CS5200 Fall 2020: Practicum 1

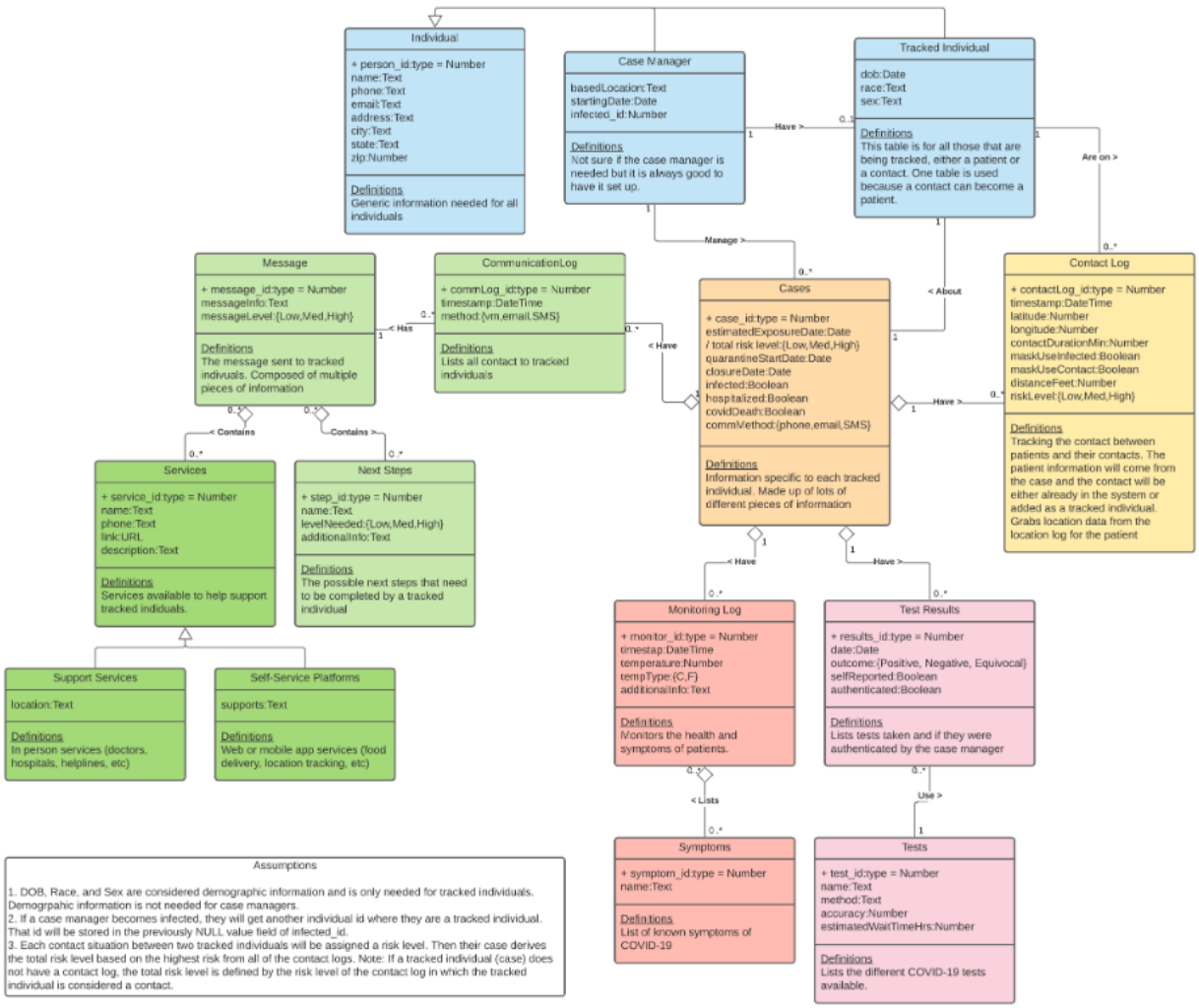
Chandra Davis, Evan Douglass

The steps we completed for Practicum 1 are detailed below. Please note that there are links to each image that requires it in each section. We decided to focus our attention on the case management aspect of the contract tracing problem.

