Reference Lab Database

Lauren Foltz

IST 659-M407

Table of Contents

I. Summary

In cases when a physician orders testing to be performed by a reference lab (the testing does not take place at the doctor's office), the reference lab needs to be able to electronically track information about the testing. Reference lab employees at the accessioning branch need to access the database in order to perform the bulk of the data entry. Reference lab employees at the testing lab need to access the information in order to provide appropriate testing and enter the testing results.

This reference lab database will house information required for medical testing. A physician at a medical practice orders a test on a patient specimen, the specimen is accessioned, and the testing lab performs the test. A result is then entered into the database and a report is sent to the ordering physician.

The code for the database was written in SQL server. A structure for all relevant information regarding patients, patient specimens, tests, etc. was created in SQL in a manner that ensures data integrity and efficiency. See page 8, the figure of the Normalized Logical Model, for a visual of the information that will be tracked.

The user interface for the database will be implemented in MS Access. This platform was chosen because many computers in the reference lab already have this program installed, and as lab employees are already familiar with MS Word and MC Excel, training on this application will be faster.

Any data question that the reference lab has should be answerable by queries via SQL or MS Access. Example data questions are below, and are addressed in detail in section VII *Answering Data Questions in SQL* and section IX *Implementation*.

1. Which tests were ordered to be run in a particular lab?
2. Which physicians ordered a particular test?
3. Which patients are residing in a particular location?
4. Which physicians are from a particular medical practice?
5. Which specimens came from a particular branch?
6. Which tests were ordered after a particular date?
7. Which specimens were ordered for a particular test?
8. Which test results are still pending (awaiting completion)?

This database is flexible, and beyond maintaining the database, the database team can make changes as the reference lab sees fit. In section XI, possible future improvements are discussed.

II. Challenges and Risks:

A particular challenge in genetic testing is how to deal with fetal specimens. Fetal specimens do not have a legal name, date of birth, address, etc., but they are genetically distinct from the patient, and confusing a maternal specimen for a fetal specimen not only compromises patient care, but also opens the reference lab up to wrongful life lawsuits.

A further complication is that with the advent of fertility drugs, more and more pregnancies involve multiple fetuses. It is essential to keep track of which fetus is which in case the patient opts for a selective termination.

Currently, branches are accessioning fetal specimens under the maternal name and accession number, but with a specimen type that indicates that this is a fetal specimen. Additionally, sometimes a branch or the testing lab will change the brother number in the accession from 0 to 1 for the fetal specimen, but this is not an adequate system, because an accession with a brother 1 is frequently used for non-fetal specimens, such as when a physician wants to order an add-on test.

Currently, the lab techs use name and accession numbers to identify which DNA to test, and specimen type isn't always apparent on a DNA tube. There have been multiple instances where a Tech in the testing lab has accidentally run the maternal specimen instead of the fetal specimen, and the doctor received wrong results. In one case, the fetal specimen was reported out as a carrier, but it was born affected.

An important feature of this database is that it has been designed to minimize the risk of confusing a maternal and fetal specimen by giving each individual specimen a unique ID. This will provide better patient care and reduce the risk of any lawsuits against the reference lab.

III. The Conceptual Model

The conceptual model describes the business data and the rules for governing their use and storage. It identifies what's important for an organization to store, and provides a framework for communicating data needs to and from business users. It uses entity-relationship modeling to codify business rules.

Business Rules

Business rules govern how data are stored and accessed. These rules help identify what information the business needs to track, as well as interactions and constraints. They are used to build a model containing entities, attributes, and relationships.

- A patient optionally has many specimens collected.
- A specimen is collected from 1 and only 1 patient.

- A specimen is accessioned by 1 and only 1 branch.
- A branch optionally accessions many specimens.

- A specimen optionally has many tests ordered.
- A test is optionally ordered on many specimens.

- A test is optionally ordered by many physicians.
- A physician optionally orders many tests.

- A physician is employed by 1 and only 1 medical practice.
- A medical practice employs 1 or many physicians.

- A test is run at 1 and only 1 testing lab.
- A testing lab optionally runs many tests.

- A patient optionally gestates multiple fetuses.
- A fetus is gestated by 1 and only 1 patient.

- A fetus optionally has many specimens collected.
- A specimen is collected from 1 and only 1 fetus.

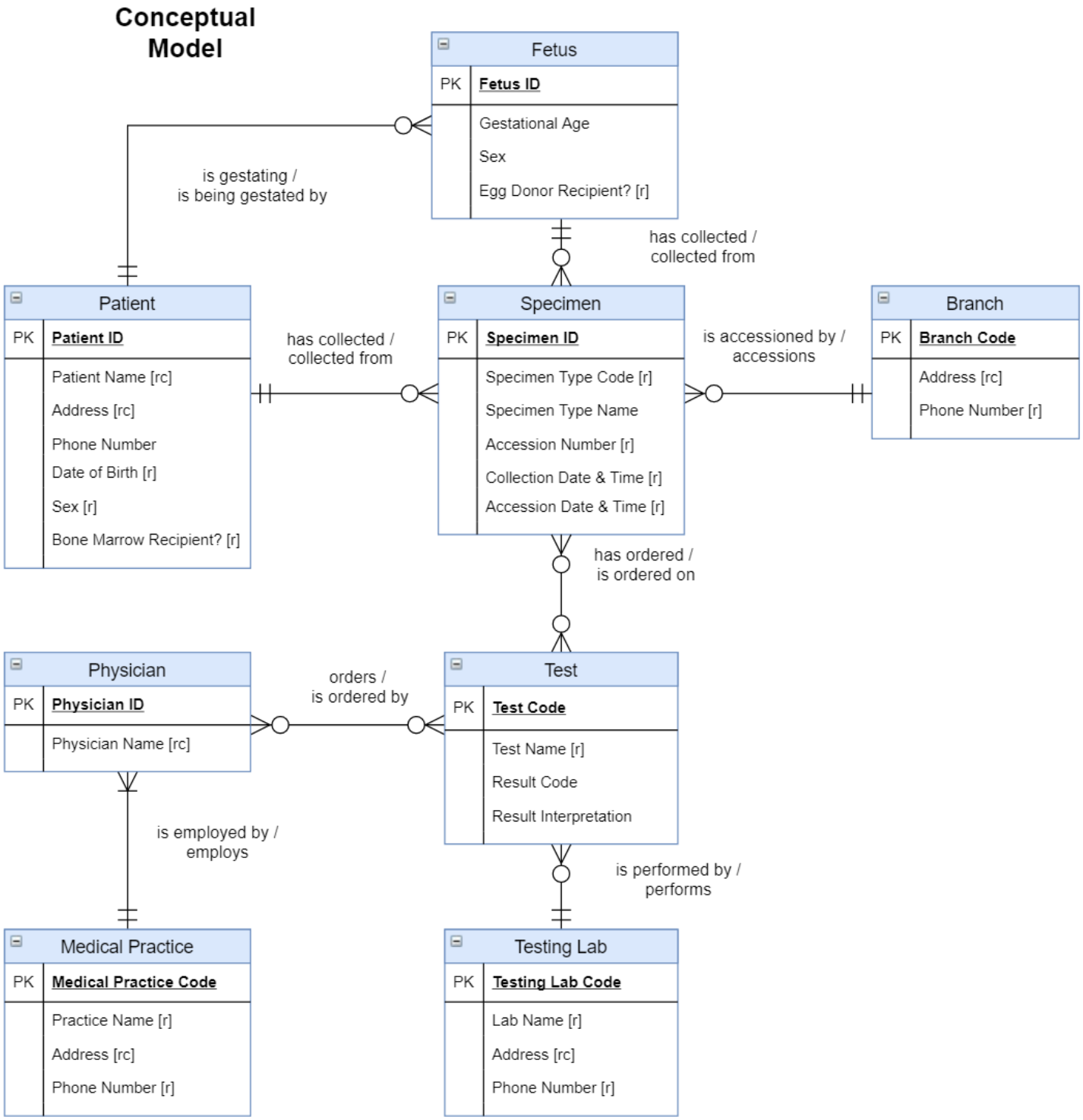
Stake Holders

Stake Holders are included below in order to define who the main people in the process are. Not all stake holders have access to the database.

- All Reference Lab employees who have access to the database are expected to comply with HIPAA and treat all patient information as confidential. More information about rules for internal use is provided in section X: *Security*.
- Reference Lab employees at the branch have access to the database and are responsible for entering the majority of the data at the time that the specimen is accessioned. Some information, such as ordering physician and tests being ordered, comes from the requisition.
- Reference Lab employees at the testing lab have access to the database so that they have all pertinent information about the specimen to perform testing.
- Patients do not have access to the database. They must receive their results from their physicians.
- Physicians do not have access to the database. Once testing is complete, the reference lab sends a report to the physician.

Diagram Explanation

The diagram on the following page consists of entities (shown in blue) and attributes (listed under each entity. These represent the data that will be collected in the database. Representative examples of data can be found in Appendix II.


Conceptual Model
Fetus
PK
Fetus ID
Gestational Age
Sex
Egg Donor Recipient? [r]
is gestating / is being gestated by
has collected / collected from
Patient
PK
Patient ID
Patient Name [rc]
Address [rc]
Phone Number
Date of Birth [r]
Sex [r]
Bone Marrow Recipient? [r]
has collected / collected from
Specimen
PK
Specimen ID
Specimen Type Code [r]
Specimen Type Name
Accession Number [r]
Collection Date & Time [r]
Accession Date & Time [r]
is accessioned by / accessions
Branch
PK
Branch Code
Address [rc]
Phone Number [r]
has ordered / is ordered on
Physician
PK
Physician ID
Physician Name [rc]
orders / is ordered by
Test
PK
Test Code
Test Name [r]
Result Code
Result Interpretation
is employed by / employs
is performed by / performs
Medical Practice
PK
Medical Practice Code
Practice Name [r]
Address [rc]
Phone Number [r]
Testing Lab
PK
Testing Lab Code
Lab Name [r]
Address [rc]
Phone Number [r]

Conceptual Model Discussion

Entity Choices:

- Fetus: The fetus is listed as a separate entity because it is genetically unique from the person gestating it. It is essential to know with certainty whether a specimen being tested came from the person gestating the fetus or from the actual fetus.

