

Applying Data Profiling and Operational Data Store Skills

Business Case Application:

You are a data analyst for a major hospital management information system (HMIS) provider. Your company has multiple system deployments over multiple hospital functions and your customers are located across North America. Recently, you were embedded within the sales and marketing function of the company, with a direct report to the Vice-President of Sales and Marketing (VP).

The VP has given you a file of current customers (e.g., HMIS.db) that was generated from an industry service bureau (ISB). Your company has entered into a contract with the ISB to provide quarterly snapshots of HMIS market share among US hospitals. The VP would like to develop a system for tracking sales leads and managing the daily/weekly/monthly efforts of the sales force, which could in turn be compared against the ISB snapshot. The VP asked you to analyze the ISB data as a possible basis for a data warehousing effort to support company strategic analysis of existing customers' HMIS architecture and potential revenue streams. The VP would like to better understand the ISB data as it relates to your customers. The VP has indicated that your recommendations and your data analysis will be used in the planning and analysis phase of re-engineering the data collection process between the point-of-contact for the HMIS customers and the sales force.

Based on your knowledge of data profiling, you will need to analyze and profile the ISB data set. You must perform column, dependency, and redundancy profiling within the ISB data set. You will also need to make recommendations with respect to an operational data store that would correspond to the VP's request. You must identify a basic data model that could yield an operational data store to meet the indicated business needs as depicted by the VP's requests. Be sure to specify the class of ODS needed to implement the data model.

All analysis and work will need to be documented in a report format that could be forwarded to the VP as your recommendations. The VP has asked that you provide your recommendations to him by the end of the week.

Deliverables:

- Relevant column profiling results for the ISB data set.
- Relevant dependency profiling results.
- Relevant redundancy profiling results.
- Potential ODS data model and update frequency.

**Example Column Data Profile:**

Attribute: *Application.LEADS.HMIS*

Type: *Text, Alphabetic (14 to 57), VCHAR(57)*

Domain: *{Computerized Practitioner Order Entry (CPOE),
Electronic Data Interchange (EDI) - Clearing House Vendor,
Enterprise EMR,
Enterprise Resource Planning,
Executive Information System}*

Pattern: *Domain contents*

Records: *20,540 (none null)*

4,036 - Electronic Data Interchange

4,051 - Enterprise Resource Planning

4,106 - Executive Information System

4,160- Computerized Practitioner Order Entry

4,187 - Enterprise (mode) (median)

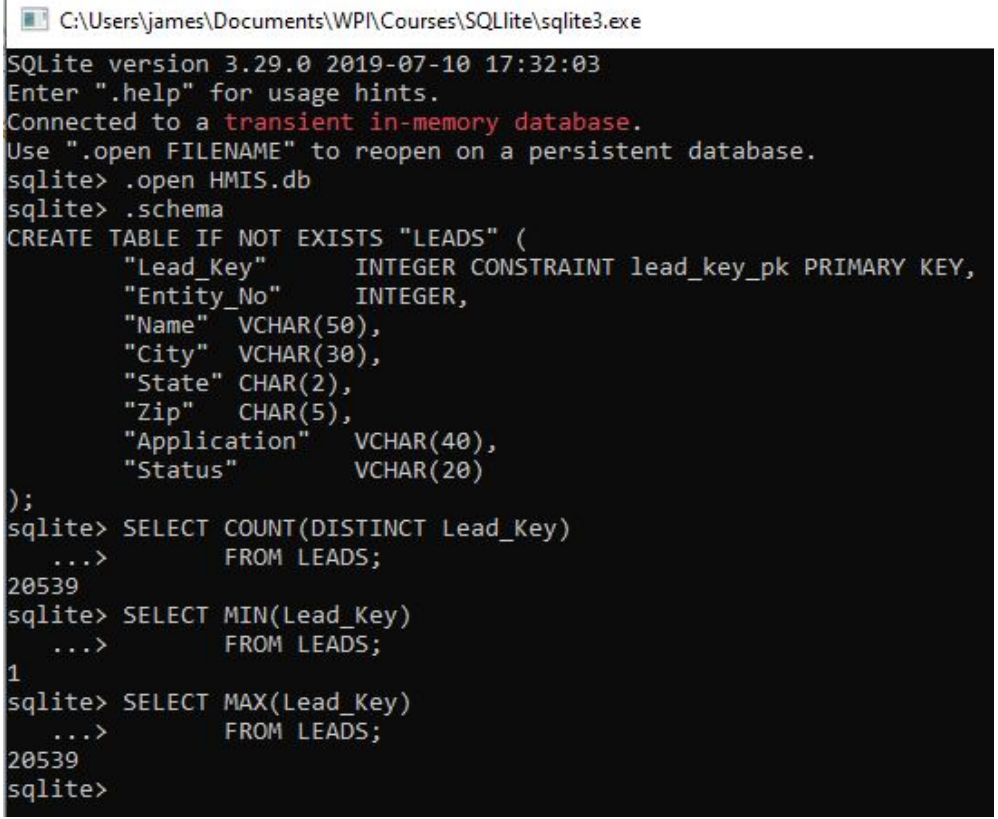
Dependencies: *Entity #, Entity Name, Zip, Phone*

