5 Permission

CREATE TABLE `Permission` (

`per\_id` int NOT NULL,

`per\_role\_id` int NOT NULL,

`per\_name` varchar(45) NOT NULL,

`per\_module` varchar(45) NOT NULL,

PRIMARY KEY (`per\_id`),

KEY `per\_role\_id\_idx` (`per\_role\_id`),

CONSTRAINT `per\_role\_id` FOREIGN KEY (`per\_role\_id`) REFERENCES `Roles` (`role\_id`));

3.2.6 Roles

CREATE TABLE `Roles` (

`role\_id` int NOT NULL,

`role\_name` varchar(45) NOT NULL,

`role\_desc` varchar(200) NOT NULL,

PRIMARY KEY (`role\_id`));

3.2.7 Stock

CREATE TABLE `Stock` (

`stk\_id` int NOT NULL,

`stk\_num` int NOT NULL,

`stk\_items` varchar(45) NOT NULL,

`stk\_type` varchar(45) NOT NULL,

`stk\_desc` varchar(200) NOT NULL,

PRIMARY KEY (`stk\_id`));

3.2.8 User

CREATE TABLE `User` (

`user\_id` int NOT NULL,

`user\_name` varchar(45) NOT NULL,

`user\_mobile` varchar(45) NOT NULL,

`user\_email` varchar(45) NOT NULL,

`user\_address` varchar(45) NOT NULL,

PRIMARY KEY (`user\_id`));

3.3 Drugs@FDA

3.3.1 ActionTypes\_Lookup

CREATE TABLE `ActionTypes\_Lookup` (

`ActionTypes\_LookupID` int NOT NULL AUTO\_INCREMENT,

`ActionTypes\_LookupDescription` varchar(100) NOT NULL,

`SupplCategoryLevel1Code` varchar(100) DEFAULT NULL,

`SupplCategoryLevel2Code` varchar(100) DEFAULT NULL,

PRIMARY KEY (`ActionTypes\_LookupID`));

3.3.2 ApplicationDocs

CREATE TABLE `ApplicationDocs` (

`ApplicationDocsID` int NOT NULL AUTO\_INCREMENT,

`ApplicationDocsTypeID` int NOT NULL,

`ApplNo` char(6) NOT NULL,

`SubmissionType` char(10) NOT NULL,

`SubmissionNo` int NOT NULL,

`ApplicationDocsTitle` varchar(100) DEFAULT NULL,

`ApplicationDocsURL` varchar(200) DEFAULT NULL,

PRIMARY KEY (`ApplicationDocsID`));

3.3.3 Applications

REATE TABLE `Applications` (

`ApplNo` char(6) NOT NULL,

`ApplType` char(5) NOT NULL,

`ApplPublicNotes` text,

`SponsorName` varchar(500) DEFAULT NULL);

3.3.4 ApplicationsDocsType\_Lookup

CREATE TABLE `ApplicationsDocsType\_Lookup` (

`ApplicationDocsType\_Lookup\_ID` int NOT NULL AUTO\_INCREMENT,

`ApplicationDocsType\_Lookup\_Description` varchar(200) NOT NULL,

PRIMARY KEY (`ApplicationDocsType\_Lookup\_ID`));

3.3.5 MarketingStatus

CREATE TABLE `MarketingStatus` (

`ApplNo` char(6) NOT NULL,

`ProductNo` char(3) NOT NULL,

`MarketingStatusID` int NOT NULL);

3.3.6 MarketingStatus\_Lookup

CREATE TABLE `MarketingStatus\_Lookup` (

`MarketingStatusID` int NOT NULL AUTO\_INCREMENT,

`MarketingStatusDescription` varchar(200) NOT NULL,

PRIMARY KEY (`MarketingStatusID`));

3.3.7 Products

CREATE TABLE `Products` (

`ApplNo` char(6) NOT NULL,

`ProductNo` char(6) NOT NULL,

`Form` varchar(255) DEFAULT NULL,

`Strength` varchar(240) DEFAULT NULL,

`ReferenceDrug` int DEFAULT NULL,

`DrugName` varchar(125) DEFAULT NULL,

`ActiveIngredient` varchar(255) DEFAULT NULL,

`ReferenceStandard` int DEFAULT NULL,

PRIMARY KEY (`ApplNo`,`ProductNo`));

3.3.8 SubmissionClass\_Lookup

CREATE TABLE `SubmissionClass\_Lookup` (

`SubmissionClassCodeID` int NOT NULL AUTO\_INCREMENT,

`SubmissionClassCode` varchar(50) NOT NULL,

`SubmissionClassCodeDescription` varchar(500) DEFAULT NULL,

PRIMARY KEY (`SubmissionClassCodeID`));

3.3.9 SubmissionPropertyType

CREATE TABLE `SubmissionPropertyType` (

`ApplNo` char(6) NOT NULL,

`SubmissionType` char(10) NOT NULL,

`SubmissionNo` int NOT NULL,

`SubmissionPropertyTypeCode` varchar(50) DEFAULT 'Orphan or NULL',

`SubmissionPropertyTypeID` int NOT NULL);

3.3.10 Submissions

CREATE TABLE `Submissions` (

`ApplNo` char(6) NOT NULL,

`SubmissionClassCodeID` int DEFAULT NULL,

`SubmissionType` char(10) NOT NULL,

`SubmissionNo` int NOT NULL,

`SubmissionStatus` char(2) DEFAULT NULL,

`SubmissionStatusDate` date DEFAULT NULL,

`SubmissionsPublicNotes` text,

`ReviewPriority` varchar(20) DEFAULT 'Standard or Priority');

3.3.11 TE

CREATE TABLE `TE` (

`ApplNo` char(6) NOT NULL,

`ProductNo` char(3) NOT NULL,

`MarketingStatusID` int NOT NULL,

`TECode` varchar(100) NOT NULL);

**3.4 FAERS**

3.4.1 DEMO23Q2

CREATE TABLE `DEMO23Q2` (`﻿primaryid` int NOT NULL, `caseid` int DEFAULT NULL, `caseversion` int DEFAULT NULL, `i\_f\_code` text, `event\_dt` text, `mfr\_dt` int DEFAULT NULL, `init\_fda\_dt` int DEFAULT NULL, `fda\_dt` int DEFAULT NULL,`rept\_cod` text, `auth\_num` text, `mfr\_num` text,`mfr\_sndr` text, `lit\_ref` text, `age` text, `age\_cod` text,`age\_grp` text, `sex` text, `e\_sub` text, `wt` text,`wt\_cod` text,`rept\_dt` int DEFAULT NULL,`to\_mfr` text, `occp\_cod` text, `reporter\_country` text, `occr\_country` text, PRIMARY KEY (`﻿primaryid`));