Attribute Choices:

- Collection Date & Time: In some cases, specimens are not collected at a branch, and therefore collection date and time are not the same as accession date and time. This attribute was included so that we have accurate information about specimen age.
    - Example: A tumor biopsy may be collected at a doctor's office or a hospital.
    - Example: A buccal swab may be collected by the patient at home, and then mailed in.
- Sex: Some genetic tests specifically target the X or Y chromosome. It is important to know if the testing is being performed on a female (typically XX) or a male (typically XY). That is why this is a required attribute (except in the cases of a fetus, because the sex may not be known).
- Bone Marrow Recipient: Genetic tests can give incorrect results on bone marrow recipients, so this is important to a patient's bone marrow status.
- Egg Donor Recipient: When a fetal specimen is collected via amniocentesis, a maternal specimen is often collected via blood and tested as well. The purpose of profiling both specimens is to make sure that the fetal specimen is, in fact, the fetal specimen, and hasn't been contaminated by maternal cells. If the Maternal Cell Contamination test is performed on the recipient of an egg donation, the results will not come out as expected. If the maternal specimen is not the recipient of an egg donation, a specimen swap should be suspected and further investigation will be required.

Relationship Choices:

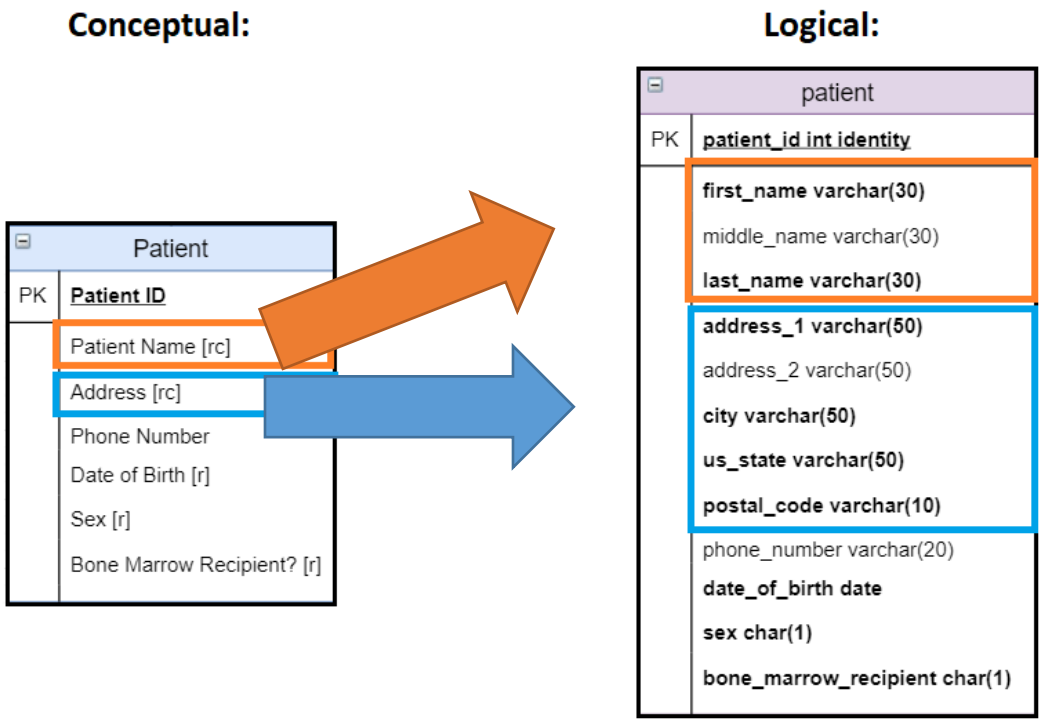
- Cardinality: The choice was made to go with "optionally many" rather than "one or many" in order to give the most flexibility.
    - We may want to enter patient information before a specimen is collected.
    - A specimen may have only one test ordered, and that test may be cancelled. We may want the specimen information to stay in the database in case the physician wants to add on a different test, or in case the first test order was deleted in error.
    - We may want all orderable tests in the database, regardless of whether they've been ordered.

IV. Logical Model

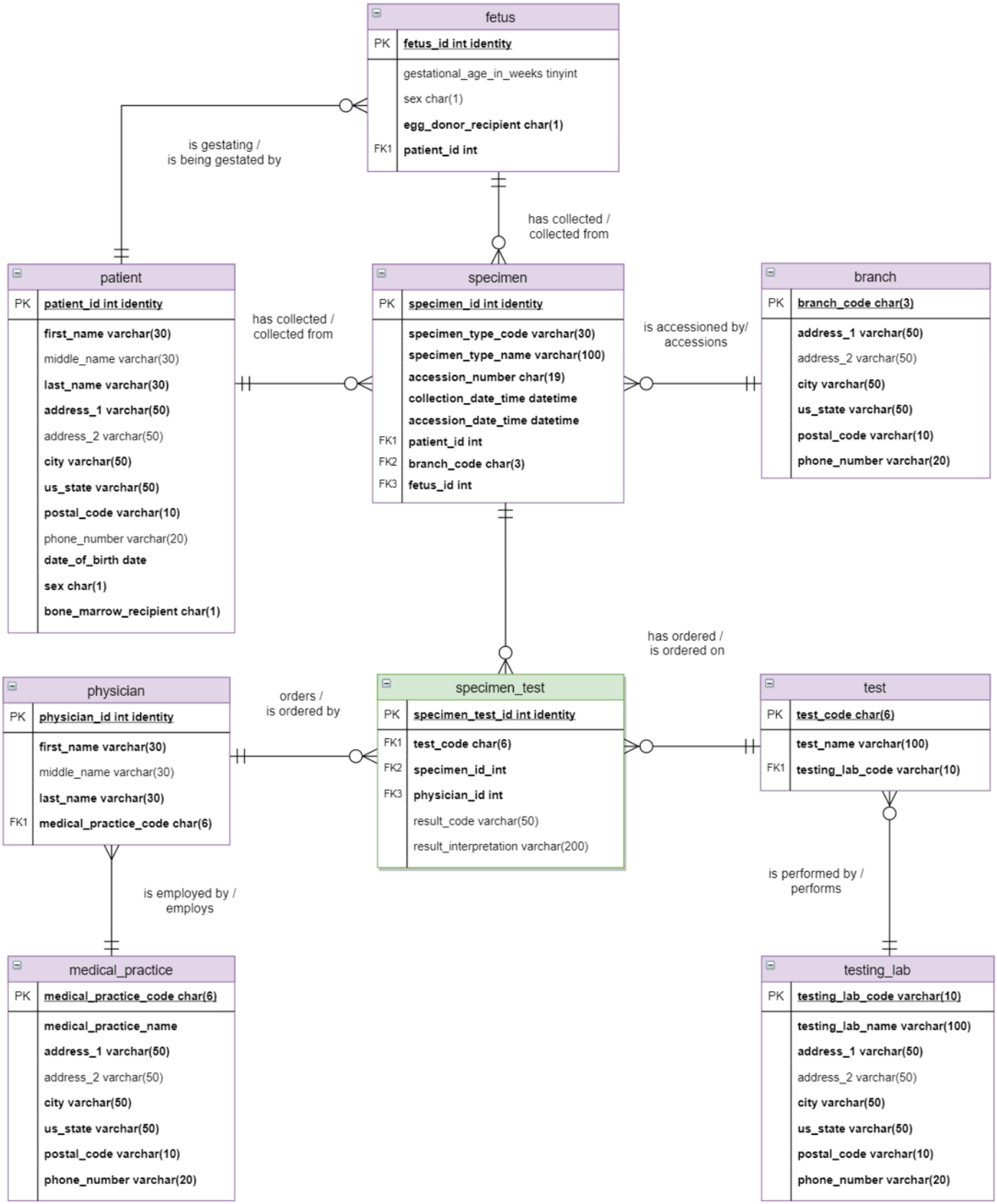
Prior to building the physical database in SQL server, the conceptual model must be converted to a normalized logical model. This required several steps:

- Data types were chosen.
  - Example: for first name “varchar(30)” was selected.
- Composite attributes had to be broken down into individual components
  - Example: Patient name was broken down into first name, middle name, last name.
- A Bridge table (shown in green on page 8) was created for many to many relationships.
- Relationships were defined using relational keys.
  - Natural Keys: Branch Code, Test Code, Medical Practice Code, and Testing Lab Code are all unique identifiers set up by the Reference Lab in advance.
  - Surrogate Keys: patient, specimen, physician, fetus, and specimen_test were given unique identifiers that have no business meaning. The choice was made not to use accession number as the primary key for the specimen table because although accession numbers are unique to patients, they are not unique to specimen type.
- The normalization process was used to eliminate multiples, redundancies, and dependencies.

The figure below shows the patient entity (in blue) and attributes (listed below) in the conceptual model being converted into the patient table (in purple) and columns (listed below) in the logical model.



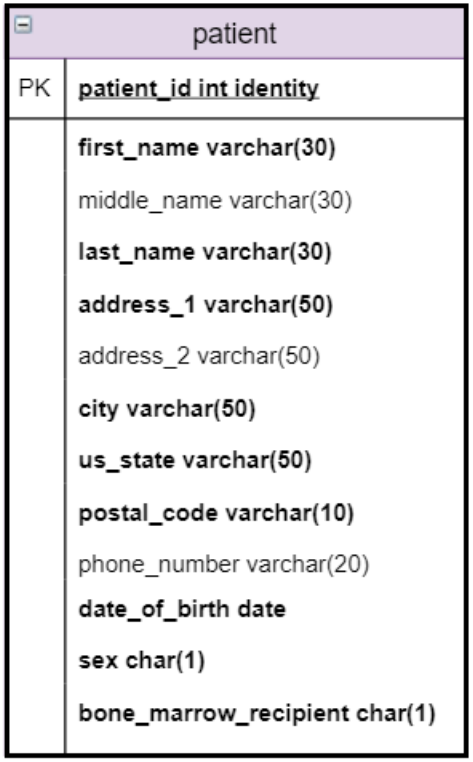
Normalized Logical Model



V. Physical Database

Each table and its columns in the normalized logical model was created in SQL using established coding syntax. The full code can be found in Appendix III, *SQL Code*. Below is a visual of the patient table being coded into SQL. For details on what the tables look like after creation in SQL, see Appendix II, *Representative Data*.

