Global Disease Tracking Database

A Reference Manual for Database Administrators

Doug Strouth

CIS228

10 December 2014

Table of Contents

Summary 3

Table creation 3

Primary Key Sequences 5

Inserting Data into Tables 5

Indexing Tables 6

Common Queries 7

Joining data from multiple tables 10

Selected single row functions 15

Group functions 20

Sub queries 23

Appendix A: Table Constraints 25

Appendix B: Database Sequences and Their Associated Columns 29

Appendix C: Table Indices 30

Appendix D: Table Descriptions 31

Appendix E: Common Simple Queries 37

Appendix F: Full Database Creation Script and All Reference Queries 38

# Summary

This document seeks to outline the structure and common uses of the Global Disease Tracking (GDT) database. The process of creating the database, the data relationships between tables, and the types of queries the database will handle are outlined in order to provide a reliable reference for administrators of the database. DBMs may wish to print or bookmark the appendices for quick reference when providing support.

# Table creation[[1]](#footnote-1)

The tables are organized in six groups in order to facilitate creation with constraints. The tables with zero foreign keys are created first and each group progresses through the tables with the most reliance on previously completed tables.

The groups are as follows:

Group 1

PATIENT

REGION

COUNTRY

DISEASE\_TYPE

AFTEREFFECT

CAT\_EVENT

SYMPTOM \_LIST

Group 2

LOCATION

DISEASE

Group 3

HEALTH\_PROFESSION

EXPECTED\_SYMPTOM

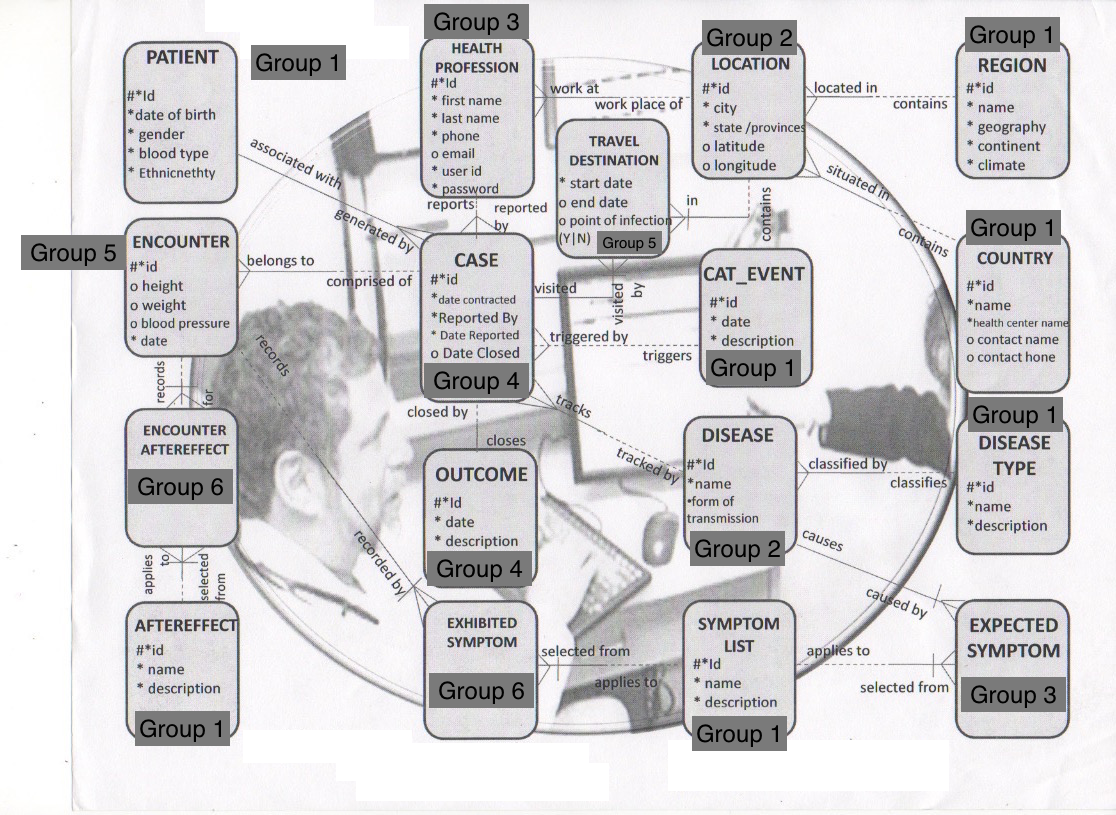
Group 4

CASE[[2]](#footnote-2)

OUTCOME

Group 5

ENCOUNTER

TRAVEL\_DESTINATION

Group 6

ENCOUNTER\_AFTEREFFECT

EXHIBITED\_SYMPTOM

Image1- A visual map of the tables and their groups

A complete list of every table, its fields, their types, and the constraint types on each field can be found in Appendix D

A list of the names of each of the constraints can be found in Appendix A. In order to accommodate the length limit in APEX for constraint names some words like location or effect were shortened.

To add a new constraint follow this example:

Ex:

*ALTER TABLE COUNTRY*

*ADD CONSTRAINT COUNTRY\_HEALTH\_CENTER;*

## Primary Key Sequences[[3]](#footnote-3)

Sequences were created for all of the primary keys of the tables in the database, except the tables consisting only of a composite primary key. All of the sequence values are five digit numbers in order to accommodate a large amount of data incrementing by one, starting with the value 1. The sequences do not have caches since there is global access to the database and number errors could occur and the sequences do not cycle in order to avoid confusion when referencing between tables.

An example of a sequence creation statement and alteration of the sequence definition is given below. The creation statement makes a purposeful mistake in order to show how to alter the sequence in the next statement and removing the cache. A full list of the sequences in the database can be found in Appendix B.

*7A)*

*CREATE SEQUENCE PATIENT\_ID\_SEQ*

*INCREMENT BY 1*

*START WITH 1*

*CACHE 5*

*NOCYCLE;*

*7B)*

*ALTER SEQUENCE PATIENT\_ID\_SEQ*

*NOCACHE;*

## Inserting Data into Tables[[4]](#footnote-4)

Insert data into the tables with an insert statement. Any of the insert statements in the database creation script serve as good examples like the one below. Just be sure to when referencing foreign keys that the data they rely upon has already been inserted.

Ex:

*INSERT INTO HEALTH\_PROFESSION*

*VALUES(HEALTH\_PROFESSION\_ID\_SEQ.NEXTVAL,2,'DANIELS','MELANIE', '(610)-459-0647', 'MDANIELS@PSYCH.ORG','MDANIELS','HOSPITABLEHOSPITAL');*

To enter data into specific columns in a table be sure to specify the columns before the values statement

Ex:

*INSERT INTO COUNTRY (COUNTRY\_ID,NAME, HEALTH\_CENTER,CONTACT\_NAME)*

*VALUES(COUNTRY\_ID\_SEQ.NEXTVAL,'AMERICA', 'BATES HOSPITAL','CONSTANCE PETERSEN');*

## Indexing Tables[[5]](#footnote-5)

A B-Tree index was added to the NAME field in the COUNTRY table because names will be a frequent search term.

To add more B-Tree indexes use the following format

Ex:

*8A B-Tree Indexes*

*CREATE INDEX COUNTRY\_NAME\_IDX*

*ON COUNTRY(NAME);*

A Bitmap index was added to the BLOOD\_TYPE field of the PATIENT table because of the limited number of options the field could be.

To add more Bitmap indexes use the following format

Ex:

*8B Bitmap Indexes*

*CREATE BITMAP INDEX PATIENT\_BLOOD\_TYPE\_IDX*

*ON PATIENT(BLOOD\_TYPE);*

A function-based index was created on the difference between the DATE\_REPORTED and DATE\_CONTRACTED in the CASE table because this is an important metric of how quickly diseases are dealt with.

Ex:

*8C Function-based indexes*

*CREATE INDEX CASE\_CONTRACT\_REPORT\_IDX*

*ON CASE(DATE\_CONTRACTED - DATE\_REPORTED);*

A list of all the indexes in the database can be found in Appendix C.

# Common Queries[[6]](#footnote-6)

This section outlines the extensive functionality of the GDT and its ability to handle complex queries that managers may have to help users troubleshoot. For the simpler queries specifically outlined in the database project outline refer to Appendix E.

The following query allows for the search of patients born within a set time period and ordered in descending order by date of birth and then by gender.

Ex:

*9A WHERE CLAUSE, 9B BETWEEN AND OPERATOR, 11Dd TO\_DATE FUNCTION, 9Ga SECONDARY SORT IN ORDER BY CLAUSE*

*SELECT DOB, PATIENT\_ID, GENDER, BLOOD\_TYPE*

*FROM PATIENT*

*WHERE DOB BETWEEN TO\_DATE('JANUARY 01,0060','MONTH DD, YY') AND TO\_DATE('DECEMBER 31, 0064','MONTH DD, YY')*

*ORDER BY DOB DESC, GENDER;*

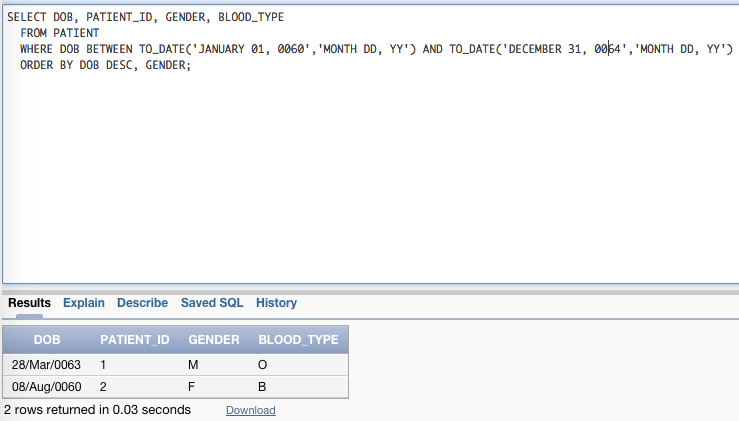
**

Figure 1

The following query will display patients with a blood type of A or O, or that are female. This could be used to quickly narrow down potential blood donors.

