

# DBMS Selection

- DBMS analysis

Choice of DBMS partially depends on the type of database we need, which depends largely on what kind of data we have and how we want to use it. A good DBMS is useful because it provides a centralized, unified view of data, giving us easy insight into where data is stored, and its current status. It also allows the use of data across applications without having to create multiple versions of the data.

---

**A good DBMS is useful because it provides a centralized, unified view of data, giving you easy insight into where data is stored, and its current status.**

DBMS is worth it, especially for large or sensitive data collections. When creating a data management software list of potential solutions, we want to consider how we will be organizing database infrastructure and how databases will be used and compare this with each type of database management system software to find the best fit.

- DBMS selection

Storing and querying massive datasets can be time consuming and expensive without the right hardware and infrastructure. BigQuery is an enterprise data warehouse that solves this problem by enabling super-fast SQL queries using the processing power of Google's infrastructure. BigQuery works great with all sizes of data, from a 100 row Excel spreadsheet to several Petabytes of data. Simply we can move data into BigQuery and it will handle the hard work. We can control access to both the project and also data based on our business needs, such as giving others the ability to view or query data.

We can access BigQuery by using the Cloud Console, by using the bq command-line tool, or by making calls to the BigQuery REST API using a variety of client libraries such as Java, .NET, or Python. There are also a variety of third-party tools that we can use to interact with BigQuery, such as visualizing the data or loading the data. BigQuery is fully-managed.

As this database design is useful for mobile applications, if the popularity of the application grows, BigQuery can handle large amounts of data easily and we can analyze data in BigQuery.

Because of the above advantages, We are using Google Cloud BigQuery as a DBMS.

## ● Hardware & Software

In BigQuery, we don't need to think about hardware and software. In BigQuery, we can just upload the data and query it. It is a truly managed service. We will be charged by storage, streaming inserts, and queries.

# Implementation & Loading

## ● DBMS Installation

We don't need to install BigQuery. To get started, We don't need to deploy any resources, such as disks or virtual machines.

BigQuery is a serverless data warehousing, Google does all resource provisioning behind the scenes, so we can focus on data and analysis rather than worrying about upgrading, securing, or managing the infrastructure.

## ● Schema creation

First created project on Google Cloud Platform

Google Cloud Platform

mohini-data225-project

DASHBOARD

ACTIVITY

RECOMMENDATIONS

Project info

Project name

mohini-data225-project

Project ID

mohini-data225-spring2021

Project number

934434978650

ADD PEOPLE TO THIS PROJECT

→

Go to project settings

Created dataset ‘CONTACT\_TRACER’ and all the required tables on BigQuery

▼

mohini-data225-spring2021

▼

CONTACT\_TRACER

ADDRESS

APP\_USER

GPS\_LOCATION

HEALTH\_TRACK\_ADMIN

LAB

NOTIFICATION

TEST

TRACKER

TRAVEL\_HISTORY

USER

Create statements:

APP\_USER

```
CREATE OR REPLACE TABLE CONTACT_TRACER.APP_USER(  
  
APP_USER_ID INT64 NOT NULL,  
  
USER_PASSWORD STRING ,  
  
FIRST_NAME STRING ,  
  
LAST_NAME STRING,
```

PHONE STRING NOT NULL

);

APP\_USER

QUERY TABLE

Schema

Details

Preview

Field name	Type	Mode	Policy tags
APP_USER_ID	INTEGER	REQUIRED	
USER_PASSWORD	STRING	NULLABLE	
FIRST_NAME	STRING	NULLABLE	
LAST_NAME	STRING	NULLABLE	
PHONE	STRING	REQUIRED	

Edit schema

USER

CREATE OR REPLACE TABLE CONTACT\_TRACER.USER(  
  
USER\_ID INT64 NOT NULL,  
  
GENDER STRING,  
  
DOB DATE NOT NULL,  
  
INFECTION\_STATUS STRING NOT NULL,  
  
ZIPCODE STRING NOT NULL,  
  
LIVING\_STATUS STRING,  
  
IMMUNITY\_COMPROMISED BOOLEAN NOT NULL,  
  
)

USER

QUERY TABLE

Schema

Details

Preview

Field name	Type	Mode	Policy tags
USER_ID	INTEGER	REQUIRED	
GENDER	STRING	NULLABLE	
DOB	DATE	REQUIRED	
INFECTION_STATUS	STRING	REQUIRED	
ZIPCODE	STRING	REQUIRED	
LIVING_STATUS	STRING	NULLABLE	
IMMUNITY_COMPR OMISED	BOOLEAN	REQUIRED	

Edit schema

HEALTH\_TRACK\_ADMIN

```
CREATE OR REPLACE TABLE CONTACT_TRACER.HEALTH_TRACK_ADMIN(  
  
ADMIN_ID INT64 NOT NULL,  
  
REG_LAT STRING NOT NULL,  
  
REG_LONG STRING NOT NULL,  
  
QUALIFICATION STRING  
)
```

HEALTH_TRACK_ADMIN			
Schema   Details   Preview			
Field name	Type	Mode	Pol
ADMIN_ID	INTEGER	REQUIRED	
REG_LAT	STRING	REQUIRED	
REG_LONG	STRING	REQUIRED	
QUALIFICATION	STRING	NULLABLE	
Edit schema			

NOTIFICATION

```
CREATE OR REPLACE TABLE CONTACT_TRACER.NOTIFICATION(  
  
NOTIFICATION_ID INT64 NOT NULL,
```

```
INFECTION_RISK STRING,  
  
APP_USER_ID INT64 NOT NULL,  
  
);
```

NOTIFICATION

QUERY

Schema

Details

Preview


Field name	Type	Mode	Pol
NOTIFICATION_ID	INTEGER	REQUIRED	
INFECTION_RISK	STRING	NULLABLE	
APP_USER_ID	INTEGER	NULLABLE	

Edit schema

TEST

```
CREATE OR REPLACE TABLE CONTACT_TRACER.TEST(  
  
TEST_ID INT64 NOT NULL,  
  
TEST_DATE DATE NOT NULL,  
  
TEST_TYPE STRING,  
  
USER_ID INT64 NOT NULL,  
  
LAB_ID INT64 NOT NULL,  
  
RESULT BOOLEAN NOT NULL,  
  
);
```

TEST

 QUERY TABLE

Schema

Details

Preview


Field name	Type	Mode	Policy type
TEST_ID	INTEGER	REQUIRED	
TEST_DATE	DATE	REQUIRED	
TEST_TYPE	STRING	NULLABLE	
USER_ID	INTEGER	REQUIRED	
LAB_ID	INTEGER	REQUIRED	
RESULT	BOOLEAN	REQUIRED	

Edit schema

LAB

```
CREATE OR REPLACE TABLE CONTACT_TRACER.LAB(  
  
  LAB_ID INT64 NOT NULL,  
  
  LAB_NAME STRING,  
  
  ZIPCODE STRING NOT NULL  
  
)
```

LAB

 QUERY TABLE

Schema

Details

Preview

Field name	Type	Mode	Policy type
LAB_ID	INTEGER	REQUIRED	
LAB_NAME	STRING	NULLABLE	
ZIPCODE	STRING	REQUIRED	

Edit schema

ADDRESS

```
CREATE OR REPLACE TABLE CONTACT_TRACER.ADDRESS(  
  
  
  
  ZIPCODE string NOT NULL,  
  
)
```

```
CITY STRING ,  
  
STATE STRING  
  
)
```

ADDRESS

QUERY

Schema

Details

Preview

Field name	Type	Mode	Policy
ZIPCODE	STRING	REQUIRED	
CITY	STRING	NULLABLE	
STATE	STRING	NULLABLE	

Edit schema

TRAVEL\_HISTORY

```
CREATE OR REPLACE TABLE CONTACT_TRACER.TRAVEL_HISTORY(  
  
COUNTRY STRING NOT NULL,  
  
USER_ID INT64 NOT NULL,  
  
);
```

TRAVEL\_HISTORY

QUERY

Schema

Details

Preview

Field name	Type	Mode	Policy
COUNTRY	STRING	REQUIRED	
USER_ID	INTEGER	REQUIRED	

Edit schema



TRACKER

```
CREATE OR REPLACE TABLE CONTACT_TRACER.TRACKER(  
  
  TRACKER_ID INT64 NOT NULL,  
  
  GPS_LOC INT64 NOT NULL,  
  
  USER_ID INT64 NOT NULL  
  
)
```

TRACKER

QUERY T

Schema

Details

Preview

Field name	Type	Mode	Policy
TRACKER_ID	INTEGER	REQUIRED	
GPS_LOC	INTEGER	REQUIRED	
USER_ID	INTEGER	REQUIRED	

Edit schema

GPS\_LOCATION

```
CREATE OR REPLACE TABLE CONTACT_TRACER.GPS_LOCATION(  
  
  GPS_LOC INT64 NOT NULL,  
  
  GPS_LAT FLOAT64 ,  
  
  GPS_LONG FLOAT64,  
  
  DATE DATETIME  
  
);
```

GPS\_LOCATION

QUERY TAI

Schema

Details

Preview

Field name	Type	Mode	Policy
GPS_LOC	INTEGER	REQUIRED	
GPS_LAT	STRING	NULLABLE	
GPS_LONG	STRING	NULLABLE	
DATE	DATETIME	NULLABLE	

Edit schema

● Data loading and conversion

Dummy data generated using <http://www.generatedata.com/>

Insert statements:

APP\_USER

```
INSERT INTO CONTACT_TRACER.APP_USER VALUES
(1, 'A0Y71IUZ3FQ', 'Charde', 'Morse', '16460329 5595'),
(2, 'THN65TJS3UA', 'Kendall', 'Freeman', '16110422 9362'),
(3, 'XP009FJL5JT', 'McKenzie', 'Hatfield', '16640323 7123'),
(4, 'PDP15LCN3CI', 'Dai', 'Buckner', '16280115 4994'),
(5, 'BQK68CQM9KL', 'Alexis', 'Gutierrez', '16631008 8429'),
(6, 'AIN56ZAU5LX', 'Colleen', 'Curtis', '16190524 4537'),
(7, 'YVP86ZPK8FB', 'Lynn', 'Merrill', '16830230 7726'),
(8, 'AFG53YFI6X0', 'Calista', 'Wade', '16740121 4080'),
(9, 'ZHF63ENF3AT', 'Melyssa', 'Gould', '16991223 0514'),
(10, 'GDM75COS3FP', 'Rashad', 'Jackson', '16820920 2533'),
(11, 'GHG63ZTD8TT', 'Karina', 'Solomon', '16210110 2164'),
(12, 'UNM49LNT1PD', 'Allistair', 'Conway', '16090823 4982'),
(13, 'RIB54YTF9FX', 'Maile', 'Lott', '16951003 2619'),
(14, 'THU98ZQE6BC', 'Ira', 'Leon', '16240417 9414'),
(15, 'BPE25ASX7HU', 'Aurora', 'Tate', '16100606 7761'),
```

(16, 'RKM44BSU2ZL', 'Alden', 'Carson', '16311106 7819'),  
(17, 'XVW86GHW4WQ', 'Aiko', 'Barrett', '16590809 1753'),  
(18, 'JQK98NHY7MX', 'Allen', 'Duke', '16090716 9643'),  
(19, 'LPA23ICJ7TD', 'Upton', 'Bradley', '16270817 6306'),  
(20, 'AID56EJH3KX', 'Lacy', 'Sharp', '16530410 4614'),  
(21, 'TYV13NJD2DU', 'Chastity', 'Bruce', '16251124 4879'),  
(22, 'PKF82IIF2FR', 'Kessie', 'Pena', '16971102 0165'),  
(23, 'WZC92ARE6XB', 'Christen', 'Bowen', '16951223 2720'),  
(24, 'RLF36HEK2DA', 'Wyatt', 'Lancaster', '16660316 5272'),  
(25, 'YGM84VJJ3GQ', 'Gabriel', 'Watson', '16040301 7072'),  
(26, 'KWW32WNB5EC', 'Judah', 'Sims', '16480123 7670'),  
(27, 'CTB61GNY9RC', 'Keith', 'Hess', '16380314 8463'),  
(28, 'SBJ29IHE80H', 'Inez', 'Aguirre', '16080530 3989'),  
(29, 'QON00QIF7YE', 'Fitzgerald', 'Browning', '16880720 2992'),  
(30, 'PCD92BCH3DQ', 'Merrill', 'Flynn', '16290127 3082'),  
(31, 'UYF910IY2SZ', 'Ainsley', 'Green', '16201017 8842'),  
(32, 'STV83FMK1KL', 'Ariel', 'Wright', '16061117 6850'),  
(33, 'SLP27AQG3DE', 'Alan', 'Frost', '16460408 2364'),  
(34, 'OWU22JAH6YE', 'Ulla', 'Kirkland', '16350225 1253'),  
(35, 'NRQ70UDO1MW', 'Regan', 'Luna', '16810813 9513'),  
(36, 'QJ037VME4XK', 'Raphael', 'Watts', '16990829 1660'),  
(37, 'PYH46RXV2CE', 'Kylan', 'Higgins', '16630430 3750'),  
(38, 'VJ042FFW3JK', 'Alan', 'Townsend', '16390118 4139'),  
(39, 'VWM63DXU5MW', 'Rosalyn', 'Torres', '16470317 1878'),  
(40, 'EME21RSC8TK', 'Zoe', 'Combs', '16290724 4293'),  
(41, 'JZV06KJP0FS', 'Julie', 'Cooke', '16681101 7984'),  
(42, 'UGJ34H0G1DG', 'Kieran', 'Walker', '16560608 3607'),  
(43, 'HXC78ECN1BG', 'Cheyenne', 'Weeks', '16921024 9695'),  
(44, 'IYC01JZQ5NL', 'Isadora', 'Beck', '16651023 7347'),  
(45, 'XFP65GCI7RE', 'Trevor', 'Pollard', '16650722 5065'),  
(46, 'DIT96ZNI6IP', 'Hadley', 'Norman', '16371129 7238'),  
(47, 'OXD61WML6PY', 'Zenias', 'Lowery', '16261005 7214'),

