ITEC 5020: Database Development Documentation (Week 3)

This document details the development of the relational database for the **"Hypotify Clinical Insights Bot"** project within the new, dedicated schema, **artificial_emr1**. This approach was chosen to ensure **no changes were made to the pre-existing artificial_emr schema**, preserving the larger dataset for future use in ITEC 5025. The schema design, normalization process, and complete SQL scripts are provided below.

1. Database Schema Design and Normalization

The database is designed to manage complex Electronic Medical Records (EMR) data efficiently, following the principles of **normalization** to eliminate data redundancy and anomalies. The design is based on the four core entities identified in the project concept documentation.

A. Normalization Process

The goal of this process is to achieve **Third Normal Form (3NF)** by systematically addressing data dependencies.

Normal Form	Required Modifications	Resulting Tables and Dependencies
Unnormalized Form (UNF)	The single table contains **repeating groups** for lab observations and diagnosis.	**EMR_MASTER:** {PatientID, PatientGender,..., AdmissionID,..., {LabName, LabValue,...}, DiagnosisCode,...}
First Normal Form (1NF)	**Remove repeating groups** and ensure all columns contain atomic, single values.	**EMR_1NF:** All attributes are present, but the table still has partial and transitive dependencies on the composite key.
Second Normal	**Remove partial dependencies** (attributes dependent on only part of the composite key). Patient demographics depend only on	**PATIENT** (PatientID is PK) **ADMISSION** (AdmissionID is PK, PatientID is FK)

Form (2NF)	PatientID; Admission details depend only on AdmissionID.	**CLINICAL_DATA** (Lab and Diagnosis data)
Third Normal Form (3NF)	**Remove transitive dependencies** (attributes dependent on a non-key attribute). DiagnosisDescription depends on DiagnosisCode.	**DIAGNOSIS** and **LAB_OBSERVATION** tables are separated to achieve the final normalized structure.

B. Final Normalized Schema (3NF)

The four resulting entities, which are **fully compatible** with the large artificial EMR dataset, form the final schema:

Entity	Primary Key (PK)	Foreign Keys (FKs)	Rationale for Data Type/Key Selection
PATIENT	PatientID (VARCHAR(50))	N/A	VARCHAR is used for flexibility, supporting the artificial patient IDs.
ADMISSION	AdmissionID (INT)	PatientID	INT provides fast integer lookups for the unique admission ID. PatientID establishes the **1:M** relationship.
DIAGNOSIS	DiagnosisRecordID (INT Auto_Inc)	AdmissionID	A **surrogate key** is used as the PK for high efficiency. AdmissionID links the diagnosis to the encounter.
LAB_OBSERVATION	LabRecordID (BIGINT Auto_Inc)	AdmissionID	BIGINT is chosen to accommodate the **107 million records** present in the full database. AdmissionID links the lab result to the encounter.

C. Entity-Relationship Diagram (ERD)

2. SQL Scripts for Development and Management

This section contains the comprehensive SQL script used to develop the database infrastructure in the isolated **artificial_emr1** schema.

A. SQL Schema Creation

The script begins by creating and utilizing the **artificial_emr1** schema, ensuring **artificial_emr is untouched**.

```
SQL

-- ------------------------------------------------------
-- DATABASE SETUP: Targeting artificial_emr1
-- ------------------------------------------------------
CREATE DATABASE IF NOT EXISTS artificial_emr1;
USE artificial_emr1;


-- Table 1: PATIENTCREATE TABLE IF NOT EXISTS PATIENT (
    PatientID VARCHAR(50) PRIMARY KEY,
    PatientGender VARCHAR(10) NOT NULL,
    PatientDateOfBirth DATE,
    PatientRace VARCHAR(50),
    PatientMaritalStatus VARCHAR(20),
    PatientLanguage VARCHAR(20),
    PatientPovertyPct DECIMAL(5, 2)
);

-- Table 2: ADMISSION
CREATE TABLE IF NOT EXISTS ADMISSION (
    AdmissionID INT PRIMARY KEY,
    PatientID VARCHAR(50) NOT NULL,
    AdmissionStartDate DATETIME,
    AdmissionEndDate DATETIME,
    FOREIGN KEY (PatientID) REFERENCES PATIENT(PatientID)
);

-- Table 3: DIAGNOSIS
CREATE TABLE IF NOT EXISTS DIAGNOSIS (
    DiagnosisRecordID INT PRIMARY KEY AUTO_INCREMENT,
    AdmissionID INT NOT NULL,
```

```
    PrimaryDiagnosisCode VARCHAR(10) NOT NULL,
    PrimaryDiagnosisDescription VARCHAR(255),
    FOREIGN KEY (AdmissionID) REFERENCES ADMISSION(AdmissionID)
);

-- Table 4: LAB_OBSERVATION
CREATE TABLE IF NOT EXISTS LAB_OBSERVATION (
    LabRecordID BIGINT PRIMARY KEY AUTO_INCREMENT,
    AdmissionID INT NOT NULL,
    LabName VARCHAR(100) NOT NULL,
    LabValue DECIMAL(10, 3),
    LabUnits VARCHAR(20),
    LabDateTime DATETIME,
    FOREIGN KEY (AdmissionID) REFERENCES ADMISSION(AdmissionID)
);
```