Ex:

*9C IN OPERATOR, 9E LOGICAL OPERATORS, 9Gb ORDER BY SELECT ORDER*

*SELECT PATIENT\_ID,BLOOD\_TYPE, GENDER*

*FROM PATIENT*

*WHERE BLOOD\_TYPE IN ('A','O')*

*OR GENDER='F'*

*ORDER BY 1 DESC;*

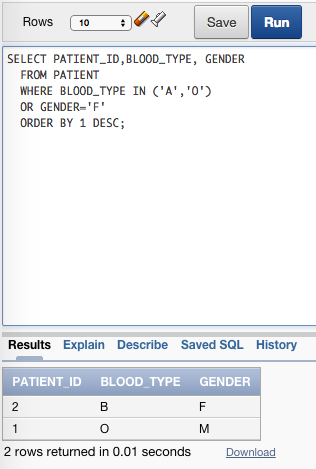


Figure 2

The following query display the full name, user id, and password of a health professional whose last name starts with B. This could come in handy when recovering passwords for doctors who are so tired they can only remember their last names.

Ex:

*9D LIKE OPERATORS, 11Ac INITCAP CASE CONVERSION,11Bh CONCAT FUNCTION*

*SELECT CONCAT(INITCAP(FIRST),CONCAT(' ',INITCAP(LAST))) AS FULL\_NAME, USER\_ID, PASSWORD*

*FROM HEALTH\_PROFESSION*

*WHERE LAST LIKE 'B%';*

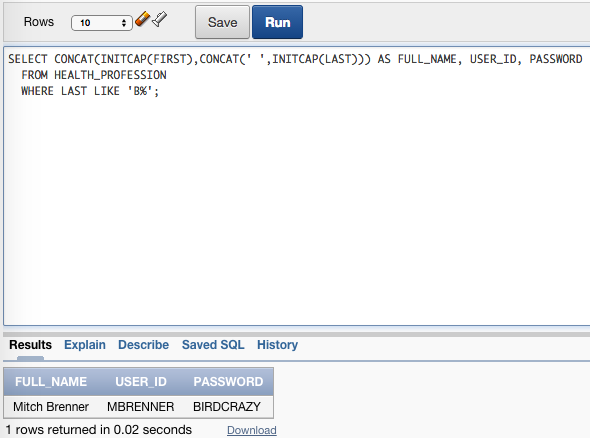


Figure 3

The following query helps quickly identify a list of patients and their disease that have yet to have their cases closed.

Ex:

*9F TREATMENT OF NULL VALUES*

*SELECT PATIENT\_ID,DISEASE\_ID,DATE\_CONTRACTED,REPORTED\_BY, DATE\_REPORTED*

*FROM CASE*

*WHERE DATE\_CLOSED IS NULL;*

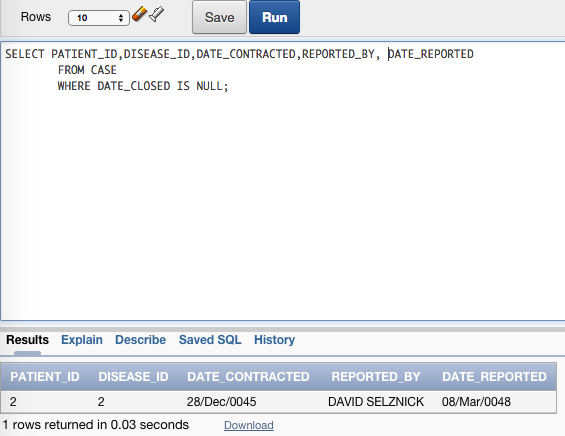
**

Figure 4

## Joining data from multiple tables[[7]](#footnote-7)

Beyond the simpler queries it will be necessary for users to display information pulled from more than one table. Common types of joins that could be seen are listed below.

A search of patients by country.

Ex:

*10A CARTESIAN JOIN*

*SELECT PATIENT\_ID, NAME*

*FROM PATIENT CROSS JOIN COUNTRY;*

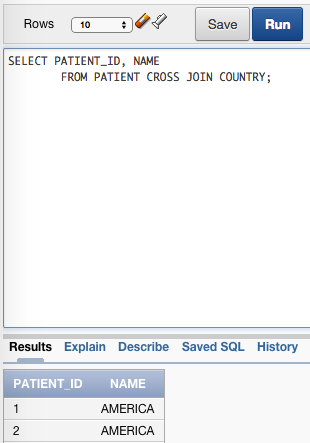


Figure 5

To join and view columns from tables with a common column use an equality join. Using a natural join statement is the easiest way to do this. This one shows the location of patient, their travel destinations, and whether these destinations are points of interest.

Ex:

*10B EQUALITY JOINS*

*SELECT LOCATION\_ID, TRAVEL\_DESTINATION\_ID, POI*

*FROM LOCATION NATURAL JOIN TRAVEL\_DESTINATION;*

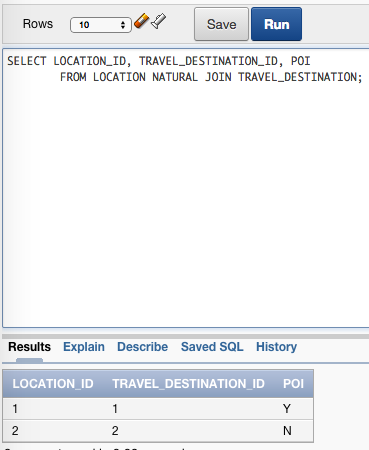


Figure 6

To join columns from tables that do not share a common column use a non-equality join.

Below is an example of a join showing the patient id and their blood pressure.

Ex:

*10C NON-EQUALITY JOIN*

*SELECT PATIENT\_ID, BLOOD\_PRESSURE*

*FROM PATIENT JOIN ENCOUNTER*

*ON DATE\_OF\_ENCOUNTER>DOB;*

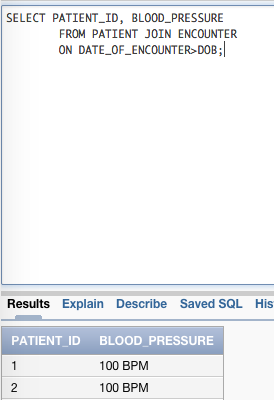
**

Figure 7

To join a table to itself and find references a self-join is used. They would typically be structured like this example, but this one is not particularly useful due to the limited info in the database at this time. DBAs may find that as tables become more fleshed out this structure may become more useful as patients switch doctors or regions

Ex:

*10D SELF JOIN*

*SELECT h.FIRST, h.LAST, p.LAST*

*FROM HEALTH\_PROFESSION h JOIN HEALTH\_PROFESSION p*

*ON p.HEALTH\_PROFESSION\_ID = r.HEALTH\_PROFESSION\_ID;*

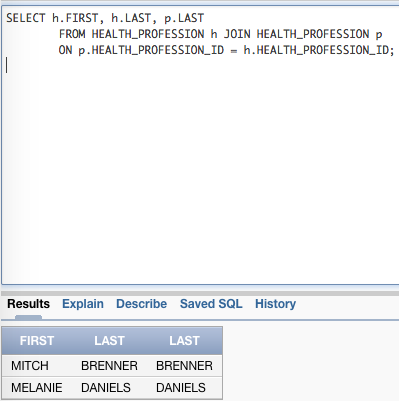


Figure 8

To display results with a null column when there is no matching record in the other table use an outer join. This outer join helps display a list of locations and the full name of the health professionals in the database.

Ex:

*10E FULL OUTER JOIN*

*SELECT LAST, FIRST, LOCATION\_ID*

*FROM LOCATION l FULL OUTER JOIN HEALTH\_PROFESSION h*

*USING (LOCATION\_ID);*

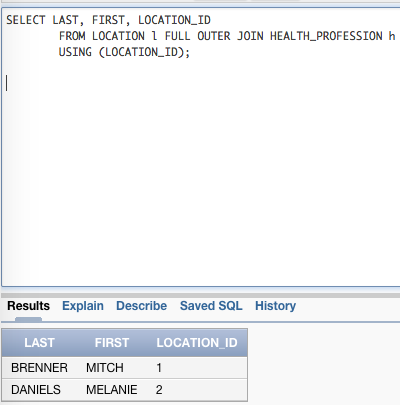


Figure 9

## Selected single row functions[[8]](#footnote-8)

Determine length of a field use the length function. The example given is the length of user passwords and their corresponding user id.