(48, 'BD002VAL1UW', 'Sybill', 'Morrison', '16220120 5248'),  
(49, 'OYM90UQI8TK', 'Cailin', 'Ballard', '16690401 6091'),  
(50, 'LSX16IFD6SY', 'Andrew', 'Pitts', '16140915 7409'),  
(51, 'RJG98TIIry', 'Varshitha', 'Dan', '16150828 0277'),  
(52, 'RJG9TIIry', 'Srihitha', 'Brown', '11230828 0277'),  
(53, 'RJG9TIIrwq', 'Dan', 'Brown', '11230821 0277'),  
(54, 'juJG9TIIry', 'Jef', 'Archer', '11230828 0282'),  
(55, 'bsJG9iIsfry', 'Dean', 'Koontz', '11230828 9873'),  
(56, 'jodkwual', 'gauhilk', 'Morse', '16460329 5562'),  
(57, 'afhjssf', 'likith', 'raj', '1623329 5562'),  
(58, 'oggspwqs', 'mahesh', 'manal', '2340329 5562'),  
(59, 'wypldthv', 'abnavk', 'kala', '1860329 5421'),  
(60, 'juosnUE', 'arnayaya', 'kapeed', '1644679 9843'),  
(61, 'AFG13RHSs', 'partheek', 'maha', '1665329 5535'),  
(71, "FCU82YIF5JC", "Zeph", "Hendricks", "(944) 750-3324"),  
(130, "JTL47JLQ30T", "Harding", "Lawson", "(984) 845-2452"),  
(110, "CJT33KKL2QA", "Thor", "Cannon", "(664) 708-3444"),  
(144, "DSZ70DQY0UX", "Blaze", "Wyatt", "(675) 502-2644"),  
(71, "MBW07DKG9PP", "Aristotle", "Reese", "(636) 494-2067"),  
(63, "FXN35YXQ8B0", "Erasmus", "Poole", "(335) 236-5361"),  
(134, "AHT74PFQ1RV", "Nathan", "Mann", "(372) 215-9035"),  
(84, "VRH84JBL0DT", "Henry", "Sharpe", "(298) 703-5487"),  
(161, "XTR33LAE2DJ", "Price", "Cleveland", "(531) 260-6978"),  
(94, "NVZ53JXV8GS", "Hector", "Acevedo", "(699) 302-9944"),  
(145, "AKY63IPG2BB", "Zeph", "Haynes", "(927) 172-6476"),

APP_USER						<a href="#">QUERY TABLE</a>	
Schema   Details <u>Preview</u>							
Row	APP_USER_ID	USER_PASSWORD	FIRST_NAME	LAST_NAME	PHONE		
1	81	YNA70LWC6DT	Ali	Fisher	(921) 367-1080		
2	439	GIE09YZT0OK	Ali	Reeves	(673) 521-0665		
3	696	ZSS31SFZ1JL	Gil	Avila	(635) 149-3885		
4	265	RKO56SKV3CN	Guy	Moss	(346) 485-6391		
5	165	UBY31YZS2LN	Guy	Cochran	(167) 103-2242		
6	128	ZBZ14POA2HQ	Guy	Barton	(610) 952-8809		
7	528	GZX56ZSH4UY	Guy	Baird	(474) 504-1076		
8	905	GDW21ENP9XD	Guy	Henson	(441) 441-8533		
9	322	JLU99ZWL0LD	Hop	Berry	(671) 307-1867		
10	630	IIB79AHN9HC	Hop	Gilmore	(240) 216-5099		
11	456	QFE11WFA0KS	Ian	Lane	(567) 343-3547		
12	557	RVU36GKP1YB	Ian	Wyatt	(656) 335-6342		
13	862	GOU07HTQ8DP	Ian	Small	(392) 710-3971		

USER

```
INSERT INTO CONTACT_TRACER.USER(USER_ID,GENDER,DOB,IMMUNITY_COMPROMISED,LIVING_STATUS,INFECTION_STATUS,ZIPCODE) VALUES

(1,"Male","1956-10-09",TRUE,"group","infected",'46833'),

(2,"Male","1972-01-13",TRUE,"isolated","suspected",'46833'),

(3,"Male","1971-06-13",TRUE,"group","not-infected", '46833'),

(4,"Male","1993-05-19",TRUE,"isolated","not-infected",'72539'),

(5,"Female","1905-06-06",FALSE,"isolated","not-infected",'99829'),

(6,"Male","1989-07-25",TRUE,"group","not-infected",'75508'),

(7,"Male","1965-01-30",TRUE,"isolated","suspected",'46833'),

(8,"Male","1951-09-26",FALSE,"group","not-infected",'75508'),

(9,"Female","1912-09-18",TRUE,"isolated","not-infected",'75508'),

(10,"Male","1987-02-14",TRUE,"group","not-infected",'72580'),

(11,"Female","1995-04-18",TRUE,"group","infected",'46833'),

(12,"Male","1961-12-31",FALSE,"group","not-infected",'15278'),

(13,"Male","1974-04-30",FALSE,"isolated","not-infected",'15278'),

(14,"Male","1986-10-06",TRUE,"isolated","suspected",'15278'),

(15,"Male","1998-04-09",TRUE,"isolated","infected",'46833'),

(16,"Male","1900-12-26",FALSE,"group","infected",'15278'),

(17,"Female","1906-12-30",FALSE,"group","not-infected",'15278'),
```

(18, "Female", "1959-04-19", FALSE, "group", "not-infected", '15278'),  
(19, "Female", "1988-10-07", TRUE, "isolated", "not-infected", '15278'),  
(20, "Male", "1980-08-24", FALSE, "group", "infected", '46833'),  
(21, "Male", "1972-07-31", FALSE, "isolated", "not-infected", '72580'),  
(22, "Female", "1951-06-24", TRUE, "isolated", "not-infected", '72580'),  
(23, "Female", "1903-09-28", FALSE, "isolated", "not-infected", '86995'),  
(24, "Male", "1952-02-22", FALSE, "isolated", "suspected", '86995'),  
(25, "Male", "1972-11-19", FALSE, "isolated", "suspected", '46833'),  
(26, "Male", "1968-12-07", TRUE, "isolated", "suspected", '86995'),  
(27, "Female", "1983-10-01", FALSE, "group", "suspected", '86995'),  
(28, "Male", "1913-08-13", FALSE, "group", "not-infected", '31365'),  
(29, "Female", "1970-09-20", FALSE, "isolated", "infected", '31365'),  
(30, "Female", "1994-02-02", FALSE, "group", "not-infected", '31365'),  
(31, "Male", "1966-10-07", FALSE, "isolated", "not-infected", '31365'),  
(32, "Male", "1968-06-30", TRUE, "group", "infected", '42042'),  
(33, "Female", "1953-02-13", TRUE, "isolated", "not-infected", '28270'),  
(34, "Male", "1957-06-11", TRUE, "group", "not-infected", '28270'),  
(35, "Female", "1953-10-13", TRUE, "group", "suspected", '28270'),  
(36, "Female", "1960-05-04", TRUE, "isolated", "not-infected", '21421'),  
(37, "Female", "1901-08-14", FALSE, "isolated", "suspected", '46833'),  
(38, "Male", "1989-10-13", FALSE, "group", "not-infected", '21421'),  
(39, "Female", "1971-06-02", FALSE, "group", "not-infected", '75508'),  
(40, "Female", "1973-12-04", TRUE, "group", "not-infected", '75508'),  
(41, "Female", "1986-09-19", FALSE, "isolated", "not-infected", '75508'),  
(42, "Male", "1950-10-19", FALSE, "group", "infected", '31365'),  
(43, "Female", "1988-03-30", TRUE, "group", "infected", '46833'),  
(44, "Male", "1917-05-01", TRUE, "isolated", "not-infected", '31365'),  
(45, "Female", "1919-12-31", FALSE, "group", "infected", '16250'),  
(46, "Female", "1902-12-20", TRUE, "isolated", "suspected", '16250'),  
(47, "Female", "1980-07-07", TRUE, "isolated", "infected", '46833'),  
(48, "Female", "1975-08-06", TRUE, "isolated", "infected", '21421'),  
(49, "Male", "1908-01-27", FALSE, "group", "suspected", '18164'),

```
(50, "Female", "1986-06-16", TRUE, "isolated", "infected", '46833'),

(51, "Female", "1977-07-26", FALSE, "group", "infected", '60704'),

(52, "Male", "1964-08-21", FALSE, "group", "suspected", '60704'),

(53, "Female", "1986-09-11", TRUE, "isolated", "not-infected", '60704'),

(54, "Male", "1916-01-13", FALSE, "isolated", "suspected", '68241'),

(55, "Male", "1916-08-28", TRUE, "isolated", "not-infected", '68241'),

(56, "Male", "1956-10-09", TRUE, "group", "infected", '46833'),

(57, "Male", "1972-01-13", TRUE, "group", "infected", '46833'),

(58, "Female", "1971-06-13", TRUE, "group", "infected", '46833'),

(59, "Male", "1993-05-19", TRUE, "group", "infected", '72539'),

(60, "Female", "1905-06-06", TRUE, "group", "infected", '99829'),

(61, "Male", "1989-07-25", TRUE, "group", "infected", '75508');
```

USER

[QUERY TABLE](#)[SHARE TABLE](#)[COPY](#)

Schema		Details		Preview				
Row	USER_ID	GENDER	DOB	INFECTION_STATUS	ZIPCODE	LIVING_STATUS	IMMUNITY_COMPROMISED	
1	51	Female	1977-07-26	infected	60704	group	false	
2	45	Female	1919-12-31	infected	16250	group	false	
3	16	Male	1900-12-26	infected	15278	group	false	
4	42	Male	1950-10-19	infected	31365	group	false	
5	20	Male	1980-08-24	infected	46833	group	false	
6	52	Male	1964-08-21	suspected	60704	group	false	
7	49	Male	1908-01-27	suspected	18164	group	false	
8	27	Female	1983-10-01	suspected	86995	group	false	
9	12	Male	1961-12-31	not-infected	15278	group	false	
10	30	Female	1994-02-02	not-infected	31365	group	false	
11	18	Female	1959-04-19	not-infected	15278	group	false	
12	17	Female	1906-12-30	not-infected	15278	group	false	
13	39	Female	1971-06-02	not-infected	75508	group	false	

HEALTH\_TRACK\_ADMIN

INSERT INTO CONTACT\_TRACER.HEALTH\_TRACK\_ADMIN VALUES

```
(901, "28.27877", "124.79873", "nurse"), (902, "88.28722", "-49.97966", "nurse"), (903, "-0.94115", "-113.463", "nurse"), (904, "34.89282", "-127.07249", "nurse"), (905, "-4.7214", "-165.48201", "nurse"), (906, "-27.08202", "70.30782", "nurse"), (907, "30.68094", "-153.3488", "nurse"), (908, "61.56295", "-100.27379", "nurse"), (909, "-56.79708", "71.71547", "nurse"), (910, "83.77932", "-136.90077", "nurse"),
```



(911, "-1.28439", "-69.11277", "nurse"), (912, "-9.4289", "-13.0715", "nurse"), (913, "19.89417", "-145.55182", "nurse"), (914, "57.48701", "13.61399", "nurse"), (915, "24.66249", "176.7693", "nurse"), (916, "-74.93157", "-31.69506", "nurse"), (917, "21.06981", "-125.93315", "nurse"), (918, "-22.49537", "131.67963", "nurse"), (919, "-69.94006", "35.10381", "nurse"), (920, "67.23856", "-172.76096", "nurse"),

(921, "-54.87321", "-175.72953", "nurse"), (922, "58.58442", "157.46987", "nurse"), (923, "86.08818", "-92.88133", "nurse"), (924, "68.29149", "-177.691", "nurse"), (925, "-42.58871", "9.62496", "nurse"), (926, "-2.28432", "128.04393", "nurse"), (927, "18.49971", "-37.8636", "nurse"), (928, "0.49677", "60.14973", "nurse"), (929, "46.5184", "-68.23192", "nurse"), (930, "38.92957", "80.68918", "nurse"),

(931, "27.28005", "-157.03269", "nurse"), (932, "-67.23288", "35.85543", "nurse"), (933, "73.1138", "-64.67499", "nurse"), (934, "12.81415", "130.60915", "nurse"), (935, "-48.56017", "132.01879", "nurse"), (936, "41.07652", "132.84653", "nurse"), (937, "73.25197", "-31.13189", "nurse"), (938, "-40.47195", "-60.52677", "nurse"), (939, "83.50963", "-86.10776", "nurse"), (940, "50.68237", "-148.02306", "nurse"),

(941, "-32.62813", "-113.66746", "nurse"), (942, "-38.83279", "-41.76623", "nurse"), (943, "85.72829", "-73.09556", "nurse"), (944, "-21.47095", "174.92648", "nurse"), (945, "-39.4847", "132.90773", "nurse"), (946, "-23.15548", "-69.50037", "nurse"), (947, "-63.00531", "-168.09954", "nurse"), (948, "79.87833", "14.29325", "nurse"), (949, "76.29567", "126.13252", "nurse"), (950, "74.81362", "-27.9019", "nurse"),

(951, "83.22024", "44.3302", "MD"), (952, "42.89511", "62.12661", "MD"), (953, "-72.24498", "-21.947", "MD"), (954, "-75.2446", "42.07758", "MD"), (955, "21.34729", "123.18368", "MD"), (956, "48.61259", "-39.10123", "MD"), (957, "-79.06474", "-127.02086", "MD"), (958, "86.67257", "91.96773", "MD"), (959, "-39.81581", "-58.92906", "MD"), (960, "62.75604", "-61.75225", "MD"),

(961, "49.8558", "141.94934", "MD"), (962, "-47.83288", "-132.43783", "MD"), (963, "59.036", "-86.04761", "MD"), (964, "-86.51371", "175.23206", "MD"), (965, "60.37106", "-178.37416", "MD"), (966, "66.87144", "56.96425", "MD"), (967, "-81.87452", "-30.66752", "MD"), (968, "-53.28987", "44.40405", "MD"), (969, "47.41057", "-134.01532", "MD"), (970, "85.27764", "-118.79551", "MD"),