Conceptual Model: UML

View the conceptual model in Lucid Chart here:

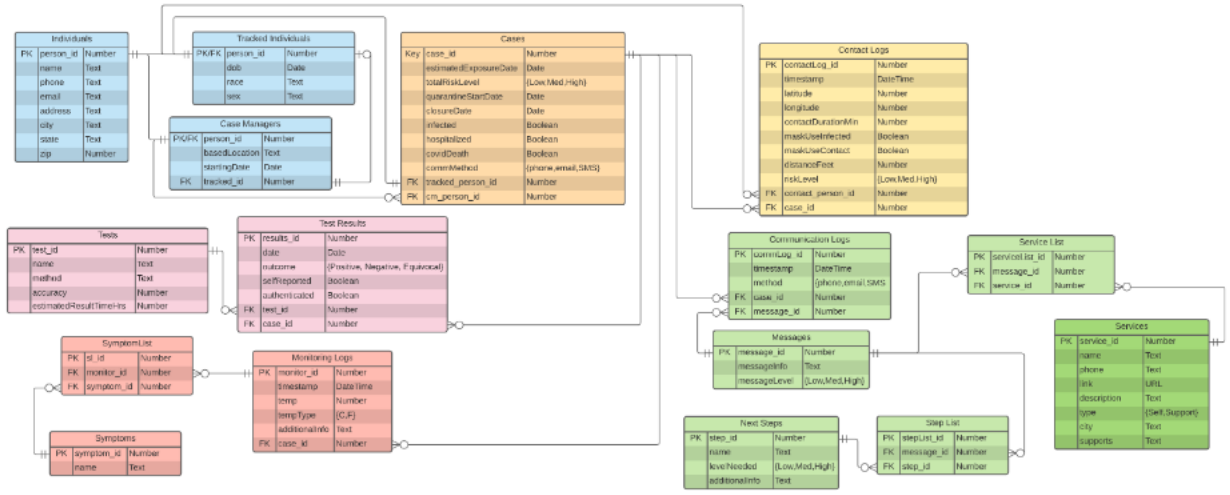
https://app.lucidchart.com/invitations/accept/5602312e-dfc3-4423-975c-47190ce6022e



Logical Model: ERD

View the logical model in Lucid Chart here:

https://app.lucidchart.com/invitations/accept/7b497cbf-268d-4a03-b1a7-822b5a844fea



Schema

View the schema in Google Docs here:

https://docs.google.com/document/d/1o8pk51aed3BJSaBcwO2EMbT8I3ru_wpGTcTN4W-DbIM/edit?usp=sharing

Individuals(person_id: Number, name: Text, phone: Text, email: Text, address: Text, city: Text, state: Text, zip: Number)

TrackedIndividuals(*person_id: Number*, dob: Date, race: Text, sex: Text)

CaseManagers(*person_id: Number*, basedCity: Text, basedState: Text, startingDate: Date, *infected_id: Number*)

Tests(test_id: Number, name: Text, method: Text, accuracy: Number, estimatedResultTimeHrs: Number)

TestResults(results_id: Number, date: Date, outcome: {Positive, Negative, Equivocal}, selfReported: Boolean, authenticated: Boolean, *test_id: Number*, *case_id: Number*)

MonitoringLogs(monitor_id: Number, timestamp: DateTime, temp: Number, tempType: {C, F}, additionalInfo: Text, *case_id: Number*)

SymptomList(sl_id: Number, *monitor_id: Number, symptom_id: Number*)

Symptoms(symptom_id: Number, name: Text)

Cases(case_id: Number, estimatedExposureDate: Date, totalRiskLevel: {Low, Med, High}, quarantineStartDate: Date, closureDate: Date, infected: Boolean, hospitalized: Boolean, covidDeath: Boolean, commMethod: {phone, email, SMS}, *cm_person_id: Number, infected_person_id: Number*)

ContactLogs(contactLog_id: Number, timestamp: DateTime, latitude: Number, longitude: Number, contactDurationMin: Number, maskUseInfected: Boolean, maskUseContact: Boolean, distanceFeet: Number, riskLevel: {Low, Med, High}, *contact_person_id: Number, case_id: Number*)

CommunicationLogs(commLog_id: Number, timestamp: DateTime, method: {phone, email, SMS}, *case_id: Number, message_id: Number*)

Messages(message_id: Number, messageLevel: {Low, Med, High}, messageInfo: Text)

StepList(stepList_id: Number, *message_id: Number, step_id: Number*)

NextSteps(step_id: Number, name: Text, levelNeeded: {Low, Med, High}, additionalInfo: Text)

ServiceList(serviceList_id: Number, *message_id: Number, service_id: Number*)

Services(service_id: Number, name: Text, phone: Text, link: Text, description: Text, type: {Self, Support}, city: Text, supports: Text)