3.4.2 DRUG23Q2

CREATE TABLE `DRUG23Q2` (`primaryid` int DEFAULT NULL, `caseid` int DEFAULT NULL,`drug\_seq` int DEFAULT NULL, `role\_cod` text, `drugname` text,`prod\_ai` text,`val\_vbm` int DEFAULT NULL, `route` text, `dose\_vbm` text,`cum\_dose\_chr` text,`cum\_dose\_unit` text,`dechal` text, `rechal` text, `lot\_num` text, `exp\_dt` text,`nda\_num` text, `dose\_amt` text, `dose\_unit` text,`dose\_form` text,`dose\_freq` text);

3.4.3 INDI23Q2

REATE TABLE `INDI23Q2` (`primaryid` int DEFAULT NULL,`caseid` int DEFAULT NULL, `indi\_drug\_seq` int DEFAULT NULL,`indi\_pt` text);

3.4.4 OUTC23Q2

CREATE TABLE `OUTC23Q2` (`primaryid` int DEFAULT NULL, `caseid` int DEFAULT NULL, `outc\_cod` text);

3.4.5 REAC23Q2

CREATE TABLE `REAC23Q2` (`primaryid` int DEFAULT NULL,`caseid` int DEFAULT NULL,`pt` text,`drug\_rec\_act` text);

3.4.6 RPSR23Q2

CREATE TABLE `RPSR23Q2` ( `primaryid` int DEFAULT NULL,`caseid` int DEFAULT NULL,`rpsr\_cod` text);

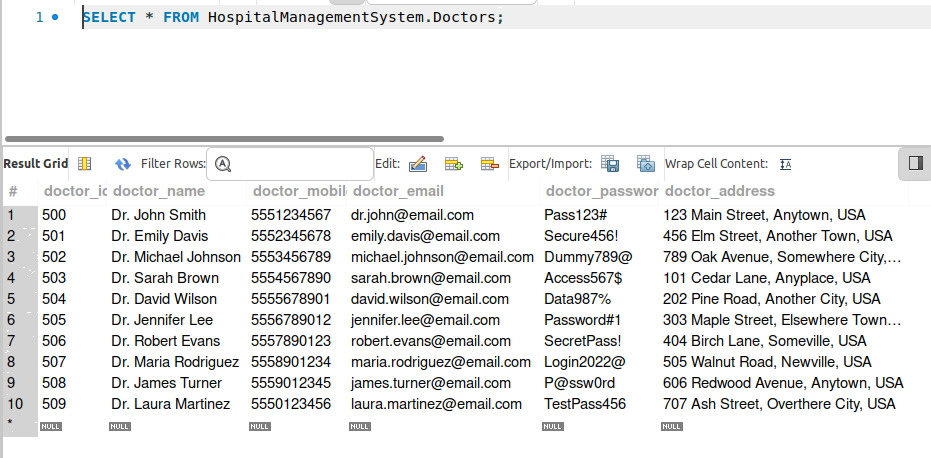
3.4.7 THER23Q2

CREATE TABLE `THER23Q2` (`primaryid` int DEFAULT NULL, `caseid` int DEFAULT NULL,`dsg\_drug\_seq` int DEFAULT NULL, `start\_dt` text,`end\_dt` text,`dur` text, `dur\_cod` text);

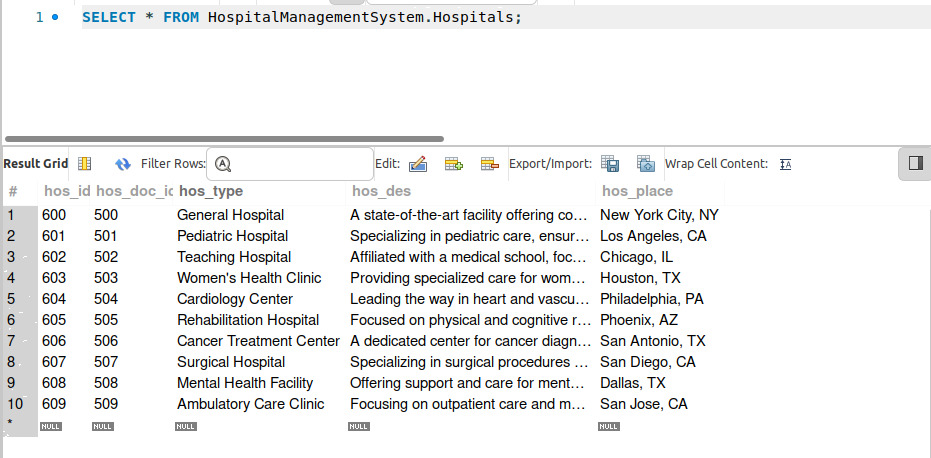
**4. Formulate 10 records and insert your one hospital management subsystem and pharmacy management subsystem. Load all data downloaded from Data@FDA and one quarter data from FAERS.**