(971, "-54.88762", "-64.46131", "MD"), (972, "6.02416", "149.63244", "MD"), (973, "18.08337", "-165.62996", "MD"), (974, "-82.61548", "-138.47179", "MD"), (975, "-56.80715", "-171.35413", "MD"), (976, "13.60743", "-107.96295", "MD"), (977, "1.05038", "-158.30089", "MD"), (978, "4.4151", "-131.85144", "MD"), (979, "25.05869", "-106.37185", "MD"), (980, "55.37065", "151.75674", "MD"),

(981, "34.06451", "61.47562", "MD"), (982, "-20.22384", "-94.06534", "MD"), (983, "-2.27461", "97.37751", "MD"), (984, "-2.92211", "67.58299", "MD"), (985, "-88.58247", "-65.28968", "MD"), (986, "54.13884", "113.19142", "MD"), (987, "17.0576", "80.64108", "MD"), (988, "-53.1760",



9", "63.21582", "MD" ), ( 989, "-1.49723", "176.14447", "MD" ), ( 990, "-44.23606", "-32.23126", "MD" ),

( 991, "85.503", "-141.45096", "MD" ), ( 992, "88.85154", "46.87509", "MD" ), ( 993, "-10.0968", "-175.11681", "MD" ), ( 994, "-80.83329", "-103.53232", "MD" ), ( 995, "-79.94455", "-109.45958", "MD" ), ( 996, "42.69032", "-122.71336", "MD" ), ( 997, "-54.89519", "-143.07906", "MD" ), ( 998, "61.23916", "-16.95275", "MD" ), ( 999, "60.33552", "61.21025", "MD" ), ( 1000, "-80.39366", "-53.18915", "MD" );

HEALTH\_TRACK\_ADMIN

Schema	Details	Preview		
Row	ADMIN_ID	REG_LAT	REG_LONG	QUALIFICATION
1	967	-81.87452	-30.66752	MD
2	1000	-80.39366	-53.18915	MD
3	956	48.61259	-39.10123	MD
4	951	83.22024	44.3302	MD
5	973	18.08337	-165.62996	MD
6	990	-44.23606	-32.23126	MD
7	983	-2.27461	97.37751	MD
8	952	42.89511	62.12661	MD
9	981	34.06451	61.47562	MD
10	987	17.0576	80.64108	MD
11	953	-72.24498	-21.947	MD
12	985	-88.58247	-65.28968	MD
13	964	-86.51371	175.23206	MD

NOTIFICATION

INSERT INTO CONTACT\_TRACER.NOTIFICATION VALUES

( 1, "High", 316 ), ( 2, "High", 955 ), ( 3, "High", 944 ), ( 4, "High", 308 ), ( 5, "High", 747 ), ( 6, "High", 972 ), ( 7, "High", 34 ), ( 8, "High", 826 ), ( 9, "High", 223 ), ( 10, "High", 827 ),

( 11, "High", 152 ), ( 12, "High", 686 ), ( 13, "High", 475 ), ( 14, "High", 180 ), ( 15, "High", 178 ), ( 16, "High", 163 ), ( 17, "High", 988 ), ( 18, "High", 197 ), ( 19, "High", 562 ), ( 20, "High", 898 ),

( 21, "High", 33 ), ( 22, "High", 681 ), ( 23, "High", 328 ), ( 24, "High", 695 ), ( 25, "High", 192 ), ( 26, "High", 615 ), ( 27, "High", 855 ), ( 28, "High", 952 ), ( 29, "High", 755 ), ( 30, "High", 189 ),

( 31, "High", 678 ), ( 32, "High", 315 ), ( 33, "High", 13 ), ( 34, "High", 121 ), ( 35, "High", 254 ), ( 36, "High", 868 ), ( 37, "High", 117 ), ( 38, "High", 855 ), ( 39, "High", 631 ), ( 40, "High", 39 ),

( 41, "High", 904 ), ( 42, "High", 173 ), ( 43, "High", 909 ), ( 44, "High", 474 ), ( 45, "High", 436 ), ( 46, "High", 218 ), ( 47, "High", 683 ), ( 48, "High", 995 ), ( 49, "High", 881 ), ( 50, "High", 957 ),

( 51, "High", 148 ), ( 52, "High", 391 ), ( 53, "High", 339 ), ( 54, "High", 82 ), ( 55, "High", 61 ), ( 56, "High", 255 ), ( 57, "High", 699 ), ( 58, "High", 133 ), ( 59, "High", 684 ), ( 60, "High", 999 ),

(61, "High", 695), (62, "High", 777), (63, "High", 505), (64, "High", 119), (65, "High", 372), (66, "High", 541), (67, "High", 261), (68, "High", 361), (69, "High", 129), (70, "High", 886), (71, "High", 398), (72, "High", 511), (73, "High", 445), (74, "High", 444), (75, "High", 49), (76, "High", 377), (77, "High", 92), (78, "High", 420), (79, "High", 608), (80, "High", 655), (81, "High", 554), (82, "High", 699), (83, "High", 11), (84, "High", 961), (85, "High", 169), (86, "High", 216), (87, "High", 385), (88, "High", 853), (89, "High", 442), (90, "High", 358), (91, "High", 261), (92, "High", 341), (93, "High", 615), (94, "High", 290), (95, "High", 13), (96, "High", 151), (97, "High", 80), (98, "High", 823), (99, "High", 190), (100, "High", 193), (101, "Low", 414), (102, "Low", 443), (103, "Low", 907), (104, "Low", 862), (105, "Low", 777), (106, "Low", 592), (107, "Low", 788), (108, "Low", 497), (109, "Low", 864), (110, "Low", 61), (111, "Low", 627), (112, "Low", 154), (113, "Low", 780), (114, "Low", 735), (115, "Low", 45), (116, "Low", 548), (117, "Low", 89), (118, "Low", 619), (119, "Low", 715), (120, "Low", 6), (121, "Low", 878), (122, "Low", 37), (123, "Low", 966), (124, "Low", 895), (125, "Low", 457), (126, "Low", 660), (127, "Low", 953), (128, "Low", 429), (129, "Low", 935), (130, "Low", 278), (131, "Low", 193), (132, "Low", 794), (133, "Low", 940), (134, "Low", 552), (135, "Low", 103), (136, "Low", 146), (137, "Low", 953), (138, "Low", 435), (139, "Low", 349), (140, "Low", 289), (141, "Low", 791), (142, "Low", 974), (143, "Low", 560), (144, "Low", 106), (145, "Low", 398), (146, "Low", 54), (147, "Low", 731), (148, "Low", 227), (149, "Low", 2), (150, "Low", 242), (151, "Low", 285), (152, "Low", 847), (153, "Low", 681), (154, "Low", 924), (155, "Low", 90), (156, "Low", 907), (157, "Low", 424), (158, "Low", 705), (159, "Low", 363), (160, "Low", 148), (161, "Low", 269), (162, "Low", 26), (163, "Low", 528), (164, "Low", 161), (165, "Low", 154), (166, "Low", 160), (167, "Low", 250), (168, "Low", 337), (169, "Low", 374), (170, "Low", 332), (171, "Low", 508), (172, "Low", 510), (173, "Low", 59), (174, "Low", 152), (175, "Low", 122), (176, "Low", 162), (177, "Low", 832), (178, "Low", 792), (179, "Low", 457), (180, "Low", 273), (181, "Low", 859), (182, "Low", 458), (183, "Low", 879), (184, "Low", 444), (185, "Low", 18), (186, "Low", 72), (187, "Low", 218), (188, "Low", 521), (189, "Low", 176), (190, "Low", 441), (191, "Low", 702), (192, "Low", 518), (193, "Low", 149), (194, "Low", 70), (195, "Low", 448), (196, "Low", 839), (197, "Low", 234), (198, "Low", 767), (199, "Low", 567), (200, "Low", 316);

NOTIFICATION				
<div>SchemaDetailsPreview</div>				
Row	NOTIFICATION_ID	INFECTION_RISK	APP_USER_ID	
1	122	Low	37	
2	123	Low	966	
3	163	Low	528	
4	101	Low	414	
5	114	Low	735	
6	134	Low	552	
7	173	Low	59	
8	147	Low	731	
9	150	Low	242	
10	199	Low	567	
11	118	Low	619	
12	141	Low	791	
13	160	Low	148	

TEST

```
INSERT INTO CONTACT_TRACER.TEST (TEST_ID,TEST_DATE,TEST_TYPE,USER_ID, LAB_ID,
RESULT) VALUES

(100,"2019-09-27", "RT-PCR",
46,17,False),(101,"2020-03-19","RT-PCR",27,195,False),(102,"2020-03-10","RT-PCR",37
1,179,False),(103,"2020-02-21","RT-PCR",388,165,False),(104,"2019-09-26","RT-PCR",1
88,42,False),(105,"2019-06-27","RT-PCR",93,47,False),(106,"2020-05-11","RT-PCR",50,
93,False),(107,"2020-01-12","RT-PCR",226,135,False),(108,"2019-11-26","RT-PCR",124,
56,False),(109,"2020-05-13","RT-PCR",1,108,False),


(110,"2019-05-20","RT-PCR",168,113,False),(111,"2019-10-09","RT-PCR",186,81,False),
(112,"2019-08-28","RT-PCR",326,82,False),(113,"2020-01-09","RT-PCR",28,119,False),(
114,"2020-01-23","RT-PCR",185,6,False),(115,"2019-10-31","RT-PCR",347,181,False),(1
16,"2019-06-03","RT-PCR",363,73,False),(117,"2019-06-26","RT-PCR",383,56,False),(11
8,"2019-07-31","RT-PCR",273,107,False),(119,"2019-09-07","RT-PCR",136,111,False),

(120,"2019-07-12","RT-PCR",131,184,False),(121,"2019-05-12","RT-PCR",235,72,False),
(122,"2019-12-09","RT-PCR",19,47,False),(123,"2019-09-23","RT-PCR",235,30,False),(1
24,"2020-05-13","RT-PCR",120,121,False),(125,"2019-10-02","RT-PCR",344,184,False),(
126,"2020-02-06","RT-PCR",245,54,False),(127,"2020-02-06","RT-PCR",15,177,False),(1
28,"2019-12-17","RT-PCR",11,71,False),(129,"2019-05-08","RT-PCR",402,187,False),

(130,"2019-12-29","RT-PCR",308,10,False),(131,"2019-06-20","RT-PCR",232,53,False),(
132,"2019-06-06","RT-PCR",272,187,False),(133,"2020-01-23","RT-PCR",123,158,False),
```

(134, "2019-06-07", "RT-PCR", 137, 75, False), (135, "2020-05-05", "RT-PCR", 364, 170, False),  
(136, "2020-02-19", "RT-PCR", 284, 10, False), (137, "2020-03-12", "RT-PCR", 392, 20, False), (138, "2019-10-01", "RT-PCR", 176, 170, False), (139, "2019-06-16", "RT-PCR", 52, 126, False),  
(140, "2019-06-05", "RT-PCR", 266, 10, False), (141, "2019-10-21", "RT-PCR", 208, 158, False),  
(142, "2019-06-10", "RT-PCR", 132, 86, False), (143, "2019-11-30", "RT-PCR", 303, 164, False),  
(144, "2019-08-14", "RT-PCR", 143, 134, False), (145, "2019-12-01", "RT-PCR", 316, 144, False),  
(146, "2019-12-29", "RT-PCR", 5, 199, False), (147, "2019-10-20", "RT-PCR", 171, 173, False),  
(148, "2019-10-02", "RT-PCR", 209, 150, False), (149, "2020-04-20", "RT-PCR", 286, 56, False),  
(150, "2020-01-08", "RT-PCR", 299, 46, False), (151, "2019-07-05", "RT-PCR", 137, 138, False),  
(152, "2020-01-25", "RT-PCR", 248, 148, False), (153, "2019-06-22", "RT-PCR", 89, 79, False), (154, "2019-05-27", "RT-PCR", 145, 155, False), (155, "2020-02-05", "RT-PCR", 72, 40, False), (156, "2020-01-12", "RT-PCR", 426, 163, False), (157, "2020-02-15", "RT-PCR", 411, 55, False), (158, "2019-09-15", "RT-PCR", 89, 166, False), (159, "2019-09-16", "RT-PCR", 192, 120, False),  
(160, "2019-08-30", "RT-PCR", 306, 13, False), (161, "2019-05-13", "RT-PCR", 437, 180, False),  
(162, "2019-10-18", "RT-PCR", 8, 19, False), (163, "2019-08-29", "RT-PCR", 33, 164, False), (164, "2019-05-26", "RT-PCR", 135, 35, False), (165, "2020-04-26", "RT-PCR", 61, 58, False), (166, "2019-11-28", "RT-PCR", 152, 42, False), (167, "2020-03-14", "RT-PCR", 438, 47, False), (168, "2020-03-04", "RT-PCR", 238, 48, False), (169, "2019-06-06", "RT-PCR", 20, 41, False),  
(170, "2019-09-15", "RT-PCR", 167, 14, False), (171, "2019-11-15", "RT-PCR", 102, 58, False), (172, "2019-07-06", "RT-PCR", 425, 57, False), (173, "2019-11-18", "RT-PCR", 400, 103, False), (174, "2019-07-29", "RT-PCR", 442, 71, False), (175, "2019-11-09", "RT-PCR", 283, 25, False), (176, "2019-08-17", "RT-PCR", 382, 105, False), (177, "2019-09-29", "RT-PCR", 296, 68, False), (178, "2019-10-28", "RT-PCR", 312, 114, False), (179, "2019-08-20", "RT-PCR", 282, 197, False),  
(180, "2020-02-01", "RT-PCR", 96, 49, False), (181, "2019-12-10", "RT-PCR", 60, 181, False), (182, "2019-05-21", "RT-PCR", 29, 17, False), (183, "2019-12-28", "RT-PCR", 332, 78, False), (184, "2019-11-12", "RT-PCR", 275, 157, False), (185, "2020-04-03", "RT-PCR", 323, 129, False), (186, "2020-01-31", "RT-PCR", 230, 147, False), (187, "2019-06-29", "RT-PCR", 203, 173, False), (188, "2020-03-21", "RT-PCR", 77, 30, False), (189, "2019-06-17", "RT-PCR", 33, 141, False),  
(190, "2019-11-17", "RT-PCR", 139, 171, False), (191, "2019-09-07", "RT-PCR", 27, 167, False),  
(192, "2019-09-08", "RT-PCR", 139, 156, False), (193, "2020-03-28", "RT-PCR", 141, 97, False),  
(194, "2019-09-07", "RT-PCR", 185, 170, False), (195, "2019-05-29", "RT-PCR", 132, 29, False),  
(196, "2020-01-23", "RT-PCR", 10, 29, False), (197, "2020-05-12", "RT-PCR", 280, 185, False), (198, "2019-06-13", "RT-PCR", 257, 171, False), (199, "2019-07-18", "RT-PCR", 128, 131, False),

TEST

 QUERY

Schema

Details

Preview

Row	TEST_ID	TEST_DATE	TEST_TYPE	USER_ID	LAB_ID	RESULT	
1	118	2019-07-31	RT-PCR	273	107	false	
2	151	2019-07-05	RT-PCR	137	138	false	
3	180	2020-02-01	RT-PCR	96	49	false	
4	186	2020-01-31	RT-PCR	230	147	false	
5	146	2019-12-29	RT-PCR	5	199	false	
6	157	2020-02-15	RT-PCR	411	55	false	
7	176	2019-08-17	RT-PCR	382	105	false	
8	149	2020-04-20	RT-PCR	286	56	false	
9	179	2019-08-20	RT-PCR	282	197	false	
10	182	2019-05-21	RT-PCR	29	17	false	
11	125	2019-10-02	RT-PCR	344	184	false	
12	130	2019-12-29	RT-PCR	308	10	false	
13	143	2019-11-30	RT-PCR	303	164	false	
...	...	...	...	...	...	...	