Table from Model:



SQL Code to Create Table in Database:

```
-- start creating rl_patient table
create table rl_patient(
-- columns for this table
    patient_id int identity,
    first_name varchar(30) not null,
    middle_name varchar(30),
    last_name varchar(30) not null,
    address_1 varchar(50) not null,
    address_2 varchar(50),
    city varchar(50) not null,
    us_state varchar(50) not null,
    postal_code varchar(10) not null,
    phone_number varchar(20),
    date_of_birth date not null,
    sex char(1) not null,
    bone_marrow_recipient char(1) not null,
-- constraints on this table
    constraint pk_rl_patient primary key (patient_id)
    )
-- end creating rl_patient table
```

Physical Database Flexibility:

If there is a need to alter a column, this can be done. For instance, the code below instructs SQL to increase the number of characters allowed in the accession_number column (within the specimen table) from 19 to 25:

```
alter table rl_specimen
      alter column accession_number char(25) not null;
```

If there is a need to set a default value in a column, this can be done. For instance, after the specimen_test table was created, the code below was run in order to set “pending” as the default value in the result_code and result_interpretations columns. This causes “pending” to appear if a test result hasn’t yet been entered.

```
alter table rl_specimen_test
add constraint df_pending default 'pending'for result_code

alter table rl_specimen_test
add constraint df_pending2 default 'pending'for result_interpretation
```

VI. Data Creation in SQL

Representative data was entered into each table using INSERT statements and listing the VALUES with which to populate each column. Below are three insert statements for the rl_ patient table. Each column name is listed, and the values to be entered into each column are listed below that in red.

```
168 insert into rl_patient
169     (first_name, middle_name, last_name, address_1, city, us_state, postal_code, phone_number, date_of_birth, sex, bone_marrow_recipient)
170     values
171     ('John', 'Gary', 'Smith', '162 Wildfell Trail', 'Cary', 'NC', '27513', '919-467-8342', '12/09/1969', 'M', 'N')
172
173 insert into rl_patient
174     (first_name, middle_name, last_name, address_1, city, us_state, postal_code, phone_number, date_of_birth, sex, bone_marrow_recipient)
175     values
176     ('Sarah', 'Johnson', 'Smith', '162 Wildfell Trail', 'Cary', 'NC', '27513', '919-523-1271', '05/12/1972', 'F', 'N')
177
178 insert into rl_patient
179     (first_name, middle_name, last_name, address_1, city, us_state, postal_code, phone_number, date_of_birth, sex, bone_marrow_recipient)
180     values
181     ('Annette', 'Lorraine', 'White', '2033 W Park Street', 'Cary', 'NC', '27511', '919-807-3256', '06/25/1982', 'F', 'N')
```

Below is an image of the rl_patient table in SQL with patient information. The table is produced by executing the select statement

```
select * from rl_patient
```

patient_id	first_name	middle_name	last_name	address_1	address_2	city	us_state	postal_code	phone_number	date_of_birth	sex	bone_marrow_recipient
1	John	Gary	Smith	162 Wildfell Trail	NULL	Cary	NC	27513	919-467-8342	1969-12-09	M	N
2	Sarah	Johnson	Smith	162 Wildfell Trail	NULL	Cary	NC	27513	919-523-1271	1972-05-12	F	N
3	Annette	Lorraine	White	2033 W Park Street	NULL	Cary	NC	27511	919-807-3256	1982-06-25	F	N
4	James	Rodger	Hume	2913 Green Hill Drive	NULL	Chapel Hill	NC	27514	919-306-8819	1974-09-18	M	N
5	Max	L.	Thompson	3604 Morningside Drive	NULL	Raleigh	NC	27607	NULL	1952-04-13	M	N
6	Sydney	R.	Jones	7 Wood Duck Court	NULL	Durham	NC	27713	919-539-4872	1995-11-01	F	N

Default Handling in Data Creation:

In the SQL code below, specimen_id 27 and 29 do not yet have test results. No result_code or result_interpretation was entered for them, and the resulting table shows “pending” as the default value. This is in contrast to specimen_id 28 and 30, who have results entered.

```
404 insert into rl_specimen_test
405     (test_code, specimen_id)
406 values
407     ('480533', '27')
408
409 insert into rl_specimen_test
410     (test_code, specimen_id, result_code, result_interpretation)
411 values
412     ('480533', '28', 'CF-HET', 'Carrier of the delta F508 Mutation')
413
414 insert into rl_specimen_test
415     (test_code, specimen_id)
416 values
417     ('489067', '29')
418
419 insert into rl_specimen_test
420     (test_code, specimen_id, result_code, result_interpretation)
421 values
422     ('511919', '30', 'FX-NEG', '21 repeats of the FMR1 gene')
```

specimen_test_id	test_code	specimen_id	ordered_date_time	result_code	result_interpretation	physician_id
1	480533	27	2018-06-13 00:00:00.000	pending	pending	1
2	480533	28	2018-06-01 00:00:00.000	CF-HET	Carrier of the delta F508 Mutation	1
3	489067	29	2018-06-15 00:00:00.000	pending	pending	2
4	511919	30	2018-06-15 00:00:00.000	FX-NEG	21 repeats of the FMR1 gene	3

Important Note: Data can also be created in Access. See section IX, *Implementation*, for details.

VII. Data Manipulation in SQL

If incorrect data was entered in error, UPDATE statements can be used to correct the data.

- Below are two examples of updates that could be performed in the rl_patient table. The first shows an updated postal code and the second shows an updated birthdate.

```
update rl_patient set postal_code = 27513 where patient_id = 3
```

```
update rl_patient set date_of_birth = '12/09/1966' where patient_id = 1
```

- Below are two examples of updates that could be performed in the rl_specimen table. The first shows an updated collection time and the second shows an updated specimen ID.

```
update rl_specimen set collection_date_time = '2018-06-13 13:22:32' where specimen_id = 2
```

```
update rl_specimen set patient_id= '2' where specimen_id = 3
```

If a record needs to be deleted, this can be done.

- Below is a visual of all tests that have been entered in the rl_test table to date.

test_code	test_name	testing_lab_code
480533	Cystic Fibrosis Profile, 32 Mutations, DNA Analysis	3151
489067	Epidermal Growth Factor Receptor (EGFR) Gene Mutation Analysis, Real-time PCR	7550
511154	Factor V Lieden Mutation, DNA Analysis	3151
511162	Factor II, DNA Analysis	3151
511200	UGT 1A1 Irinotecan Toxicity	3151
511238	MTHFR, DNA Analysis	3151
511402	Maternal Cell Contamination	3151
511426	Tay-Sachs Disease, Biochemical, Leukocytes	4817
511919	Fragile X Syndrome, FISH	1284
511966	InSight: PrenatalAneuploid Evaluation, Microarray	8129

- If the reference lab decides it will no longer offer the UGT test, the test can be deleted using the statement below, and the test will no longer appear on the table.

```
delete from rl_test where test_code = 511200
```

Important Note: Data can also be manipulated in Access. See section IX, *Implementation*, for details.

VIII. Answering Data Questions in SQL

Select statements can be executed in SQL in order to answer data questions. Below are several examples.

Data Question #1: Which tests were ordered to be run in a particular lab?

Below is a select statement that brings back a list of all tests run in lab 3151, Molecular Genetics.

```
select * from rl_test where testing_lab_code = 3151
```

	test_code	test_name	testing_lab_code
1	480533	Cystic Fibrosis Profile, 32 Mutations, DNA Anal...	3151
2	511154	Factor V Lieden Mutation, DNA Analysis	3151
3	511162	Factor II, DNA Analysis	3151
4	511238	MTHFR, DNA Analysis	3151
5	511402	Maternal Cell Contamination	3151

Data Question # 3: Which patients are residing in a particular location?

Below is a select statement that brings back a list of patient names and phone numbers for all patients residing in Cary, NC.

```
select first_name, last_name, phone_number from rl_patient where city = 'cary'
```

	first_name	last_name	phone_number
1	John	Smith	919-467-8342
2	Sarah	Smith	919-523-1271
3	Annette	White	919-807-3256

Data Question # 8: Which test results are still pending (awaiting completion)?

Below is a select statement that brings back all records for which there are no results in the system.

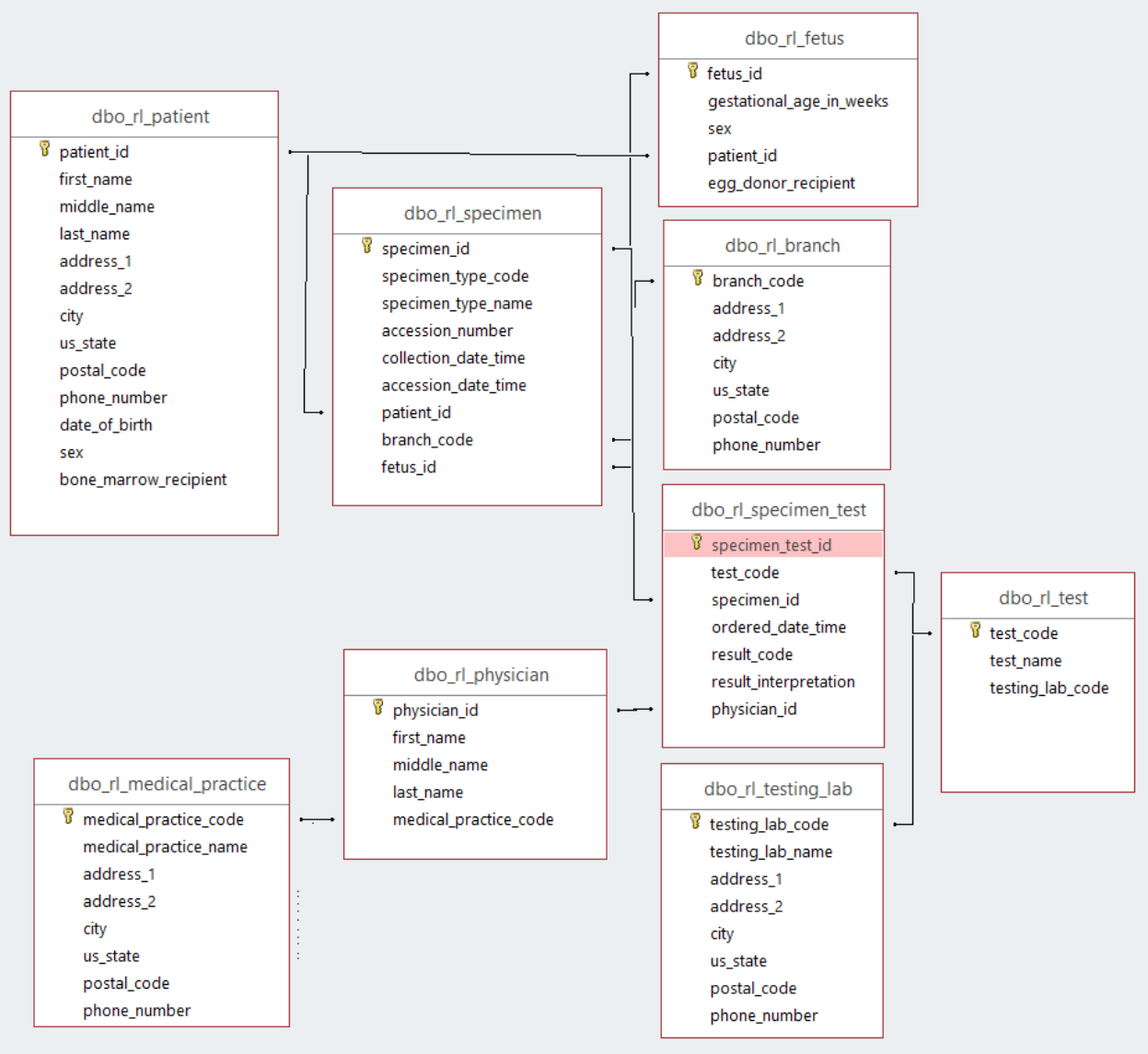
```
select * from rl_specimen_test where result_code = 'pending'
```