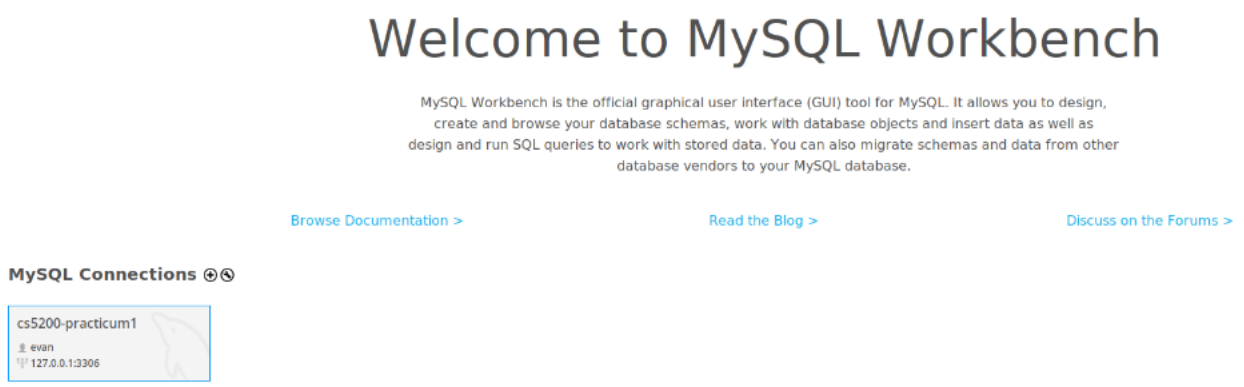
Creating Database Tables

Should you wish to inspect the scripts that create the database and populate data, they can be found at:

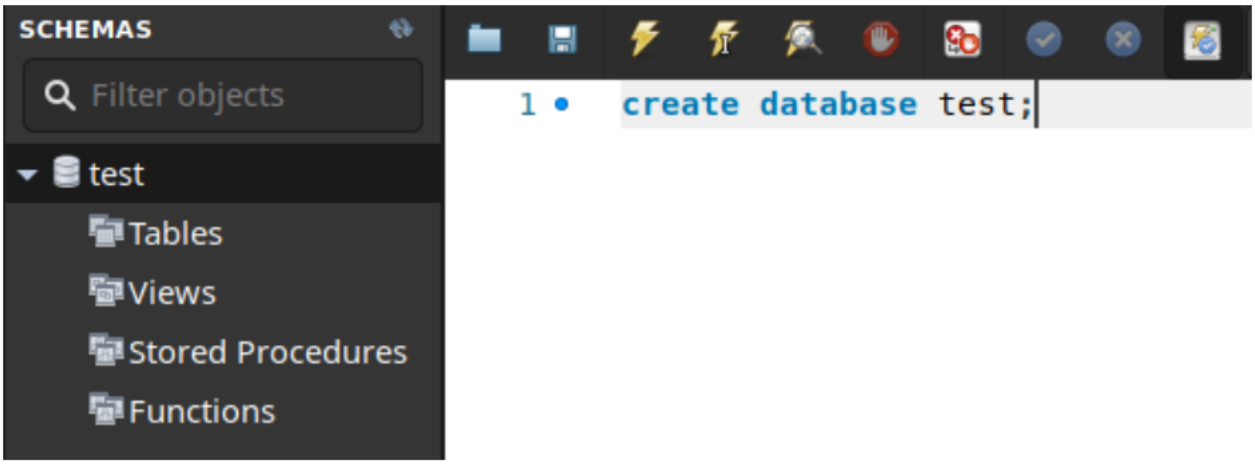
https://github.com/eldss-classwork/databases-practicum1-scripts

The following images will show a progression from an empty database through table creation in MySQL Workbench.