**Exercise:**

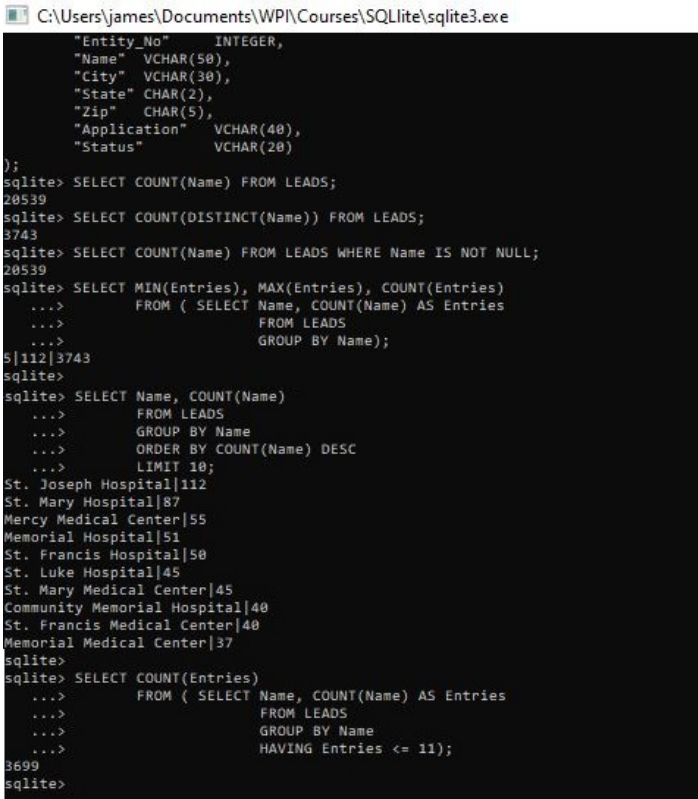
- 1) Using the CreateDBHMIS.sql script, create the HMIS.db database in SQLite (found in Week 03 Module, Tutorial Two Assignment), or download HMIS.db from the Tutorial Two Assignment.
- 2) Using the HMIS.db database file, profile each of the columns in the LEADS table.
- 3) Identify dependency profiling among the columns in the LEADS table.
- 4) Identify redundancy profiling among the columns in the LEADS table.
- 5) Using the LEADS table of the HMIS.db, identify other tables and columns that salesmen may need to propose an overall ODS data model. Identify the ODS class based on the update frequency needed and justify your proposal recommendation according to Module 3 content.
- 6) Organize deliverables 1 to 5 into a single PDF and upload it via the Tutorial Two assignment link by the due date.

