

DATAWAREHOUSE PROJECT REPORT: Improved Patient Treatment through Smart Healthcare

Team Members:

Syed Saddat Raza – 14962

Muhammad Sadiq Vayani – 14827

Acquiring the business knowledge:

I went through multiple platforms for acquiring business knowledge related to smart healthcare which includes YouTube, Wikipedia, Twitter, different websites and annual reports as well. I gained knowledge that how in the world we are using smart technology and real time transforming of data for analysis or for action to be taken which is vital for Health sector. I learned about different Smart Healthcare system currently working in good hospitals and in developing country which is very helpful for patients living in those part of the world.

Further the important knowledge I acquired, is about the use of AI, robotics and online surgery which is still far from current real world right now, but it is the next aim for smart Healthcare. It focuses on that with minimum human / doctor's intervention patient can be treated/ cured.

Therefore, in case of any emergency or more patients arriving to Hospital just like in corona times people can be cured easily. Further analytics being performed on recent patients' history with same disease can be helpful to patients and they can be better at decision making.

Assumption and Knowledge used in designing ERD:

I have taken many assumptions and drilled down my business knowledge. My ERD revolves around that all patients' current sensors are being monitored using smart healthcare device and in case of any threat, device vibrates and give notification so person can precaution steps. Further through this ERD our knowledge is that all data regarding which doctor visiting which hospital, his whole history and reputation, currently treating which patients. Further also showing ambulance details related to each hospital. So, patient can make good decision regarding their healthcare. In ERD we have also assumed higher authorities will have access of all data like which patient being treated his whole medical history and real-time data, further knowledge of which hospital and doctors' approach is good. Further data scientist can do Data analytics on being data generated which help hospital, authorities, doctors, and patients to take better steps in future. According to my assumption patient, doctors, hospitals and higher authority will be using

some smart healthcare app to access their respective data and make smart decision regarding it and should be well informed.

Data Generation/Acquiring Process:

I have generated my own data according to my ERD / dimension modelling using **PYTHON Jupyter Notebook**.

I have looped my data according to number of rows I used random method to generate random numbers , created one function and manipulated data to generate random dates according to my Minimum or maximum date.

I have created List for string row, write some 10-20 string values according to my thinking and then by using loop among these 20 values data being generated randomly into these rows.

Just like in below python code I have generated random 1000 values through loop for Ambulance_Req table.

- **Pid** generated between 0 and 999 using random.randint function, so data randomly generated from 0 to 999 in Pid column.
- **Datetime** generated for dates between “2020-1-1” to “2020-12-1”, using different variable of startdate, end date, days_between_dates and by using randrange and timedelta function to generate random dates between date specified.
- **Longitude** values generated between -180 and 181 using randint function to generate random values for Longitude column.
- **Latitude** values generated between -80 and 81 using randint function to generate random values for Latitude column.
- **Type** column name which was early named as **Is_emergency** have random value of 0 or 1 to be randomly generated for type column either 0 or 1.

Data.append used to put all row values in sequence with Pid as first value.

Pd.DataFrame to convert list of data to data-frame simultaneously

```

data = []
flag = ['requested','automatically triggered']

for i in range(1000):
    start_date = datetime.date(2020, 1, 1)
    end_date = datetime.date(2020, 12, 1)
    time_between_dates = end_date - start_date
    days_between_dates = time_between_dates.days
    random_number_of_days = random.randrange(days_between_dates)
    random_date = start_date + datetime.timedelta(days=random_number_of_days)
    pid = random.randint(0,9999)
    longitude = random.randint(-180,181)
    latitude = random.randint(-80,81)

    hours = random.randint(2,10)

    is_emergency = flag[random.randint(0,1)]

    data.append([pid, random_date, longitude, latitude, is_emergency])
    ambulance_req = pd.DataFrame(data,columns = ['P_ID', 'datetime', 'longitude', 'latitude','type'])

```

```

ambulance_req

```

```

:

```

	P_ID	datetime	longitude	latitude	type
0	500	2020-01-02	-157	63	automatically triggered
1	8747	2020-05-21	-154	-15	automatically triggered
2	3334	2020-09-16	76	33	requested
3	2177	2020-11-12	112	-47	automatically triggered
4	2195	2020-01-21	26	-80	requested
...
995	4781	2020-08-19	-141	-64	requested

Like above I have generated one of my table Ambulance_Req, I have also generated my other tables using same Python technique.

Now in the I have converted my Dataframe for all tables to CSV file using code below:

```

disease.to_csv(r'disease.csv', index = False)

doctors.to_csv(r'doctors.csv',index=False)
patients.to_csv(r'patients.csv',index=False)
tests.to_csv(r'tests.csv',index=False)
patients_test.to_csv(r'patients-test.csv',index=False)
ambulance_req.to_csv(r'ambulance_req.csv',index=False)

```

Hive Commands Executed:

I have uploaded my data to Hive. My repo link I have given on lms.