B. Data Population (Sample Records)

This script populates the tables with **more than five sample records** for demonstration purposes.

```
SQL

-- Insert into PATIENT (6 records)
INSERT INTO PATIENT (PatientID, PatientGender, PatientDateOfBirth, PatientRace,
PatientMaritalStatus, PatientLanguage, PatientPovertyPct) VALUES
('P001', 'Female', '1955-03-15', 'White', 'Married', 'English', 12.50),
('P002', 'Male', '1980-11-20', 'African American', 'Single', 'English', 25.10),
('P003', 'Female', '1992-07-01', 'Asian', 'Divorced', 'Spanish', 18.00),
('P004', 'Male', '1975-01-25', 'White', 'Married', 'English', 9.30),
('P005', 'Female', '1960-05-10', 'African American', 'Widowed', 'English', 31.75),
('P006', 'Male', '2000-02-02', 'White', 'Single', 'Icelandic', 5.00);

-- Insert into ADMISSION (7 records)
INSERT INTO ADMISSION (AdmissionID, PatientID, AdmissionStartDate,
AdmissionEndDate) VALUES
(1001, 'P001', '2023-01-10 10:00:00', '2023-01-15 15:30:00'),
(1002, 'P001', '2023-05-20 14:00:00', '2023-05-21 09:00:00'),
(1003, 'P002', '2023-08-01 11:30:00', '2023-08-05 10:00:00'),
(1004, 'P003', '2023-09-15 08:00:00', '2023-09-17 12:00:00'),
(1005, 'P004', '2023-10-05 18:00:00', '2023-10-10 13:00:00'),
```

```
(1006, 'P004', '2024-01-10 09:00:00', '2024-01-11 11:00:00'),
(1007, 'P005', '2024-02-14 16:00:00', '2024-02-20 14:00:00');

-- Insert into DIAGNOSIS (7 records)
INSERT INTO DIAGNOSIS (AdmissionID, PrimaryDiagnosisCode,
PrimaryDiagnosisDescription) VALUES
(1001, 'I10', 'Essential (primary) hypertension'),
(1002, 'J45.909', 'Unspecified asthma, uncomplicated'),
(1003, 'E11.9', 'Type 2 diabetes mellitus without complications'),
(1004, 'N39.0', 'Urinary tract infection, site not specified'),
(1005, 'I50.9', 'Heart failure, unspecified'),
(1006, 'R10.9', 'Unspecified abdominal pain'),
(1007, 'E78.5', 'Hyperlipidemia, unspecified');

-- Insert into LAB_OBSERVATION (10 records)
INSERT INTO LAB_OBSERVATION (AdmissionID, LabName, LabValue, LabUnits,
LabDateTime) VALUES
(1001, 'METABOLIC: SODIUM', 138.0, 'mmol/L', '2023-01-11 08:00:00'),
(1001, 'METABOLIC: POTASSIUM', 4.1, 'mmol/L', '2023-01-11 08:00:00'),
(1003, 'METABOLIC: GLUCOSE', 215.5, 'mg/dL', '2023-08-01 12:00:00'),
(1003, 'CBC: HEMOGLOBIN', 14.2, 'g/dL', '2023-08-02 09:00:00'),
(1005, 'METABOLIC: BUN', 35.0, 'mg/dL', '2023-10-06 10:00:00'),
(1007, 'METABOLIC: GLUCOSE', 98.0, 'mg/dL', '2024-02-15 07:00:00'),
(1007, 'CBC: WHITE BLOOD CELL COUNT', 11.5, '10^3/uL', '2024-02-15 07:00:00'),
(1007, 'CBC: HEMATOCRIT', 40.0, '%', '2024-02-15 07:00:00'),
(1006, 'METABOLIC: CREATININE', 1.1, 'mg/dL', '2024-01-10 15:00:00'),
(1006, 'METABOLIC: ALBUMIN', 4.5, 'g/dL', '2024-01-10 15:00:00');
```