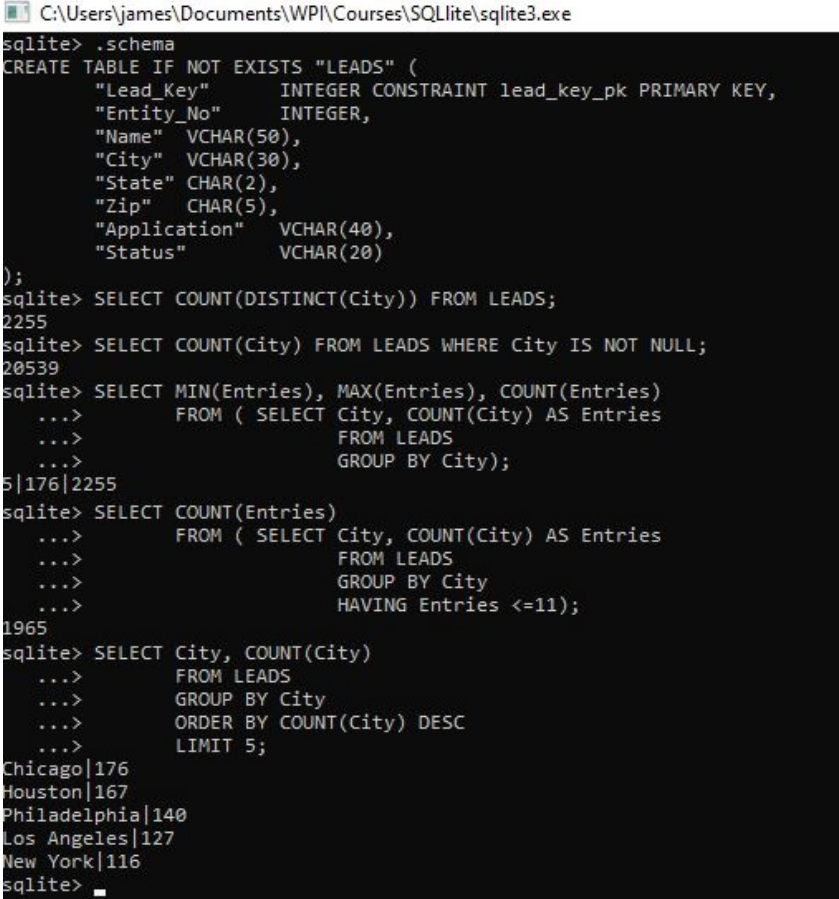
Answers:

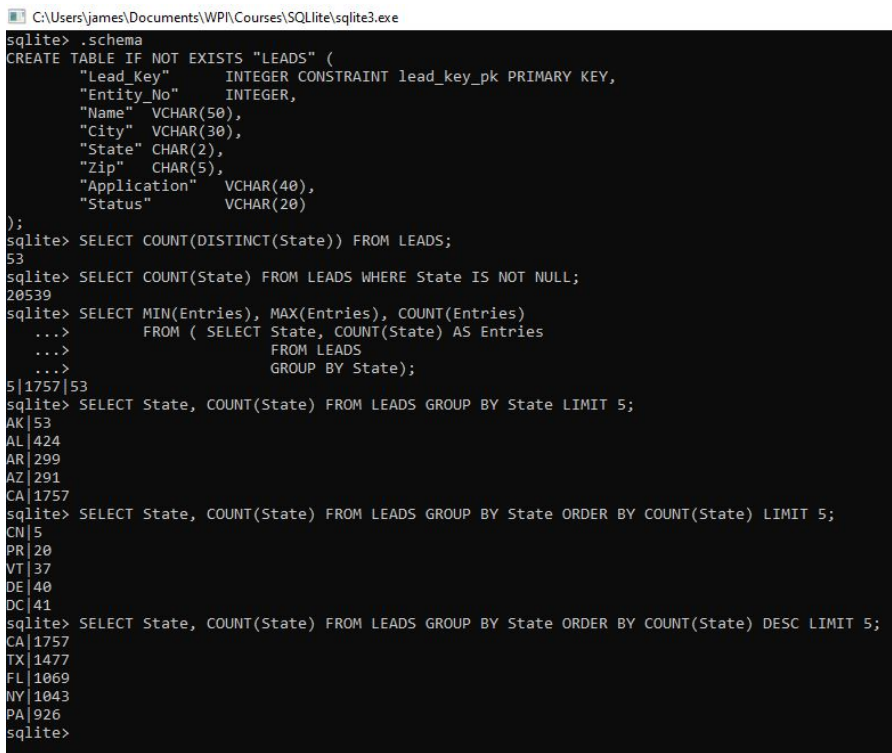
Column profiling on Lead_Key, Entity_No, Name, City, State, Zip, Application, and Status, as well as dependency and redundancy profiling, should follow content in the tables below.