specimen_test_id	test_code	specimen_id	ordered_date_time	result_code	result_interpretation	physician_id
1	480533	27	2018-06-13 00:00:00.000	pending	pending	1
3	489067	29	2018-06-15 00:00:00.000	pending	pending	2
5	511402	31	2018-06-17 00:00:00.000	pending	pending	4
7	511966	33	2018-06-19 00:00:00.000	pending	pending	5
8	511162	28	2018-06-28 00:00:00.000	pending	pending	1
9	511154	28	2018-06-28 00:00:00.000	pending	pending	1
10	511238	28	2018-06-28 00:00:00.000	pending	pending	1
11	511154	35	2018-06-28 00:00:00.000	pending	pending	2

Important Note: Data questions can also be answered in Access. See section IX, *Implementation*, for details.

IX. Implementation

An ODBC was utilized to connect SQL to Access. All tables were imported into Access and relationships were established (figure below).

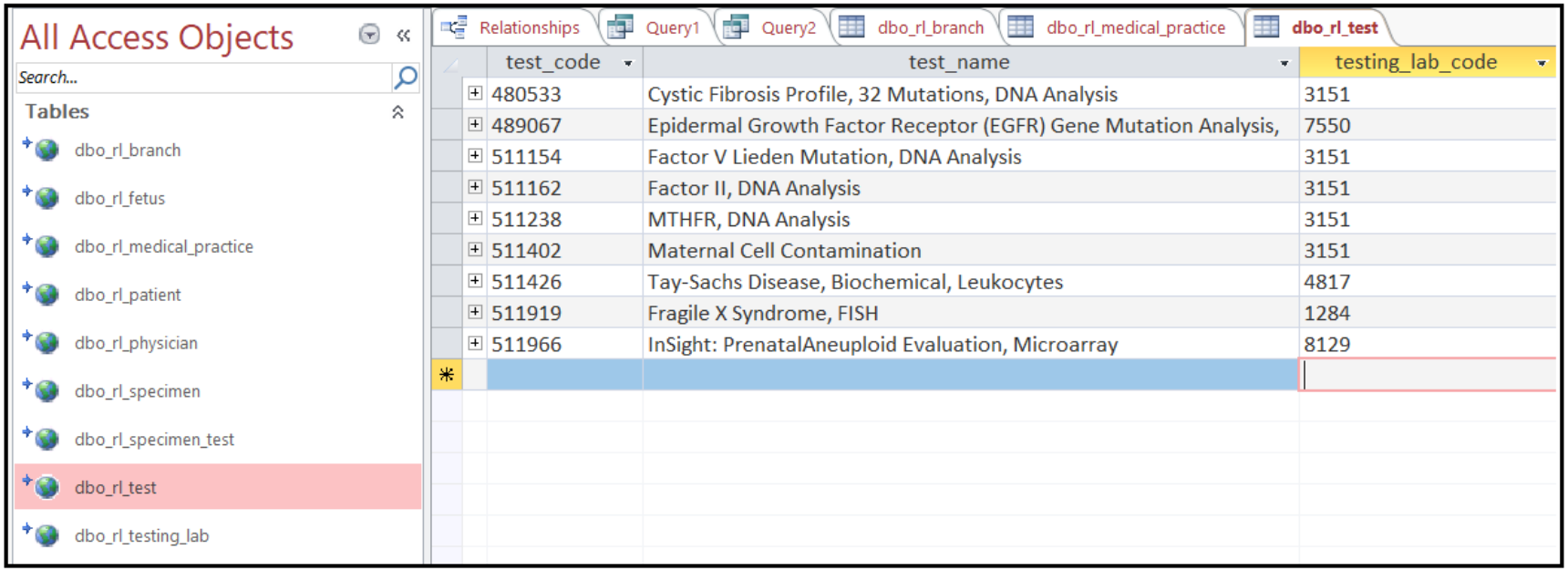


This interface allows users to do several things in Access:

1. Tables can be viewed.
2. Data can be entered.
3. Data can be updated.
4. Queries can be performed (and saved).
5. Forms can be generated.
6. Reports can be generated.

These actions will be discussed in detail on the following pages.

1. Tables can be viewed. Below, the user has clicked on the rl_test table, and all information in the table can be viewed.



test_code	test_name	testing_lab_code
480533	Cystic Fibrosis Profile, 32 Mutations, DNA Analysis	3151
489067	Epidermal Growth Factor Receptor (EGFR) Gene Mutation Analysis,	7550
511154	Factor V Lieden Mutation, DNA Analysis	3151
511162	Factor II, DNA Analysis	3151
511238	MTHFR, DNA Analysis	3151
511402	Maternal Cell Contamination	3151
511426	Tay-Sachs Disease, Biochemical, Leukocytes	4817
511919	Fragile X Syndrome, FISH	1284
511966	InSight: PrenatalAneuploid Evaluation, Microarray	8129

This is the same information a user would see in SQL using the select statement "`select * from rl_test`"

test_code	test_name	testing_lab_code
480533	Cystic Fibrosis Profile, 32 Mutations, DNA Analysis	3151
489067	Epidermal Growth Factor Receptor (EGFR) Gene Mutation Analysis, ...	7550
511154	Factor V Lieden Mutation, DNA Analysis	3151
511162	Factor II, DNA Analysis	3151
511238	MTHFR, DNA Analysis	3151
511402	Maternal Cell Contamination	3151
511426	Tay-Sachs Disease, Biochemical, Leukocytes	4817
511919	Fragile X Syndrome, FISH	1284
511966	InSight: PrenatalAneuploid Evaluation, Microarray	8129

2. Data can be entered. Below, an additional patient was added to Access.

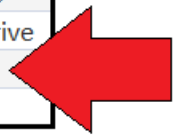
patient_id	first_name	middle_name	last_name	address_1
1	John	Gary	Smith	162 Wildfell Trail
2	Sarah	Johnson	Smith	162 Wildfell Trail
3	Annette	Lorraine	White	2033 W Park Street
4	James	Rodger	Hume	2913 Green Hill Drive
5	Max	L.	Thompson	3604 Morningside Drive
6	Sydney	R.	Jones	525 Walnut Court
7	Bryce		Day	413 Melinda Drive

SQL was automatically updated:

patient_id	first_name	middle_name	last_name	address_1
1	John	Gary	Smith	162 Wildfell Trail
2	Sarah	Johnson	Smith	162 Wildfell Trail
3	Annette	Lorraine	White	2033 W Park Street
4	James	Rodger	Hume	2913 Green Hill Drive
5	Max	L.	Thompson	3604 Morningside Drive
6	Sydney	R.	Jones	525 Walnut Court
7	Bryce	NULL	Day	413 Melinda Drive

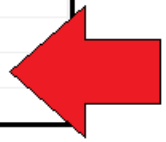
3. Data can be updated. Below, the user has clicked on the patient table and updated the street address for a patient.

patient_id	first_name	middle_name	last_name	address_1
1	John	Gary	Smith	162 Wildfell Trail
2	Sarah	Johnson	Smith	162 Wildfell Trail
3	Annette	Lorraine	White	2033 W Park Street
4	James	Rodger	Hume	2913 Green Hill Drive
5	Max	L.	Thompson	3604 Morningside Drive
6	Sydney	R.	Jones	315 Buckfield Lane
7	Bryce		Day	413 Melinda Drive



SQL was automatically updated:

patient_id	first_name	middle_name	last_name	address_1
1	John	Gary	Smith	162 Wildfell Trail
2	Sarah	Johnson	Smith	162 Wildfell Trail
3	Annette	Lorraine	White	2033 W Park Street
4	James	Rodger	Hume	2913 Green Hill Drive
5	Max	L.	Thompson	3604 Morningside Drive
6	Sydney	R.	Jones	315 Buckfield Lane
7	Bryce	NULL	Day	413 Melinda Drive