**4.1 Hospital Management System:**

4.1.1 Doctors



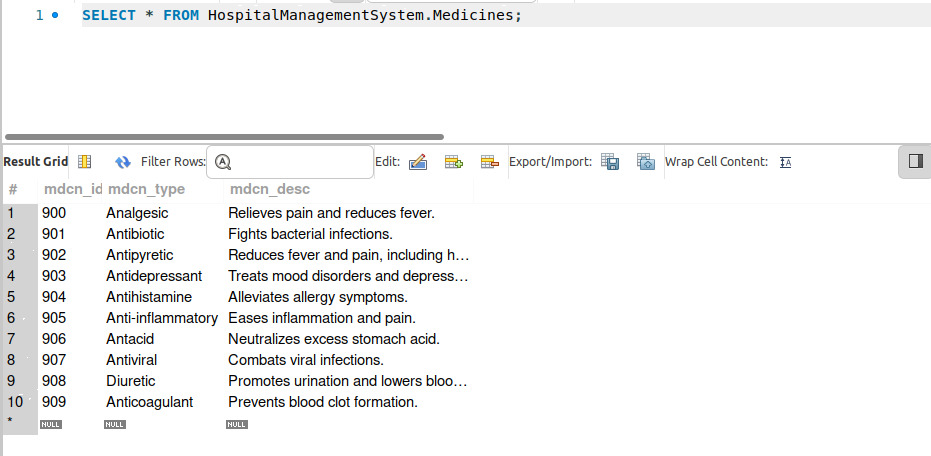
4.1.2 Hospitals



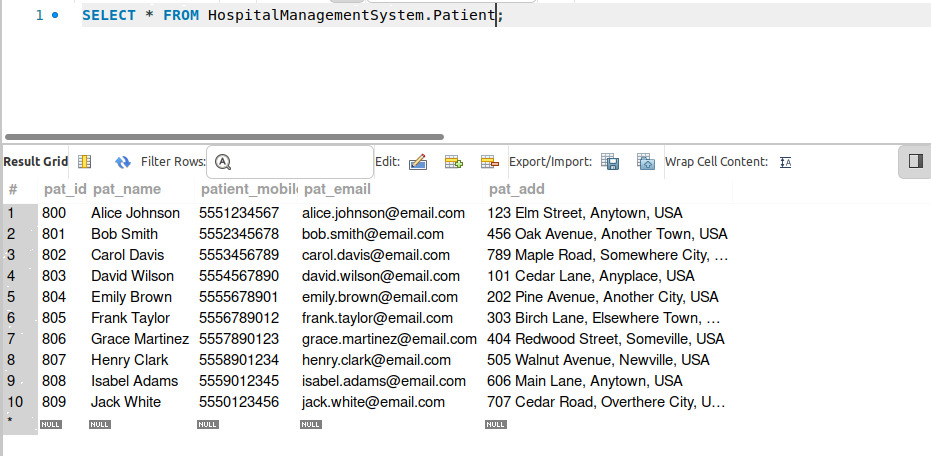
4.1.3 Login



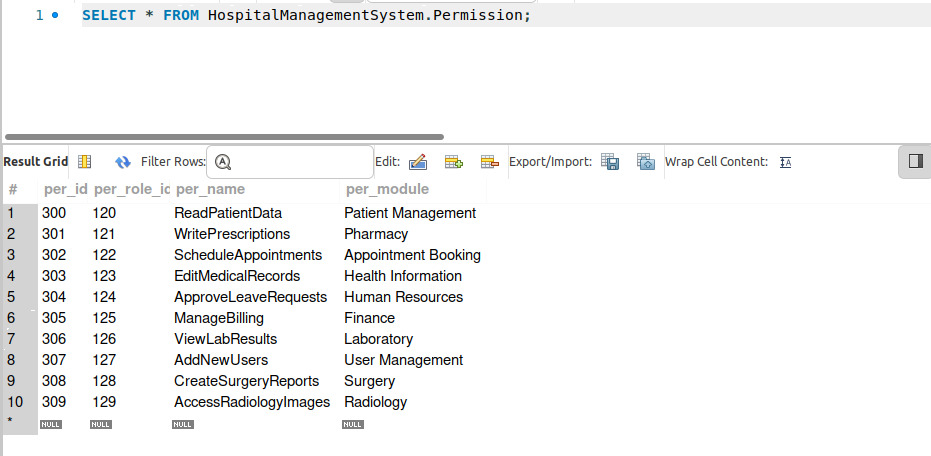
4.1.4 Medicines



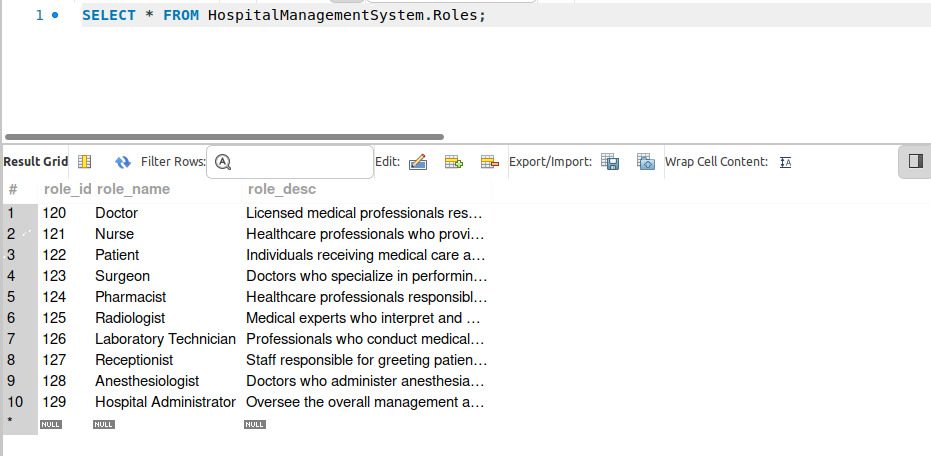
4.1.5 Patient



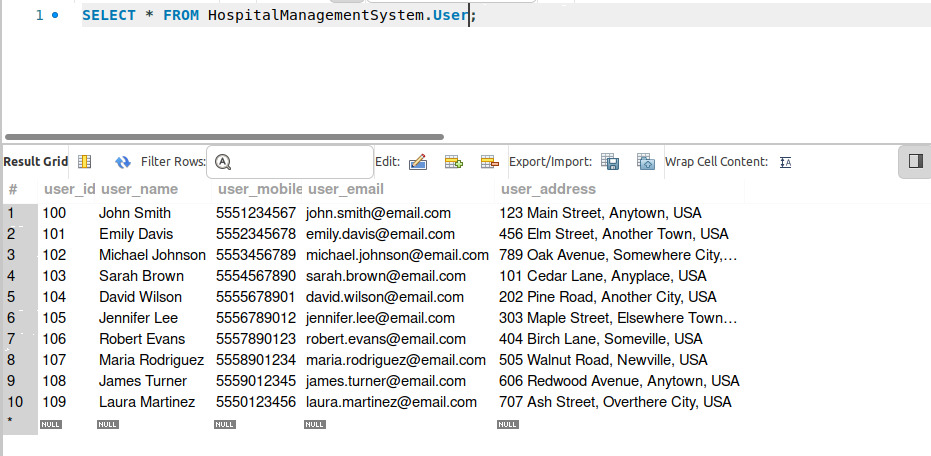
4.1.6 Permission



4.1.7 Roles

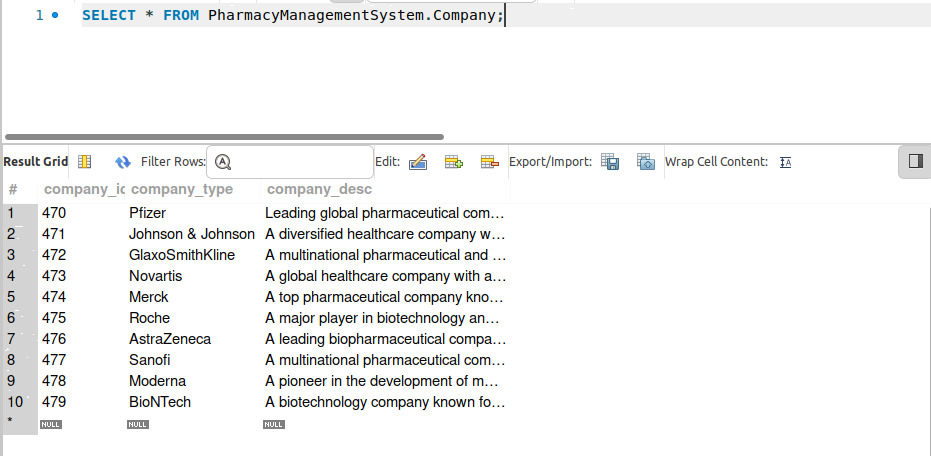


4.1.8 User

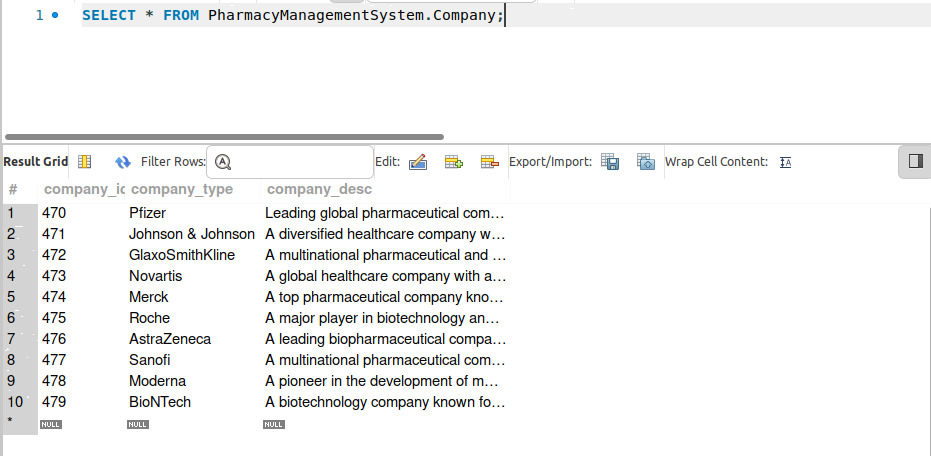


4.2 Pharmacy Management System

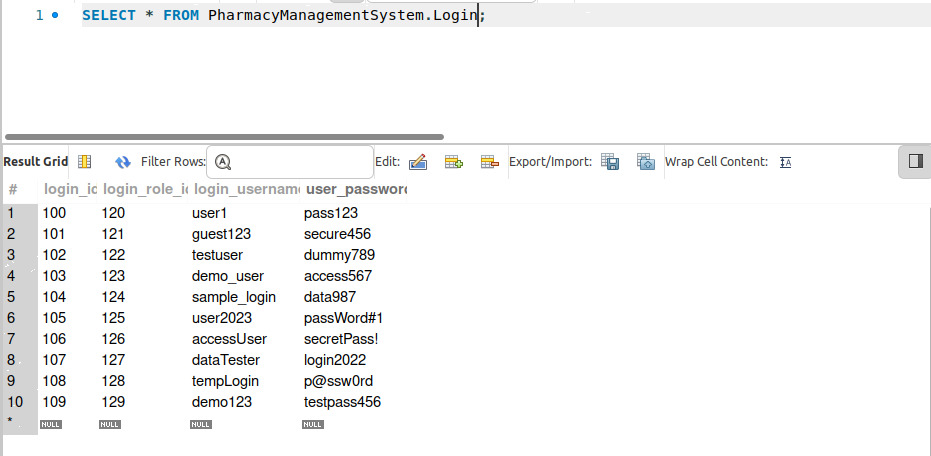
4.2.1 Company



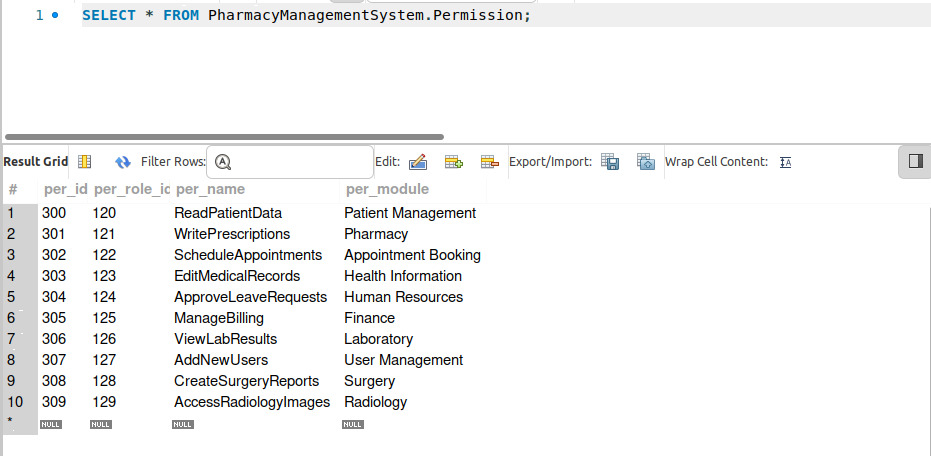
4.2.2 Inventory



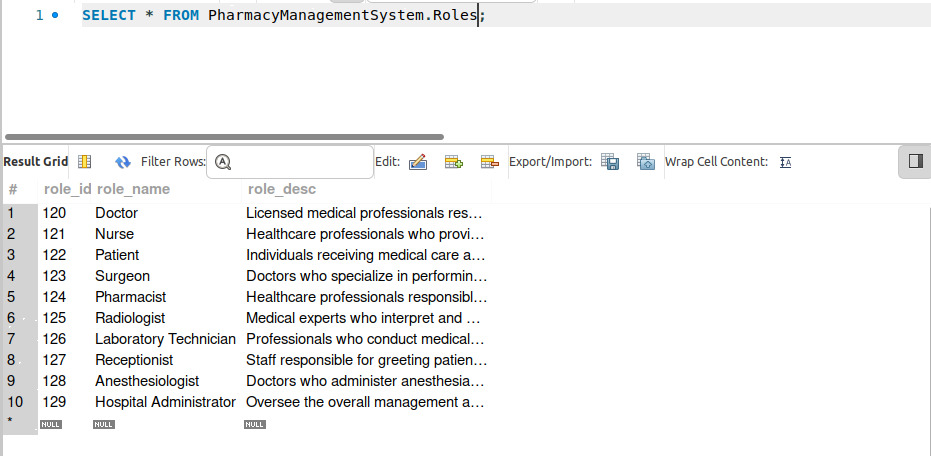
4.2.3 Login



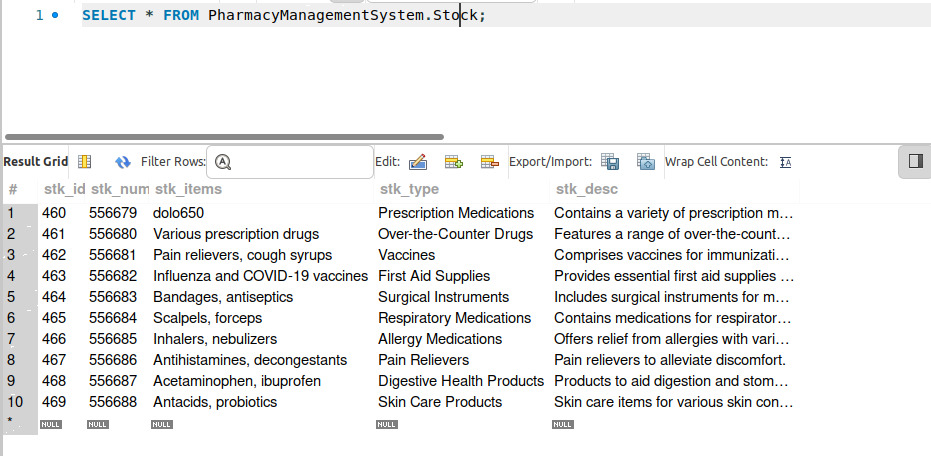
4.2.4 Permission