Attribute	HMIS.LEADS.Lead_Key
Type	INTEGER
Domain	Numeric 1 to 20539
Pattern	Sequentially incremented by 1 for each record
# Records	20,539 unique; Minimum – 1; Maximum – 20539; No instances are Null; The mode is 1; Records 10,269 and 10,270 are the median entries.
Dependencies	None
Redundancies	None
SQL Scripts Used	 <pre> C:\Users\james\Documents\WPI\Courses\SQLite\sqlite3.exe SQLite version 3.29.0 2019-07-10 17:32:03 Enter ".help" for usage hints. Connected to a transient in-memory database. Use ".open FILENAME" to reopen on a persistent database. sqlite> .open HMIS.db sqlite> .schema CREATE TABLE IF NOT EXISTS "LEADS" ("Lead_Key" INTEGER CONSTRAINT lead_key_pk PRIMARY KEY, "Entity_No" INTEGER, "Name" VARCHAR(50), "City" VARCHAR(30), "State" CHAR(2), "Zip" CHAR(5), "Application" VARCHAR(40), "Status" VARCHAR(20)); sqlite> SELECT COUNT(DISTINCT Lead_Key) ...> FROM LEADS; 20539 sqlite> SELECT MIN(Lead_Key) ...> FROM LEADS; 1 sqlite> SELECT MAX(Lead_Key) ...> FROM LEADS; 20539 sqlite> </pre>

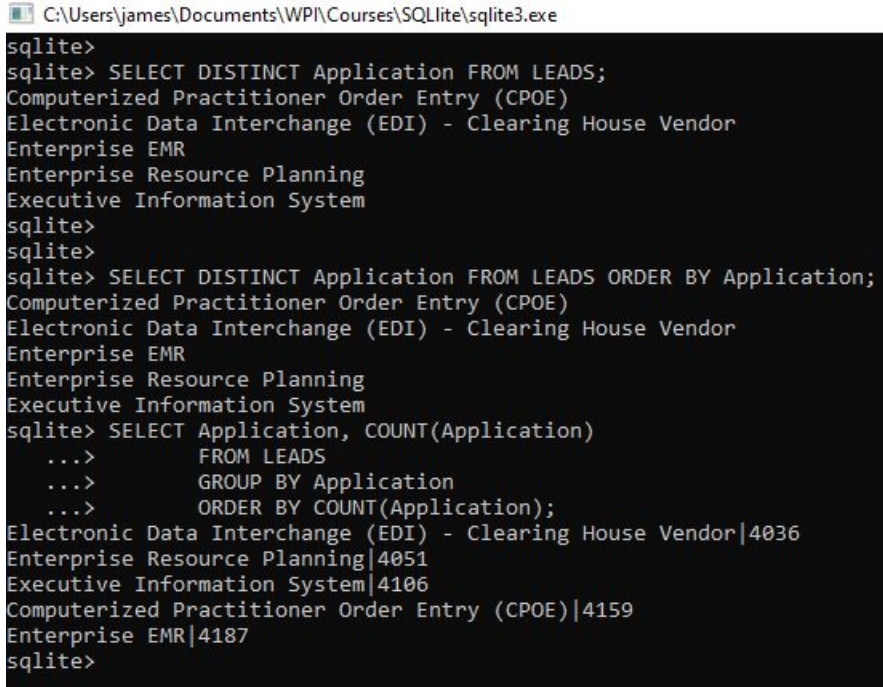
Attribute	HMIS.LEADS.Entity_No
Type	INTEGER
Domain	Numeric
Pattern	Randomly incremented for unique hospital names sorted alphabetically; Repeats where a hospital has multiple HMISs
# Records	20,539 total with duplicates; Minimum = 2; Maximum = 100063557; No instances are null; 4,003 distinct; Entity_No 2619, 2621, 22161, 42056, and 100055263 all have 9 duplicates (mode); Entity_No 20297 is the median.
Dependencies	Lead_Key, Name
Redundancies	Name
SQL Scripts Used	<pre> sqlite> sqlite> .schema CREATE TABLE IF NOT EXISTS "LEADS" ("Lead_Key" INTEGER CONSTRAINT lead_key_pk PRIMARY KEY, "Entity_No" INTEGER, "Name" VARCHAR(50), "City" VARCHAR(30), "State" CHAR(2), "Zip" CHAR(5), "Application" VARCHAR(40), "Status" VARCHAR(20)); sqlite> SELECT COUNT(DISTINCT Entity_No) ...> FROM LEADS; 4003 sqlite> SELECT MIN(Entity_No) ...> FROM LEADS; 2 sqlite> SELECT MAX(Entity_No) ...> FROM LEADS; 100063557 sqlite> sqlite> sqlite> SELECT Entity_No ...> FROM LEADS ...> WHERE Lead_Key = 10269 OR Lead_Key = 10270; 20297 20297 sqlite> sqlite> SELECT Entity_No, Entries ...> FROM (SELECT Entity_No, COUNT(Entity_No) AS Entries ...> FROM LEADS ...> GROUP BY Entity_No ...> HAVING Entries > 5 ...> ORDER BY Entries DESC) ...> ORDER BY Entries DESC; 2619 9 2621 9 22161 9 42056 9 100055263 9 </pre>