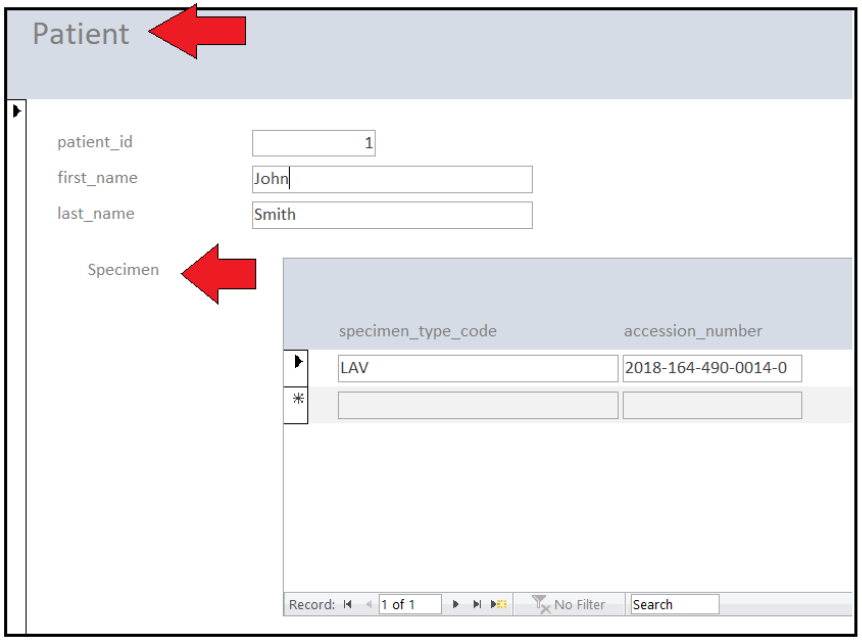


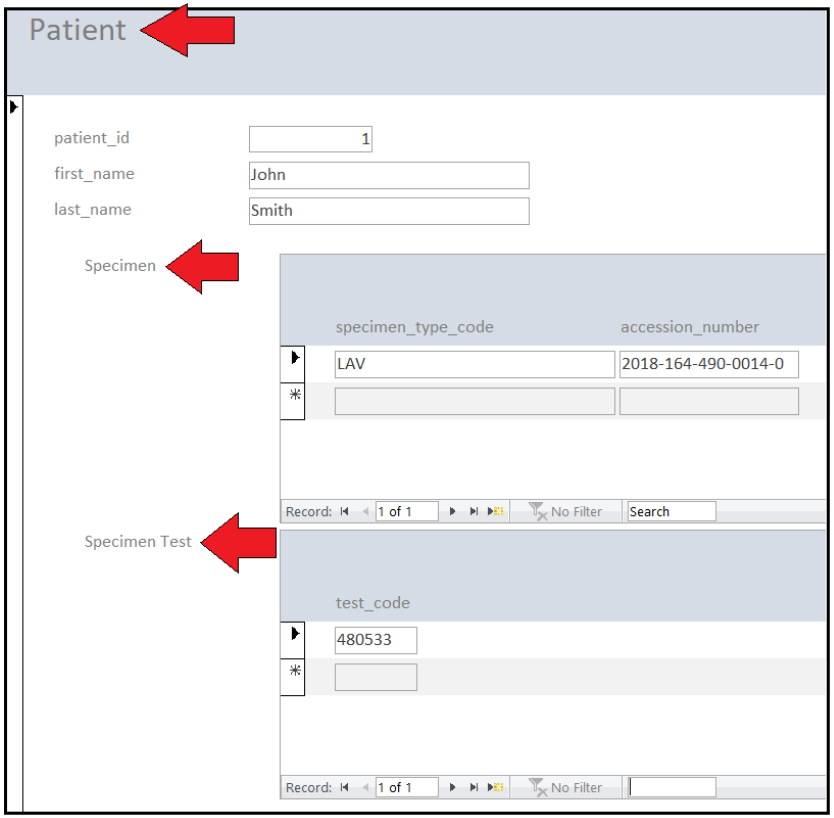
4. **Queries** can be performed. Below, a query was written that brings back:
    a. From the patient table: first and last name
    b. From the specimen table: specimen type code and accession number
    c. From the test table: test name
    d. From the specimen-test table: test code and result code

first_name	last_name	accession_number	specimen_type_code	test_code	result_code
John	Smith	2018-164-490-0014-0	LAV	480533	pending
Sarah	Smith	2018-152-490-0029-0	LAV	480533	CF-HET
Annette	White	2018-166-725-0001-0	FFPE	489067	pending
James	Hume	2018-166-335-0012-0	LAV	511919	FX-NEG
Max	Thompson	2018-168-237-0056-0	YEL	511402	pending
Sydney	Jones	2018-170-525-0001-0	LAV	511426	NO MCC
Sydney	Jones	2018-170-525-0001-0	AMD	511966	pending
Sarah	Smith	2018-152-490-0029-0	LAV	511162	pending
Sarah	Smith	2018-152-490-0029-0	LAV	511154	pending
Sarah	Smith	2018-152-490-0029-0	LAV	511238	pending
Annette	White	2018-179-725-0024-0	BUC	511154	pending

5. **Forms** can be generated. If viewing tables does not meet business needs, forms can be created. Below are two example forms.

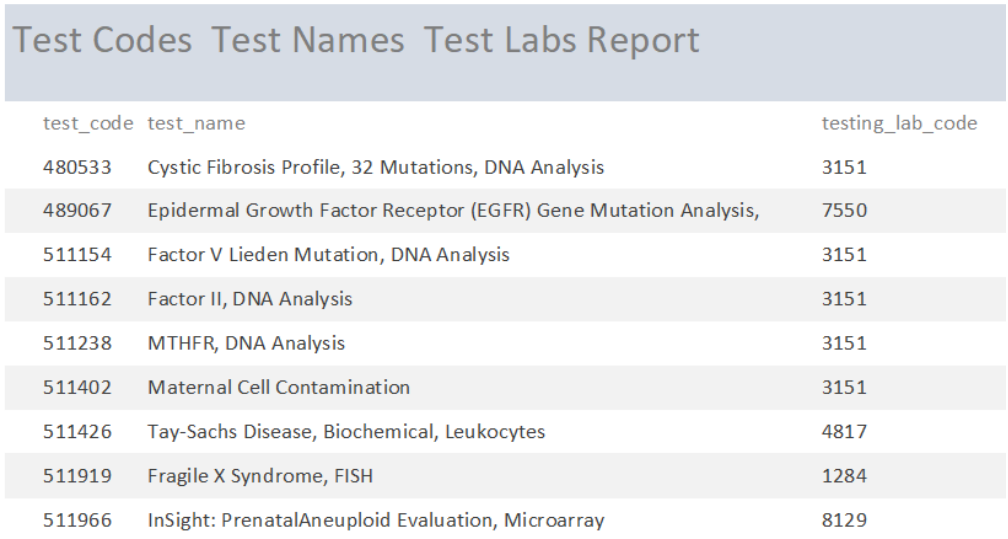
- On the left, columns from the Patient and Specimen tables were included in the form.
- On the right, the same form was created, and then a column from the Specimen Test table was added.





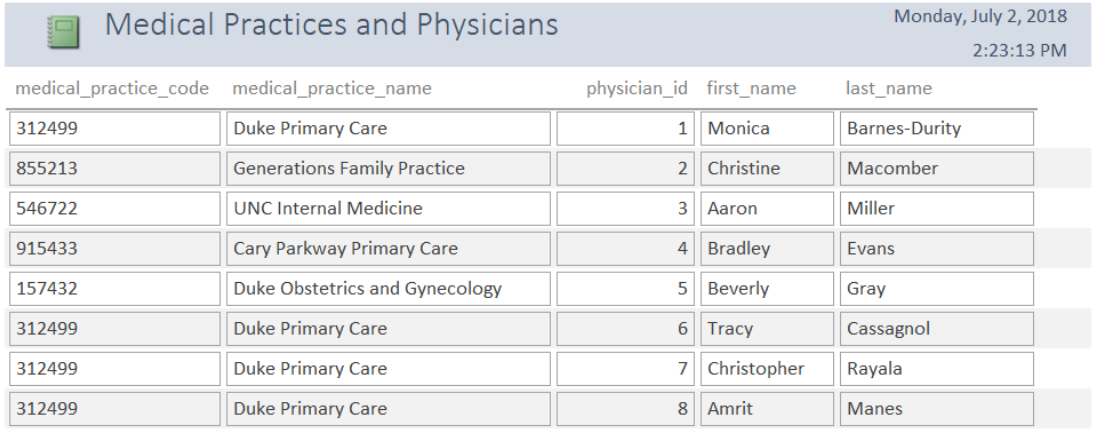
Forms are similar to tables in that changes to records can be made, and these changes will automatically be made in SQL. Forms can be formatted in various ways to meet business needs and make data manipulation user friendly.

6. **Reports** can be generated. These are read only and hard copies can be printed.
    - Top Image: A report listing test codes, names, and the code of the testing lab that performs each test.
    - Bottom Image: An example of a report listing Medical Practices and Physicians.



Test Codes Test Names Test Labs Report

test_code	test_name	testing_lab_code
480533	Cystic Fibrosis Profile, 32 Mutations, DNA Analysis	3151
489067	Epidermal Growth Factor Receptor (EGFR) Gene Mutation Analysis,	7550
511154	Factor V Lieden Mutation, DNA Analysis	3151
511162	Factor II, DNA Analysis	3151
511238	MTHFR, DNA Analysis	3151
511402	Maternal Cell Contamination	3151
511426	Tay-Sachs Disease, Biochemical, Leukocytes	4817
511919	Fragile X Syndrome, FISH	1284
511966	InSight: PrenatalAneuploid Evaluation, Microarray	8129



Medical Practices and Physicians

Monday, July 2, 2018
2:23:13 PM

medical_practice_code	medical_practice_name	physician_id	first_name	last_name
312499	Duke Primary Care	1	Monica	Barnes-Durity
855213	Generations Family Practice	2	Christine	Macomber
546722	UNC Internal Medicine	3	Aaron	Miller
915433	Cary Parkway Primary Care	4	Bradley	Evans
157432	Duke Obstetrics and Gynecology	5	Beverly	Gray
312499	Duke Primary Care	6	Tracy	Cassagnol
312499	Duke Primary Care	7	Christopher	Rayala
312499	Duke Primary Care	8	Amrit	Manes

X. Security

- Due to HIPAA, it is of the utmost importance to protect patient information and maintain patient privacy.
- Before being granted access, employees must sign the company's policy acknowledging that they are only authorized to look up information needed to perform their job. It is not permissible to look up information about themselves, family members, or friends. Additionally, employees must acknowledge that they may not discuss patient results outside of work. Furthermore, employees must not share their login information with anyone. Failure to follow the company policy can result in termination.
- Users will be prompted to change their passwords every three months. If a user's password isn't changed within 7 days of the prompt, the user will be locked out.
- Supervisors are responsible for notifying IT when an employee separates, so that privileges can be revoked.

XI. Possible Future Improvements

This database is flexible, and beyond maintaining the database, the database team can make changes as the reference lab sees fit. Below are some possible improvements that could be made in the future.

