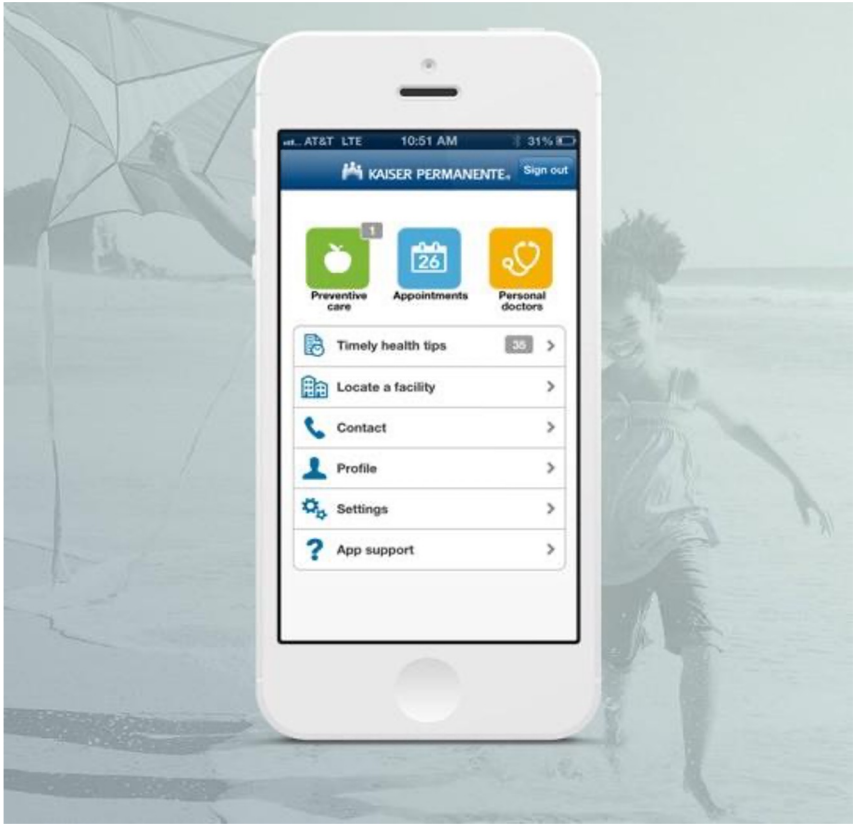
DBMS PROJECT

Team members:
Jhaanvi Golani
Shaheen Nazar
Pallavi Saboo
Komal Srivastava
Kalyani Ravi

Agenda

- Business Scenario
- Swim Lane Diagram
- ER Diagram
- SQL Queries
- Advanced SQL Queries
- Visualization
- Conclusion