Attribute	HMIS.LEADS.Name
Type	VARCHAR(50)
Domain	Alphabet; Names of hospitals having HMIS
Pattern	Repeats where a hospital has multiple HMIS ($5 \leq n \leq 11$); Repeats across hospitals similarly named in different cities across states; 3,699 hospitals have unique names
# Records	20,539 total with duplicates; No instances are NULL; 3,743 distinct hospital names; 'St. Joseph Hospital' is duplicated most across cities and states (mode); ; 'McDonough District Hospital' and 'McDowell ARH Hospital' are the medians.
Dependencies	Lead-Key, Entity_No
Redundancies	Entity_No
SQL Scripts Used	 <pre> C:\Users\james\Documents\WPI\Courses\SQLite\sqlite3.exe "Entity_No" INTEGER, "Name" VARCHAR(50), "City" VARCHAR(30), "State" CHAR(2), "Zip" CHAR(5), "Application" VARCHAR(40), "Status" VARCHAR(20)); sqlite> SELECT COUNT(Name) FROM LEADS; 20539 sqlite> SELECT COUNT(DISTINCT(Name)) FROM LEADS; 3743 sqlite> SELECT COUNT(Name) FROM LEADS WHERE Name IS NOT NULL; 20539 sqlite> SELECT MIN(Entries), MAX(Entries), COUNT(Entries) ...> FROM (SELECT Name, COUNT(Name) AS Entries ...> FROM LEADS ...> GROUP BY Name); \$[112]3743 sqlite> sqlite> SELECT Name, COUNT(Name) ...> FROM LEADS ...> GROUP BY Name ...> ORDER BY COUNT(Name) DESC ...> LIMIT 10; St. Joseph Hospital 112 St. Mary Hospital 87 Mercy Medical Center 55 Memorial Hospital 51 St. Francis Hospital 50 St. Luke Hospital 45 St. Mary Medical Center 45 Community Memorial Hospital 40 St. Francis Medical Center 40 Memorial Medical Center 37 sqlite> sqlite> SELECT COUNT(Entries) ...> FROM (SELECT Name, COUNT(Name) AS Entries ...> FROM LEADS ...> GROUP BY Name ...> HAVING Entries <= 11); 3699 sqlite> </pre>