LAB

```
INSERT INTO CONTACT_TRACER.LAB VALUES (1, "In", "29778"),

(2, "erat", "67908"),

(3, "mus", "16250"),

(4, "dolor", "98995"),

(5, "etu", "65104"),

(6, "vitae", "42042"),

(7, "at", "15278"),

(8, "tellus", "72580"),

(9, "facilisi", "28270"),

(10, "Pellentesque", "15278"),

(11, "lacus", "67908"),

(12, "id, ", "77000"),

(13, "est, ", "47822"),

(14, "allgou", "15278"),

(15, "nisl", "46833"),

(16, "mauris, ", "31365"),

(17, "Fusce", "86995"),

(18, "nunc", "75508"),

(19, "dignissim", "65104"),
```

(20, "mollis", "65104"),  
(21, "malesuada", "71102"),  
(22, "libero", "35024"),  
(23, "dui", "98995"),  
(24, "Mauris", "98995"),  
(25, "at", "72580"),  
(26, "inpoguy", "72580"),  
(27, "cursus", "72580"),  
(28, "dolor", "72539"),  
(29, "rhoncus.", "75508"),  
(30, "commodo", "72580"),  
(31, "odiolis", "72539"),  
(32, "lacusay", "47822"),  
(33, "consequat", "33887"),  
(34, "erat", "72539"),  
(35, "etpoy", "42042"),  
(36, "malesuada", "16503"),  
(37, "nonummy", "65104"),  
(38, "vel", "21421"),  
(39, "Aliquam", "42042"),  
(40, "nec", "67908"),  
(41, "utstar", "16503"),  
(42, "Nulla", "86995"),  
(43, "Mauris", "99829"),  
(44, "enimlo", "67908"),  
(45, "bibendum", "77000"),  
(46, "ligula", "77000"),  
(47, "Utrogger", "77000"),  
(48, "parturient", "29778"),  
(49, "apoil", "21421"),  
(50, "risus", "72580"),

LAB				
<div>SchemaDetailsPreview</div>				
Row	LAB_ID	LAB_NAME	ZIPCODE	
1	59	In	570732	
2	128	In	1112	
3	93	In	457918	
4	1	In	29778	
5	90	a,	3844	
6	73	ac	1675	
7	7	at	15278	
8	25	at	72580	
9	70	et	4949	
10	137	et	B4V 4L5	
11	145	et	56332	
12	96	eu	14829	
13	22		76633	

ADDRESS

```
INSERT INTO CONTACT_TRACER.ADDRESS VALUES ("16503","Gaithersburg","MD"),
("21421","New Orleans","LA"),
("46833","Flint","Michigan"),
("47822","Columbia","MO"),
("35024","Montgomery","Alabama"),
("99829","Ketchikan","Alaska"),
("31365","Helena","MT"),
("68241","Meridian","ID"),
("60704","Kapolei","Hawaii"),
("29778","Pike Creek","DE"),
("72539","Savannah","GA"),
("77000","Kearney","NE"),
("28270","Jacksonville","FL"),
("16250","San Antonio","Texas"),
("42042","South Portland","Maine"),
("71102","Augusta","Maine"),
("15278","Grand Island","NE"),
("65104","Houston","TX"),
("72580","Fayetteville","Arkansas"),
```



```
("67908", "Reno", "NV"),

("75508", "Norfolk", "Virginia"),

("33887", "Allentown", "PA"),

("86995", "Harrisburg", "PA"),

("98995", "Fort Worth", "TX"),

("18164", "Chattanooga", "TN");
```

ADDRESS				
Schema   Details <a href="#">Preview</a>				
Row	ZIPCODE	CITY	STATE	
1	29778	Pike Creek	DE	
2	28270	Jacksonville	FL	
3	72539	Savannah	GA	
4	68241	Meridian	ID	
5	21421	New Orleans	LA	
6	16503	Gaithersburg	MD	
7	47822	Columbia	MO	
8	31365	Helena	MT	
9	77000	Kearney	NE	
10	15278	Grand Island	NE	
11	67908	Reno	NV	
12	33887	Allentown	PA	
13	22202	Arlington	VA	

TRAVEL\_HISTORY

```
INSERT INTO CONTACT_TRACER.TRAVEL_HISTORY VALUES

("Ghana", 1), ("Rwanda", 2), ("Japan", 3), ("Tajikistan", 4), ("Uganda", 5), ("Mauritania", 6)

, ("Senegal", 7), ("Belize", 8), ("Saint Pierre and Miquelon", 9), ("Micronesia", 10);

INSERT INTO CONTACT_TRACER.TRAVEL_HISTORY VALUES ("French

Polynesia", 11), ("Barbados", 12), ("Kenya", 13), ("Croatia", 14), ("Mauritius", 15), ("Congo

(Brazzaville)", 16), ("Montserrat", 17), ("Guyana", 18), ("Nicaragua", 19), ("Bahamas", 20);

INSERT INTO CONTACT_TRACER.TRAVEL_HISTORY VALUES

("Georgia", 21), ("Iraq", 22), ("Zambia", 23), ("Swaziland", 24), ("Zambia", 25), ("Japan", 26

), ("Barbados", 27), ("Turkmenistan", 28), ("Slovakia", 29), ("Iraq", 30);

INSERT INTO CONTACT_TRACER.TRAVEL_HISTORY VALUES

("Bangladesh", 31), ("Israel", 32), ("Montenegro", 33), ("United Arab

Emirates", 34), ("Sint Maarten", 35), ("Austria", 36), ("Mongolia", 37), ("Palestine, State

of", 38), ("Bangladesh", 39), ("Ukraine", 40);
```



```
INSERT INTO CONTACT_TRACER.TRAVEL_HISTORY VALUES ("Russian Federation",41), ("Liberia",42), ("Malta",43), ("Mauritania",44), ("Italy",45), ("Malaysia",46), ("Nauru",47), ("Guyana",48), ("Croatia",49), ("Madagascar",50);

INSERT INTO CONTACT_TRACER.TRAVEL_HISTORY VALUES ("Indonesia",51), ("Liberia",52), ("Georgia",53), ("Albania",54), ("French Southern Territories",55), ("Malaysia",56), ("Turkey",57), ("Tokelau",58), ("Comoros",59), ("Thailand",60);

INSERT INTO CONTACT_TRACER.TRAVEL_HISTORY VALUES ("China",61), ("Denmark",62), ("Ethiopia",63), ("Montserrat",64), ("Sweden",65), ("Chile",66), ("Philippines",67), ("New Caledonia",68), ("Cuba",69), ("Yemen",70);

INSERT INTO CONTACT_TRACER.TRAVEL_HISTORY VALUES ("Suriname",71), ("Saint Barthélemy",72), ("Rwanda",73), ("Canada",74), ("Sudan",75), ("Sri Lanka",76), ("Malaysia",77), ("Western Sahara",78), ("Côte D'Ivoire (Ivory Coast)",79), ("Puerto Rico",80);

INSERT INTO CONTACT_TRACER.TRAVEL_HISTORY VALUES ("Bahamas",81), ("Guatemala",82), ("United Kingdom (Great Britain)",83), ("Belarus",84), ("British Indian Ocean Territory",85), ("Jersey",86), ("Trinidad and Tobago",87), ("Monaco",88), ("Myanmar",89), ("Bahamas",90);

INSERT INTO CONTACT_TRACER.TRAVEL_HISTORY VALUES ("Martinique",91), ("Malawi",92), ("Qatar",93), ("Malta",94), ("Armenia",95), ("Papua New Guinea",96), ("Curaçao",97), ("Netherlands",98), ("Isle of Man",99), ("Ghana",100);
```

TRAVEL_HISTORY			
<div>SchemaDetailsPreview</div>			
Row	COUNTRY	USER_ID	
1	Chad	438	
2	Chad	567	
3	Chad	721	
4	Cuba	69	
5	Cuba	305	
6	Cuba	230	
7	Cuba	427	
8	Cuba	534	
9	Fiji	171	
10	Fiji	101	
11	Fiji	392	
12	Fiji	471	
13	Fiji	770	

TRACKER

INSERT INTO CONTACT\_TRACER.TRACKER VALUES

(1, 174, 20),

(2, 174, 23),

(3, 174, 10),

(4, 174, 5),

(5, 153, 44),

(6, 26, 1),

(7, 26, 40),

(8, 34, 23),

(9, 42, 3),

(10, 49, 30),

(11, 30, 24),

(12, 18, 3),

(13, 47, 36),

(14, 36, 36),

(15, 86, 39),

(16, 115, 3),

(17, 42, 46),

(18, 131, 26),

(19, 68, 38),

(20, 52, 35),

(21, 92, 13),

(22, 198, 8),

(23, 58, 3),

(24, 95, 21),

(25, 67, 4),

(26, 7, 35),

(27, 154, 3),

(28, 188, 23),

(29, 116, 55),

(30, 185, 54),  
(31, 89, 19),  
(32, 59, 30),  
(33, 79, 33),  
(34, 168, 9),  
(35, 186, 49),  
(36, 54, 5),  
(37, 54, 40),  
(38, 39, 9),  
(39, 18, 55),  
(40, 159, 49),  
(41, 106, 10),  
(42, 84, 21),  
(43, 58, 4),  
(44, 108, 55),  
(45, 48, 27),  
(46, 181, 7),  
(47, 5, 54),  
(48, 164, 12),  
(49, 31, 40),  
(50, 94, 53),  
(51, 184, 19),  
(52, 30, 17),  
(53, 95, 27),  
(54, 175, 22),  
(55, 193, 5),  
(56, 55, 28),  
(57, 140, 15),  
(58, 138, 24),  
(59, 26, 14),  
(60, 69, 34),  
(61, 35, 3),

(62, 108, 35),  
(63, 123, 47),  
(64, 89, 22),  
(65, 184, 40),  
(66, 127, 3),  
(67, 117, 3),  
(68, 133, 3),  
(69, 81, 46),  
(70, 53, 41),  
(71, 105, 40),  
(72, 83, 32),  
(73, 101, 44),  
(74, 23, 3),  
(75, 47, 3),  
(76, 168, 36),  
(77, 96, 35),  
(78, 26, 15),

TRACKER				
<div>SchemaDetailsPreview</div>				
Row	TRACKER_ID	GPS_LOC	USER_ID	
1	6	26	1	
2	104	111	2	
3	12	18	3	
4	27	154	3	
5	68	133	3	
6	290	45	3	
7	66	127	3	
8	118	76	3	
9	115	189	3	
10	299	161	3	
11	111	25	3	
12	211	134	3	
13	33	133	3	

GPS\_LOCATION

```
INSERT INTO CONTACT_TRACER.GPS_LOCATION VALUES
(1, "-34.72092", "152.80613", "2020-05-27 11:10:18"),
(2, "-8.97384", "62.66872", "2020-05-26 15:42:24"),
(3, "-80.70657", "176.32768", "2021-04-22 06:17:37"),
(4, "69.60905", "101.08713", "2020-09-04 21:21:43"),
(5, "21.26325", "-88.96061", "2021-11-02 08:21:06");
```

GPS\_LOCATION

Schema

Details

Preview

Row	GPS_LOC	GPS_LAT	GPS_LONG	DATE
1	12	-30.95478	8.43323	2021-01-29T23:01:00
2	112	-30.95478	8.423	2021-01-29T23:01:00
3	59	-38.16435	16.86402	2022-01-04T00:04:32
4	159	-48.16435	16.86402	2022-01-04T00:04:32
5	31	63.71132	55.7328	2021-07-27T22:12:00
6	131	63.711	55.7328	2021-07-27T22:12:00
7	46	-53.26431	-128.36983	2020-12-18T01:14:16
8	146	-53.26431	-128.36983	2020-12-18T01:14:16
9	62	-26.49651	75.78811	2021-09-04T08:18:48
10	162	-26.49651	75.711	2021-09-04T08:18:48
11	17	20.5259	-107.72261	2020-05-16T22:29:00
12	117	20.5259	-107.72261	2020-05-16T22:29:00
13	7	21.8888	75.88756	2020-11-28T17:48:00

# Testing & Evaluation

\*Constraint, CRUD operations

When the application is under execution, the end-user mainly utilizes the ‘CRUD’ operations facilitated by the DB Tool.

**C: Create** – When user ‘Save’ any new transaction, ‘Create’ operation is performed.

**R: Retrieve** – When user ‘Search’ or ‘View’ any saved transaction, ‘Retrieve’ operation is performed.

**U: Update** – When user ‘Edit’ or ‘Modify’ an existing record, the ‘Update’ operation of DB is performed.

**D: Delete** – When a user ‘Remove’ any record from the system, ‘Delete’ operation of DB is performed.

Any database operation performed by the end-user is always one of the above four.