Ex:

*11Bb LENGTH FUNCTION*

*SELECT USER\_ID,LENGTH(PASSWORD)*

*FROM HEALTH\_PROFESSION;*

**

Figure 10

To select only part of a varchar2 field use the substring function. If the first 3 letters of a doctor’s last name were needed for prescription verification this query could be used.

Ex:

*11Ba SUBSTRING FUNCTION*

*SELECT SUBSTR(LAST,1,3)*

*FROM HEALTH\_PROFESSION;*

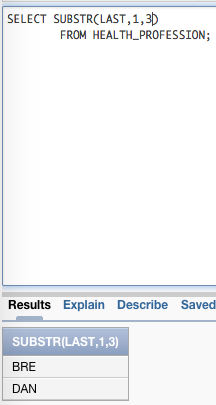
**

Figure 11

Rounding numbers could also be useful and would take this format.

Ex:

*11Ca ROUND FUNCTION*

*SELECT ROUND(PATIENT\_ID, 1)*

*FROM PATIENT;*

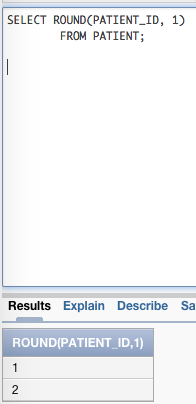
**

Figure 12

Similarly exponent operations would be entered like thus.

Ex:

*11Ca POWER FUNCTION*

*SELECT POWER(PATIENT\_ID, 3)*

*FROM PATIENT;*

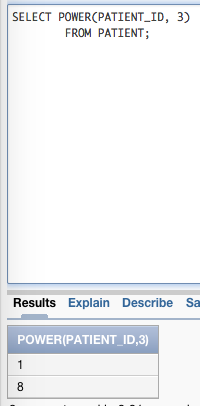
**

Figure 13

Date measurements are important when figuring out how urgent a case might be. This statement will add 3 months to the contraction date and round it to the last day of the month or a specific day of the week

Ex:

*11Db ADD\_MONTHS FUNCTION, 11Dc LAST\_DAY AND NEXT\_DAY FUNCTION*

*SELECT CASE\_ID, PATIENT\_ID, DATE\_CONTRACTED, LAST\_DAY(ADD\_MONTHS(DATE\_CONTRACTED,3) )"EXPECTED CLOSE DATE"*

*FROM CASE;*

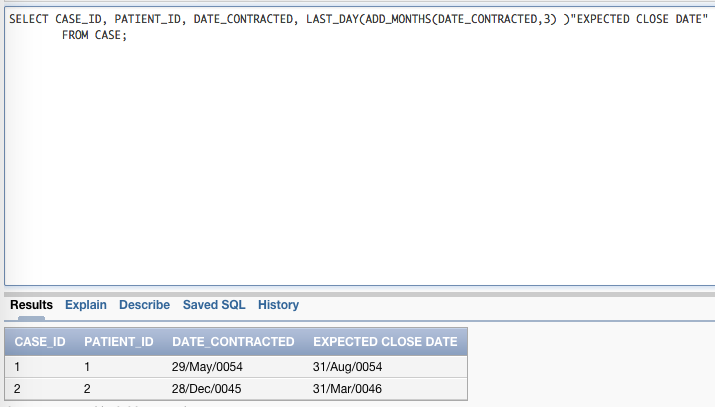
**

Figure 14

*SELECT CASE\_ID, PATIENT\_ID, DATE\_CONTRACTED, NEXT\_DAY(ADD\_MONTHS(DATE\_CONTRACTED,3), 1 )"EXPECTED CLOSE DATE"*

*FROM CASE;*

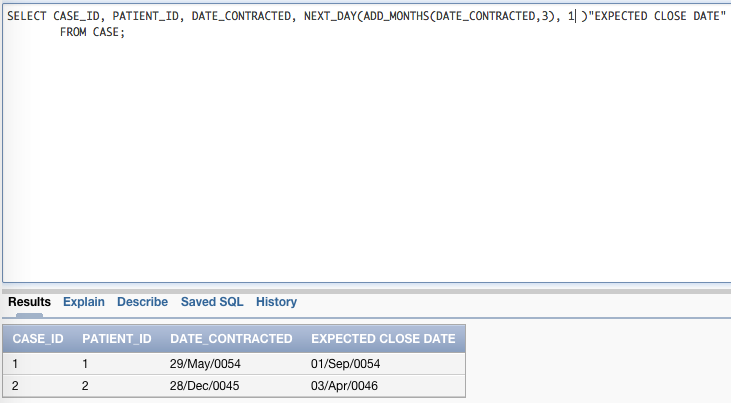
**

Figure 15

## Group functions[[9]](#footnote-9)

The number of patients in the database could be determined using the next query.

Ex:

*12C COUNT FUNCTION*

*SELECT COUNT(PATIENT\_ID) "NUMBER OF PATIENTS"*

*FROM PATIENT;*

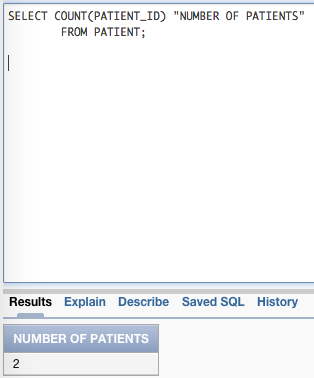


Figure 16

The max or min functions can be used to manipulate numeric data such as patient ids or number of visits

Ex:

*12E MIN FUNCTION, 12D MAX FUNCTION*

*SELECT BLOOD\_TYPE, MIN(PATIENT\_ID)*

*FROM PATIENT*

*GROUP BY BLOOD\_TYPE;*

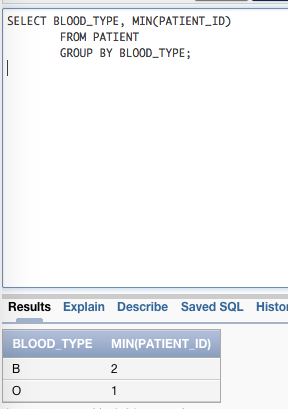
**

Figure 17

*SELECT BLOOD\_TYPE, MAX(PATIENT\_ID)*

*FROM PATIENT*

*GROUP BY BLOOD\_TYPE;*

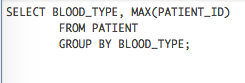


Figure 18

## Sub queries[[10]](#footnote-10)

More advanced users may need help creating single-row or multiple row queries.

To run a single-row sub query use a format like the one below

Ex:

*13A SINGLE-ROW SUB QUERY*

*SELECT BLOOD\_TYPE, PATIENT\_ID*

*FROM PATIENT*

*WHERE PATIENT\_ID=(SELECT MAX(PATIENT\_ID)*

*FROM PATIENT);*

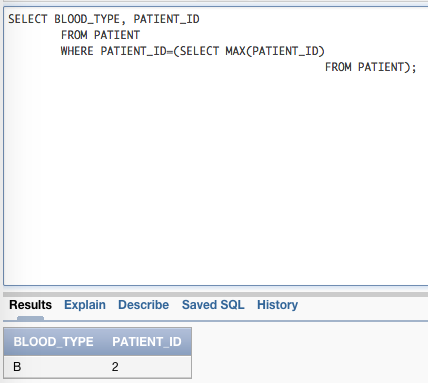
**

Figure 19

Multiple-row sub queries are a little more complicated. This example will cross-reference all patients and their blood types that are in the patient table and display them if they are also in the case table. Not all patients might have had a case file opened so this quickly narrows down that data.

Ex:

*13B MULTIPLE-ROW SUB QUERY*

*SELECT BLOOD\_TYPE, PATIENT\_ID*

*FROM PATIENT*

*WHERE PATIENT\_ID IN (SELECT PATIENT\_ID*

*FROM CASE);*

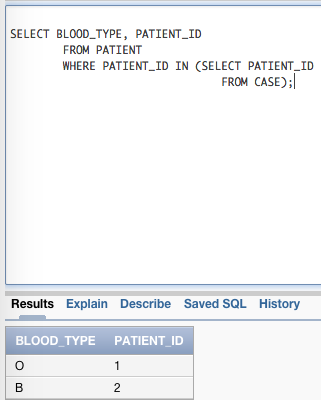
**

Figure 20

# Appendix A: Table Constraints

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Group** | **Table Name** | **Fields** | **Constraint Type** | **Constraint Name** |
| 1 | PATIENT | PATIENT\_ID | PRIMARY KEY | PATIENT\_ID\_PK |
|  |  | GENDER | CHECK ‘M’, ‘F’ | PATIENT\_GENDER\_CK |
|  |  |  |  |  |
| 1 | REGION | REGION\_ID | PRIMARY KEY | REGION\_ID\_PK |
|  |  | NAME | UNIQUE, NOT NULL | REGION\_NAME\_UK |
| 1 | COUNTRY | COUNTRY\_ID | PRIMARY KEY | COUNTRY\_ID\_PK |
|  |  | NAME | UNIQUE, NOT NULL | COUNTRY\_NAME\_UK |
| 1 | DISEASE\_TYPE | DISEASE\_TYPE\_ID | PRIMARY KEY | DISEASE\_TYPE\_ID\_PK |
| 1 | AFTEREFFECT | AFTEREFFECT\_ID | PRIMARY KEY | AFTEREFFECT\_ID\_PK |
| 1 | CAT\_EVENT | CAT\_EVENT\_ID | PRIMARY KEY | CAT\_EVENT\_ID\_PK |
| 1 | SYMPTOM\_LIST | SYMPTOM\_ID | PRIMARY KEY | SYMPTOM\_LIST\_ID\_PK |
|  |  |  |  |  |
| 2 | LOCATION | LOCATION\_ID | PRIMARY KEY | DISEASE\_ID\_PK |
|  |  | REGION\_ID | FOREIGN KEY | DISEASE\_TYPE\_ID\_FK |
| 2 | DISEASE | DISEASE\_ID | PRIMARY KEY | HEALTH\_PROFESSION\_ID\_PK |
|  |  | DISEASE\_TYPE\_ID | FOREIGN KEY | HEALTH\_PROFESSION\_LOC\_ID\_FK |
|  |  |  |  |  |
| 3 | HEALTH\_PROFESSION | HEALTH\_PROFESSION\_ID | PRIMARY KEY | HEALTH\_PROFESSION\_ID\_PK |
|  |  | LOCATION\_ID | FOREIGN KEY | HEALTH\_PROFESSION\_LOC\_ID\_FK |
|  |  | USER\_ID | UNIQUE | HEALTH\_PROFESSION\_USER\_ID\_UK |
| 3 | EXPECTED\_SYMPTOM | SYMPTOM\_ID, DISEASE\_ID | PRIMARY KEY | EXPECTED\_SYMPTOM\_PK |
|  |  | SYMPTOM\_ID | FOREIGN KEY | EXPECTED\_SYMPTOM\_SYMPTOM\_ID\_FK |
|  |  | DISEASE\_ID | FOREIGN KEY | EXPECTED\_SYMPTOM\_DISEASE\_ID\_FK |
|  |  |  |  |  |
| 4 | CASE | CASE\_ID | PRIMARY KEY | CASE\_ID\_PK |
|  |  | PATIENT\_ID | FOREIGN KEY | CASE\_PATIENT\_ID\_FK |
|  |  | HEALTH\_PROFESSION\_ID | FOREIGN KEY | CASE\_HEALTH\_PROFESSION\_ID\_FK |
|  |  | DISEASE\_ID | FOREIGN KEY | CASE\_DISEASE\_ID\_FK |
|  |  | CAT\_EVENT\_ID | FOREIGN KEY | CASE\_CAT\_EVEN\_ID\_FK |
|  |  | OUTCOME\_ID | FOREIGN KEY | CASE\_OUTCOME\_ID\_FK |
| 4 | OUTCOME | OUTCOME\_ID | PRIMARY KEY | OUTCOME\_ID\_PK |
|  |  | CASE\_ID | FOREIGN KEY | OUTCOME\_CASE\_ID\_FK |
| 5 | ENCOUNTER | ENCOUNTER\_ID | PRIMARY KEY | ENCOUNTER\_ID\_PK |
|  |  | CASE\_ID | FOREIGN KEY | ENCOUNTER\_CASE\_ID\_FK |
| 5 | TRAVEL\_DESTINATION | TRAVEL\_DESTINATION\_ID | PRIMARY KEY | TRAVEL\_DESTINATION\_ID\_PK |
|  |  | LOCATION\_ID | FOREIGN KEY | TRAVEL\_DESTINATION\_LOC\_ID\_FK |
|  |  | POI | CHECK ‘Y’, ’N' | TRAVEL\_DESTINATION\_POI\_CK |
|  |  |  |  |  |
| 6 | ENCOUNTER\_AFTEREFFECT | AFTEREFFECT\_ID, ENCOUNTER\_ID | PRIMARY KEY | ENCOUNTER\_AFTEREFFECT\_PK |
|  |  | AFTEREFFECT\_ID | FOREIGN KEY | ENCOUNTER\_AFTEREFFECT\_AE\_ID\_FK |
|  |  | ENCOUNTER\_ID | FOREIGN KEY | ENCOUNTER\_AFTEREFFECT\_E\_ID\_FK |
| 6 | EXHIBITED\_SYMPTOM | SYMPTOM\_ID, ENCOUNTER\_ID | PRIMARY KEY | EXHIBITED\_SYMPTOM\_PK |
|  |  | ENCOUNTER\_ID | FOREIGN KEY | EXHIBITED\_SYMPTOM\_E\_ID\_FK |
|  |  | SYMPTOM\_ID | FOREIGN KEY | EXHIBITED\_SYMPTOM\_S\_ID\_FK |

# Appendix B: Database Sequences and Their Associated Columns

|  |  |  |  |
| --- | --- | --- | --- |
| **Group** | **Table name** | **Sequence Name** | **Column Using Sequence** |
| 1 | PATIENT | PATIENT\_ID\_SEQ | PATIENT\_ID |
| 1 | REGION | REGION\_ID\_SEQ | REGION\_ID |
| 1 | COUNTRY | COUNTRY\_ID\_SEQ | COUNTRY\_ID |
| 1 | DISEASE\_TYPE | DISEASE\_TYPE\_ID\_SEQ | DISEASE\_TYPE\_ID |
| 1 | AFTEREFFECT | AFTEREFFECT\_ID\_SEQ | AFTEREFFECT\_ID |
| 1 | CAT\_EVENT | CAT\_EVENT\_ID\_SEQ | CAT\_EVENT\_ID |
| 1 | SYMPTOM\_LIST | SYMPTOM\_LIST\_ID\_SEQ | SYMPTOM\_LIST\_ID |
|  |  |  |  |
| 2 | LOCATION | LOCATION\_ID\_SEQ | LOCATION\_ID |
| 2 | DISEASE | DISEASE\_ID\_SEQ | DISEASE\_ID |
|  |  |  |  |
| 3 | HEALTH\_PROFESSION | HEALTH\_PROFESSION\_ID\_SEQ | HEALTH\_PROFESSION\_ID |
|  |  |  |  |
| 4 | CASE | CASE\_ID\_SEQ | CASE\_ID |
| 4 | OUTCOME | OUTCOME\_ID\_SEQ | OUTCOME\_ID |
|  |  |  |  |
| 5 | ENCOUNTER | ENCOUNTER\_ID\_SEQ | ENCOUNTER\_ID |
| 5 | TRAVEL\_DESTINATION | TRAVEL\_DESTINATION\_ID\_SEQ | TRAVEL\_DESTINATION\_ID |

# Appendix C: Table Indices

|  |  |  |  |
| --- | --- | --- | --- |
| **Group** | **Table** | **Index Name** | **Index Type** |
| 1 | PATIENT | PATIENT\_ID\_IDX | B-Tree |
| 1 | PATIENT | PATIENT\_BLOOD\_TYPE\_IDX | Bitmap |
| 4 | CASE | CASE\_CONTRACT\_REPORT\_IDX | Function-Based |