Attribute	HMIS.LEADS.City
Type	VARCHAR(50)
Domain	Alphabet; Names of cities where hospitals are located; None are NULL
Pattern	Repeats where a hospital has multiple HMIS ($5 \leq n \leq 11$); Repeats where a city has multiple hospitals. Repeats across similarly named cities in different states; 1,965 cities have unique names across all states.
# Records	20,539 total with duplicates; 2,255 distinct city names; 'Chicago' is the city with the hospital HMISs (mode); 'Long Beach' is the median.
Dependencies	Lead-Key, Entity_No, Name
Redundancies	Zip
SQL Scripts Used	 <pre> C:\Users\james\Documents\WPI\Courses\SQLite\sqlite3.exe sqlite> .schema CREATE TABLE IF NOT EXISTS "LEADS" ("Lead_Key" INTEGER CONSTRAINT lead_key_pk PRIMARY KEY, "Entity_No" INTEGER, "Name" VARCHAR(50), "City" VARCHAR(30), "State" CHAR(2), "Zip" CHAR(5), "Application" VARCHAR(40), "Status" VARCHAR(20)); sqlite> SELECT COUNT(DISTINCT(City)) FROM LEADS; 2255 sqlite> SELECT COUNT(City) FROM LEADS WHERE City IS NOT NULL; 20539 sqlite> SELECT MIN(Entries), MAX(Entries), COUNT(Entries) ...> FROM (SELECT City, COUNT(City) AS Entries ...> FROM LEADS ...> GROUP BY City); 5 176 2255 sqlite> SELECT COUNT(Entries) ...> FROM (SELECT City, COUNT(City) AS Entries ...> FROM LEADS ...> GROUP BY City ...> HAVING Entries <=11); 1965 sqlite> SELECT City, COUNT(City) ...> FROM LEADS ...> GROUP BY City ...> ORDER BY COUNT(City) DESC ...> LIMIT 5; Chicago 176 Houston 167 Philadelphia 140 Los Angeles 127 New York 116 sqlite> </pre>

Attribute	HMIS.LEADS.State
Type	VARCHAR(2)
Domain	Alphabet; Names of U.S. states {AK, AL, ..., WY} where hospitals are located as well as DC (District of Columbia), PR (Puerto Rico), and CN (Canada); None are NULL
Pattern	Repeats when a hospital has multiple HMIS ($5 \leq n \leq 11$); Repeats when a state has multiple cities with multiple hospitals.
# Records	20,539 total with duplicates; 53 distinct states; CN includes 5 entries; PR has 20 entries; DC has 41 entries; CA is the state with the most hospital HMISs (mode); MO is the median, being in records 10,269 and 10,270.
Dependencies	Lead_Key, Entity_No, Name, City
Redundancies	Zip
SQL Scripts Used	 <pre> C:\Users\james\Documents\WPI\Courses\SQLite\sqlite3.exe sqlite> .schema CREATE TABLE IF NOT EXISTS "LEADS" ("Lead_Key" INTEGER CONSTRAINT lead_key_pk PRIMARY KEY, "Entity_No" INTEGER, "Name" VARCHAR(50), "City" VARCHAR(30), "State" CHAR(2), "Zip" CHAR(5), "Application" VARCHAR(40), "Status" VARCHAR(20)); sqlite> SELECT COUNT(DISTINCT(State)) FROM LEADS; 53 sqlite> SELECT COUNT(State) FROM LEADS WHERE State IS NOT NULL; 20539 sqlite> SELECT MIN(Entries), MAX(Entries), COUNT(Entries) ...> FROM (SELECT State, COUNT(State) AS Entries ...> FROM LEADS ...> GROUP BY State); 5 1757 53 sqlite> SELECT State, COUNT(State) FROM LEADS GROUP BY State LIMIT 5; AK 53 AL 424 AR 299 AZ 291 CA 1757 sqlite> SELECT State, COUNT(State) FROM LEADS GROUP BY State ORDER BY COUNT(State) LIMIT 5; CN 5 PR 20 VT 37 DE 40 DC 41 sqlite> SELECT State, COUNT(State) FROM LEADS GROUP BY State ORDER BY COUNT(State) DESC LIMIT 5; CA 1757 TX 1477 FL 1069 NY 1043 PA 926 sqlite> </pre>