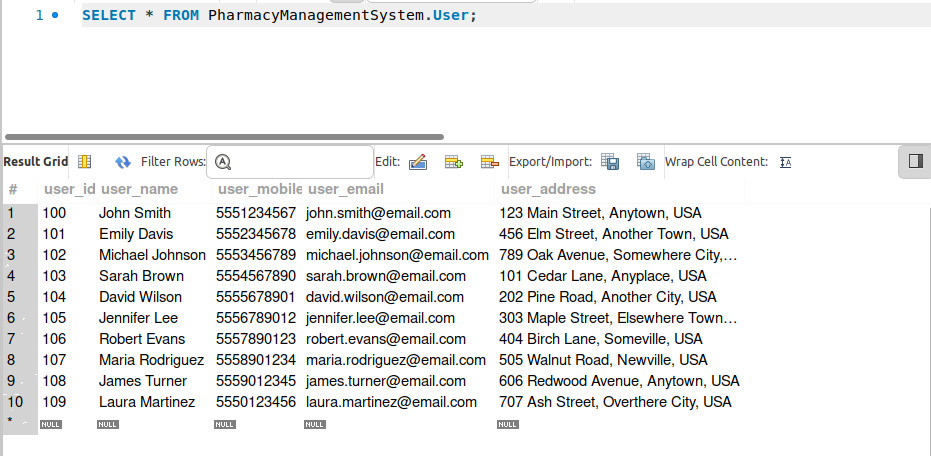
4.2.5 Roles



4.2.6 Stock

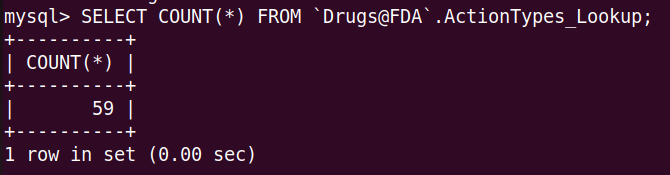


4.2.7 User

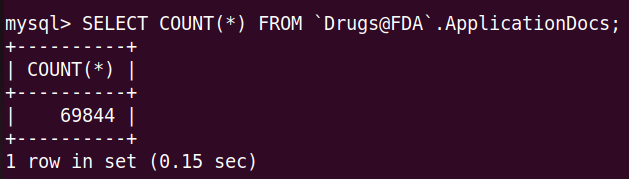


**4.3 Drugs@FDA**

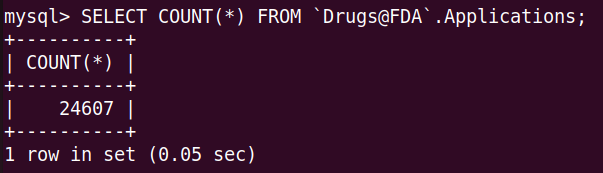
4.3.1 ActionTypes\_Lookup



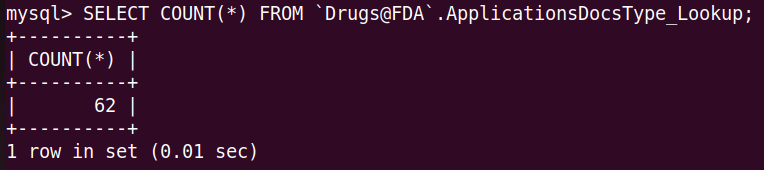
4.3.2 ApplicationDocs



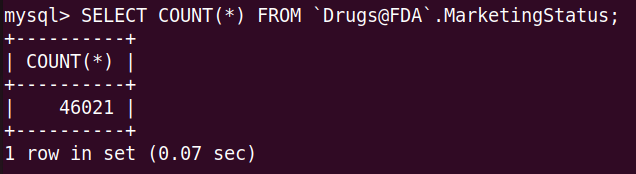
4.3.3 Applications



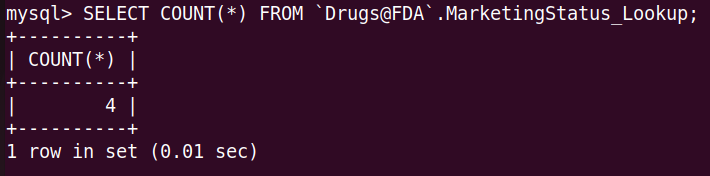
4.3.4 ApplicationsDocsType\_Lookup



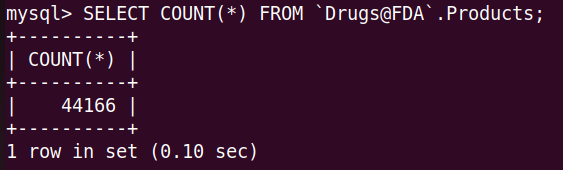
4.3.5 MarketingStatus



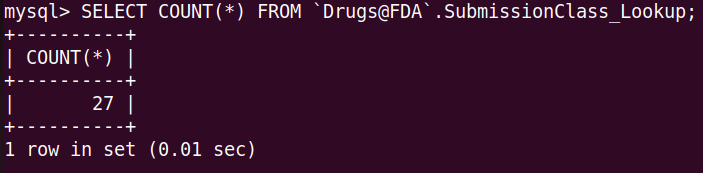
4.3.6 MarketingStatus\_Lookup



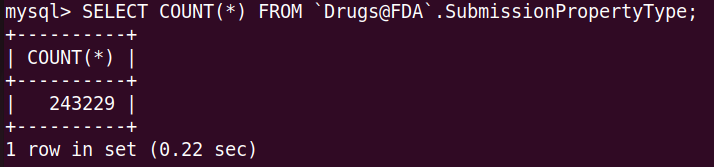
4.3.7 Products



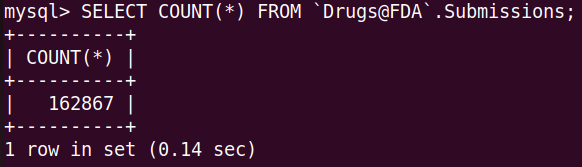