- Add a column to the specimen_test table that auto-populates the result date and time as the result is being entered. Then a Stored Procedure could be written to calculate how long testing took, by calculating the difference between the accession date/time and the result date/time.
- Add a specimen_type_name table and enter a list of specimen type codes with their corresponding descriptions. This way, the specimen_type_names don't have to be manually entered into the specimen_test table each time.
- Change the accession number to allow for more characters. This would allow for additional information to describe the specimen.
    - Add -FA if the accession is for a specimen collected from Fetus A, Add -FB for a specimen collected from Fetus B, etc.
    - Add -MC if the specimen is being used as a maternal control.
    - Add -PC if the specimen is being used as a paternal control.

XII. Appendix I: Glossary

Amnio: Amniotic fluid containing fetal cells. This specimen type is collected using a process called amniocentesis.

Branch: The location where specimens have accession numbers assigned.

Carrier: A person who has the ability to pass on a disease but who is not affected by the disease.

HIPAA: The Health Insurance Portability and Accountability Act of 1996.

Julian Date: The day of the year, between 1 and 366. January 1 has a Julian Date of 1, January 2 has a Julian date of 2, and so on. The Julian date is important because the there is a published turn-around-time for each test, and having this number in the accession number helps the testing lab track when patients are expecting results.

Specimen: A specimen is a biological component that is removed from a patient or fetus for testing.

XIII. Appendix II: Representative Data:

Specimen

specimen_id	specimen_type_code	specimen_type_name	accession_number	collection_date_time	accession_date_time	patient_id	branch_code	fetus_id
27	LAV	whole blood in lavender-top (EDTA) tube	2018-164-490-0014-0	2018-06-13 13:22:00.000	2018-06-13 13:26:00.000	1	490	NULL
28	LAV	whole blood in lavender-top (EDTA) tube	2018-152-490-0029-0	2018-06-01 10:03:32.000	2018-06-01 10:04:05.000	2	490	NULL
29	FFPE	Formalin-fixed, paraffin-embedded tissue block	2018-166-725-0001-0	2018-06-15 13:10:32.000	2018-06-15 13:15:05.000	3	725	NULL
30	LAV	whole blood in lavender-top (EDTA) tube	2018-166-335-0012-0	2018-06-15 11:05:18.000	2018-06-15 14:10:27.000	4	335	NULL
31	YEL	whole blood in yellow-top (ACD) tube	2018-168-237-0056-0	2018-06-17 09:15:37.000	2018-06-17 09:21:53.000	5	237	NULL
32	LAV	whole blood in lavender-top (EDTA) tube	2018-170-525-0001-0	2018-06-19 08:47:17.000	2018-06-19 08:54:05.000	6	525	NULL
33	AMD	Amniotic Fluid	2018-170-525-0001-0	2018-06-17 14:03:57.000	2018-06-19 08:54:05.000	6	525	1
35	BUC	Buccal Swab Kit	2018-179-725-0024-0	2018-06-26 09:14:02.000	2018-06-28 09:54:16.000	3	725	NULL

Specimen Test

specimen_test_id	test_code	specimen_id	ordered_date_time	result_code	result_interpretation	physician_id
1	480533	27	2018-06-13 00:00:00.000	pending	pending	1
2	480533	28	2018-06-01 00:00:00.000	CF-HET	Carrier of the delta F508 Mutation	1
3	489067	29	2018-06-15 00:00:00.000	pending	pending	2
4	511919	30	2018-06-15 00:00:00.000	FX-NEG	21 repeats of the FMR1 gene	3
5	511402	31	2018-06-17 00:00:00.000	pending	pending	4
6	511426	32	2018-06-19 00:00:00.000	NO MCC	No Maternal Cell Contamination detected	5
7	511966	33	2018-06-19 00:00:00.000	pending	pending	5
8	511162	28	2018-06-28 00:00:00.000	pending	pending	1
9	511154	28	2018-06-28 00:00:00.000	pending	pending	1
10	511238	28	2018-06-28 00:00:00.000	pending	pending	1
11	511154	35	2018-06-28 00:00:00.000	pending	pending	2

Medical Practice

medical_practice_code	medical_practice_name	address_1	address_2	city	us_state	postal_code	phone_number
157432	Duke Obstetrics and Gynecology	5704 Fayetteville Street	NULL	Durham	NC	27713	919-684-4673
312499	Duke Primary Care	10950 Chapel Hill Road	NULL	Morrisville	NC	27560	919-980-4219
546722	UNC Internal Medicine	1181 Weaver Dairy Road	Suite 250	Chapel Hill	NC	27514	984-215-4340
855213	Generations Family Practice	1021 Darrington Drive	Suite 101	Cary	NC	27513	919-852-3999
915433	Cary Parkway Primary Care	3701 NW Cary Parkway	NULL	Cary	NC	27513	919-235-6415

Physician

physician_id	first_name	middle_name	last_name	medical_practice_code
1	Monica	NULL	Barnes-Durity	312499
2	Christine	NULL	Macomber	855213
3	Aaron	J.	Miller	546722
4	Bradley	NULL	Evans	915433
5	Beverly	A.	Gray	157432
6	Tracy	NULL	Cassagnol	312499
7	Christopher	Z	Rayala	312499
8	Amrit	K	Manes	312499

Branch

branch_code	address_1	address_2	city	us_state	postal_code	phone_number
237	3850 Ed Drive	NULL	Raleigh	NC	27612	919-571-6514
335	5324 McFarland Drive	100	Durham	NC	27707	919-401-0180
490	601 Kiesler Drive	NULL	Cary	NC	27518	919-233-6750
525	2945 S. Miami Blvd.	#100	Durham	NC	27703	919-361-2165
725	3601 Davis Drive	NULL	Morrisville	NC	27560	919-245-8813

Test

test_code	test_name	testing_lab_code
480533	Cystic Fibrosis Profile, 32 Mutations, DNA Analysis	3151
489067	Epidermal Growth Factor Receptor (EGFR) Gene Mut...	7550
511154	Factor V Lieden Mutation, DNA Analysis	3151
511162	Factor II, DNA Analysis	3151
511200	UGT 1A1 Irinotecan Toxicity	3151
511238	MTHFR, DNA Analysis	3151
511402	Maternal Cell Contamination	3151
511426	Tay-Sachs Disease, Biochemical, Leukocytes	4817
511919	Fragile X Syndrome, FISH	1284
511966	InSight: PrenatalAneuploid Evaluation, Microarray	8129

Patient

patient_id	first_name	middle_name	last_name	address_1	address_2	city	us_state	postal_code	phone_number	date_of_birth	sex	bone_marrow_recipient
1	John	Gary	Smith	162 Wildfell Trail	NULL	Cary	NC	27513	919-467-8342	1969-12-09	M	N
2	Sarah	Johnson	Smith	162 Wildfell Trail	NULL	Cary	NC	27513	919-523-1271	1972-05-12	F	N
3	Annette	Lorraine	White	2033 W Park Street	NULL	Cary	NC	27511	919-807-3256	1982-06-25	F	N
4	James	Rodger	Hume	2913 Green Hill Drive	NULL	Chapel Hill	NC	27514	919-306-8819	1974-09-18	M	N
5	Max	L.	Thompson	3604 Morningside Drive	NULL	Raleigh	NC	27607	NULL	1952-04-13	M	N
6	Sydney	R.	Jones	7 Wood Duck Court	NULL	Durham	NC	27713	919-539-4872	1995-11-01	F	N

Testing Lab

testing_lab_code	testing_lab_name	address_1	address_2	city	us_state	postal_code	phone_number
1284	Cytogenetics	1904 T.W. Alexander Drive	NULL	RTP	NC	27709	800-533-0318
3151	Molecular Genetics	1912 T.W. Alexander Drive	NULL	RTP	NC	27709	800-533-0567
4817	Biochemical Genetics	1912 T.W. Alexander Drive	NULL	RTP	NC	27709	800-533-0567
7550	Molecular Oncology	1904 T.W. Alexander Drive	NULL	RTP	NC	27709	800-533-0318
8129	Prenatal Genomics Screening	1912 T.W. Alexander Drive	NULL	RTP	NC	27709	800-533-0567