So, using CRUD operation we are testing our database.

# Query Descriptions

## Queries for objectives

1. Using SELECT CASE/WHEN to determine risk status of a person based on his/her infection status and immunity compromised or not.

```
SELECT USER_ID, INFECTION_STATUS, IMMUNITY_COMPROMISED,  
  
CASE WHEN INFECTION_STATUS = 'suspected' AND IMMUNITY_COMPROMISED = TRUE  
  
THEN 'This user is at high risk'  
  
WHEN INFECTION_STATUS = 'suspected' AND IMMUNITY_COMPROMISED = FALSE  
  
THEN 'This user is at moderate risk'  
  
WHEN INFECTION_STATUS = 'infected'  
  
THEN 'This user is Covid-19 positive'  
  
WHEN INFECTION_STATUS = 'not-infected' AND IMMUNITY_COMPROMISED = TRUE  
  
THEN 'This user is at low risk'  
  
ELSE 'This User is safe'  
  
END AS Alert  
  
FROM CONTACT_TRACER.USER
```

Query complete (0.0 sec elapsed, cached)

Job information

Results

JSON

Execution details

Row	USER_ID	INFECTION_STATUS	IMMUNITY_COMPROMISED	Alert	
1	51	infected	false	This user is Covid-19 positive	
2	45	infected	false	This user is Covid-19 positive	
3	16	infected	false	This user is Covid-19 positive	
4	42	infected	false	This user is Covid-19 positive	
5	20	infected	false	This user is Covid-19 positive	
6	52	suspected	false	This user is at moderate risk	
7	49	suspected	false	This user is at moderate risk	
8	27	suspected	false	This user is at moderate risk	
9	12	not-infected	false	This User is safe	
10	30	not-infected	false	This User is safe	
11	18	not-infected	false	This User is safe	
12	17	not-infected	false	This User is safe	
13	39	not-infected	false	This User is safe	
14	28	not-infected	false	This User is safe	
15	38	not-infected	false	This User is safe	
16	8	not-infected	false	This User is safe	
17	68	infected	false	This user is Covid-19 positive	
18	122	infected	false	This user is Covid-19 positive	
19	80	infected	false	This user is Covid-19 positive	
20	142	infected	false	This user is Covid-19 positive	
21	119	infected	false	This user is Covid-19 positive	
22	94	infected	false	This user is Covid-19 positive	

2. List of labs that provide the more than 1 immunity tests.

```
SELECT Count(T.TEST_ID) AS Test_Count, L.LAB_NAME AS Lab_Name
FROM CONTACT_TRACER.LAB AS L, CONTACT_TRACER.TEST AS T
WHERE T.LAB_ID = L.LAB_ID
GROUP BY L.LAB_ID, L.LAB_NAME
HAVING COUNT(T.TEST_ID) > 1
```

Query complete (0.0 sec elapsed, cached)

Job information

Results

JSON

Execution details

Row	Test_Count	Lab_Name	
1	3	Pellentesque	
2	2	vitae diam. Proin dolor. Nulla	
3	2	commodo	
4	2	rhoncus.	
5	3	Utrogger	
6	2	Fusce	
7	2	Nulla	

3. Identify the locations of the infected persons

```
SELECT U.USER_ID, G.GPS_LAT, G.GPS_LONG
```



```
FROM `mohini-data225-spring2021.CONTACT_TRACER.GPS_LOCATION` AS G

INNER JOIN `mohini-data225-spring2021.CONTACT_TRACER.TRACKER` AS T

ON G.GPS_LOC = T.GPS_LOC

INNER JOIN `mohini-data225-spring2021.CONTACT_TRACER.USER` AS U

ON T.USER_ID = U.USER_ID

WHERE U.INFECTION_STATUS = "infected"
```

Query complete (0.4 sec elapsed, 27.7 KB processed)

Job informationResultsJSONExecution details

Row	USER_ID	GPS_LAT	GPS_LONG	
1	32	45.37975	-160.22205	
2	47	28.94	-18.613	
3	20	72.42477	-117.69821	
4	50	-84.01231	-176.84174	
5	15	60.19581	-149.34752	
6	32	-55.14301	1.18273	
7	47	21.26325	-88.96061	
8	11	25.34848	-154.75583	
9	47	-16.785	29.09286	
10	42	-73.57437	-14.8258	
11	47	-69.90588	-92.48282	
12	50	-69.90588	-92.48282	
13	11	-84.47124	-32.95661	
14	1	47.43498	-19.29981	
15	15	47.43498	-19.29981	
16	42	24.75319	59.71643	
17	11	-25.01256	-111.8108	

4. Infected persons travel history

```
SELECT U.USER_ID, T.COUNTRY AS Country_travelled_to

FROM `mohini-data225-spring2021.CONTACT_TRACER.USER` AS U

INNER JOIN `mohini-data225-spring2021.CONTACT_TRACER.TRAVEL_HISTORY` AS T

ON U.USER_ID = T.USER_ID

WHERE U.INFECTION_STATUS = 'infected'
```

GROUP BY U.USER\_ID, T.COUNTRY

Query complete (0.4 sec elapsed, 35 KB processed)

Job informationResultsJSONExecution d

Row	USER_ID	Country_travelled_to	
1	51	Indonesia	
2	45	Italy	
3	16	Congo (Brazzaville)	
4	42	Liberia	
5	20	Bahamas	
6	68	New Caledonia	
7	122	Germany	
8	80	Puerto Rico	
9	142	India	
10	119	Marshall Islands	
11	94	Malta	
12	63	Ethiopia	
13	101	Fiji	
14	116	Cambodia	
15	66	Chile	
16	76	Sri Lanka	
17	67	Philippines	

5. Find non-infected users and their living situation who have come in contact with a infected person (CONTACT TRACING) based on GPS location

```
SELECT U.USER_ID, U.LIVING_STATUS

FROM CONTACT_TRACER.USER AS U, CONTACT_TRACER.TRACKER AS T,
CONTACT_TRACER.GPS_LOCATION AS G

WHERE U.USER_ID = T.USER_ID

AND T.GPS_LOC = G.GPS_LOC

AND U.INFECTION_STATUS != 'infected'

AND T.GPS_LOC IN (

    SELECT T1.GPS_LOC

    FROM CONTACT_TRACER.USER AS U1, CONTACT_TRACER.TRACKER AS T1

    WHERE U1.USER_ID = T1.USER_ID

    AND U1.INFECTION_STATUS = 'infected'

)
```

Query complete (0.0 sec elapsed, cache

Job information   [Results](#)   JSON

Row	USER_ID	LIVING_STATUS	
1	3	group	
2	4	isolated	
3	4	isolated	
4	5	isolated	
5	7	isolated	
6	10	group	
7	14	isolated	
8	19	isolated	
9	23	isolated	
10	24	isolated	
11	36	isolated	
12	40	group	
13	40	group	

6. Number of infected cases per city

```
SELECT COUNT(U.USER_ID) AS infected_case_count, A.CITY
FROM CONTACT_TRACER.USER AS U
INNER JOIN CONTACT_TRACER.ADDRESS AS A
ON U.ZIPCODE = A.ZIPCODE
WHERE U.INFECTION_STATUS = 'infected'
GROUP BY A.CITY
```

Query complete (0.2 sec elapsed, cached)			
Job information		Results	JSON   Execution
Row	infected_case_count	CITY	
1	1	Kapolei	
2	1	San Antonio	
3	1	Grand Island	
4	2	Helena	
5	10	Flint	
6	1	South Portland	
7	1	Savannah	
8	1	Ketchikan	
9	1	Norfolk	
10	1	New Orleans	

**View:**

A view is a virtual table defined by a SQL query. When we create a view, we query it in the same way we query a table. When a user queries the view, the query results contain data only from the tables and fields specified in the query that defines the view.

Also we can give a view access to a dataset that is known as creating an authorized view in BigQuery. An authorized view lets us share query results with particular users and groups without giving them access to the underlying tables. We can also use the view's SQL query to restrict the columns (fields) the users are able to query.

Here, we are creating a view for the user’s test and test result conducted by the lab.

```
CREATE VIEW CONTACT_TRACER.USERDATA AS (SELECT USER.USER_ID AS UserID, LAB.LAB_NAME
AS LabName, TEST.TEST_TYPE AS TestType , TEST.TEST_DATE as TestDate, ADDRESS.STATE
as State,

CASE WHEN TEST.RESULT = False THEN 'Negative'

ELSE 'Positive'

END AS TestResult

FROM CONTACT_TRACER.USER , CONTACT_TRACER.TEST ,
CONTACT_TRACER.LAB, CONTACT_TRACER.ADDRESS

WHERE LAB.LAB_ID= TEST.LAB_ID AND

TEST.USER_ID = USER.USER_ID AND

USER.ZIPCODE=ADDRESS.ZIPCODE

ORDER BY UserID)
```

Query complete (0.0 sec elapsed, cached)

Job information   [Results](#)   JSON

Row	UserID	LabName	TestType	TestDate	TestResult	
1	8	dignissim	RT-PCR	2019-10-18	Negative	
2	10	rhoncus.	RT-PCR	2020-01-23	Negative	
3	19	Utrogger	RT-PCR	2019-12-09	Negative	
4	20	utstar	RT-PCR	2019-06-06	Negative	
5	29	Fusce	RT-PCR	2019-05-21	Negative	
6	46	Fusce	RT-PCR	2019-09-27	Negative	
7	61	vitae diam. Proin dolor. Nulla	RT-PCR	2020-04-26	Negative	
8	72	nec	RT-PCR	2020-02-05	Negative	
9	77	commodo	RT-PCR	2020-03-21	Negative	
10	93	Utrogger	RT-PCR	2019-06-27	Negative	
11	96	apoil	RT-PCR	2020-02-01	Negative	
12	102	vitae diam. Proin dolor. Nulla	RT-PCR	2019-11-15	Negative	
13	132	rhoncus.	RT-PCR	2019-05-29	Negative	
14	135	etpoy	RT-PCR	2019-05-26	Negative	
15	152	Nulla	RT-PCR	2019-11-28	Negative	
16	167	allgou	RT-PCR	2019-09-15	Negative	
17	185	vitae	RT-PCR	2020-01-23	Negative	
18	188	Nulla	RT-PCR	2019-09-26	Negative	
19	235	commodo	RT-PCR	2019-09-23	Negative	
20	238	parturient	RT-PCR	2020-03-04	Negative	
21	245	Aenean eget metus. In	RT-PCR	2020-02-06	Negative	
22	266	Pellentesque	RT-PCR	2019-06-05	Negative	
23	283	at	RT-PCR	2019-11-09	Negative	
24	284	Pellentesque	RT-PCR	2020-02-19	Negative	
25	296	vestibulum nec, euismod in, dolor. Fusce feugiat. Lorem ipsum dolor	RT-PCR	2019-09-29	Negative	
26	299	ligula	RT-PCR	2020-01-08	Negative	
27	306	est,	RT-PCR	2019-08-30	Negative	
28	308	Pellentesque	RT-PCR	2019-12-29	Negative	
29	392	mollis	RT-PCR	2020-03-12	Negative	
30	437	nisl arcu iaculis enim, sit	RT-PCR	2019-05-13	Negative	
31	438	Utrogger	RT-PCR	2020-03-14	Negative	
32	451	et, magna. Praesent interdum ligula	Antibody	2020-01-18	Positive	
33	514	Nunc	Antibody	2020-05-03	Positive	
34	636	dapibus gravida. Aliquam tincidunt,	Antibody	2019-08-19	Positive	
35	734	arcu iaculis enim, sit	Antibody	2019-09-10	Positive	
36	773	conubia	Antibody	2019-07-08	Positive	
37	888	in consectetuer ipsum	Antibody	2019-11-28	Positive	
38	899	sapien, gravida non, sollicitudin a, malesuada	Antibody	2019-09-19	Positive	

Stored Procedure:

Stored procedures execute multiple statements in one request. Stored procedures also allow us to build sophisticated workflows using control logic such as IF...THEN, and WHILE, and save these tasks as stored procedures within BigQuery so that they can be shared with other users, who can then invoke these procedures with a CALL command.

We can also use variables to apply parameters to customize execution of the script, as well as use temporary tables to stage data at the session level. BigQuery users who are familiar with procedural languages will find it easy to use these new capabilities to accelerate data migration or execute complex ETL logic—as well as save time with logic reuse.

Here we are creating stored procedure to get number of infected cases for 'Michigan' state

```
CREATE OR REPLACE PROCEDURE CONTACT_TRACER.GetCovidCases (  
  
IN STATE_IN STRING, OUT COUNT INT64  
  
)
```

```
BEGIN
```

```
SET COUNT = (SELECT COUNT(U.USER_ID)
```

```
FROM CONTACT_TRACER.USER AS U
```

```
INNER JOIN CONTACT_TRACER.ADDRESS AS A
```

```
ON U.ZIPCODE = A.ZIPCODE
```

```
WHERE U.INFECTION_STATUS = 'infected' AND
```

```
A.STATE = STATE_IN
```

```
);
```

```
END;
```

```
DECLARE COUNT INT64;
```

```
DECLARE STATE_NAME STRING DEFAULT 'Michigan';
```

```
CALL CONTACT_TRACER.GetCovidCases(STATE_NAME, COUNT);
```

```
SELECT STATE_NAME, COUNT
```

Row	STATE_NAME	COUNT	
1	Michigan	10	

# Contact Tracing for Covid-19

## Team:

### Project Group 6

Mohini Patil - 015359188

Samhitha Upadhyaya - 015276092

Shwetarani - 015277964

## \*M Introduction

Contact tracing in the context of COVID-19 requires identifying persons who may have been exposed to a person with COVID-19 and monitoring them up daily for 14 days from the last point of exposure. Contacts should remain in self-quarantine during the 14-day monitoring period in order to limit the possibility of exposing other people to infection should they become ill, since COVID-19 transmission can occur before symptoms develop.

## \*M How the system will work?

The designated health officials will receive the information immediately via the internet, analyse the data, and use the system's communication and messaging tools to respond. The designated health officials can monitor incoming reports through a Web interface.