# Appendix D: Table Descriptions

| **Group** | **Table name** | **Fields** | **Type** | **Constraints[[11]](#footnote-11)** | **Foreign key to which table(s)** |
| --- | --- | --- | --- | --- | --- |
| 1 | PATIENT | PATIENT\_ID | NUMBER | PRIMARY KEY [[12]](#footnote-12) | CASE |
|  |  | DOB | DATE | NOT NULL[[13]](#footnote-13) |  |
|  |  | GENDER | CHAR | CHECK ‘M’, ‘F’[[14]](#footnote-14) |  |
|  |  | BLOOD\_TYPE | VARCHAR2 | NOT NULL |  |
|  |  | ETHNICITY | VARCHAR2 | NOT NULL |  |
|  |  |  |  |  |  |
| 1 | REGION | REGION\_ID | NUMBER | PRIMARY KEY | LOCATION |
|  |  | NAME | VARCHAR2 | UNIQUE, NOT NULL[[15]](#footnote-15) |  |
|  |  | GEOGRAPHY | VARCHAR2 | NOT NULL |  |
|  |  | CONTINENT | VARCHAR2 | NOT NULL |  |
|  |  | CLIMATE | VARCHAR2 | NOT NULL |  |
| 1 | COUNTRY | COUNTRY\_ID | NUMBER | PRIMARY KEY | LOCATION |
|  |  | NAME | VARCHAR2 | NOT NULL |  |
|  |  | HEALTH\_CENTER | VARCHAR2 | NOT NULL |  |
|  |  | CONTACT\_NAME | VARCHAR2 |  |  |
|  |  | CONTACT\_PHONE | VARCHAR2 |  |  |
| 1 | DISEASE\_TYPE | DISEASE\_TYPE\_ID | NUMBER | PRIMARY KEY | DISEASE |
|  |  | NAME | VARCHAR2 | NOT NULL |  |
|  |  | DESCRIPTION | VARCHAR2 | NOT NULL |  |
| 1 | AFTEREFFECT | AFTEREFFECT\_ID | NUMBER | PRIMARY KEY | ENCOUNTER\_AFTEREFFECT |
|  |  | NAME | VARCHAR2 | NOT NULL |  |
|  |  | DESCRIPTION | VARCHAR2 | NOT NULL |  |
| 1 | CAT\_EVENT | CAT\_EVENT\_ID | NUMBER | PRIMARY KEY | CASE |
|  |  | DATE\_OF\_EVENT | DATE | NOT NULL |  |
|  |  | DESCRIPTION | VARCHAR2 | NOT NULL |  |
| 1 | SYMPTOM\_LIST | SYMPTOM\_ID | NUMBER | PRIMARY KEY | EXPECTED\_SYMPTOM, EXHIBITED\_SYMPTOM |
|  |  | NAME | VARCHAR2 | NOT NULL |  |
|  |  | DESCRIPTION | VARCHAR2 | NOT NULL |  |
|  |  |  |  |  |  |
| 2 | LOCATION | LOCATION\_ID | NUMBER | PRIMARY KEY | HEALTH\_PROFESSION, TRAVEL\_DESTINATION |
|  |  | REGION\_ID | NUMBER | FOREIGN KEY[[16]](#footnote-16) |  |
|  |  | CITY | VARCHAR2 | NOT NULL |  |
|  |  | STATE | CHAR 2 PLACES | NOT NULL |  |
|  |  | LATITUDE | VARCHAR2 |  |  |
|  |  | LONGITUDE | VARCHAR2 |  |  |
| 2 | DISEASE | DISEASE\_ID | NUMBER | PRIMARY KEY | CASE, EXPECTED\_SYMPTOM |
|  |  | DISEASE\_TYPE\_ID | NUMBER | FOREIGN KEY |  |
|  |  | NAME | VARCHAR2 | NOT NULL |  |
|  |  | TRANSMISSION\_FORM | VARCHAR2 |  |  |
|  |  |  |  |  |  |
| 3 | HEALTH\_PROFESSION | HEALTH\_PROFESSION\_ID | NUMBER | PRIMARY KEY | CASE |
|  |  | LOCATION\_ID | NUMBER | FOREIGN KEY |  |
|  |  | LAST | VARCHAR2 | NOT NULL |  |
|  |  | FIRST | VARCHAR2 | NOT NULL |  |
|  |  | PHONE | VARCHAR2 | NOT NULL |  |
|  |  | EMAIL | VARCHAR2 |  |  |
|  |  | USER\_ID | VARCHAR2 | UNIQUE |  |
|  |  | PASSWORD | VARCHAR2 |  |  |
| 3 | EXPECTED\_SYMPTOM |  |  |  |  |
|  |  | SYMPTOM\_ID | NUMBER | PRIMARY KEY |  |
|  |  | DISEASE\_ID | NUMBER | PRIMARY KEY |  |
|  |  |  |  |  |  |
| 4 | CASE | CASE\_ID | NUMBER | PRIMARY KEY | TRAVEL\_DESTINATION |
|  |  | PATIENT\_ID | NUMBER | FOREIGN KEY |  |
|  |  | HEALTH\_PROFESSION\_ID | NUMBER | FOREIGN KEY |  |
|  |  | DISEASE\_ID | NUMBER | FOREIGN KEY |  |
|  |  | CAT\_EVENT\_ID | NUMBER | FOREIGN KEY |  |
|  |  | DATE\_CONTRACTED | DATE | NOT NULL |  |
|  |  | REPORTED\_BY | VARCHAR2 | NOT NULL |  |
|  |  | DATE\_REPORTED | DATE | NOT NULL |  |
|  |  | BETWEEN\_CONTRACTION\_AND\_REPORT[[17]](#footnote-17) |  |  |  |
|  |  | DATE\_CLOSED | DATE |  |  |
|  |  | OUTCOME\_ID | NUMBER | FOREIGN KEY |  |
| 4 | OUTCOME | OUTCOME\_ID | NUMBER | PRIMARY KEY | CASE |
|  |  | CASE\_ID | NUMBER | FOREIGN KEY |  |
|  |  | DATE\_OF\_OUTCOME | DATE | NOT NULL |  |
|  |  | DESCRIPTION | VARCHAR2 | NOT NULL |  |
|  |  |  |  |  |  |
| 5 | ENCOUNTER | ENCOUNTER\_ID | NUMBER | PRIMARY KEY | ENCOUNTER\_AFTEREFFECT |
|  |  | CASE\_ID | NUMBER | FOREIGN KEY |  |
|  |  | HEIGHT | VARCHAR2 |  |  |
|  |  | WEIGHT | NUMBER |  |  |
|  |  | BLOOD\_PRESSURE | VARCHAR2 |  |  |
|  |  | DATE\_OF\_ENCOUNTER | DATE | NOT NULL |  |
| 5 | TRAVEL\_DESTINATION | TRAVEL\_DESTINATION\_ID | NUMBER | PRIMARY KEY |  |
|  |  | LOCATION\_ID | NUMBER | FOREIGN KEY |  |
|  |  | START\_DATE | DATE | NOT NULL |  |
|  |  | END\_DATE | DATE |  |  |
|  |  | POI | CHAR | CHECK ‘Y’, ’N' |  |
|  |  |  |  |  |  |
| 6 | ENCOUNTER\_AFTEREFFECT |  |  |  |  |
|  |  | AFTEREFFECT\_ID | NUMBER | PRIMARY KEY |  |
|  |  | ENCOUNTER\_ID | NUMBER | PRIMARY KEY |  |
| 6 | EXHIBITED\_SYMPTOM |  |  |  |  |
|  |  | ENCOUNTER\_ID | NUMBER | PRIMARY KEY |  |
|  |  | SYMPTOM\_ID | NUMBER | PRIMARY KEY |  |

# Appendix E: Common Simple Queries

SELECT PATIENT WITH MULTIPLE DISEASES

*SELECT PATIENT\_ID, CASE\_ID, DISEASE\_ID*

*FROM PATIENT NATURAL JOIN CASE FULL OUTER JOIN DISEASE*

*USING (DISEASE\_ID)*

*ORDER BY PATIENT\_ID DESC;*

SELECT PATIENTS PER HEALTH PROFESSIONAL GROUPED BY REGION

*SELECT PATIENT\_ID, HEALTH\_PROFESSION\_ID, REGION\_ID*

*FROM CASE NATURAL JOIN HEALTH\_PROFESSION NATURAL JOIN LOCATION;*

LIST OF DISEASES WITHIN A REGION

*SELECT DISEASE\_ID, REGION\_ID*

*FROM DISEASE NATURAL JOIN CASE NATURAL JOIN HEALTH\_PROFESSION NATURAL JOIN LOCATION*

*ORDER BY REGION\_ID;*

LIST OF SYMPTOMS AND RELATED DISEASES

*SELECT SYMPTOM\_ID, NAME, DESCRIPTION, DISEASE\_ID*

*FROM EXPECTED\_SYMPTOM NATURAL JOIN SYMPTOM\_LIST;*

# Appendix F: Full Database Creation Script and All Reference Queries

/\* RESET TABLE STRUCTURE \*/

DROP TABLE PATIENT CASCADE CONSTRAINTS;

DROP TABLE REGION CASCADE CONSTRAINTS;

DROP TABLE COUNTRY CASCADE CONSTRAINTS;

DROP TABLE DISEASE\_TYPE CASCADE CONSTRAINTS;

DROP TABLE AFTEREFFECT CASCADE CONSTRAINTS;

DROP TABLE CAT\_EVENT CASCADE CONSTRAINTS;

DROP TABLE SYMPTOM\_LIST CASCADE CONSTRAINTS;

DROP TABLE LOCATION CASCADE CONSTRAINTS;

DROP TABLE DISEASE CASCADE CONSTRAINTS;

DROP TABLE HEALTH\_PROFESSION CASCADE CONSTRAINTS;

DROP TABLE EXPECTED\_SYMPTOM CASCADE CONSTRAINTS;

DROP TABLE CASE CASCADE CONSTRAINTS;

DROP TABLE OUTCOME CASCADE CONSTRAINTS;

DROP TABLE ENCOUNTER CASCADE CONSTRAINTS;

DROP TABLE TRAVEL\_DESTINATION CASCADE CONSTRAINTS;

DROP TABLE ENCOUNTER\_AFTEREFFECT CASCADE CONSTRAINTS;

DROP TABLE EXHIBITED\_SYMPTOM CASCADE CONSTRAINTS;

/\*RESET SEQUENCES\*/

DROP SEQUENCE PATIENT\_ID\_SEQ;

DROP SEQUENCE REGION\_ID\_SEQ;

DROP SEQUENCE COUNTRY\_ID\_SEQ;

DROP SEQUENCE DISEASE\_TYPE\_ID\_SEQ;

DROP SEQUENCE AFTEREFFECT\_ID\_SEQ;

DROP SEQUENCE CAT\_EVENT\_ID\_SEQ;

DROP SEQUENCE SYMPTOM\_LIST\_ID\_SEQ;

DROP SEQUENCE LOCATION\_ID\_SEQ;

DROP SEQUENCE DISEASE\_ID\_SEQ;

DROP SEQUENCE HEALTH\_PROFESSION\_ID\_SEQ;

DROP SEQUENCE CASE\_ID\_SEQ;

DROP SEQUENCE OUTCOME\_ID\_SEQ;

DROP SEQUENCE ENCOUNTER\_ID\_SEQ;

DROP SEQUENCE TRAVEL\_DESTINATION\_ID\_SEQ;