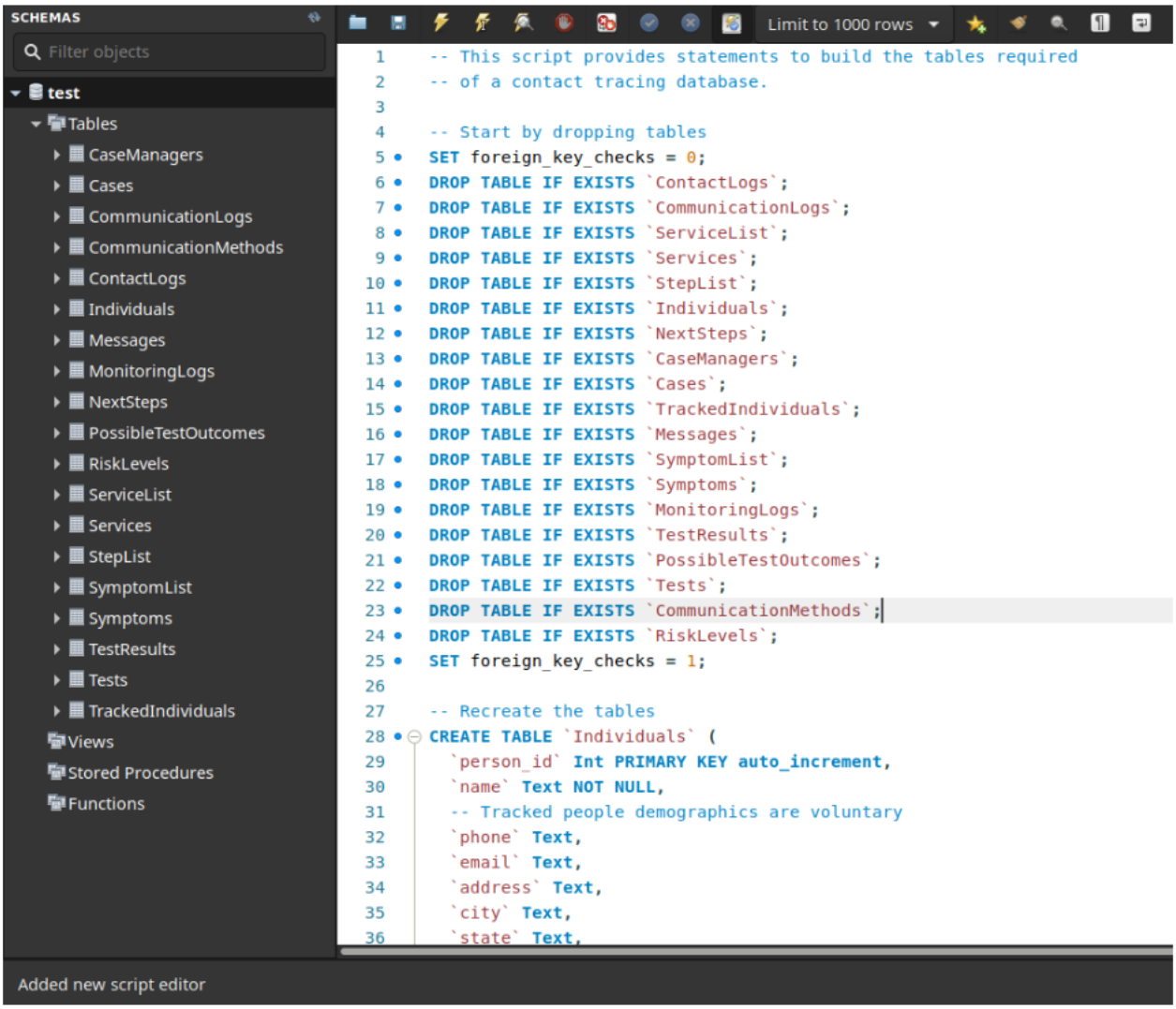
The MySQL Workbench start screen.



The newly created, empty `test` database.



The `test` database after table creation.



The following photos provide a detailed look at the schema of each table as it was created in MySQL.

```
1    DESCRIBE CaseManagers;
```

Filter Rows:

#	Field	Type	Null	Key	Default	Extra
1	person_id	int(11)	NO	PRI	NULL	
2	basedCity	text	NO		NULL	
3	basedState	text	NO		NULL	
4	startingDate	date	NO		NULL	
5	infected_id	int(11)	YES	UNI	NULL	

```
1    DESCRIBE Cases;
```

Filter Rows: Export: Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	estimatedExposureDate	date	NO		NULL	
2	quarantineStartDate	date	YES		NULL	
3	closureDate	date	YES		NULL	
4	case_id	int(11)	NO	PRI	NULL	auto_increment
5	totalRiskLevel_id	int(11)	NO	MUL	NULL	
6	commMethod_id	int(11)	NO	MUL	NULL	
7	cm_person_id	int(11)	NO	MUL	NULL	
8	tracked_person_id	int(11)	NO	UNI	NULL	
9	infected	tinyint(1)	YES		0	
10	hospitalized	tinyint(1)	YES		0	
11	covidDeath	tinyint(1)	YES		0	

```
1  DESCRIBE CommunicationLogs;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	commLog_id	int(11)	NO	PRI	NULL	auto_increment
2	timestamp	datetime	NO		NULL	
3	method_id	int(11)	NO	MUL	NULL	
4	case_id	int(11)	NO	MUL	NULL	
5	message_id	int(11)	NO	MUL	NULL	

```
1  DESCRIBE CommunicationMethods;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	commMethod_id	int(11)	NO	PRI	NULL	auto_increment
2	commMethod	text	NO	UNI	NULL	

```
1  DESCRIBE ContactLogs;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	contactLog_id	int(11)	NO	PRI	NULL	auto_increment
2	timestamp	datetime	NO		NULL	
3	latitude	decimal…	NO		NULL	
4	longitude	decimal…	NO		NULL	
5	contactDurationMin	int(11)	NO		NULL	
6	maskUseInfected	tinyint(1)	NO		NULL	
7	maskUseContact	tinyint(1)	NO		NULL	
8	distanceFeet	int(11)	NO		NULL	
9	risk_id	int(11)	NO	MUL	NULL	
10	contact_person_id	int(11)	NO	MUL	NULL	
11	case_id	int(11)	NO	MUL	NULL	