Once the designated health officials analyse the data with COVID-19 systems, they will notify you of the necessary things to do, such as 14 days of self-isolation, wearing a surgical mask, distancing, and test via e-mail and SMS messages. The designated health officials could communicate with remote health officials in that locality to intensify the awareness of COVID-19 guidelines such as the washing of hands, avoid handshaking, social distancing, etc.

If you test positive for COVID-19, the designated health official will ask you to consent to your handshake information via a digital form. A child's parent, guardian, or care will be asked for consent if tested positive.

The filled information enables designated health officials to call close contacts to advise them on what to do. Health officials will not disclose the name of the infected person. Also, the database will only be available to authorised health officials.

## Purpose

The project aims at applying the concepts of database design and modeling to create a Contact Tracer database design for COVID 19 from initial planning of database to implementation and testing of the database. It also aims to design a business model for contact Tracing of COVID 19 system which allows different users to use this database. The contact tracer database designed, can be used for mobile application development.

## Description of the Project

Contact Tracing is used to control the spread of disease enforced by respective health departments, preventing further spread of COVID-19. The aim is to create a Contact Tracer database for COVID 19 in order to store different information related to users using contact tracer application. The main idea is to keep track of information about people who have come in contact with the patient infected by COVID 19. The GPS location tracker is used to track the user's location and notify them. The database also contains the information about tests conducted in a lab at a particular city.

## Database Initial Study

- Analysis & requirements



Data Analysis and requirements is the first step in conceptual design, here we are identifying the characteristics of data elements. Data element characteristics should contain the following information. The database design should generate the contact tracing information, that is used to control the spread of disease. The database is designed for those who want to use the contact tracing information.

The Database is designed to track information about people who have come in contact with the patient infected by COVID 19. Using GPS coordinates for a given date, infected people within a particular radius of their region are tracked. Notifications are sent to individuals when there is a chance of that user being contacted by an infected person. Further, Lab test results are used to know whether the person is infected with COVID-19. Based on the Lab test results the infection status gets updated.

## ● Problems and Constraints

The model is designed such that it can be used in contact tracing for the Covid-19 pandemic, and some additional features can be applied so that it can be used for general contact tracing as well for analytical purposes.

The application users, who use this application can be categorised into user and health track admin. The user is the one who is monitored and tracked to check if they have come in contact with any infected user. The user's infection status is given based on the tests conducted in the lab, which is further verified by the health track admin.

Health track admin is the one who can see the information about users and they get notified if the user is infected, in a region where the health check admin holds responsibility.

Users are tracked to check if they have come in contact with an infected user. They are designated as either infected, not-infected or suspected based on the infection status which is given to them based on tests conducted in a lab at a particular city.

Users will also be notified if they are at risk of contracting Covid-19. The risk can be calculated using triggers. The risk of having contracted Covid-19 depends on the following factors

- Individuals having medical conditions which compromises their immunity.
- Individual's travel history within a given period of time.
- Individuals coming in contact with other infected individuals based on GPS tracking.

A user is tracked based on GPS coordinates with "gpsLocation" attribute that uniquely identifies the coordinates of the person on that particular date while considering the virus to be present where that infected individual has been at least one day.

## Objectives

The main objective of our project is to give people a sense of security by preventing further spread of COVID 19 disease, which is achieved by keeping track of individuals who have come in contact with the infected person. From the overall objective perspective we try to achieve the following:

- Determine infection risk status of particular individuals.
- List of labs that provide the maximum number of immunity tests. \*
- Identify the location with the highest number of risks.

#(most infected users from particular gps\_location)

- Find the country name which has the highest number of infected status.

# user with infected status as infected travelled to which country

- Find non-infected persons and their living status, who is in contact with the infected person.

# Find non-infected users and their living situation who have come in contact with a

# infected person (CONTACT TRACING) based on GPS location and date

- Number of infected cases per city

## **\*S.Scope & Boundaries of project**

The Scope of the project includes to quickly locate infected patients, and help them to isolate from others to prevent the further spread of the disease and also to identify people with whom the infected person was in close contact using GPS Tracker. The Contact Tracer Database can be used in other applications with some additional enhancements.It can also be used to achieve generic contact tracing.

\*\*Application

\*\*Survey

## **Database Design Approach**

The Top Down Design Approach is followed,where the data sets are identified,and for those data sets ER Model is designed by defining Entities and their attributes,along with relationships between different entities and Relational Schema is designed.

## **Conceptual Design**

The Conceptual Design is a high-level design process which involves describing entities,attributes,relationships and constraints for COVID 19 Contact Tracer Database.

- \*\*Elaborate

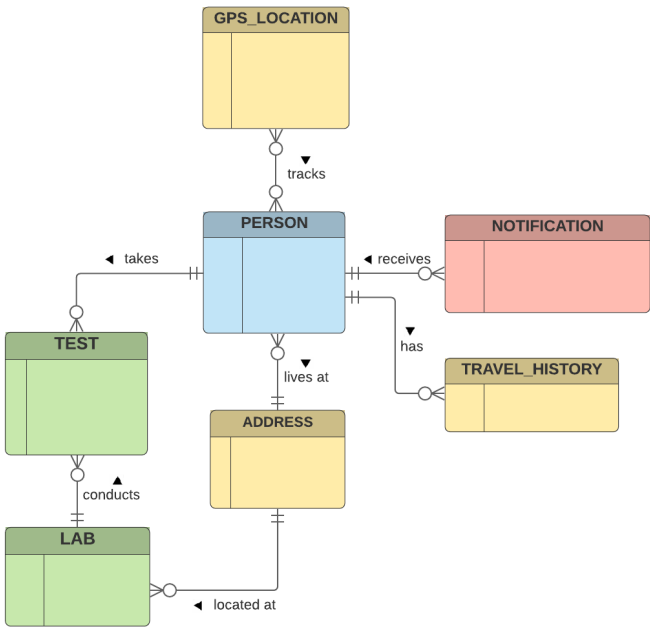
## **Business Rules**

- An APP\_USER can be a HEALTH\_TRACK\_ADMIN and each HEALTH\_TRACK\_ADMIN is an app user.
- An APP\_USER can be a USER and each USER belongs to an app user.
- Each USER can have one address and each ADDRESS belongs to many users.
- Each USER can take many tests and each TEST is carried out on a particular user.
- A Lab can conduct many tests and each TEST is carried out in one lab.
- Each USER can have many tracking GPS locations and each GPS location is tracked by many users.
- Each USER can have many travel histories and each TRAVEL HISTORY belongs to one user.
- An APP\_USER can receive many notifications about his risk of contracting COVID-19 and each NOTIFICATION is received by one app\_user.
- Each LAB is located at a particular address and each ADDRESS belongs to many labs.

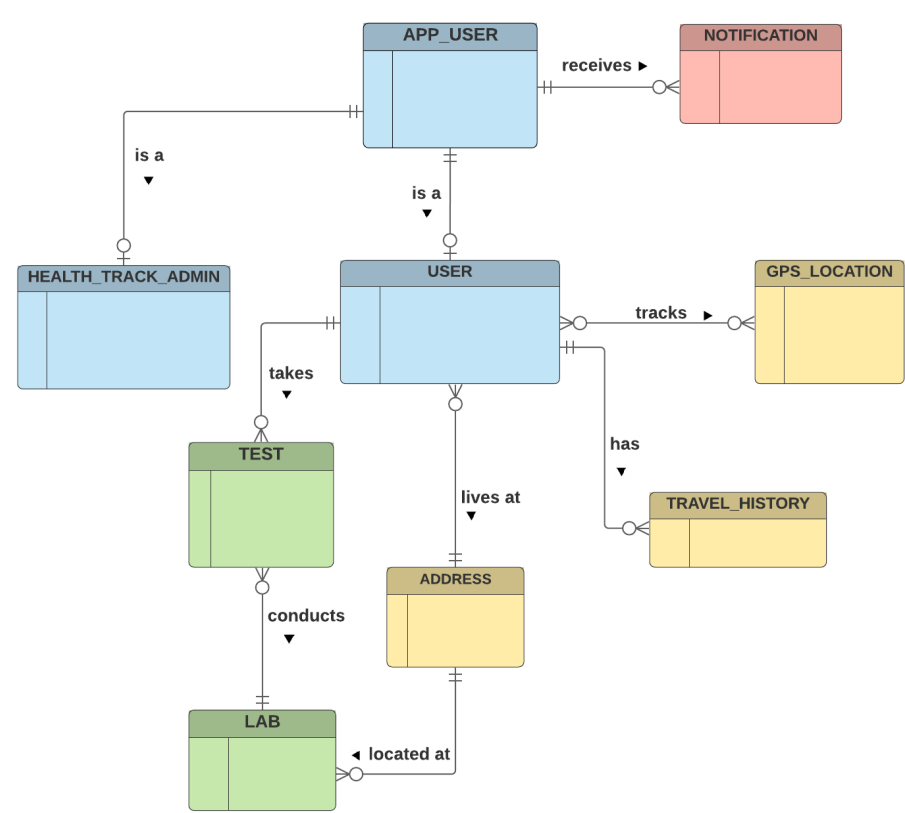
Entity Relationship Table:

ENTITY	RELATIONSHIP	CONNECTIVITY	ENTITY
ADDRESS	lives at	1:M	USER
USER	takes	1:M	TEST
LAB	conducts	1:M	TEST
USER	tracks	M:N	GPS_LOCATION
USER	has	1:M	TRAVEL_HISTORY
APP_USER	receives	1:M	NOTIFICATION
APP_USER	is a	1:1	USER
APP_USER	is a	1:1	HEALTH_TRACK_ADMIN
ADDRESS	loacted at	1:M	LAB

Conceptual Model Showing various Entities and Attributes



Revised Conceptual Model Showing Various Entities and Attributes

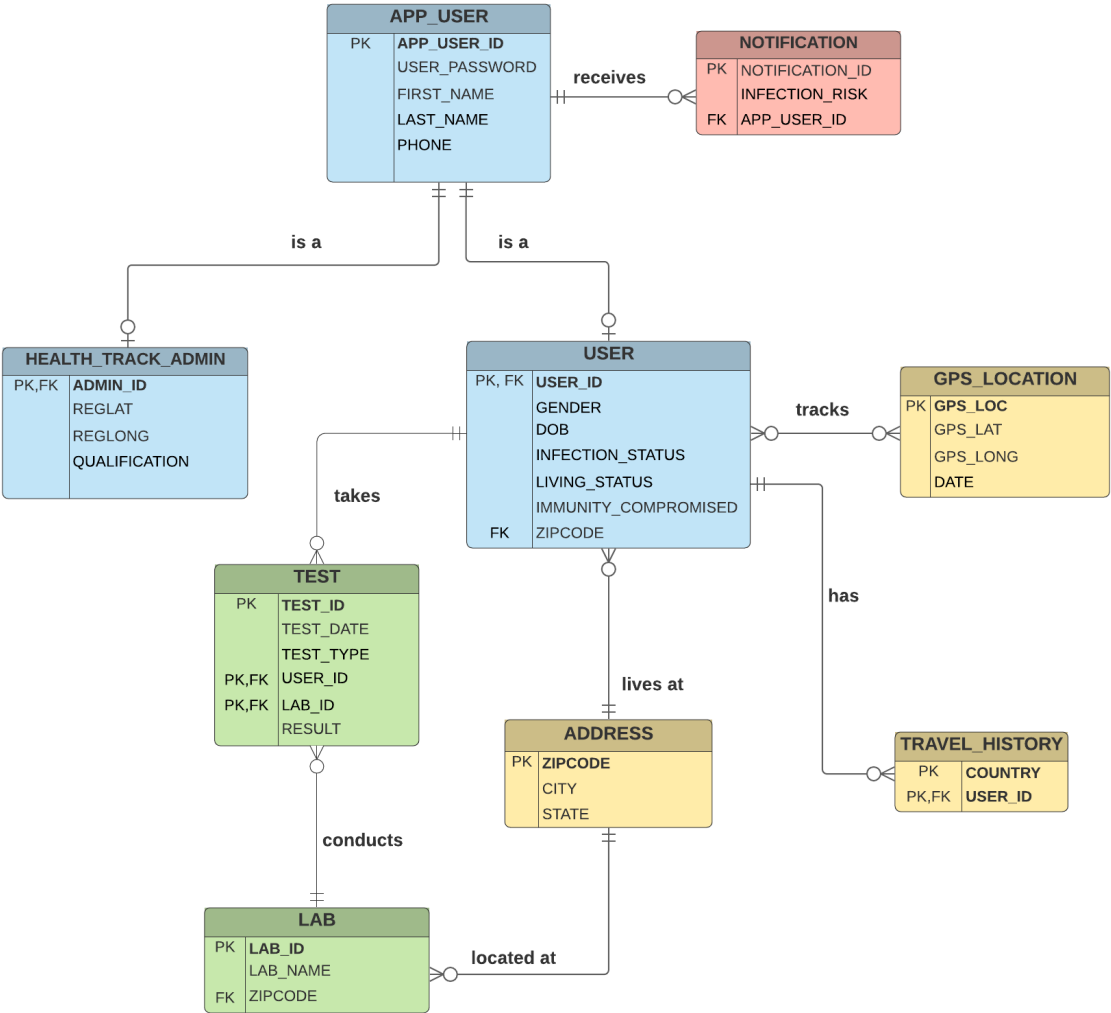


Developing an Entity Relational Diagram

The process of developing the ER model is an iterative process which involves defining entities, attributes, relationships and constraints. In order to develop a workable ER model we start designing the initial ER Model, which will lead us to the Final ER model.

ENTITY	ATTRIBUTES	PRIMARY	FOREIGN KEY
USER	USER_ID GENDER DOB INFECTION_STATUS PHONE_NUMBER LIVING_STATUS IMMUNITY_COMPROMISED ZIPCODE	USER_ID	ZIPCODE
LAB	LAB_ID LAB_NAME ZIPCODE	LAB_ID	ZIPCODE
NOTIFICATION	NOTIFICATION_ID INFECTION_RISK APP_USER_ID	NOTIFICATION_ID	USER_ID
TEST	TEST_ID TEST_DATE TEST_TYPE USER_ID RESULT LAB_ID	TEST_ID LAB_ID USER_ID	LAB_ID USER_ID
GPS_LOCATION	GPS_LOC GPS_LAT GPS_LONG DATE	GPS_LOC	NONE
ADDRESS	ZIPCODE CITY STATE	ZIPCODE	NONE
TRAVEL_HISTORY	COUNTRY USER_ID	COUNTRY USER_ID	USER_ID
APP_USER	APP_USER_ID USER_PASSWORD FIRST_NAME LAST_NAME PHONE	USER_ID	NONE
HEALTH_TRACK_ADMIN	ADMIN_ID REGLAT REGLONG QUALIFICATION	ADMIN_ID	ADMIN_ID

Initial Entity Relationship Model



Logical Design

Logical design requires to map entities,relationships and constraints defined in the conceptual model to relational database.The logical design is independent of different physical-level details.