/\* 1 Table creations \*/

/\* GROUP 1 \*/

/\* 2E NOT NULL CONSTRAINT \*/

/\*2A PRIMARY KEY \*/

/\*2D CHECK CONSTRAINT \*/

CREATE TABLE PATIENT

(PATIENT\_ID NUMBER(5),

DOB DATE NOT NULL,

GENDER CHAR(1) NOT NULL,

BLOOD\_TYPE VARCHAR2(5) NOT NULL,

ETHNICITY VARCHAR2(20) NOT NULL,

CONSTRAINT PATIENT\_ID\_PK PRIMARY KEY (PATIENT\_ID),

CONSTRAINT PATIENT\_GENDER\_CK CHECK (GENDER IN('M','F')));

/\* 2C UNIQUE CONSTRAINT \*/

CREATE TABLE REGION

(REGION\_ID NUMBER(5),

NAME VARCHAR2(20) NOT NULL,

GEOGRAPHY VARCHAR2(20) NOT NULL,

CONTINENT VARCHAR2(20) NOT NULL,

CLIMATE VARCHAR2(20) NOT NULL,

CONSTRAINT REGION\_ID\_PK PRIMARY KEY(REGION\_ID),

CONSTRAINT REGION\_NAME\_UK UNIQUE(NAME)

);

CREATE TABLE COUNTRY

(COUNTRY\_ID NUMBER(5),

NAME VARCHAR2(20) NOT NULL,

HEALTH\_CENTER VARCHAR2(20) NOT NULL,

CONTACT\_NAME VARCHAR2(20),

CONTACT\_PHONE VARCHAR2(20),

CONSTRAINT COUNTRY\_ID\_PK PRIMARY KEY(COUNTRY\_ID)

);

CREATE TABLE DISEASE\_TYPE

(DISEASE\_TYPE\_ID NUMBER(5),

NAME VARCHAR2(20) NOT NULL,

DESCRIPTION VARCHAR2(50) NOT NULL,

CONSTRAINT DISEASE\_TYPE\_ID\_PK PRIMARY KEY(DISEASE\_TYPE\_ID)

);

CREATE TABLE AFTEREFFECT

(AFTEREFFECT\_ID NUMBER(5),

NAME VARCHAR2(20) NOT NULL,

DESCRIPTION VARCHAR2(50) NOT NULL,

CONSTRAINT AFTEREFFECT\_ID\_PK PRIMARY KEY(AFTEREFFECT\_ID)

);

CREATE TABLE CAT\_EVENT

(CAT\_EVENT\_ID NUMBER(5),

DATE\_OF\_EVENT DATE,

DESCRIPTION VARCHAR2(50),

CONSTRAINT CAT\_EVENT\_ID\_PK PRIMARY KEY(CAT\_EVENT\_ID)

);

CREATE TABLE SYMPTOM\_LIST

(SYMPTOM\_ID NUMBER(5),

NAME VARCHAR2(20) NOT NULL,

DESCRIPTION VARCHAR2(50) NOT NULL,

CONSTRAINT SYMPTOM\_LIST\_ID\_PK PRIMARY KEY (SYMPTOM\_ID)

);

/\* GROUP 2 \*/

/\* 2B FOREIGN KEY \*/

CREATE TABLE LOCATION

( LOCATION\_ID NUMBER(5),

REGION\_ID NUMBER(5),

CITY VARCHAR2(20) NOT NULL,

STATE CHAR(2) NOT NULL,

LATITUDE VARCHAR2(20),

LONGITUDE VARCHAR2(20),

CONSTRAINT LOCATION\_ID\_PK PRIMARY KEY (LOCATION\_ID),

CONSTRAINT LOCATION\_REGION\_ID\_FK FOREIGN KEY (REGION\_ID)

REFERENCES REGION(REGION\_ID)

);

CREATE TABLE DISEASE

(DISEASE\_ID NUMBER(5),

DISEASE\_TYPE\_ID NUMBER(5),

NAME VARCHAR2(20) NOT NULL,

TRANSMISSION\_FORM VARCHAR2(50),

CONSTRAINT DISEASE\_ID\_PK PRIMARY KEY (DISEASE\_ID),

CONSTRAINT DISEASE\_TYPE\_ID\_FK FOREIGN KEY(DISEASE\_TYPE\_ID)

REFERENCES DISEASE\_TYPE(DISEASE\_TYPE\_ID)

);

/\*GROUP 3 \*/

CREATE TABLE HEALTH\_PROFESSION

(HEALTH\_PROFESSION\_ID NUMBER(5),

LOCATION\_ID NUMBER(5),

LAST VARCHAR2(20) NOT NULL,

FIRST VARCHAR2(20) NOT NULL,

PHONE VARCHAR2(20) NOT NULL,

EMAIL VARCHAR2(20),

USER\_ID VARCHAR2(20) NOT NULL,

PASSWORD VARCHAR2(20) NOT NULL,

CONSTRAINT HEALTH\_PROFESSION\_ID\_PK PRIMARY KEY (HEALTH\_PROFESSION\_ID),

CONSTRAINT HEALTH\_PROFESSION\_LOC\_ID\_FK FOREIGN KEY (LOCATION\_ID)

REFERENCES LOCATION(LOCATION\_ID),

CONSTRAINT HEALTH\_PROFESSION\_USER\_ID\_UK UNIQUE(USER\_ID)

);

CREATE TABLE EXPECTED\_SYMPTOM

(

SYMPTOM\_ID NUMBER(5),

DISEASE\_ID NUMBER(5),

CONSTRAINT EXPECTED\_SYMPTOM\_PK PRIMARY KEY (SYMPTOM\_ID, DISEASE\_ID),

CONSTRAINT EXPECTED\_SYMPTOM\_SYMPTOM\_ID\_FK FOREIGN KEY (SYMPTOM\_ID)

REFERENCES SYMPTOM\_LIST (SYMPTOM\_ID),

CONSTRAINT EXPECTED\_SYMPTOM\_DISEASE\_ID\_FK FOREIGN KEY (DISEASE\_ID)

REFERENCES DISEASE (DISEASE\_ID)

);

/\*GROUP 4 \*/

/\* 4 HANDLING VIRTUAL COLUMNS \*/

CREATE TABLE CASE

(CASE\_ID NUMBER(5),

PATIENT\_ID NUMBER(5),

HEALTH\_PROFESSION\_ID NUMBER(5),

DISEASE\_ID NUMBER(5),

CAT\_EVENT\_ID NUMBER(5),

DATE\_CONTRACTED DATE NOT NULL,

REPORTED\_BY VARCHAR2(20) NOT NULL,

DATE\_REPORTED DATE NOT NULL,

BETWEEN\_CONTRACTION\_AND\_REPORT AS (DATE\_REPORTED - DATE\_CONTRACTED),

DATE\_CLOSED DATE,

OUTCOME\_ID NUMBER(5),

CONSTRAINT CASE\_ID\_PK PRIMARY KEY (CASE\_ID),

CONSTRAINT CASE\_PATIENT\_ID\_FK FOREIGN KEY (PATIENT\_ID)

REFERENCES PATIENT(PATIENT\_ID),

CONSTRAINT CASE\_HEALTH\_PROFESSION\_ID\_FK FOREIGN KEY (HEALTH\_PROFESSION\_ID)

REFERENCES HEALTH\_PROFESSION(HEALTH\_PROFESSION\_ID),

CONSTRAINT CASE\_DISEASE\_ID\_FK FOREIGN KEY (DISEASE\_ID)

REFERENCES DISEASE(DISEASE\_ID),

CONSTRAINT CASE\_CAT\_EVEN\_ID\_FK FOREIGN KEY (CAT\_EVENT\_ID)

REFERENCES CAT\_EVENT(CAT\_EVENT\_ID)

);

CREATE TABLE OUTCOME

(OUTCOME\_ID NUMBER(5),

CASE\_ID NUMBER(5) NOT NULL,

DATE\_OF\_OUTCOME DATE NOT NULL,

DESCRIPTION VARCHAR2(50),

CONSTRAINT OUTCOME\_ID\_PK PRIMARY KEY (OUTCOME\_ID),

CONSTRAINT OUTCOME\_CASE\_ID\_FK FOREIGN KEY (CASE\_ID)

REFERENCES CASE (CASE\_ID)

);

/\*SPECIAL ENTRY TO ADD OUTCOME\_ID AS A FOREIGN KEY TO CASE SINCE THEY WERE MADE SIMULTANEOUSLY \*/

ALTER TABLE CASE

ADD CONSTRAINT CASE\_OUTCOME\_ID\_FK FOREIGN KEY (OUTCOME\_ID)

REFERENCES OUTCOME (OUTCOME\_ID);

/\* GROUP 5 \*/

CREATE TABLE ENCOUNTER

(ENCOUNTER\_ID NUMBER(5),

CASE\_ID NUMBER(5) NOT NULL,

HEIGHT VARCHAR2(20),

WEIGHT VARCHAR2(20),

BLOOD\_PRESSURE VARCHAR2(20),

DATE\_OF\_ENCOUNTER DATE NOT NULL,

CONSTRAINT ENCOUNTER\_ID\_PK PRIMARY KEY (ENCOUNTER\_ID),

CONSTRAINT ENCOUNTER\_CASE\_ID\_FK FOREIGN KEY (CASE\_ID)

REFERENCES CASE(CASE\_ID)

);

CREATE TABLE TRAVEL\_DESTINATION

(TRAVEL\_DESTINATION\_ID NUMBER(5),

LOCATION\_ID NUMBER(5) NOT NULL,

START\_DATE DATE NOT NULL,

END\_DATE DATE,

POI CHAR(1),

CONSTRAINT TRAVEL\_DESTINATION\_ID\_PK PRIMARY KEY (TRAVEL\_DESTINATION\_ID),

CONSTRAINT TRAVEL\_DESTINATION\_LOC\_ID\_FK FOREIGN KEY (LOCATION\_ID)