Below showing screenshot of queries being generated:

The screenshot shows the Hue Hive Editor interface. The query editor contains the following SQL query:

```
1 SELECT * FROM patient_test,patients where patient_test.patient_id=patients.patient_id;
```

Below the query editor, there are buttons for "Execute", "Save as...", "Explain", "Format", and "New query". The "Results" tab is selected, displaying a table with 7 rows and 6 columns:

	patient_test.patient_id	patient_test.test_id	patient_test.date	patient_test.result	patients.patient_id	patients.patient_name
1	0	2	2020-04-11	postive	0	eren
2	0	3	2020-09-20	negative	0	eren
3	0	3	2020-07-21	postive	0	eren
4	3	3	2020-09-09	negative	3	Talha
5	4	0	2020-01-04	postive	4	Daniyal
6	5	4	2020-08-28	negative	5	Askari
7	7	1	2020-04-02	negative	7	Muneeb

The screenshot shows the Hue Hive Editor interface. The query editor contains the following SQL query:

```
1 SELECT COUNT(*) FROM patients;
```

Below the query editor, there are buttons for "Execute", "Save as...", "Explain", "Format", and "New query". The "Results" tab is selected, displaying a table with 1 row and 1 column:

	_c0
1	69663

Activities Google Chrome Mon 13:57

Hue-Hive-Query x hive hue cant find x Solved: User Cloud x Sakai: DATAWARE x (103) WhatsApp x Hadoop Mapred x

localhost:8889/beeswax/execute/query/24#query/results

HUE Query Editors Data Browsers Workflows Search Security

Hive Editor Query Editor My Queries Saved Queries History

Assist Settings

default

Tables (6)

- ambulance_req
- disease
- doctors
- patient_test
- patients
- test

```
1 SELECT COUNT(*) FROM patient_test;
```

Execute Save as... Explain Format or create a New query

Recent queries Query Log Columns Results Chart

	_c0
1	10000

Activities Google Chrome Mon 14:00

Hue-Hive-Query x full outer join x SQL FULL OUT x Solved: User Cloud x Sakai: DATAWARE x (104) WhatsApp x Hadoop Mapred x

localhost:8889/execute/query/27#query/results

HUE Query Editors Data Browsers Workflows Search Security

Hive Editor Query Editor My Queries Saved Queries History

Assist Settings

default

Tables (6)

- ambulance_req
 - datetime (string)
 - latitude (smallint)
 - longitude (smallint)
 - p_id (smallint)
 - type (string)
- disease
- doctors
- patient_test
- patients
- test

```
1 SELECT * FROM ambulance_req FULL OUTER JOIN patients ON patients.patient_id=ambulance_req.p_id;
```

Execute Save as... Explain Format or create a New query

Recent queries Query Log Columns Results Chart

	latitude	ambulance_req.type	patients.patient_id	patients.patient_name	patients.email	patients.age	patients.gender
1		NULL	0	eren	Testemal0@gmail.com	56	M
2		NULL	1	Sam	Testemal1@gmail.com	41	F
3		NULL	2	Sam	Testemal2@gmail.com	27	M
4		NULL	3	Taiha	Testemal3@gmail.com	66	M
5		NULL	4	Daniyal	Testemal4@gmail.com	62	M
6		NULL	5	Askari	Testemal5@gmail.com	29	F
7		NULL	6	Moiz	Testemal6@gmail.com	61	F

localhost:8889/accounts/logout/

Activities Google Chrome Mon 14:02

localhost:8889/beeswax/execute/query/29#query/results

HUE Query Editors Data Browsers Workflows Search Security

Hive Editor Query Editor My Queries Saved Queries History

Assist Settings

Tables (6)

- ambulance_req
 - datetime (string)
 - latitude (smallint)
 - longitude (smallint)
 - p_id (smallint)
 - type (string)
- disease
 - disease_id (bigint)
 - disease_name (string)
 - disease_type (string)
 - recovery_rate (smallint)
- doctors
 - doctor_contact_no (int)
 - doctor_id (bigint)
 - doctor_name (string)
 - specialty (string)
- patient_test
 - date (string)
 - patient_id (smallint)
 - result (string)
 - test_id (smallint)

```
1 SELECT * FROM disease;
```

Execute Save as... Explain Format or create a New query

Recent queries Query Log Columns Results Chart

	disease.disease_id	disease.disease_name	disease.disease_type
1	0	Cancer	*Mesothelioma is associated with simian virus 40
2	1	Diabetes mellitus type 1	Seasonal affective disorder is associated with Epstein-Barr virus.
3	2	Sarcoidosis	Stomach cancer is associated with the bacterium Helicobacter pylori.
4	3	Cancer	Oropharyngeal cancer can be caused by human papillomaviruses.
5	4		*Liver cancer. Hepatocellular carcinoma can be caused by hepatitis B virus
6	5	Cancer	
7	6	Cancer	Leukemia. Adult T-cell leukemia can be caused by human T-cell leukemia virus-1.