Entity, Attributes & Constraints

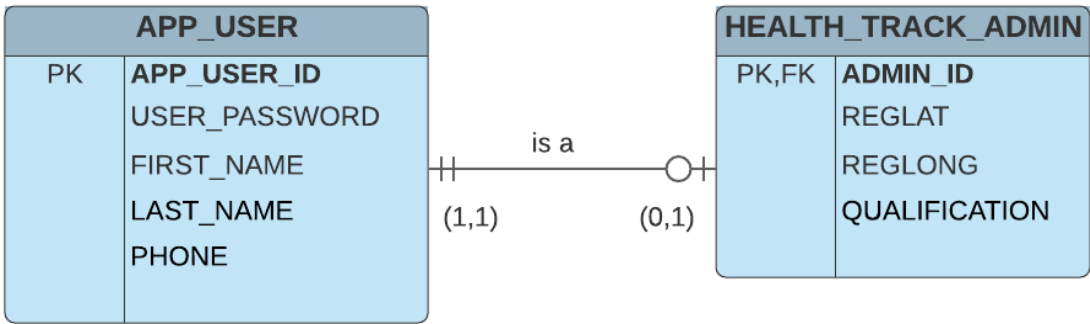
Entity and their attributes:

ENTITY	ATTRIBUTES	PRIMARY	FOREIGN KEY
USER	USER_ID GENDER DOB INFECTION_STATUS PHONE_NUMBER LIVING_STATUS IMMUNITY_COMPROMISED ZIPCODE	USER_ID	ZIPCODE
LAB	LAB_ID LAB_NAME ZIPCODE	LAB_ID	ZIPCODE
NOTIFICATION	NOTIFICATION_ID INFECTION_RISK APP_USER_ID	NOTIFICATION_ID	USER_ID
TEST	TEST_ID TEST_DATE TEST_TYPE USER_ID RESULT LAB_ID	TEST_ID LAB_ID USER_ID	LAB_ID USER_ID

GPS_LOCATION	GPS_LOC GPS_LAT GPS_LONG DATE	GPS_LOC	NONE
ADDRESS	ZIPCODE CITY STATE	ZIPCODE	NONE
TRAVEL_HISTORY	COUNTRY USER_ID	COUNTRY USER_ID	USER_ID
APP_USER	APP_USER_ID USER_PASSWORD FIRST_NAME LAST_NAME PHONE	USER_ID	NONE
HEALTH_TRACK_ADMIN	ADMIN_ID REGLAT REGLONG QUALIFICATION	ADMIN_ID	ADMIN_ID

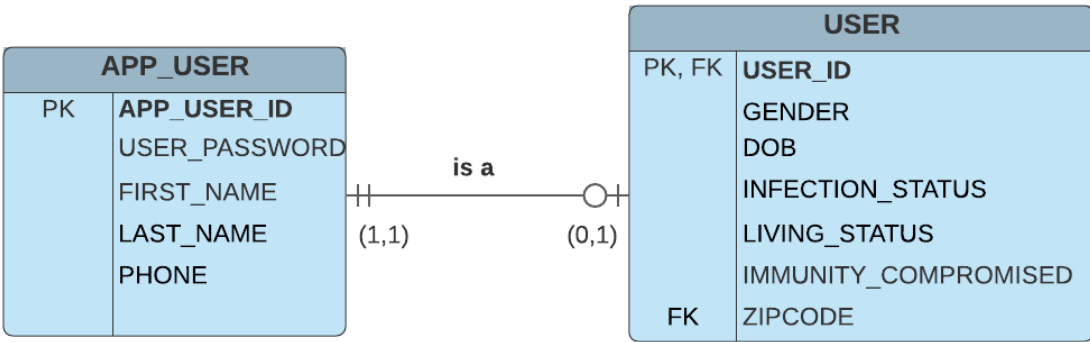
Mapping Conceptual Model to Logical Model

APP\_USER to HEALTH\_TRACK\_ADMIN:



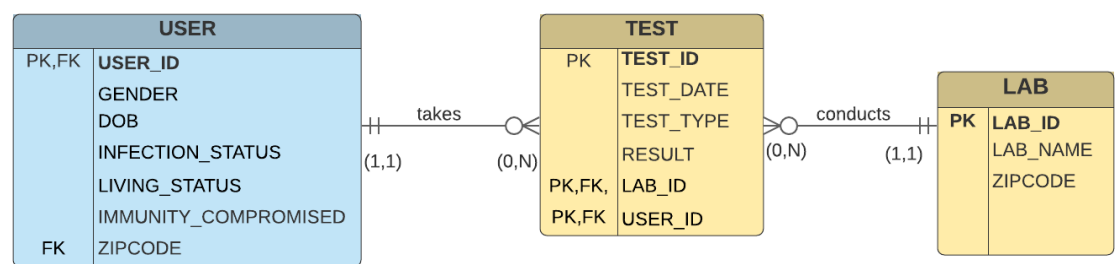
An APP\_USER can be a HEALTH\_TRACK\_ADMIN and each HEALTH\_TRACK\_ADMIN is an app user. Strong relationships exist between APP\_USER and HEALTH\_TRACK\_ADMIN entities. There is a 1:1 relationship between APP\_USER and HEALTH\_TRACK\_ADMIN. The cardinalities are represented by writing (1,1) next to APP\_USER and (0,1) next to HEALTH\_TRACK\_ADMIN. There can be an APP\_USER, who is not a HEALTH\_TRACK\_ADMIN. Therefore HEALTH\_TRACK\_ADMIN is optional for APP\_USER.

APP\_USER to USER:



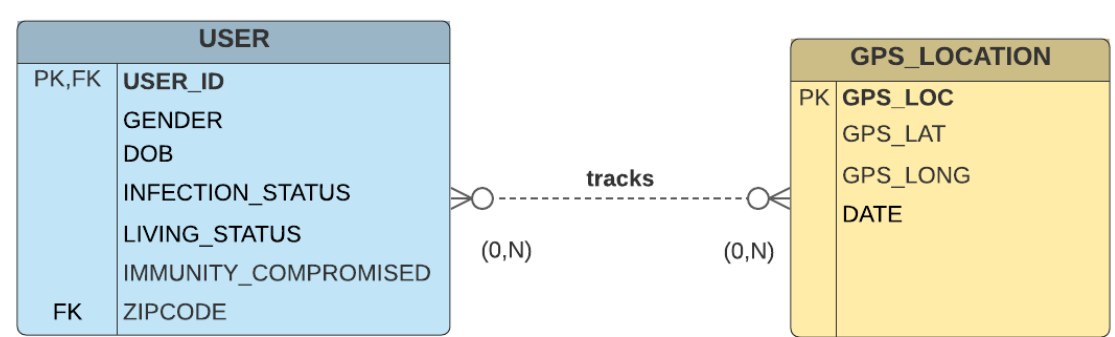
An APP\_USER can be a USER and each USER belongs to an app user. Strong relationships exist between APP\_USER and USER entities. There is a 1:1 relationship between APP\_USER and USER. The cardinalities are represented by writing (1,1) next to APP\_USER and (0,1) next to USER. There can be an APP\_USER, who is not a USER. Therefore USER is optional for APP\_USER.

USER to LAB:

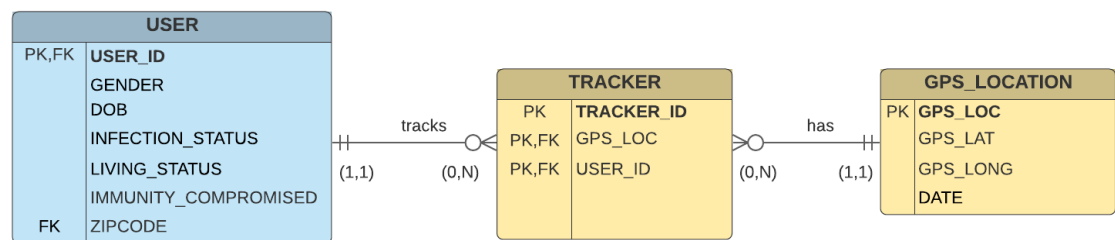


A USER can take many tests and each TEST is carried out on one user. LAB can conduct many tests and each TEST is carried out in one lab. Since there exists an M:N relationship between PERSON and LAB, we build a composite entity TEST to show 1:M relationship with the parent entities PERSON and LAB. Here TEST entity is optional for both TEST and LAB.

USER to GPS\_LOCATION:



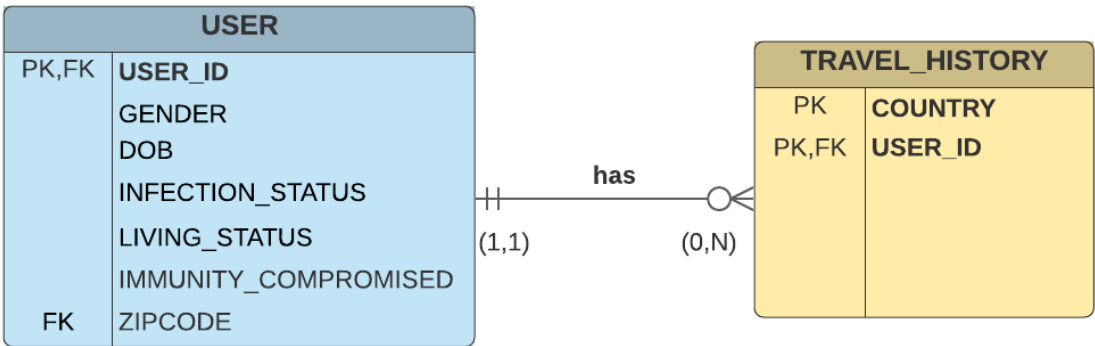
There exists an M:N relationship between USER entity and GPS\_LOCATION entity, so we define an associative entity TRACKER to show 1:M relationship with parent entities USER and GPS\_Location.



A USER can have many tracking GPS locations and each GPS\_LOCATION is tracked by many user. Here TRACKER is a weak entity and it is optional for both USER and GPS\_LOCATION. Strong relationship exists between USER and GPS\_LOCATION associated with TRACKER entity.

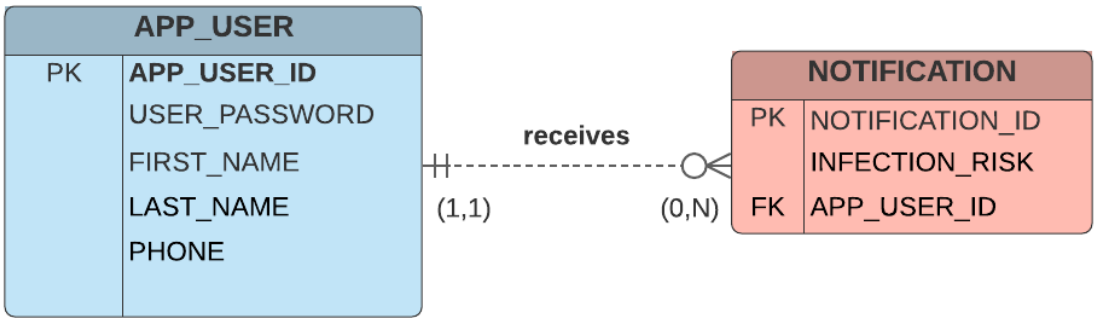


USER to TRAVEL\_HISTORY:



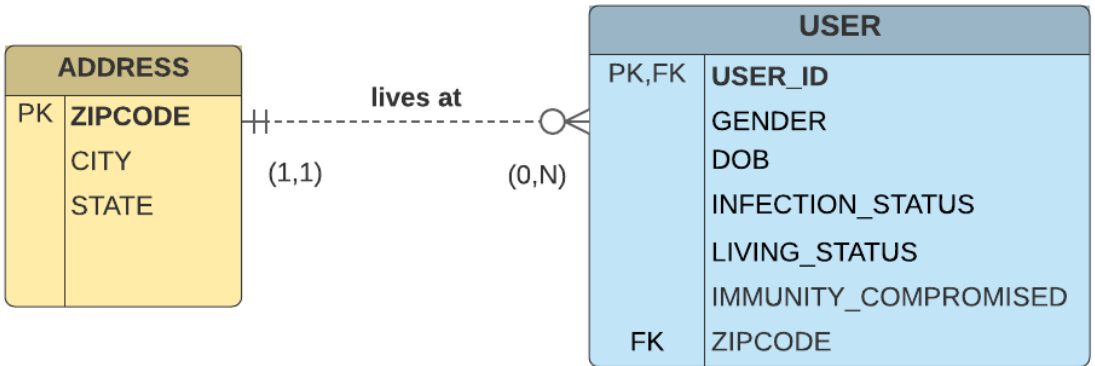
A USER can have many travel histories and each TRAVEL HISTORY belongs to one USER. There exists 1:M relationship between USER and TRAVEL\_HISTORY. The cardinalities are represented by writing (1,1) next to USER and (0,N) next to TRAVEL\_HISTORY. There can be a USER who does not have TRAVEL\_HISTORY. Therefore TRAVEL\_HISTORY is optional for USER.