REFERENCES LOCATION(LOCATION\_ID),

CONSTRAINT TRAVEL\_DESTINATION\_POI\_CK CHECK (POI IN ('Y','N'))

);

/\*GROUP 6 \*/

CREATE TABLE ENCOUNTER\_AFTEREFFECT

(

AFTEREFFECT\_ID NUMBER(5) ,

ENCOUNTER\_ID NUMBER(5) ,

CONSTRAINT ENCOUNTER\_AFTEREFFECT\_ID\_PK PRIMARY KEY (AFTEREFFECT\_ID, ENCOUNTER\_ID),

CONSTRAINT ENCOUNTER\_AFTEREFFECT\_AE\_ID\_FK FOREIGN KEY (AFTEREFFECT\_ID)

REFERENCES AFTEREFFECT(AFTEREFFECT\_ID),

CONSTRAINT ENCOUNTER\_AFTEREFFECT\_E\_ID\_FK FOREIGN KEY (ENCOUNTER\_ID)

REFERENCES ENCOUNTER(ENCOUNTER\_ID)

);

CREATE TABLE EXHIBITED\_SYMPTOM

(

ENCOUNTER\_ID NUMBER(5) ,

SYMPTOM\_ID NUMBER(5),

CONSTRAINT EXHIBITED\_SYMPTOM\_PK PRIMARY KEY (ENCOUNTER\_ID, SYMPTOM\_ID),

CONSTRAINT EXHIBITED\_SYMPTOM\_E\_ID\_FK FOREIGN KEY (ENCOUNTER\_ID)

REFERENCES ENCOUNTER(ENCOUNTER\_ID),

CONSTRAINT EXHIBITED\_SYMPTOM\_S\_ID\_FK FOREIGN KEY (SYMPTOM\_ID)

REFERENCES SYMPTOM\_LIST(SYMPTOM\_ID)

);

/\*PRIMARY KEY SEQUENCE CREATION\*/

/\* 7A SEQUENCE CREATION \*/

CREATE SEQUENCE PATIENT\_ID\_SEQ

INCREMENT BY 1

START WITH 1

CACHE 5

NOCYCLE;

/\* 7B ALTERING SEQUENCE \*/

ALTER SEQUENCE PATIENT\_ID\_SEQ

NOCACHE;

/\*ALL OTHER SEQUENCE CREATIONS \*/

CREATE SEQUENCE REGION\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE COUNTRY\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE DISEASE\_TYPE\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE AFTEREFFECT\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE CAT\_EVENT\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE SYMPTOM\_LIST\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE LOCATION\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE DISEASE\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE HEALTH\_PROFESSION\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE CASE\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE OUTCOME\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE ENCOUNTER\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE TRAVEL\_DESTINATION\_ID\_SEQ

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

/\* 3 INSERT STATEMENTS \*/

INSERT INTO PATIENT

VALUES (PATIENT\_ID\_SEQ.NEXTVAL, '28-MAR-63', 'M','O','RUSSIAN');

INSERT INTO PATIENT

VALUES (PATIENT\_ID\_SEQ.NEXTVAL, '08-AUG-60', 'F','B','GERMAN');

INSERT INTO PATIENT

VALUES (PATIENT\_ID\_SEQ.NEXTVAL, '03-MAR-03', 'M','A','ASIAN');

INSERT INTO REGION

VALUES(REGION\_ID\_SEQ.NEXTVAL,'WEST COAST','WEST','NORTH AMERICA','VARIED');

INSERT INTO REGION

VALUES(REGION\_ID\_SEQ.NEXTVAL,'DESERT STATES','MIDWEST','NORTH AMERICA','VERY WARM');

INSERT INTO COUNTRY

VALUES(COUNTRY\_ID\_SEQ.NEXTVAL, 'AMERICA', 'GREEN MANORS ','ANTHONY EDWARDS', '(123)-123-1234');

INSERT INTO COUNTRY (COUNTRY\_ID,NAME, HEALTH\_CENTER,CONTACT\_NAME)

VALUES(COUNTRY\_ID\_SEQ.NEXTVAL,'AMERICA', 'BATES HOSPITAL','CONSTANCE PETERSEN');

INSERT INTO DISEASE\_TYPE

VALUES(DISEASE\_TYPE\_ID\_SEQ.NEXTVAL, 'VIRUS','QUICK INFECTION');

INSERT INTO DISEASE\_TYPE

VALUES(DISEASE\_TYPE\_ID\_SEQ.NEXTVAL, 'BRAIN INFECTION', 'INCREASED AGGRESIVENESS');

INSERT INTO AFTEREFFECT

VALUES(AFTEREFFECT\_ID\_SEQ.NEXTVAL, 'SORES', 'BIG RED, ALL OVER');

INSERT INTO AFTEREFFECT

VALUES(AFTEREFFECT\_ID\_SEQ.NEXTVAL,'HEADACHE','ONLY ON RT. SIDE');

INSERT INTO CAT\_EVENT

VALUES(CAT\_EVENT\_ID\_SEQ.NEXTVAL,'09-MAY-58', 'MASSIVE OUTBREAK');

INSERT INTO CAT\_EVENT

VALUES(CAT\_EVENT\_ID\_SEQ.NEXTVAL, '28-JUL-59','ACCIDENTAL DEATH');

INSERT INTO SYMPTOM\_LIST

VALUES(SYMPTOM\_LIST\_ID\_SEQ.NEXTVAL, 'STAB WOUNDS', 'POINTED AND ALL OVER');

INSERT INTO SYMPTOM\_LIST

VALUES(SYMPTOM\_LIST\_ID\_SEQ.NEXTVAL,'INSANITY','CUCKOO CRAZY');

INSERT INTO LOCATION

VALUES(LOCATION\_ID\_SEQ.NEXTVAL, 1, 'BODEGA BAY', 'CA', '38.333250','-123.048057');

INSERT INTO LOCATION

VALUES(LOCATION\_ID\_SEQ.NEXTVAL, 2, 'FAIRVALE','AZ','34.048928','-111.093731');

INSERT INTO DISEASE

VALUES(DISEASE\_ID\_SEQ.NEXTVAL,1,'BIRD FLU','BIRD ATTACKS');

INSERT INTO DISEASE

VALUES(DISEASE\_ID\_SEQ.NEXTVAL,2,'HULKISM','BIRD ATTACKS');

INSERT INTO HEALTH\_PROFESSION

VALUES(HEALTH\_PROFESSION\_ID\_SEQ.NEXTVAL,1,'BRENNER','MITCH', '(603) 650-5000','MBRENNER@PSYCH.ORG','MBRENNER','BIRDCRAZY');

INSERT INTO HEALTH\_PROFESSION

VALUES(HEALTH\_PROFESSION\_ID\_SEQ.NEXTVAL,2,'DANIELS','MELANIE', '(610)-459-0647', 'MDANIELS@PSYCH.ORG','MDANIELS','HOSPITABLEHOSPITAL');

INSERT INTO EXPECTED\_SYMPTOM

VALUES(1,1);

INSERT INTO EXPECTED\_SYMPTOM

VALUES(2,2);

INSERT INTO CASE(CASE\_ID,PATIENT\_ID,HEALTH\_PROFESSION\_ID,DISEASE\_ID,CAT\_EVENT\_ID,DATE\_CONTRACTED,REPORTED\_BY,DATE\_REPORTED,DATE\_CLOSED)

VALUES(CASE\_ID\_SEQ.NEXTVAL,1,1,1,1,'29-MAY-54','HARRY WARNER','29-MAR-63','16-NOV-12');

INSERT INTO CASE(CASE\_ID,PATIENT\_ID,HEALTH\_PROFESSION\_ID,DISEASE\_ID,CAT\_EVENT\_ID,DATE\_CONTRACTED,REPORTED\_BY,DATE\_REPORTED)

VALUES(CASE\_ID\_SEQ.NEXTVAL,2,2,2,2,'28-DEC-45','DAVID SELZNICK','08-MAR-48');

INSERT INTO OUTCOME

VALUES(OUTCOME\_ID\_SEQ.NEXTVAL,1,'09-SEP-61', 'DESTRUCTION OF TOWN');

/\*BECAUSE OF FOREIGN KEY RESTRICTIONS THIS STATEMENT HAD TO BE ADDED AFTER THE INSERT STATEMENT\*/

UPDATE CASE

SET OUTCOME\_ID=1

WHERE CASE\_ID=1;

INSERT INTO ENCOUNTER

VALUES(ENCOUNTER\_ID\_SEQ.NEXTVAL,1,'6 FT', '180 LBS','100 BPM','22-JUL-64');

INSERT INTO ENCOUNTER

VALUES(ENCOUNTER\_ID\_SEQ.NEXTVAL,2,'5FT 4IN','115 LBS','85 BPM', '14-JUL-66');

INSERT INTO TRAVEL\_DESTINATION

VALUES(TRAVEL\_DESTINATION\_ID\_SEQ.NEXTVAL,1,'1-JUN-56','22-DEC-56','Y');