Fetus

fetus_id	gestational_age_in_weeks	sex	patient_id	egg_donor_recipient
1	9	M	6	N

XIV. Appendix III: SQL Code

```
-- create all tables in reference lab model

-- start creating rl_patient table
create table rl_patient(
-- columns for this table
	patient_id int identity,
	first_name varchar(30) not null,
	middle_name varchar(30),
	last_name varchar(30) not null,
	address_1 varchar(50) not null,
	address_2 varchar(50),
	city varchar(50) not null,
	us_state varchar(50) not null,
	postal_code varchar(10) not null,
	phone_number varchar(20),
	date_of_birth date not null,
	sex char(1) not null,
	bone_marrow_recipient char(1) not null,
-- constraints on this table
	constraint pk_rl_patient primary key (patient_id)
	)
-- end creating rl_patient table


-- start creating rl_medical_practice table
create table rl_medical_practice(
-- columns for this table
	medical_practice_code char(6),
	medical_practice_name varchar(50) not null,
	address_1 varchar(50) not null,
	address_2 varchar(50),
	city varchar(50) not null,
	us_state varchar(50) not null,
	postal_code varchar(10) not null,
	phone_number varchar(20),
-- constraints on this table
	constraint pk_rl_medical_practice primary key (medical_practice_code)
	)
-- end creating rl_medical_practice table
```

```
-- start creating rl_testing_lab table
create table rl_testing_lab(
-- columns for this table
	testing_lab_code varchar(10),
	testing_lab_name varchar(100) not null,
	address_1 varchar(50) not null,
	address_2 varchar(50),
	city varchar(50) not null,
	us_state varchar(50) not null,
	postal_code varchar(10) not null,
	phone_number varchar(20),
-- constraints on this table
	constraint pk_rl_testing_lab primary key (testing_lab_code)
	)
-- end creating testing_lab table

-- start creating rl_branch table
create table rl_branch(
-- columns for this table
	branch_code char(3),
	address_1 varchar(50) not null,
	address_2 varchar(50),
	city varchar(50) not null,
	us_state varchar(50) not null,
	postal_code varchar(10) not null,
	phone_number varchar(20) not null,
-- constraints on this table
	constraint pk_rl_branch primary key (branch_code)
	)
-- end creating rl_branch table

-- start creating rl_test table
create table rl_test(
--columns for this table
	test_code char(6),
	test_name varchar(100) not null,
	testing_lab_code varchar(10) not null,
--constraints on this table
	constraint pk_rl_test primary key (test_code),
	constraint fk1_rl_test foreign key (testing_lab_code) references rl_testing_lab(testing_lab_code)
	)
--end creating rl_test table
```

```
-- start creating rl_fetus table
create table rl_fetus(
--columns for this table
	fetus_id int identity,
	gestational_age_in_weeks tinyint,
	sex char(1),
	patient_id int,
	egg_donor_recipient char(1) not null,
--constraints on this table
	constraint pk_rl_fetus primary key (fetus_id),
	constraint fk1_rl_fetus foreign key (patient_id) references rl_patient(patient_id)
	)
--end creating rl_fetus table


-- start creating rl_specimen table
create table rl_specimen(
--columns for this table
	specimen_id int identity,
	specimen_type_code varchar(30) not null,
	specimen_type_name varchar(100),
	accession_number char(19) not null,
	collection_date_time datetime not null,
	accession_date_time datetime not null,
	patient_id int,
	branch_code char(3),
	fetus_id int,
--constraints on this table
	constraint pk_rl_specimen primary key (specimen_id),
	constraint fk1_rl_specimen foreign key (patient_id) references rl_patient(patient_id),
	constraint fk2_rl_specimen foreign key (branch_code) references rl_branch(branch_code),
	constraint fk3_rl_specimen foreign key (fetus_id) references rl_fetus(fetus_id)
	)
--end creating rl_specimen table
```

```
-- start creating specimen_test table
create table rl_specimen_test(
--columns for this table
	specimen_test_id int identity,
	test_code char(6),
	specimen_id int,
	ordered_date_time datetime,
	result_code varchar(50),
	result_interpretation varchar(200),
--constraints on this table
	constraint pk_rl_specimen_test primary key (specimen_test_id),
	constraint fk1_rl_specimen_test foreign key (test_code) references rl_test(test_code),
	constraint fk2_rl_specimen_test foreign key (specimen_id) references rl_specimen(specimen_id),
	)
--end creating rl_specimen_test table

-- alter specimen_test table - add column
alter table rl_specimen_test
add physician_id int

-- alter specimen_test table - add constraint
alter table rl_specimen_test
add constraint fk3_rl_specimen_test foreign key (physician_id) references rl_physician(physician_id)


-- start creating rl_phyician table
create table rl_physician(
--columns for this table
	physician_id int identity,
	first_name varchar(30) not null,
	middle_name varchar(30),
	last_name varchar(30) not null,
	medical_practice_code char(6),
--constraints on this table
	constraint pk_rl_physician primary key (physician_id),
	constraint fk1_rl_physician foreign key (medical_practice_code) references
rl_medical_practice(medical_practice_code),
	)
--end creating rl_physician table

-- end creating all tables in reference lab model
```

```
-- start adding data to all tables in reference lab database

-- start adding data to the rl_patient table
insert into rl_patient
	(first_name, middle_name, last_name, address_1, city, us_state, postal_code, phone_number, date_of_birth, sex,
bone_marrow_recipient)
	values
	('John', 'Gary', 'Smith', '162 Wildfell Trail', 'Cary', 'NC', '27513', '919-467-8342', '12/09/1969', 'M', 'N')

insert into rl_patient
	(first_name, middle_name, last_name, address_1, city, us_state, postal_code, phone_number, date_of_birth, sex,
bone_marrow_recipient)
	values
	('Sarah', 'Johnson', 'Smith', '162 Wildfell Trail', 'Cary', 'NC', '27513', '919-523-1271', '05/12/1972', 'F', 'N')

insert into rl_patient
	(first_name, middle_name, last_name, address_1, city, us_state, postal_code, phone_number, date_of_birth, sex,
bone_marrow_recipient)
	values
	('Annette', 'Lorraine', 'White', '2033 W Park Street', 'Cary', 'NC', '27511', '919-807-3256', '06/25/1982', 'F', 'N')

insert into rl_patient
	(first_name, middle_name, last_name, address_1, city, us_state, postal_code, phone_number, date_of_birth, sex,
bone_marrow_recipient)
	values
	('James', 'Rodger', 'Hume', '2913 Green Hill Drive', 'Chapel Hill', 'NC', '27514', '919-306-8819', '09/18/1974', 'M',
'N')

insert into rl_patient
	(first_name, middle_name, last_name, address_1, city, us_state, postal_code, date_of_birth, sex,
bone_marrow_recipient)
	values
	('Max', 'L.', 'Thompson', '3604 Morningside Drive', 'Raleigh', 'NC', '27607', '04/13/1952', 'M', 'N')

insert into rl_patient
	(first_name, middle_name, last_name, address_1, city, us_state, postal_code, phone_number, date_of_birth, sex,
bone_marrow_recipient)
	values
	('Sydney', 'R.', 'Jones', '7 Wood Duck Court', 'Durham', 'NC', '27713', '919-539-4872', '11/01/1995', 'F', 'N')
-- end adding data to rl_patient table
```

```
-- start adding data in rl_medical practice table
insert into rl_medical_practice
       (medical_practice_code, medical_practice_name, address_1, address_2, city, us_state, postal_code, phone_number)
       values
       ('312499', 'Duke Primary Care', '10950 Chapel Hill Road', null, 'Morrisville', 'NC', '27560', '919-980-4219')

insert into rl_medical_practice
       (medical_practice_code, medical_practice_name, address_1, address_2, city, us_state, postal_code, phone_number)
       values
       ('855213', 'Generations Family Practice', '1021 Darrington Drive', 'Suite 101', 'Cary', 'NC', '27513', '919-852-
3999')

insert into rl_medical_practice
       (medical_practice_code, medical_practice_name, address_1, address_2, city, us_state, postal_code, phone_number)
       values
       ('546722', 'UNC Internal Medicine', '1181 Weaver Dairy Road', 'Suite 250', 'Chapel Hill', 'NC', '27514', '984-215-
4340')

insert into rl_medical_practice
       (medical_practice_code, medical_practice_name, address_1, address_2, city, us_state, postal_code, phone_number)
       values
       ('915433', 'Cary Parkway Primary Care', '3701 NW Cary Parkway', null, 'Cary', 'NC', '27513', '919-235-6415')

insert into rl_medical_practice
       (medical_practice_code, medical_practice_name, address_1, address_2, city, us_state, postal_code, phone_number)
       values
       ('157432', 'Duke Obsetrics and Gynecology', '5704 Fayetteville Street', null, 'Durham', 'NC', '27713', '919-684-
4673')
-- end adding data in rl_medical practice table
```

```sql
-- start adding data into rl_testing lab table
insert into rl_testing_lab
	(testing_lab_code, testing_lab_name, address_1, address_2, city, us_state, postal_code, phone_number)
	values
	('3151', 'Molecular Genetics', '1912 T.W. Alexander Drive', null, 'RTP', 'NC', '27709', '800-533-0567')

insert into rl_testing_lab
	(testing_lab_code, testing_lab_name, address_1, address_2, city, us_state, postal_code, phone_number)
	values
	('7550', 'Molecular Oncology', '1904 T.W. Alexander Drive', null, 'RTP', 'NC', '27709', '800-533-0318')

insert into rl_testing_lab
	(testing_lab_code, testing_lab_name, address_1, address_2, city, us_state, postal_code, phone_number)
	values
	('1284', 'Cytogenetics', '1904 T.W. Alexander Drive', null, 'RTP', 'NC', '27709', '800-533-0318')

insert into rl_testing_lab
	(testing_lab_code, testing_lab_name, address_1, address_2, city, us_state, postal_code, phone_number)
	values
	('4817', 'Biochemical Genetics', '1912 T.W. Alexander Drive', null, 'RTP', 'NC', '27709', '800-533-0567')

insert into rl_testing_lab
	(testing_lab_code, testing_lab_name, address_1, address_2, city, us_state, postal_code, phone_number)
	values
	('8129', 'Prenatal Genomics Screening', '1912 T.W. Alexander Drive', null, 'RTP', 'NC', '27709', '800-533-0567')
-- end adding data into rl_testing lab table
```

```
-- start adding data into branch table
insert into rl_branch
       (branch_code, address_1, address_2, city, us_state, postal_code, phone_number)
       values
       ('490', '601 Kiesler Drive', null, 'Cary', 'NC', '27518', '919-233-6750')

insert into rl_branch
       (branch_code, address_1, address_2, city, us_state, postal_code, phone_number)
       values
       ('725', '3601 Davis Drive', null, 'Morrisville', 'NC', '27560', '919-245-8813')

insert into rl_branch
       (branch_code, address_1, address_2, city, us_state, postal_code, phone_number)
       values
       ('335', '5324 McFarland Drive', '100', 'Durham', 'NC', '27707', '919-401-0180')

insert into rl_branch
       (branch_code, address_1, address_2, city, us_state, postal_code, phone_number)
       values
       ('237', '3850 Ed Drive', null, 'Raleigh', 'NC', '27612', '919-571-6514')

insert into rl_branch
       (branch_code, address_1, address_2, city, us_state, postal_code, phone_number)
       values
       ('525', '2945 S. Miami Blvd.', '#100', 'Durham', 'NC', '27703', '919-361-2165')
-- end adding data into rl_branch table
```

```
--start adding data into rl_test table
insert into rl_test
       (test_code, test_name, testing_lab_code)
       values
       ('480533', 'Cystic Fibrosis Profile, 32 Mutations, DNA Analysis', '3151')