APP\_USER to NOTIFICATIONS:



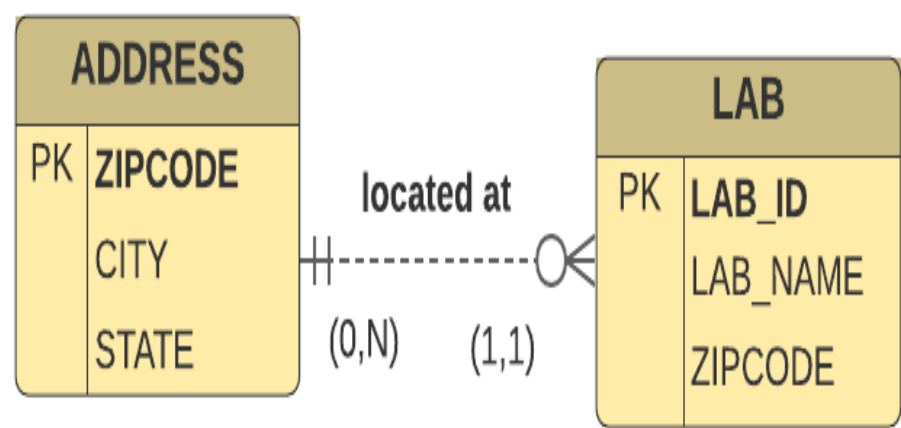
An APP\_USER can receive many notifications and each NOTIFICATION is received by one app user. There can be an APP\_USER, who does not receive any NOTIFICATION. Therefore, NOTIFICATION is optional for APP\_USER. The cardinalities are represented by (1,1) next to APP\_USER and (0,N) next to NOTIFICATION.

ADDRESS to USER:



A USER lives at one address and an ADDRESS can be assigned to many user.Weak relationship exists between USER and ADDRESS entities. There is a 1:M relationship between ADDRESS and USER. The cardinalities are represented by writing (1,1) next to ADDRESS and (0,N) next to USER. There can be an ADDRESS, which is not assigned to any user. Therefore USER is optional for ADDRESS.

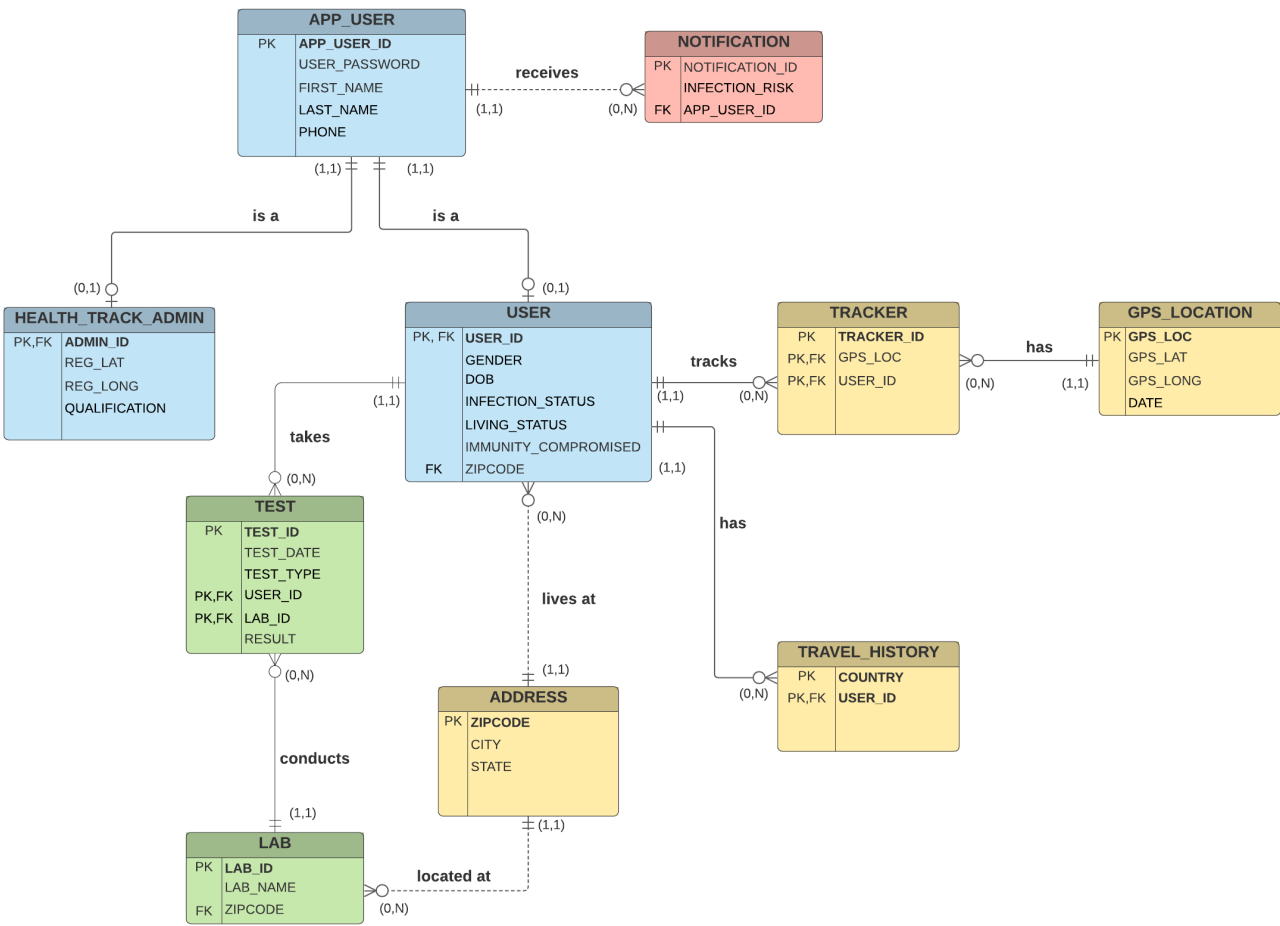
**ADDRESS to LAB:**



A LAB is located at a particular address and each ADDRESS belongs to many labs.There exists a 1:M relationship between ADDRESS and LAB.The cardinalities are represented by writing (1,1) next to ADDRESS and (0,N) next to LAB. There can be an ADDRESS which is not associated with any LAB.Therefore LAB is optional for ADDRESS.

**Logical Model**

# we can add data dictionary as well



Normalization and schemas:

Relational Schema

The Relational schema is developed from the logical model. The relation with the attributes is represented below,where the primary key attributes are written in bold and underlined.

APP\_USER(**APP\_USER\_ID**, USER\_PASSWORD, FIRST\_NAME, LAST\_NAME, PHONE)

HEALTH\_TRACK\_ADMIN (**ADMIN\_ID**, REG\_LAT, REG\_LONG, QUALIFICATION)

USER (**USER\_ID**, GENDER, DOB, INFECTION\_STATUS,LIVING\_STATUS,IMMUNITY\_COMPROMISED, ZIP\_CODE )

NOTIFICATION(**NOTIFICATION\_ID**,INFECTION\_RISK,APP\_USER\_ID)

GPS\_LOCATION(**GPS\_LOC**,GPS\_LAT,GPS\_LONG,DATE)

TRACKER( **TRACKER\_ID**, GPS\_LOC,USER\_ID)

TRAVEL\_HISTORY (**COUNTRY**, **USER\_ID**)

TEST (TEST\_ID, TEST\_TYPE, TEST\_DATE,USER\_UD, LAB\_ID, RESULT)

LAB (LAB\_ID, LAB\_NAME, ZIPCODE)

ADDRESS (ZIPCODE, CITY, STATE)

Constraints :

Primary key should be unique and not null

User should have Unique Phone Number

Infection Status: {infected,suspected,not infected}

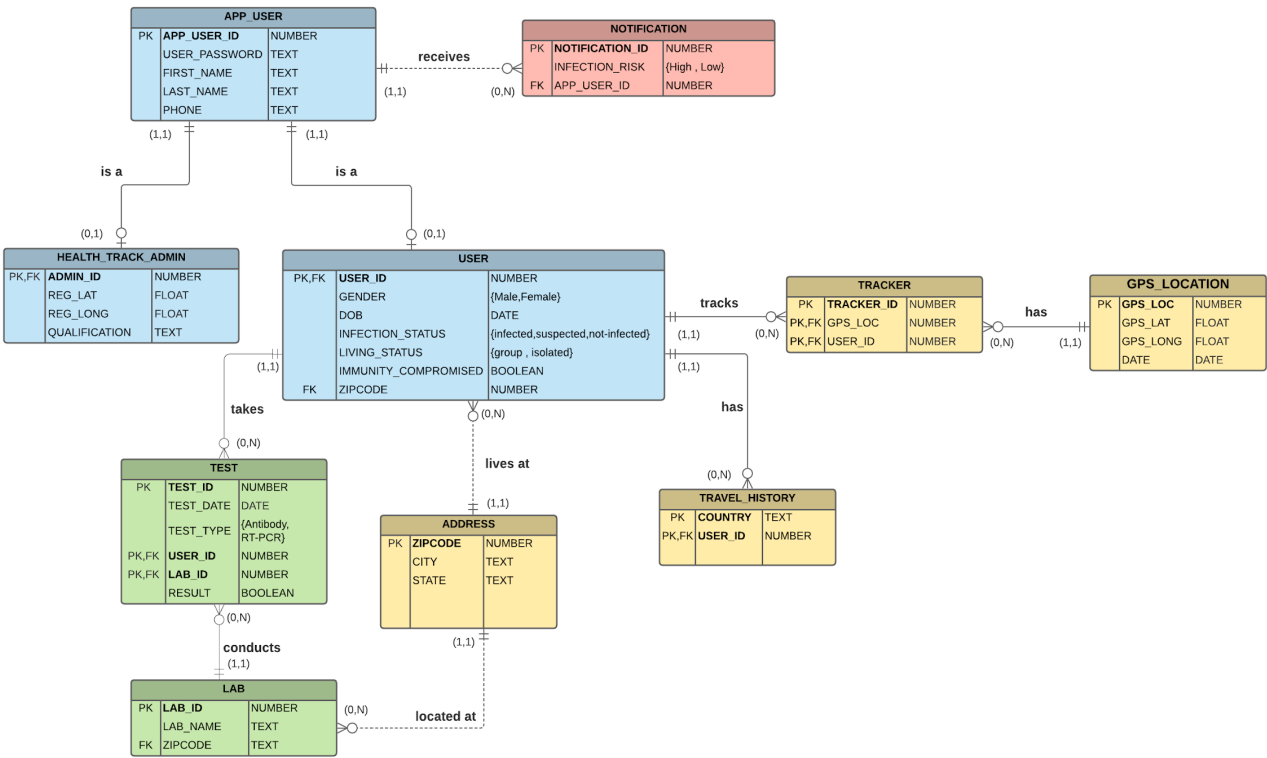
Living Status: {group,isolated}

Infection Risk: {high,low}

Test Type: {Antibody,RT-PCR}

Physical design

in the physical design = we will add all the access control(example our 225 project(IAM), dataset, tables access) like that



DBMS Selection

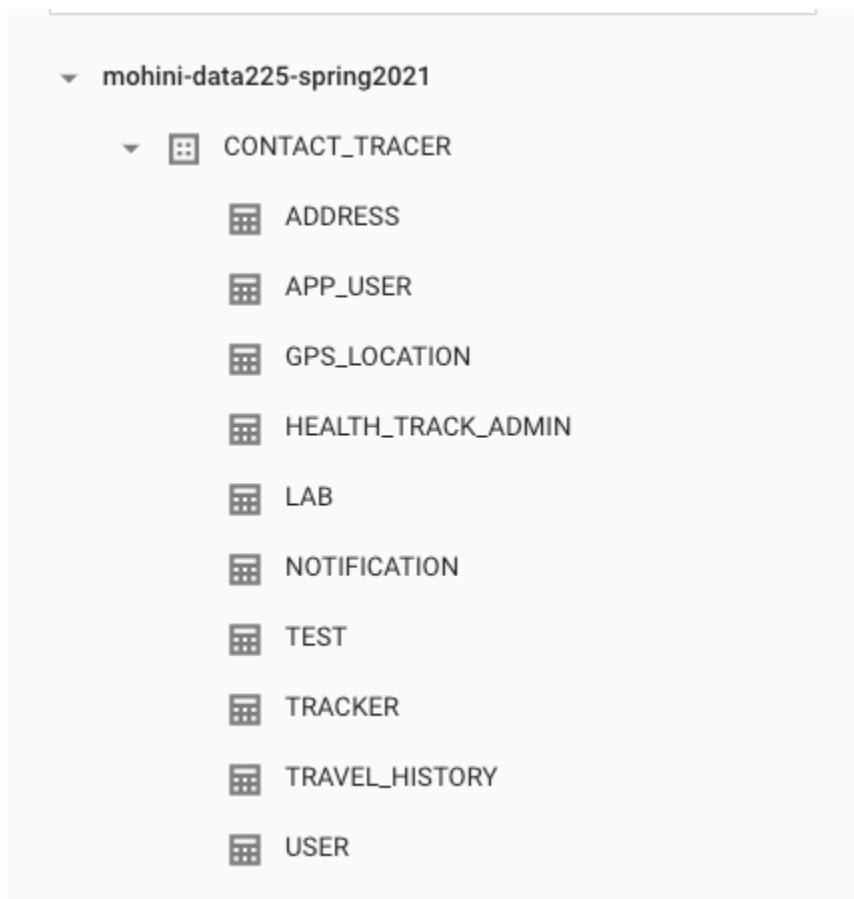
- DBMS analysis
- DBMS selection
- Hardware & Software

## Implementation & Loading

- **Schema creation**

Created project on Google Cloud Platform

Created dataset ‘CONTACT\_TRACER’ and all the required tables on BigQuery



Create statements

- **Data loading and conversion**

Insert statements

Dummy data inserted using <http://www.generatedata.com/>

## Testing & Evaluation

Constraint, CRUD operations

## Query Descriptions

## Conclusion

This database system can be used to build mobile applications.

## Future Work :

### \*M Challenges

There are societal fears of privacy breaches, provision of wrong medical advice based on self-reported symptoms, and the systematic exclusion of some members of society who cannot access the system. So, the need to have sufficient regulatory oversight and ethical issues surrounding privacy, security, transparency, and accountability.

As well, there is the possibility that marginalized and disadvantaged groups particularly those in the low middle-income will be more likely to be excluded. Again, the use of the system will incur costs such as hardware, and software costs, training costs, and require continuous user support.

Moreover, there is currently limited evidence to evaluate the effectiveness of digital tools for COVID-19 response like such a proposed system, despite several countries and areas having deployed. So should not be considered as 'stand-alone solutions' for contact tracing but rather as complementary tools for manual contact tracing.