INSERT INTO TRAVEL\_DESTINATION

VALUES(TRAVEL\_DESTINATION\_ID\_SEQ.NEXTVAL,2,'19-DEC-69','21-JUN-72','N');

INSERT INTO ENCOUNTER\_AFTEREFFECT

VALUES(1,1);

INSERT INTO ENCOUNTER\_AFTEREFFECT

VALUES(2,2);

INSERT INTO EXHIBITED\_SYMPTOM

VALUES(1,1);

INSERT INTO EXHIBITED\_SYMPTOM

VALUES(2,2);

/\* 5 MODIFY EXISTING ROWS \*/

/\* 5A UPDATE COMMAND \*/

UPDATE PATIENT

SET GENDER='F'

WHERE PATIENT\_ID=3;

/\* 5B SUBSTITUTION VARIABLES \*/

/\* 6 DELETING ROWS \*/

DELETE FROM PATIENT

WHERE PATIENT\_ID=3;

/\* 8 INDEX CREATION \*/

/\* 8A B-TREE \*/

CREATE INDEX COUNTRY\_NAME\_IDX

ON COUNTRY(NAME);

/\* 8B BITMAP \*/

CREATE BITMAP INDEX PATIENT\_BLOOD\_TYPE\_IDX

ON PATIENT(BLOOD\_TYPE);

/\* 8C FUNCTION-BASED \*/

CREATE INDEX CASE\_CONTRACT\_REPORT\_IDX

ON CASE(DATE\_CONTRACTED - DATE\_REPORTED);

/\*END OF DATABASE CREATION SECTION \*/

/\* 9 QUERIES FOR RESTRICTING ROWS AND SORTING DATA \*/

/\*9A WHERE CLAUSE, 9B BETWEEN AND OPERATOR, 11Dd TO\_DATE FUNCTION, 9Ga SECONDARY SORT IN ORDER BY CLAUSE \*/

SELECT DOB, PATIENT\_ID, GENDER, BLOOD\_TYPE

FROM PATIENT

WHERE DOB BETWEEN TO\_DATE('JANUARY 01, 60','MONTH DD, YY') AND TO\_DATE('DECEMBER 31, 64','MONTH DD, YY')

ORDER BY DOB DESC, GENDER;

/\* 9C IN OPERATOR, 9E LOGICAL OPERATORS, 9Gb ORDER BY SELECT ORDER\*/

SELECT PATIENT\_ID,BLOOD\_TYPE, GENDER

FROM PATIENT

WHERE BLOOD\_TYPE IN ('A','O')

OR GENDER='F'

ORDER BY 1 DESC;

/\* 9D LIKE OPERATORS, 11Ac INITCAP CASE CONVERSION,11Bh CONCAT FUNCTION \*/

SELECT CONCAT(INITCAP(FIRST),CONCAT(' ',INITCAP(LAST))) AS FULL\_NAME, USER\_ID, PASSWORD

FROM HEALTH\_PROFESSION

WHERE LAST LIKE 'B%';

/\*9F TREATMENT OF NULL VALUES \*/

SELECT PATIENT\_ID,DISEASE\_ID,DATE\_CONTRACTED,REPORTED\_BY,DATE\_REPORTED

FROM CASE

WHERE DATE\_CLOSED IS NULL;

/\*10A CARTESIAN JOIN \*/

SELECT PATIENT\_ID, NAME

FROM PATIENT CROSS JOIN COUNTRY;

/\*10B EQUALITY JOIN \*/

SELECT LOCATION\_ID, TRAVEL\_DESTINATION\_ID, POI

FROM LOCATION NATURAL JOIN TRAVEL\_DESTINATION;

/\*10C NON-EQUALITY JOIN \*/

SELECT PATIENT\_ID, BLOOD\_PRESSURE

FROM PATIENT JOIN ENCOUNTER

ON DATE\_OF\_ENCOUNTER>DOB;

/\*10D SELF JOIN \*/

SELECT h.FIRST, h.LAST, p.LAST

FROM HEALTH\_PROFESSION h JOIN HEALTH\_PROFESSION p

ON p.HEALTH\_PROFESSION\_ID = h.HEALTH\_PROFESSION\_ID;

/\*10E OUTER JOIN \*/

SELECT LAST, FIRST, LOCATION\_ID

FROM LOCATION l FULL OUTER JOIN HEALTH\_PROFESSION h

USING (LOCATION\_ID);

/\*11Bb LENGTH FUNCTION\*/

SELECT USER\_ID,LENGTH(PASSWORD)

FROM HEALTH\_PROFESSION;

/\*11Ba SUBSTRING FUNCTION \*/

SELECT SUBSTR(LAST,1,3)

FROM HEALTH\_PROFESSION;

/\* 11Ca ROUND FUNCTION \*/

SELECT ROUND(PATIENT\_ID, 1)

FROM PATIENT;

/\* 11Ca POWER FUNCTION \*/

SELECT POWER(PATIENT\_ID, 3)

FROM PATIENT;

/\* 11Db ADD\_MONTHS FUNCTION, 11Dc LAST\_DAY AND NEXT\_DAY FUNCTION \*/

SELECT CASE\_ID, PATIENT\_ID, DATE\_CONTRACTED, LAST\_DAY(ADD\_MONTHS(DATE\_CONTRACTED,3) )"EXPECTED CLOSE DATE"

FROM CASE;

SELECT CASE\_ID, PATIENT\_ID, DATE\_CONTRACTED, NEXT\_DAY(ADD\_MONTHS(DATE\_CONTRACTED,3), MON )"EXPECTED CLOSE DATE"

FROM CASE;

/\* 12C COUNT FUNCTION \*/

SELECT COUNT(PATIENT\_ID) "NUMBER OF PATIENTS"

FROM PATIENT;

/\* 12E MIN FUNCTION, 12D MAX FUNCTION \*/

SELECT BLOOD\_TYPE, MIN(PATIENT\_ID)

FROM PATIENT

GROUP BY BLOOD\_TYPE;

SELECT BLOOD\_TYPE, MAX(PATIENT\_ID)

FROM PATIENT

GROUP BY BLOOD\_TYPE;

/\*13A SINGLE-ROW SUB QUERY \*/

SELECT BLOOD\_TYPE, PATIENT\_ID

FROM PATIENT

WHERE PATIENT\_ID=(SELECT MAX(PATIENT\_ID)

FROM PATIENT);

/\*13B MULTIPLE-ROW SUB QUERY \*/

SELECT BLOOD\_TYPE, PATIENT\_ID

FROM PATIENT

WHERE PATIENT\_ID IN (SELECT PATIENT\_ID

FROM CASE);

/\*SELECT PATIENT WITH MULTIPLE DISEASES\*/

SELECT PATIENT\_ID, CASE\_ID, DISEASE\_ID

FROM PATIENT NATURAL JOIN CASE FULL OUTER JOIN DISEASE

USING (DISEASE\_ID)

ORDER BY PATIENT\_ID DESC;

/\*SELECT PATIENTS PER HEALTH PROFESSIONAL GROUPED BY REGION \*/

SELECT PATIENT\_ID, HEALTH\_PROFESSION\_ID, REGION\_ID

FROM CASE NATURAL JOIN HEALTH\_PROFESSION NATURAL JOIN LOCATION;

/\*LIST OF DISEASES WITHIN A REGION \*/

SELECT DISEASE\_ID, REGION\_ID

FROM DISEASE NATURAL JOIN CASE NATURAL JOIN HEALTH\_PROFESSION NATURAL JOIN LOCATION

ORDER BY REGION\_ID;

/\*LIST OF SYMPTOMS AND RELATED DISEASES \*/

SELECT SYMPTOM\_ID, NAME, DESCRIPTION, DISEASE\_ID

FROM EXPECTED\_SYMPTOM NATURAL JOIN SYMPTOM\_LIST;

1. This section meets section 1 of the database requirements for full details refer to Appendix D [↑](#footnote-ref-1)
2. This group also contains an alter statement to add a foreign key to the CASE table since CASE and OUTCOME are linked and so are all the following tables. Adding the foreign key in the CASE creation statement would have returned an error [↑](#footnote-ref-2)
3. The following section meets section 7 of the database criteria [↑](#footnote-ref-3)
4. The following section covers section 3 of the database criteria [↑](#footnote-ref-4)
5. The following section meets section 8 of the database criteria [↑](#footnote-ref-5)
6. This section meets several criteria requested by the database guidelines. The specific sections are mentioned before the SQL code in every example. [↑](#footnote-ref-6)
7. The following section primarily meets the requirements of section 10 of the database requirements. [↑](#footnote-ref-7)
8. The following section primarily meets the requirements of section 11 [↑](#footnote-ref-8)
9. The following section primarily meets the section 12 requirements [↑](#footnote-ref-9)
10. The following section primarily meets the requirements of section 13 [↑](#footnote-ref-10)
11. The constraint names can be found in Appendix A [↑](#footnote-ref-11)
12. This meets requirement 2A of the database [↑](#footnote-ref-12)
13. This meets requirement 2E of the database [↑](#footnote-ref-13)
14. This meets requirement 2D of the database [↑](#footnote-ref-14)
15. This meets requirement 2C of the Database [↑](#footnote-ref-15)
16. This meets requirement 2B of the database [↑](#footnote-ref-16)
17. This is an example of a virtual column per part 4 of the requirements of the database [↑](#footnote-ref-17)