4.3.8 SubmissionClass\_Lookup



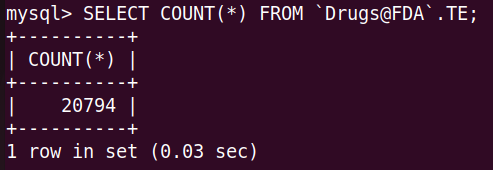
4.3.9 SubmissionPropertyType



4.3.10 Submissions

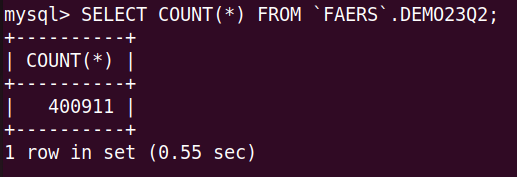


4.3.11 TE

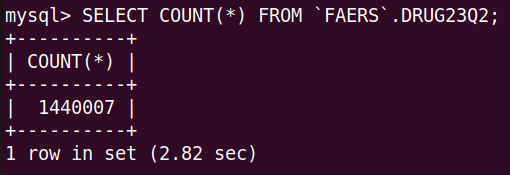


**4.4 FAERS**

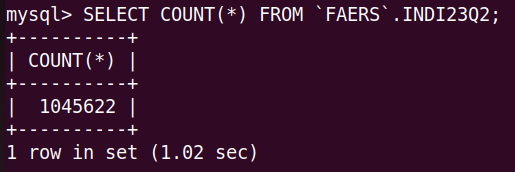
4.4.1 DEMO23Q2



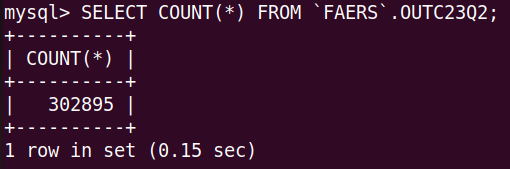
4.4.2 DRUG23Q2



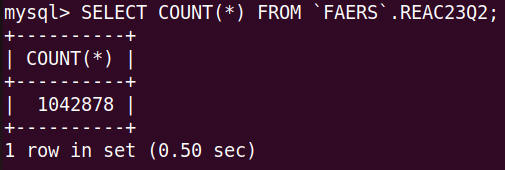
4.4.3 INDI23Q2



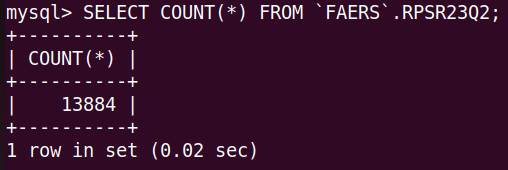
4.4.4 OUTC23Q2



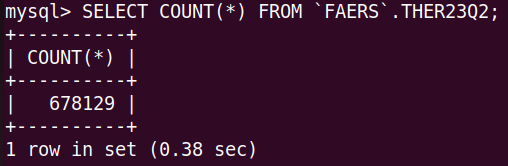
4.4.5 REAC23Q2



4.4.6 RPSR23Q2



4.4.7 THER23Q2



**5 & 6. Formulate in English at least 10 realistic queries you believe would**

be useful to somebody using you database. Formulate all of your queries using relational algebra.