```
1    DESCRIBE Individuals;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	person_id	int(11)	NO	PRI	NULL	auto_increment
2	name	text	NO		NULL	
3	phone	text	YES		NULL	
4	email	text	YES		NULL	
5	address	text	YES		NULL	
6	city	text	YES		NULL	
7	state	text	YES		NULL	
8	zip	int(11)	YES		NULL	

```
1    DESCRIBE Messages;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	message_id	int(11)	NO	PRI	NULL	auto_increment
2	messageLevel_id	int(11)	NO	MUL	NULL	
3	messageInfo	text	NO		NULL	

```
1    DESCRIBE MonitoringLogs;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	monitor_id	int(11)	NO	PRI	NULL	auto_increment
2	timestamp	datetime	NO		NULL	
3	temp	decimal…	NO		NULL	
4	tempType	text	NO		NULL	
5	additionalInfo	text	YES		NULL	
6	case_id	int(11)	NO	MUL	NULL	

```
1  DESCRIBE NextSteps;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	step_id	int(11)	NO	PRI	NULL	auto_increment
2	name	text	NO		NULL	
3	levelNeeded_id	int(11)	NO	MUL	NULL	
4	additionalInfo	text	NO		NULL	

```
1  DESCRIBE PossibleTestOutcomes;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	outcome_id	int(11)	NO	PRI	NULL	auto_increment
2	outcome	text	NO		NULL	

```
1  DESCRIBE RiskLevels;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	risk_id	int(11)	NO	PRI	NULL	auto_increment
2	risk	text	NO	UNI	NULL	

```
1  DESCRIBE ServiceList;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	serviceList_id	int(11)	NO	PRI	NULL	auto_increment
2	message_id	int(11)	NO	MUL	NULL	
3	service_id	int(11)	NO	MUL	NULL	

```
1    DESCRIBE Services;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	service_id	int(11)	NO	PRI	NULL	auto_increment
2	name	text	NO		NULL	
3	phone	text	NO		NULL	
4	link	text	NO		NULL	
5	description	text	NO		NULL	
6	type	text	NO		NULL	
7	city	text	NO		NULL	
8	supports	text	NO		NULL	

```
1    DESCRIBE StepList;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	stepList_id	int(11)	NO	PRI	NULL	auto_increment
2	message_id	int(11)	NO	MUL	NULL	
3	step_id	int(11)	NO	MUL	NULL	

```
1    DESCRIBE SymptomList;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	sl_id	int(11)	NO	PRI	NULL	auto_increment
2	monitor_id	int(11)	NO	MUL	NULL	
3	symptom_id	int(11)	NO	MUL	NULL	

```
1    DESCRIBE Symptoms;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	symptom_id	int(11)	NO	PRI	NULL	auto_increment
2	name	text	NO		NULL	

```
1  DESCRIBE TestResults;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	results_id	int(11)	NO	PRI	NULL	auto_increment
2	date	date	NO		NULL	
3	outcome_id	int(11)	NO	MUL	NULL	
4	selfReported	tinyint(1)	NO		NULL	
5	authenticated	tinyint(1)	NO		0	
6	test_id	int(11)	NO	MUL	NULL	
7	case_id	int(11)	NO	MUL	NULL	

```
1  DESCRIBE Tests;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	test_id	int(11)	NO	PRI	NULL	auto_increment
2	name	text	NO	UNI	NULL	
3	method	text	NO		NULL	
4	accuracy	decimal…	NO		NULL	
5	estimatedResultTimeHrs	int(11)	NO		NULL	

```
1  DESCRIBE TrackedIndividuals;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	Field	Type	Null	Key	Default	Extra
1	person_id	int(11)	NO	PRI	NULL	
2	dob	date	YES		NULL	
3	race	text	YES		NULL	
4	sex	text	YES		NULL	

Populating The Database

The script for populating data into our database can be found at the following link:

https://github.com/eldss-classwork/databases-practicum1-scripts/blob/master/populate.sql

Demonstrations that the data was loaded correctly can be found in the queries below. That is, they return the data that was loaded.

Queries

The script for running all of the queries can be found at the following link:

https://github.com/eldss-classwork/databases-practicum1-scripts/blob/master/queries.sql

The following is the screenshots of the query results.

Query 1