insert into rl_test
       (test_code, test_name, testing_lab_code)
       values
       ('489067', 'Epidermal Growth Factor Receptor (EGFR) Gene Mutation Analysis,
       Real-time PCR', '7550')

insert into rl_test
       (test_code, test_name, testing_lab_code)
       values
       ('511919', 'Fragile X Syndrome, FISH', '1284')

insert into rl_test
       (test_code, test_name, testing_lab_code)
       values
       ('511426', 'Tay-Sachs Disease, Biochemical, Leukocytes', '4817')

insert into rl_test
       (test_code, test_name, testing_lab_code)
       values
       ('511966', 'InSight: PrenatalAneuploid Evaluation, Microarray', '8129')

insert into rl_test
       (test_code, test_name, testing_lab_code)
       values
       ('511402', 'Maternal Cell Contamination', '3151')
-- end adding data into rl_test table
```

```
-- start adding data into rl_physician table
insert into rl_physician
       (first_name, middle_name, last_name, medical_practice_code)
       values
       ('Monica', null, 'Barnes-Durity', '312499')

insert into rl_physician
       (first_name, middle_name, last_name, medical_practice_code)
       values
       ('Christine', null, 'Macomber', '855213')

insert into rl_physician
       (first_name, middle_name, last_name, medical_practice_code)
       values
       ('Aaron', 'J.', 'Miller', '546722')

insert into rl_physician
       (first_name, middle_name, last_name, medical_practice_code)
       values
       ('Bradley', null, 'Evans', '915433')

insert into rl_physician
       (first_name, middle_name, last_name, medical_practice_code)
       values
       ('Beverly', 'A.', 'Gray', '157432')
-- end adding data to physician table


-- start adding data into rl_fetus table
insert into rl_fetus
       (gestational_age_in_weeks, sex, patient_id, egg_donor_recipient)
       values
       ('9', 'M', '6', 'N')
-- end adding data into rl_fetus table

-- start adding data into rl_specimen table
insert into rl_specimen
       (specimen_type_code, specimen_type_name, accession_number, collection_date_time, accession_date_time, patient_id,
branch_code, fetus_id)
       values
       ('LAV', 'whole blood in lavender-top (EDTA) tube', '2018-164-490-0014-0', '6/13/2018 13:22', '6/13/2018 13:26', '1',
'490', null)
```

```
insert into rl_specimen
	(specimen_type_code, specimen_type_name, accession_number, collection_date_time, accession_date_time, patient_id,
branch_code, fetus_id)
	values
	('LAV', 'whole blood in lavender-top (EDTA) tube', '2018-152-490-0029-0', '2018-06-01 10:03:32', '2018-06-01
10:04:05', '2', '490', null)

insert into rl_specimen
	(specimen_type_code, specimen_type_name, accession_number, collection_date_time, accession_date_time, patient_id,
branch_code, fetus_id)
	values
	('FFPE', 'Formalin-fixed, paraffin-embedded tissue block', '2018-166-725-0001-0', '2018-06-15 13:10:32', '2018-06-15
13:15:05', '3', '725', null)

insert into rl_specimen
	(specimen_type_code, specimen_type_name, accession_number, collection_date_time, accession_date_time, patient_id,
branch_code, fetus_id)
	values
	('LAV', 'whole blood in lavender-top (EDTA) tube', '2018-166-335-0012-0', '2018-06-15 11:05:18', '2018-06-15
14:10:27', '4', '335', null)

insert into rl_specimen
	(specimen_type_code, specimen_type_name, accession_number, collection_date_time, accession_date_time, patient_id,
branch_code, fetus_id)
	values
	('YEL', 'whole blood in yellow-top (ACD) tube', '2018-168-237-0056-0', '2018-06-17 09:15:37', '2018-06-17 09:21:53',
'5', '237', null)

insert into rl_specimen
	(specimen_type_code, specimen_type_name, accession_number, collection_date_time, accession_date_time, patient_id,
branch_code, fetus_id)
	values
	('LAV', 'whole blood in lavender-top (EDTA) tube', '2018-170-525-0001-0', '2018-06-19 08:47:17', '2018-06-19
08:54:05', '6', '525', null)

insert into rl_specimen
	(specimen_type_code, specimen_type_name, accession_number, collection_date_time, accession_date_time, patient_id,
branch_code, fetus_id)
	values
	('AMD', 'Amniotic Fluid', '2018-170-525-0001-0', '2018-06-17 14:03:57', '2018-06-19 08:54:05', '6', '525', '1')

-- end adding data to rl_specimen table
```

```
-- alter rl_specimen_test table so that if not results have been entered, a default value of 'pending' shows up
alter table rl_specimen_test
add constraint df_pending_rc default 'pending'for result_code
alter table rl_specimen_test
add constraint df_pending_ri default 'pending'for result_interpretation

-- start adding data to rl_specimen_test table
insert into rl_specimen_test
       (test_code, specimen_id)
values
       ('480533', '27')

insert into rl_specimen_test
       (test_code, specimen_id, result_code, result_interpretation)
values
       ('480533', '28', 'CF-HET', 'Carrier of the delta F508 Mutation')

insert into rl_specimen_test
       (test_code, specimen_id)
values
       ('489067', '29')

insert into rl_specimen_test
       (test_code, specimen_id, result_code, result_interpretation)
values
       ('511919', '30', 'FX-NEG', '21 repeats of the FMR1 gene')

insert into rl_specimen_test
       (test_code, specimen_id)
values
       ('511402', '31')
insert into rl_specimen_test
       (test_code, specimen_id, result_code, result_interpretation)
values
       ('511426', '32', 'NO MCC', 'No Maternal Cell Contamination detected')
insert into rl_specimen_test
       (test_code, specimen_id)
values
       ('511966', '33')
-- end adding data to sample_result table
```

```
-- add ordered_date_time to specimens in specimen_test table
update rl_specimen_test set ordered_date_time = '6/13/18' where specimen_id = 27
update rl_specimen_test set ordered_date_time = '6/01/18' where specimen_id = 28
update rl_specimen_test set ordered_date_time = '6/15/18' where specimen_id = 29
update rl_specimen_test set ordered_date_time = '6/15/18' where specimen_id = 30
update rl_specimen_test set ordered_date_time = '6/17/18' where specimen_id = 31
update rl_specimen_test set ordered_date_time = '6/19/18' where specimen_id = 32
update rl_specimen_test set ordered_date_time = '6/19/18' where specimen_id = 33

-- add physician_id to specimens in specimen_test table
update rl_specimen_test set physician_id = '1' where specimen_id = 27
update rl_specimen_test set physician_id = '1' where specimen_id = 28
update rl_specimen_test set physician_id = '2' where specimen_id = 29
update rl_specimen_test set physician_id = '3' where specimen_id = 30
update rl_specimen_test set physician_id = '4' where specimen_id = 31
update rl_specimen_test set physician_id = '5' where specimen_id = 32
update rl_specimen_test set physician_id = '5' where specimen_id = 33


-- add more tests for lab 3151 to perform
insert into rl_test
       (test_code, test_name, testing_lab_code)
       values
       ('511162', 'Factor II, DNA Analysis', '3151')

insert into rl_test
       (test_code, test_name, testing_lab_code)
       values
       ('511154', 'Factor V Lieden Mutation, DNA Analysis', '3151')

insert into rl_test
       (test_code, test_name, testing_lab_code)
       values
       ('511238', 'MTHFR, DNA Analysis', '3151')

insert into rl_test
       (test_code, test_name, testing_lab_code)
       values
       ('511200', 'UGT 1A1 Irinotecan Toxicity', '3151')
```

```
-- add additional physicians to an existing practice
insert into rl_physician
       (first_name, middle_name, last_name, medical_practice_code)
       values
       ('Tracy', null, 'Cassagnol', '312499')

insert into rl_physician
       (first_name, middle_name, last_name, medical_practice_code)
       values
       ('Christopher', 'Z', 'Rayala', '312499')

insert into rl_physician
       (first_name, middle_name, last_name, medical_practice_code)
       values
       ('Amrit', 'K', 'Manes', '312499')


-- add more test orders onto an existing sample
insert into rl_specimen_test
       (test_code, specimen_id)
values
       ('511162', '28')

insert into rl_specimen_test
       (test_code, specimen_id)
values
       ('511154', '28')

insert into rl_specimen_test
       (test_code, specimen_id)
values
       ('511238', '28')


-- add new specimen onto existing patient
insert into rl_specimen
       (specimen_type_code, specimen_type_name, accession_number, collection_date_time, accession_date_time, patient_id,
branch_code, fetus_id)
       values
       ('BUC', 'Buccal Swab Kit', '2018-179-725-0024-0', '2018-06-26 09:14:02', '2018-06-28 09:54:16', '3', '725', null)
```

```sql
-- fix typo
update rl_medical_practice set medical_practice_name = 'Duke Obstetrics and Gynecology' where medical_practice_code = 157432

-- add new test onto new specimen
insert into rl_specimen_test
       (test_code, specimen_id)
values
       ('511154', '35')

-- add ordered_date_time to specimens in specimen_test table
update rl_specimen_test set ordered_date_time = '6/28/18' where specimen_test_id = 8
update rl_specimen_test set ordered_date_time = '6/28/18' where specimen_test_id = 9
update rl_specimen_test set ordered_date_time = '6/28/18' where specimen_test_id = 10
update rl_specimen_test set ordered_date_time = '6/28/18' where specimen_test_id = 11


-- add physician_id to specimens in specimen_test table
update rl_specimen_test set physician_id = '1' where specimen_id = 28
update rl_specimen_test set physician_id = '2' where specimen_id = 35


-- select all records in the test table that were run by a particular lab
select * from rl_test where testing_lab_code = 3151

-- select all physician IDs from the specimen_test table who ordered a particular test
select physician_id from rl_specimen_test where test_code = 511154

-- select records from patient table, but just for patients in the city of Cary
select first_name, last_name, phone_number from rl_patient where city = 'cary'

-- select all records from specimen test table with no results (results are still pending)
select * from rl_specimen_test where result_code = 'pending'

-- delete the UGT test - the lab no longer offers this test
delete from rl_test where test_code = 511200
```

XIV. Reflections:

- While working on my database project in SQL, it all started to come together. It went from “oh, this is so tedious” to “oh, this is fun. I think I got bitten by the bug.”
- It was a huge pain that “sample” was a reserved word, and I decided to use “specimen” instead. I had to re-do a lot of things. Thank goodness for *Ctrl H* in MS Word.
- I actually got all of my tables in without errors, and my data generally went in fine too. The more time-consuming part was when I realized I had made a typo in the data, and I had to figure out what kind of SQL statement to use to fix things. That turned out okay though, since the project was calling for SQL statements showing UPDATE, ALTER, etc.
- I definitely learned a lot about what goes on behind the scenes. Regardless of how much or how little I use SQL and Access in the future, I have an added appreciation of what can be done and the effort it takes.
- If I had to do it all again, I would very carefully review my name choices before adding my tables in SQL, since it’s not so easy to change them after they’ve been added.