1. **Query : Retrieve distinct doctor names from Doctor Table by combining Patient and Medicines Table.**

SELECT DISTINCT doctor\_name

FROM Doctors

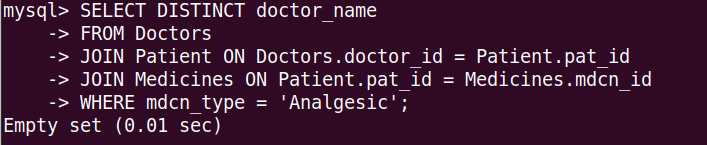
JOIN Patient ON Doctors.doctor\_id = Patient.pat\_id

JOIN Medicines ON Patient.pat\_id = Medicines.mdcn\_id

WHERE mdcn\_type = 'Analgesic';

Expression: π doctor\_name(σ mdcn\_type='Analgesic' (Doctors ⨝ (Patient ⨝ Medicines)))

Result : Empty set



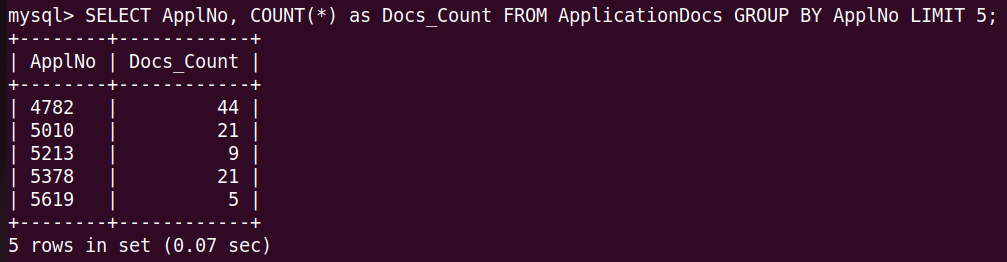
1. **Query: Write a query to display the count of ApplNo from ApplicationDocs of Drugs@FDA schema limitedby 5.**

SELECT ApplNo, COUNT(\*) as Docs\_Count

FROM ApplicationDocs

GROUP BY ApplNo

LIMIT 5;

Output: 