```
1    -- How many communications have been sent to people in each exposure level group?
2 •  SELECT r.risk, COUNT(cl.`timestamp`) AS numCommunications
3    FROM CommunicationLogs AS cl, Cases AS c, RiskLevels AS r
4    WHERE cl.case_id=c.case_id AND c.totalRiskLevel_id=r.risk_id
5    GROUP BY r.risk
6    ORDER BY r.risk_id;
```

Filter Rows: Export: Wrap Cell Content:

#	risk	numCommunications
1	Low	39
2	Medium	32
3	High	29

Query 2

```
1    -- List the number of COVID deaths by state, in descending order of deaths.
2 •  SELECT i.state, COUNT(*) AS deaths
3    FROM Cases AS c, Individuals AS i
4    WHERE c.tracked_person_id=i.person_id AND c.covidDeath=1
5    GROUP BY i.state
6    ORDER BY deaths DESC;
```

Filter Rows: Export: Wrap Cell Content:

#	state	deaths
1	Imo	3
2	São Paulo	1
3	Gye	1

Query 3

```
1   -- List the individual name, type of test, test method, test result,
2   -- and date of test for all recorded tests; order by date taken.
3 • SELECT i.`name`, t.`name`, t.method, pto.outcome, tr.`date`
4   FROM Individuals AS i, Tests AS t, TestResults AS tr, Cases AS c, PossibleTestOutcomes AS pto
5   WHERE
6       tr.test_id=t.test_id
7       AND tr.case_id=c.case_id
8       AND tr.outcome_id=pto.outcome_id
9       AND c.tracked_person_id=i.person_id
10  ORDER BY tr.`date`;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	name	name	method	outcome	date
1	Gareth Flores	Experimental Test 4	Nasal Swab	Positive	2020-02-05
2	Gareth Flores	Experimental Test 3	Nasal Swab	Negative	2020-02-28
3	Bradley Rocha	Experimental Test 3	Nasal Swab	Negative	2020-03-15
4	Ciaran Jacobs	Experimental Test 3	Nasal Swab	Positive	2020-03-22
5	Bradley Rocha	Experimental Test 2	Cheek Swab	Positive	2020-03-24
6	Bradley Rocha	Experimental Test 3	Nasal Swab	Negative	2020-03-28
7	Macy Wise	Experimental Test 1	Nasal Swab	Positive	2020-04-04
8	Knox Henderson	Experimental Test 3	Nasal Swab	Negative	2020-04-04
9	Macy Wise	Experimental Test 4	Nasal Swab	Negative	2020-05-03
10	Elmo Burns	Experimental Test 2	Cheek Swab	Positive	2020-05-05
11	Knox Henderson	Experimental Test 1	Nasal Swab	Negative	2020-05-13
12	Elmo Burns	Experimental Test 3	Nasal Swab	Negative	2020-06-09
13	Bertha Acosta	Experimental Test 2	Cheek Swab	Negative	2020-06-30
14	Bertha Acosta	Experimental Test 1	Nasal Swab	Negative	2020-07-09
15	Murphy Leon	Experimental Test 2	Cheek Swab	Positive	2020-07-15
16	Lydia Ramirez	Experimental Test 4	Nasal Swab	Positive	2020-08-06
17	Carl Perry	Experimental Test 3	Nasal Swab	Positive	2020-08-18
18	Lydia Ramirez	Experimental Test 2	Cheek Swab	Negative	2020-09-17
19	Carl Perry	Experimental Test 1	Nasal Swab	Positive	2020-09-18
20	Carl Perry	Experimental Test 2	Cheek Swab	Positive	2020-09-29