Attribute	HMIS.LEADS.Zip
Type	VARCHAR(5)
Domain	Alpha-Numeric; U.S. and Canadian postal zip codes where hospitals are located; None are NULL
Pattern	Repeats when a hospital has multiple HMIS ($5 \leq n \leq 11$); Repeats when multiple hospitals with HMISs are located in the same postal zip code.
# Records	20,539 total with duplicates; 3556 distinct zip codes; CN zip code H3G1A has 5 entries; Hospitals with HMISs in zip code 77030 has the most entries (mode); Zip codes 53188 and 53209 are the medians.
Dependencies	Lead_Key, Entity_No, Name
Redundancies	City, State
SQL Scripts Used	<pre> C:\Users\james\Documents\WPI\Courses\SQLite\sqlite3.exe sqlite> .schema CREATE TABLE IF NOT EXISTS "LEADS" ("Lead_Key" INTEGER CONSTRAINT lead_key_pk PRIMARY KEY, "Entity_No" INTEGER, "Name" VARCHAR(50), "City" VARCHAR(30), "State" CHAR(2), "Zip" CHAR(5), "Application" VARCHAR(40), "Status" VARCHAR(20)); sqlite> SELECT COUNT(DISTINCT(Zip)) FROM LEADS; 3556 sqlite> SELECT COUNT(Zip) FROM LEADS WHERE Zip IS NOT NULL; 20539 sqlite> SELECT MIN(Entries), MAX(Entries), COUNT(Entries) ...> FROM (SELECT Zip, COUNT(Zip) AS Entries ...> FROM LEADS ...> GROUP BY Zip); 5 53 3556 sqlite> SELECT Zip, COUNT(Zip) ...> FROM LEADS ...> GROUP BY Zip ...> ORDER BY COUNT(Zip) ASC ...> LIMIT 5; 10011 5 10034 5 10035 5 10037 5 10305 5 sqlite> SELECT Zip, COUNT(Zip) ...> FROM LEADS ...> GROUP BY Zip ...> ORDER BY COUNT(Zip) DESC ...> LIMIT 5; 77030 53 78229 44 75235 37 40202 33 70115 30 sqlite> sqlite> SELECT Zip, COUNT(Zip) ...> FROM LEADS ...> WHERE State = "CN" ...> GROUP BY Zip ...> ORDER BY COUNT(Zip); H3G1A 5 sqlite> </pre>

Attribute	HMIS.LEADS.Application
Type	VARCHAR(40)
Domain	Alphabet; Type of HMIS application {Computerized Practitioner Order Entry (CPOE), Electronic Data Interchange (EDI) – Clearing House Vendor, Enterprise EMR, Enterprise Resource Planning, Executive Information System}; None are NULL
Pattern	Repeats when multiple hospitals have the same application
# Records	20,539 total with duplicates; {Computerized Practitioner Order Entry (CPOE)=4159, Electronic Data Interchange (EDI) – Clearing House Vendor=4036, Enterprise EMR=4187 (mode & median), Enterprise Resource Planning=4051, Executive Information System=4106}
Dependencies	Lead_Key, Entity_No, Name
Redundancies	None
SQL Scripts Used	 <pre> C:\Users\james\Documents\WPI\Courses\SQLite\sqlite3.exe sqlite> sqlite> SELECT DISTINCT Application FROM LEADS; Computerized Practitioner Order Entry (CPOE) Electronic Data Interchange (EDI) - Clearing House Vendor Enterprise EMR Enterprise Resource Planning Executive Information System sqlite> sqlite> sqlite> SELECT DISTINCT Application FROM LEADS ORDER BY Application; Computerized Practitioner Order Entry (CPOE) Electronic Data Interchange (EDI) - Clearing House Vendor Enterprise EMR Enterprise Resource Planning Executive Information System sqlite> SELECT Application, COUNT(Application) ...> FROM LEADS ...> GROUP BY Application ...> ORDER BY COUNT(Application); Electronic Data Interchange (EDI) - Clearing House Vendor 4036 Enterprise Resource Planning 4051 Executive Information System 4106 Computerized Practitioner Order Entry (CPOE) 4159 Enterprise EMR 4187 sqlite> </pre>