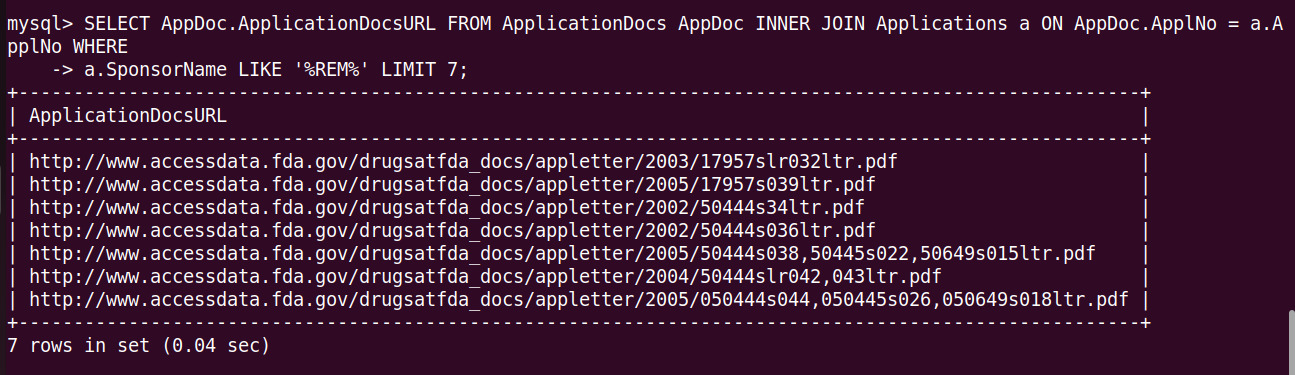
Algebraic Expression:

π ApplNo, COUNT() as Docs\_Count(σ True (ApplicationDocs) ÷ π ApplNo, COUNT() as Docs\_Count(σ True (ApplicationDocs)))

1. **Query: Write a query to display ApplicationDocsURL using an inner join with Applications table limited by 7.**

SELECT AppDoc.ApplicationDocsURL FROM ApplicationDocs AppDoc INNER JOIN Applications A ON AppDoc.ApplNo = A.ApplNo WHERE A.SponsorName LIKE '%REM%' LIMIT 7;

Result:



Relational Expression:

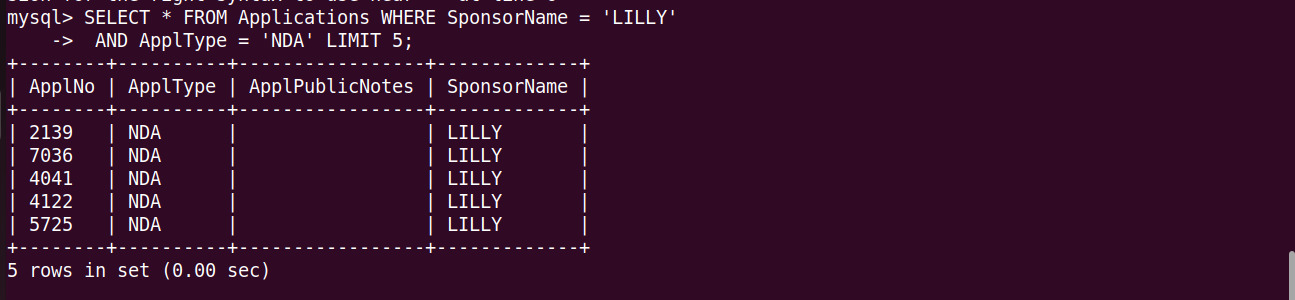
π ApplicationDocsURL(σ SponsorName LIKE '%REM%' (ApplicationDocs ⨝[ApplNo = ApplNo] Applications))

1. **Query: Write a query to get the data from Applications table considering the SponsorName as LILLY.**

SELECT \* FROM Applications WHERE SponsorName = 'LILLY'

AND ApplType = 'NDA' LIMIT 5;

Result:



Relational Expression: σ SponsorName = 'LILLY' AND ApplType = 'NDA' (Applications) ⨝ (π \* (Applications) LIMIT 5)

1. **Query: Write a query to display ApplicationDocssTypeID, ApplicationDocsTypID, ApplicationsDocsTitle and Lookup description from ApplicationDocs table using the ApplNo as 5010 limited by 5.**

SELECT ad.ApplicationDocsTypeID, ad.ApplicationDocsTitle,

adl.ApplicationDocsType\_Lookup\_Description

FROM ApplicationDocs ad

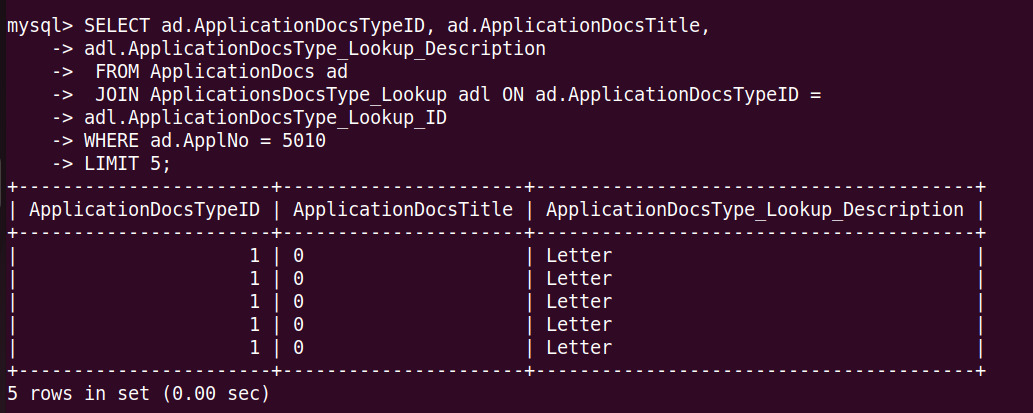
JOIN ApplicationsDocsType\_Lookup adl ON ad.ApplicationDocsTypeID =

adl.ApplicationDocsType\_Lookup\_ID

WHERE ad.ApplNo = 5010

LIMIT 5;

Result:



Relational Expression:

πApplicationDocsTypeID,ApplicationDocsTitle,ApplicationDocsType\_Lookup\_Description(σApplNo=5010(ApplicationDocs⨝[ApplicationDocsTypeID= ApplicationDocsType\_Lookup\_ID] ApplicationDocsType\_Lookup) LIMIT 5)

1. **Query: Write a View ApplDocs\_Detail using ApplicationDocs and INNER JOIN and retrieve all the possible values limited by 5.**