Activities Google Chrome Mon 14:04

localhost:8889/beeswax/execute/query/30#query/results

HUE Query Editors Data Browsers Workflows Search Security

Hive Editor Query Editor My Queries Saved Queries History

Assist Settings

Tables (6)

- ambulance_req
 - datetime (string)
 - latitude (smallint)
 - longitude (smallint)
 - p_id (smallint)
 - type (string)
- disease
 - disease_id (bigint)
 - disease_name (string)
 - disease_type (string)
 - recovery_rate (smallint)
- doctors
 - doctor_contact_no (int)
 - doctor_id (bigint)
 - doctor_name (string)
 - specialty (string)
- patient_test
 - date (string)
 - patient_id (smallint)
 - result (string)
 - test_id (smallint)

```
1 SELECT COUNT(*) FROM doctors;
```

Execute Save as... Explain Format or create a New query

Recent queries Query Log Columns Results Chart

	_c0
1	600

Activities Google Chrome Mon 14:07

Hue-Hive-Query x full outer join - Go x SQL FULL OUTER x Solved: User Cloud x Sakai: DATAWARE x Hadoop Mapred... x

localhost:8889/beeswax/execute/query/31#query/results

HUE Query Editors Data Browsers Workflows Search Security

Hive Editor Query Editor My Queries Saved Queries History

Assist Settings

default

Tables (6)

- ambulance_req
 - datetime (string)
 - latitude (smallint)
 - longitude (smallint)
 - p_id (smallint)
 - type (string)
- disease
 - disease_id (bigint)
 - disease_name (string)
 - disease_type (string)
 - recovery_rate (smallint)
- doctors
 - doctor_contact_no (int)
 - doctor_id (bigint)
 - doctor_name (string)
 - specialty (string)
- patient_test
 - date (string)
 - patient_id (smallint)
 - result (string)
 - test_id (smallint)

```
1 SELECT * FROM patients FULL OUTER JOIN doctors on patients.doctor_id=doctors.doctor_id;
```

Execute Save as... Explain Format or create a New query

Recent queries Query Log Columns Results Chart

	patients.patient_id	patients.patient_name	patients.email	patients.age	patients.gender	patients.contactnumber	pat
1	53012	Salman	Testemai53012@gmail.com	33	F	3445289889	C
2	65991	Baqar	Testemai65991@gmail.com	63	M	3421089389	C
3	2393	Askari	Testemai2393@gmail.com	73	F	3464455775	C
4	46130	Salman	Testemai46130@gmail.com	34	M	3643576533	C
5	67537	Falco	Testemai67537@gmail.com	60	M	3027965814	C
6	28224	Falco	Testemai28224@gmail.com	25	M	3112891535	C
7	931	Daniyal	Testemai931@gmail.com	70	F	3699839715	C

Activities Google Chrome Mon 14:07

Hue-Hive-Query x full outer join - Go x SQL FULL OUTER x Solved: User Cloud x Sakai: DATAWARE x Hadoop Mapred... x

localhost:8889/beeswax/execute/query/32#query/results

HUE Query Editors Data Browsers Workflows Search Security

Hive Editor Query Editor My Queries Saved Queries History

Assist Settings

default

Tables (6)

- ambulance_req
 - datetime (string)
 - latitude (smallint)
 - longitude (smallint)
 - p_id (smallint)
 - type (string)
- disease
 - disease_id (bigint)
 - disease_name (string)
 - disease_type (string)
 - recovery_rate (smallint)
- doctors
 - doctor_contact_no (int)
 - doctor_id (bigint)
 - doctor_name (string)
 - specialty (string)
- patient_test
 - date (string)
 - patient_id (smallint)
 - result (string)
 - test_id (smallint)

```
1 SELECT * FROM patients;
```

Execute Save as... Explain Format or create a New query

Recent queries Query Log Columns Results Chart

	patients.patient_id	patients.patient_name	patients.email	patients.age	patients.gender	patients.contactnumber	patient
1	0	eren	Testemai0@gmail.com	56	M	3204329069	239
2	1	Sam	Testemai1@gmail.com	41	F	3042336837	399
3	2	Sam	Testemai2@gmail.com	27	M	3234201891	152
4	3	Talha	Testemai3@gmail.com	66	M	3013851539	237
5	4	Daniyal	Testemai4@gmail.com	62	M	3578759119	531
6	5	Askari	Testemai5@gmail.com	29	F	3158960425	306
7	6	Moiz	Testemai6@gmail.com	61	F	3648128578	432

Activities Google Chrome Mon 14:09

localhost:8889/beeswax/execute/query/33#query/results

HUE Query Editors Data Browsers Workflows Search Security

Hive Editor Query Editor My Queries Saved Queries History

Assist Settings

Tables (6) Q

- ambulance_req
 - datetime (string)
 - latitude (smallint)
 - longitude (smallint)
 - p_id (smallint)
 - type (string)
- disease
 - disease_id (bigint)
 - disease_name (string)
 - disease_type (string)
 - recovery_rate (smallint)
- doctors
 - doctor_id (bigint)
 - doctor_name (string)
 - specialty (string)
- patient_test
 - date (string)
 - patient_id (smallint)
 - result (string)
 - test_id (smallint)
- patients

```
1 SELECT COUNT(*) from ambulance_req;
```

Execute Save as... Explain Format or create a New query

Recent queries Query Log Columns Results Chart

	_c0
1	1000