Attribute	HMIS.LEADS.Status
Type	VARCHAR(40)
Domain	Alphabet; Status of HMIS application {Contracted/Not Yet Installed, Installation in Process, Live and Operational, Not Automated, Not Reported, Not Yet Contracted, To Be Replaced} at a given hospital; None are NULL
Pattern	Repeats when multiple hospitals have the same application status
# Records	20,539 total with duplicates; { Contracted/Not Yet Installed=1,118, Installation in Process=229, Live and Operational=6,923, Not Automated=11,041 (mode & median), Not Reported=724, Not Yet Contracted=193, To Be Replaced=311}
Dependencies	Application
Redundancies	None
SQL Scripts Used	<pre> sqlite> sqlite> .schema CREATE TABLE IF NOT EXISTS "LEADS" ("Lead_Key" INTEGER CONSTRAINT lead_key_pk PRIMARY KEY, "Entity_No" INTEGER, "Name" VARCHAR(50), "City" VARCHAR(30), "State" CHAR(2), "Zip" CHAR(5), "Application" VARCHAR(40), "Status" VARCHAR(20)); sqlite> SELECT COUNT(DISTINCT(Status)) FROM LEADS; 7 sqlite> SELECT Status, COUNT(Status) ...> FROM LEADS ...> GROUP BY Status ...> ORDER BY Status; Contracted/Not Yet Installed 1118 Installation in Process 229 Live and Operational 6923 Not Automated 11041 Not Reported 724 Not Yet Contracted 193 To be Replaced 311 sqlite> sqlite> SELECT COUNT(Status) FROM LEADS WHERE Status IS NOT NULL; 20539 sqlite> </pre>

Answers:

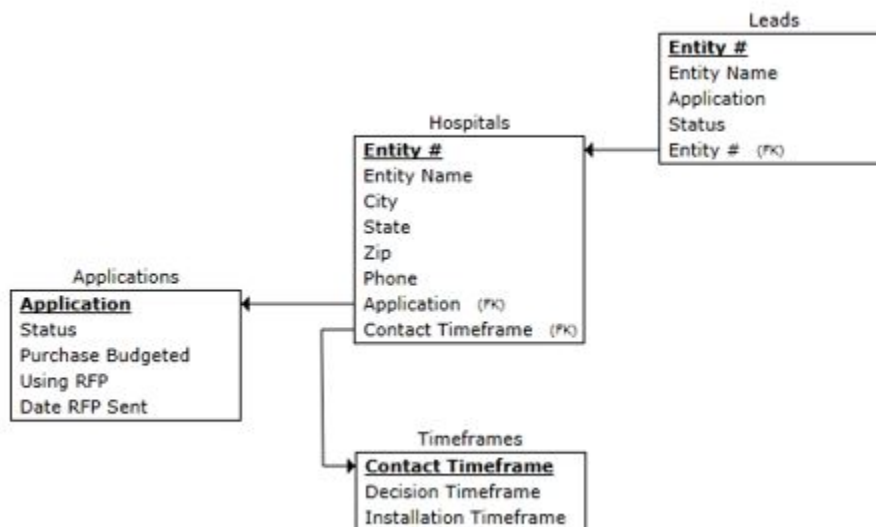
ODS data model and update frequency

Individual student ODS data model responses will vary, yet the response should include at least the minimum number of attributes found in the HMIS.db. Likewise, the ODS update frequency response (Class I, II, or III) can vary among students, but should be supported in the context of the student's response.

```

sqlite> .schema
CREATE TABLE IF NOT EXISTS "LEADS" (
  "Lead_Key"      INTEGER CONSTRAINT lead_key_pk PRIMARY
  "Entity_No"     INTEGER,
  "Name"          VARCHAR(50),
  "City"          VARCHAR(30),
  "State"         CHAR(2),
  "Zip"           CHAR(5),
  "Application"   VARCHAR(40),
  "Status"        VARCHAR(20)
);
  
```

Sample ODS data model example:



Class I systems provide synchronous or near-synchronous updates.

Class II ODS requires more frequent updates (perhaps hourly) to reflect changes.

Class III ODS is updated daily and provides reports about business transactions for that day, such as sales totals or orders filled.

Class IV ODS adds capacity for more interaction between the data warehouse or data mart and the ODS.