CREATE VIEW ApplDocs\_Detail AS

SELECT

appdocs.ApplicationDocsTitle,

appdocs.ApplicationDocsURL,

app.SponsorName

FROM ApplicationDocs AS appdocs

INNER JOIN Applications AS app

ON appdocs.ApplNo = app.ApplNo

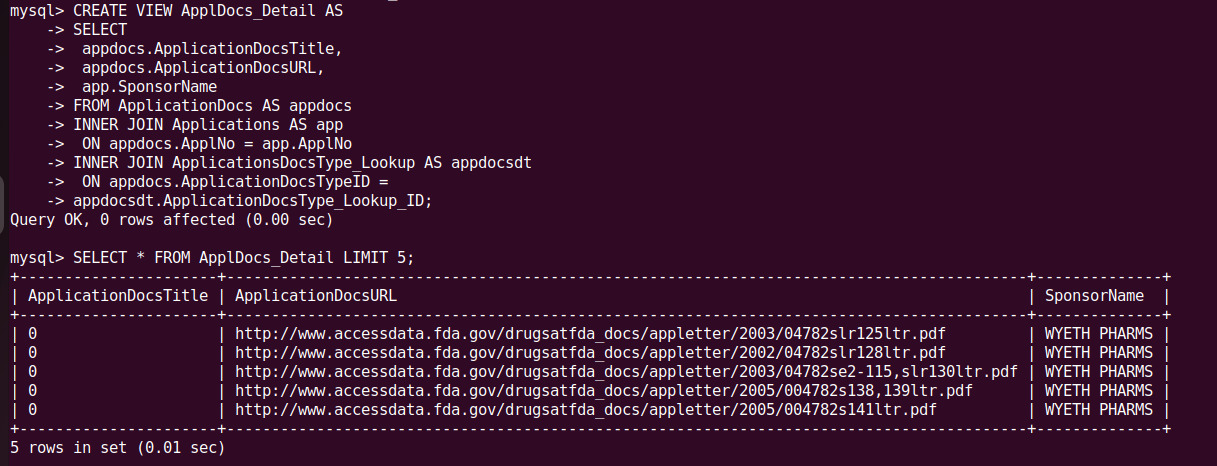
INNER JOIN ApplicationsDocsType\_Lookup AS appdocsdt

ON appdocs.ApplicationDocsTypeID =

appdocsdt.ApplicationDocsType\_Lookup\_ID;

SELECT \* FROM ApplDocs\_Detail LIMIT 5;

Result:



Relational Expression:

Let's denote the creation of the "ApplDocs\_Detail" view as V1:

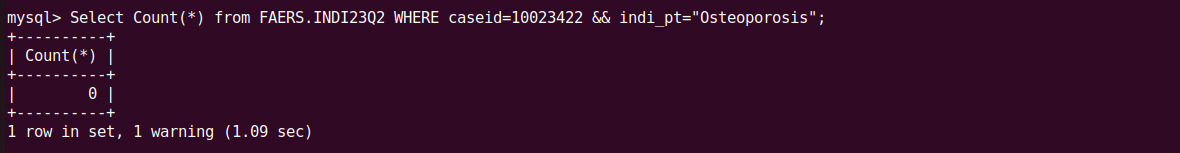
V1:= πApplicationDocsTitle, ApplicationDocsURL, SponsorName((ApplicationDocs ⨝[ApplNo = ApplNo] Applications) ⨝[ApplicationDocsTypeID = ApplicationDocsType\_Lookup\_ID] ApplicationsDocsType\_Lookup)

Now, the SQL query to select from the "ApplDocs\_Detail" view:

π \* (ApplDocs\_Detail) LIMIT 5

1. **Query: Display the count of all rows where caseid=10023422 and indi\_pt value is Osteoporosis.**

Select Count(\*) from FAERS.INDI23Q2 WHERE caseid=10023422 && indi\_pt="Osteoporosis";

Result: 

Relational Expression:

σ (caseid = 10023422 and indi\_pt = "Osteoporosis") (π \*) (FAERS.INDI23Q2)

1. Query: Retrieve all the values considering the caseid 10023422.

Select \* from FAERS.INDI23Q2 WHERE caseid=10023422;

Result:



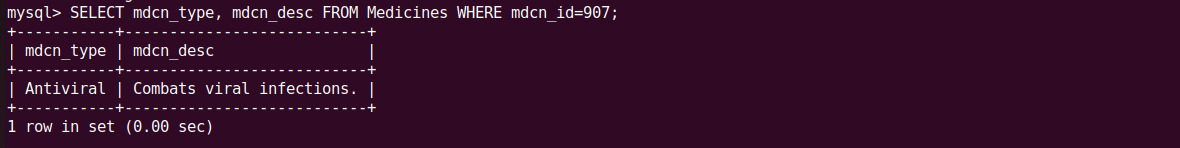
Relational Expression:

σ (caseid = 10023422) (π \*) (FAERS.INDI23Q2)

1. Query: Retrieve medicine type and description using the medicine id 907.

SELECT mdcn\_type, mdcn\_desc FROM Medicines WHERE mdcn\_id=907;

Result:



Relational Expression:

σ (mdcn\_id = 907) (π mdcn\_type, mdcn\_desc (Medicines))

1. Query: Display the SubmissionNo count along with the values grouped by SubmissionNo.

SELECT SubmissionNo, COUNT(\*) as Sub\_Count

FROM ApplicationDocs

GROUP BY SubmissionNo

LIMIT 5;

Result:



Relational Expression:

π SubmissionNo, COUNT() as Sub\_Count (γ SubmissionNo; COUNT() (ApplicationDocs))

**7. What have I learnt from the project?**

Relational systems and SQL enable users to retrieve specific subsets of data, aggregate data, and answer complex questions by composing queries.

Here in the project, I dealt with four different types of schemas and their ER Diagrams, converting them into real-time tables with the real-time data. The Bradley Hospital Management involved a huge datasets integration where I had to use multiple complex queries of MySQL. I had to load huge amount of data through SQL Load which took a pretty much less time which cannot be done with manually inserting the data.

Relational database systems are powerful tools for organizing and managing large volumes of data. They provide an efficient and structured way to store, retrieve, and manipulate data. Relational systems and SQL enable users to retrieve specific subsets of data, aggregate data, and answer complex questions by composing queries. Relational systems excel at managing data with complex relationships through techniques like JOIN operations, allowing data from multiple tables to be combined for analysis.

I had to interact with the database uniting all the schemas, performing data operations, retrieving, using DDL, DML. The relational algebra used in the project gives a deeper understanding of the all the queries.

Based on the experience with this project, I am quite sure that developing a real-time is not much difficult if everything is implemented in a right way.

**References:**

<https://www.fda.gov/drugs/drug-approvals-and-databases/drugsfda-data-files>

<https://www.freeprojectz.com/entity-relationship/hospital-management-system-er-diagram>

<https://www.freeprojectz.com/entity-relationship/pharmacy-management-system-er-diagram>