C. Queries and Database Management Operations

The following SQL demonstrates the required retrieval and management operations on the small sample dataset.

```
SQL

-- ------------------------------------------------
-- QUERIES AND MANAGEMENT OPERATIONS
-- ------------------------------------------------

-- 1. List data based on a single key: Retrieve all admissions for Patient P004.
SELECT * FROM ADMISSION WHERE PatientID = 'P004';
```

```sql
-- 2. List data based on a number of keys and criteria: Retrieve lab data for P002 where glucose is > 200.



SELECT
    P.PatientID, L.LabName, L.LabValue, L.LabUnits
FROM PATIENT P
JOIN ADMISSION A ON P.PatientID = A.PatientID
JOIN LAB_OBSERVATION L ON A.AdmissionID = L.AdmissionID
WHERE P.PatientID = 'P002' AND L.LabName = 'METABOLIC: GLUCOSE' AND L.LabValue > 200;

-- 3. Search for records that contain unique characteristics: Search for all patients with 'hypertension'.
SELECT
    P.PatientID, P.PatientGender, D.PrimaryDiagnosisDescription
FROM PATIENT P
JOIN ADMISSION A ON P.PatientID = A.PatientID
JOIN DIAGNOSIS D ON A.AdmissionID = D.AdmissionID
WHERE D.PrimaryDiagnosisDescription LIKE '%hypertension%';

-- 4. Calculate the total number of records given a unique characteristic: Total count for 'METABOLIC: GLUCOSE'.
SELECT COUNT(LabRecordID) AS TotalGlucoseRecords
FROM LAB_OBSERVATION
WHERE LabName = 'METABOLIC: GLUCOSE';

-- 5. Add new records: Adds new Patient P007 and Admission 1008.
INSERT INTO PATIENT (PatientID, PatientGender, PatientDateOfBirth, PatientRace, PatientMaritalStatus, PatientLanguage, PatientPovertyPct) VALUES
('P007', 'Male', '1945-09-12', 'Asian', 'Married', 'English', 15.00);

INSERT INTO ADMISSION (AdmissionID, PatientID, AdmissionStartDate, AdmissionEndDate) VALUES
(1008, 'P007', '2024-03-01 12:00:00', '2024-03-03 14:00:00');

-- 6. Update current record: Update Patient P005's Marital Status.
UPDATE PATIENT
SET PatientMaritalStatus = 'Single'
```

```
WHERE PatientID = 'P005';

-- 7. Remove a record: Remove the most recently added patient (P007) and admission (1008).
DELETE FROM DIAGNOSIS WHERE AdmissionID = 1008;
DELETE FROM LAB_OBSERVATION WHERE AdmissionID = 1008;
DELETE FROM ADMISSION WHERE AdmissionID = 1008;
DELETE FROM PATIENT WHERE PatientID = 'P007';
```

References:

Kartoun, U. (2018). EMRBots: a 100,000-patient database [Dataset]. Figshare. https://doi.org/10.6084/m9.figshare.7040198

Elmasri, R., & Navathe, S. B. (2016). Fundamentals of Database Systems (7th ed.). Pearson.

Health Level Seven International (HL7). (2023). HL7 FHIR Standards for Electronic Medical Records. https://www.hl7.org/fhir/

Murphy, S. N., Weaver, C. A., & Mendis, M. (2019). Electronic health records and clinical data warehouses. In J. H. Holmes (Ed.), Clinical Research Informatics. Springer. https://doi.org/10.1007/978-3-319-98779-0_6

IBM. (2022). Best Practices for Designing Relational Databases in Healthcare Systems. IBM Knowledge Center.

GoogleGemini 2.5 Flash

ChatGPT 5