Query 4

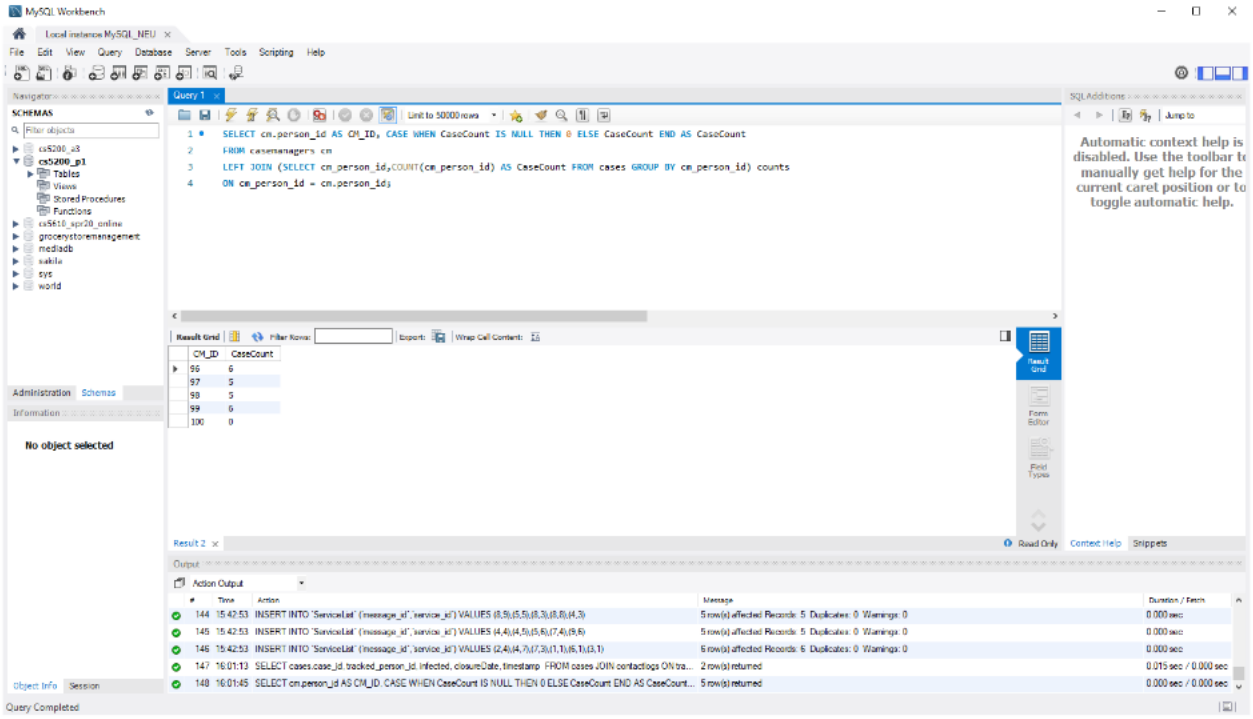
```
1   -- Get the name, phone, and email of individuals who did not provide
2   -- daily check-ins while in quarantine (14 days).
3 • SELECT i.`name`, i.phone, i.email, COUNT(*) AS `messages sent`
4   FROM Individuals AS i, Cases AS c, MonitoringLogs AS ml
5   WHERE
6       c.tracked_person_id=i.person_id
7       AND ml.case_id=c.case_id
8       AND c.closureDate IS NOT NULL
9   GROUP BY i.`name`
10  HAVING `messages sent` < 14;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	name	phone	email	messages sent
1	Bertha Acosta	(640) 327-1743	rutrum@sitamet.org	11
2	Carl Perry	(426) 177-3217	orci.lacus.vestibulum@euismodma…	8
3	Ciaran Jacobs	(960) 864-9603	Ut.tincidunt.orci@vel.co.uk	10
4	Elmo Burns	(521) 702-4435	Donec.at@in.edu	11
5	Gareth Flores	(140) 387-5572	facilisi.Sed@Quisque.org	9
6	Knox Henderson	(377) 684-0047	sodales.elit.erat@aliquetdiamSed.ca	6
7	Lydia Ramirez	(309) 418-8256	taciti.sociosqu.ad@odioPhasellusat…	9
8	Macy Wise	(878) 442-1017	dolor.dolor@Nullainterdum.net	13
9	Murphy Leon	(327) 972-7628	vitae.purus.gravida@feugiatnec.com	8

Query 5

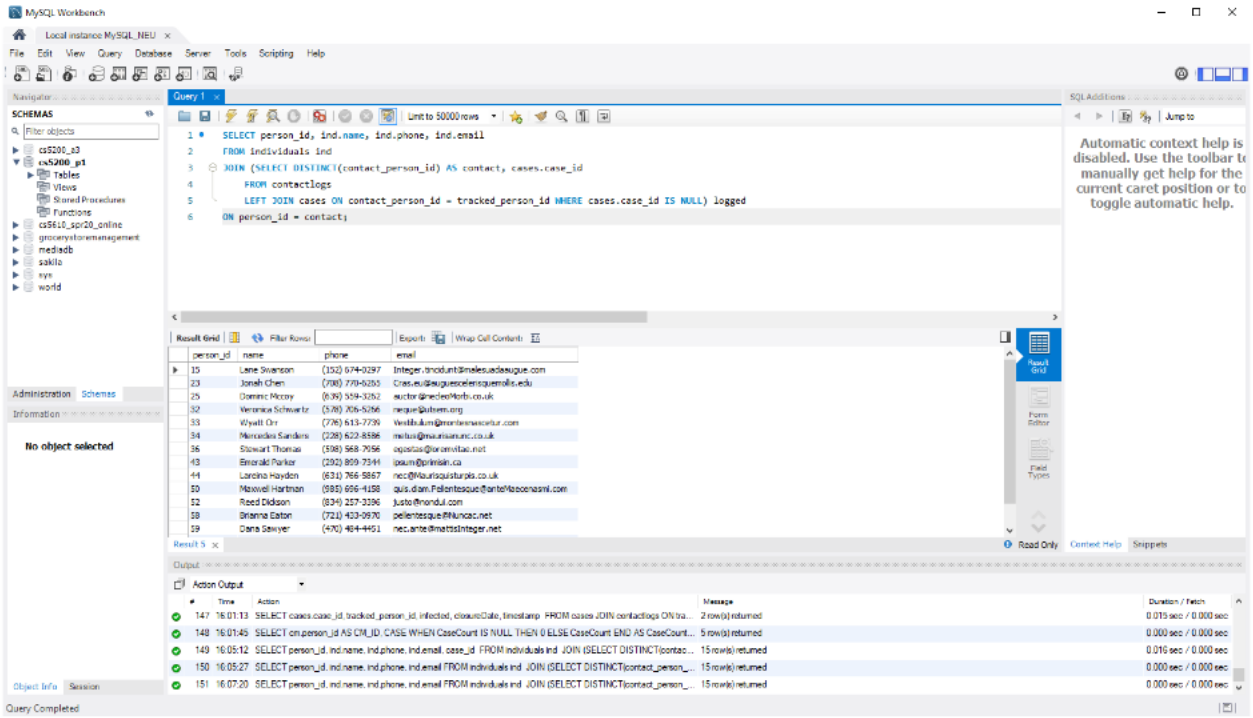
– Get the id and number of cases assigned to each case manager.



Query 6

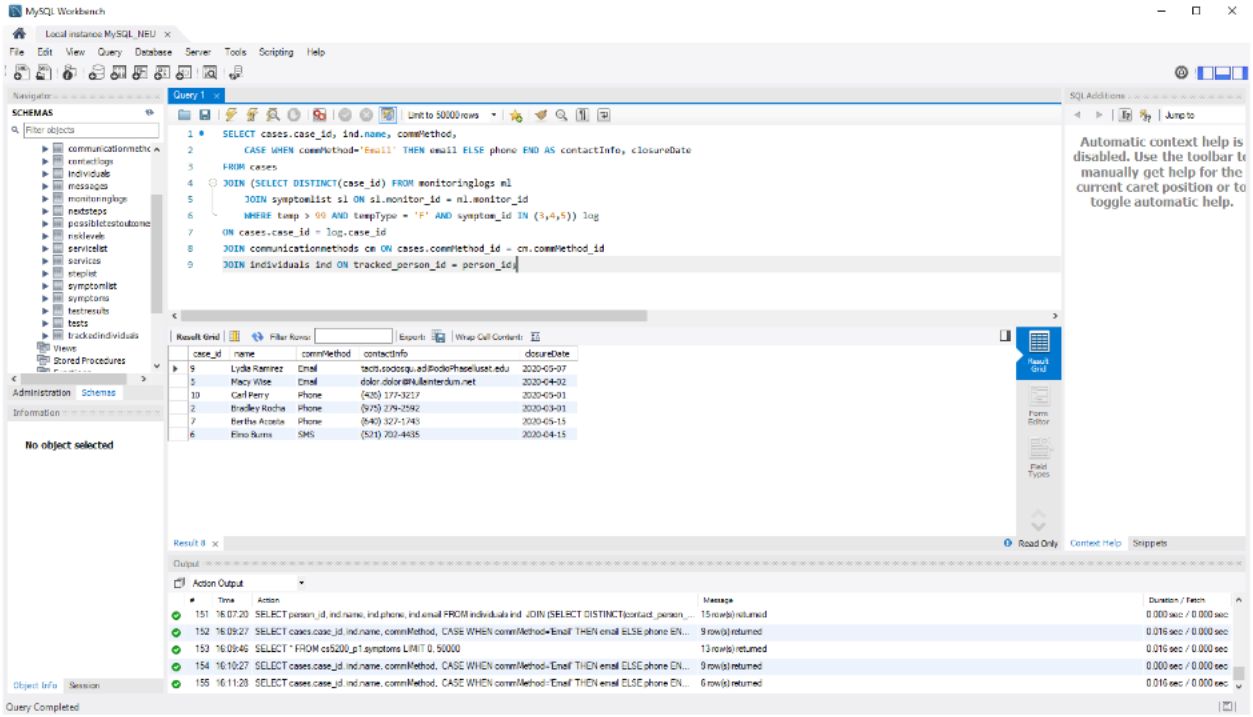
– Get the id, name, phone, and email of individuals who have been

– in contact with a tracked individual but do not have a case yet.



Query 7

– Get the case, name, preferred communication method, contact information,
– and case closure date for tracked individuals who have a temperature over
– 99 degrees F and are experiencing a cough or breathing problems.



Query 8

– Find the id of individuals that were not infected and have been re-exposed.
– Include the date the original case was closed, and the new exposure date.