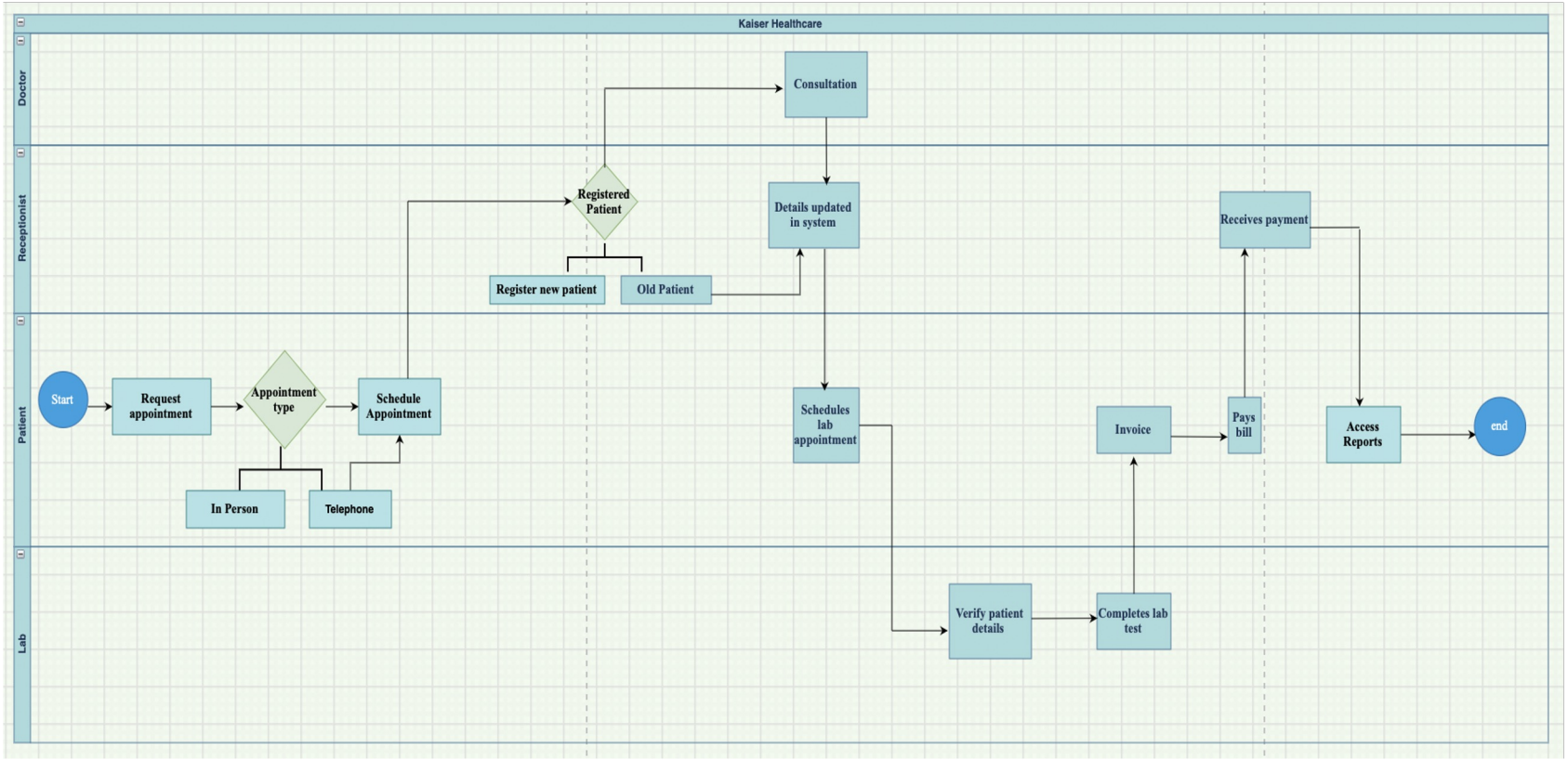
Business Scenario

Ramie got an admit from SCU and is required to upload immunization records. Her healthcare provider is Kaiser. She wants to consult a doctor and get all the immunization requirements fulfilled. So, she books a telephone consultation appointment for 5:30 PM via the Kaiser app. The system updates the doctor's calendar and blocks the 5:30 PM slot for Ramie's consultation. The doctor calls Ramie at the scheduled time. Ramie explains the immunization requirements. The doctor ordered five lab tests on the Kaiser portal based on the patient's medical history. After this call, Ramie gets another call from Kaiser to schedule lab tests. The caller asks for a preferred location and time slot. Ramie scheduled the appointment for the next day at 10:30 AM.

The next day at the lab reception, Ramie shows her confirmation details on the app. Ramie gets a token number. The receptionist asks Ramie to wait until an automated system announces her token number. Soon, the automated system announces Ramie's token number and asks her to go to lab station 5. After reaching the lab station, Ramie shows her Kaiser ID to the lab official. The lab official confirms the appointment details in the system and conducts lab test.

After two days, Ramie receives an email stating that her reports are ready along with the payment link. She makes the payment and receives her lab test results on the registered email ID. The lab results are also recorded under 'Medical Record' in the Kaiser database.

Swim Lane Diagram



ER Diagram

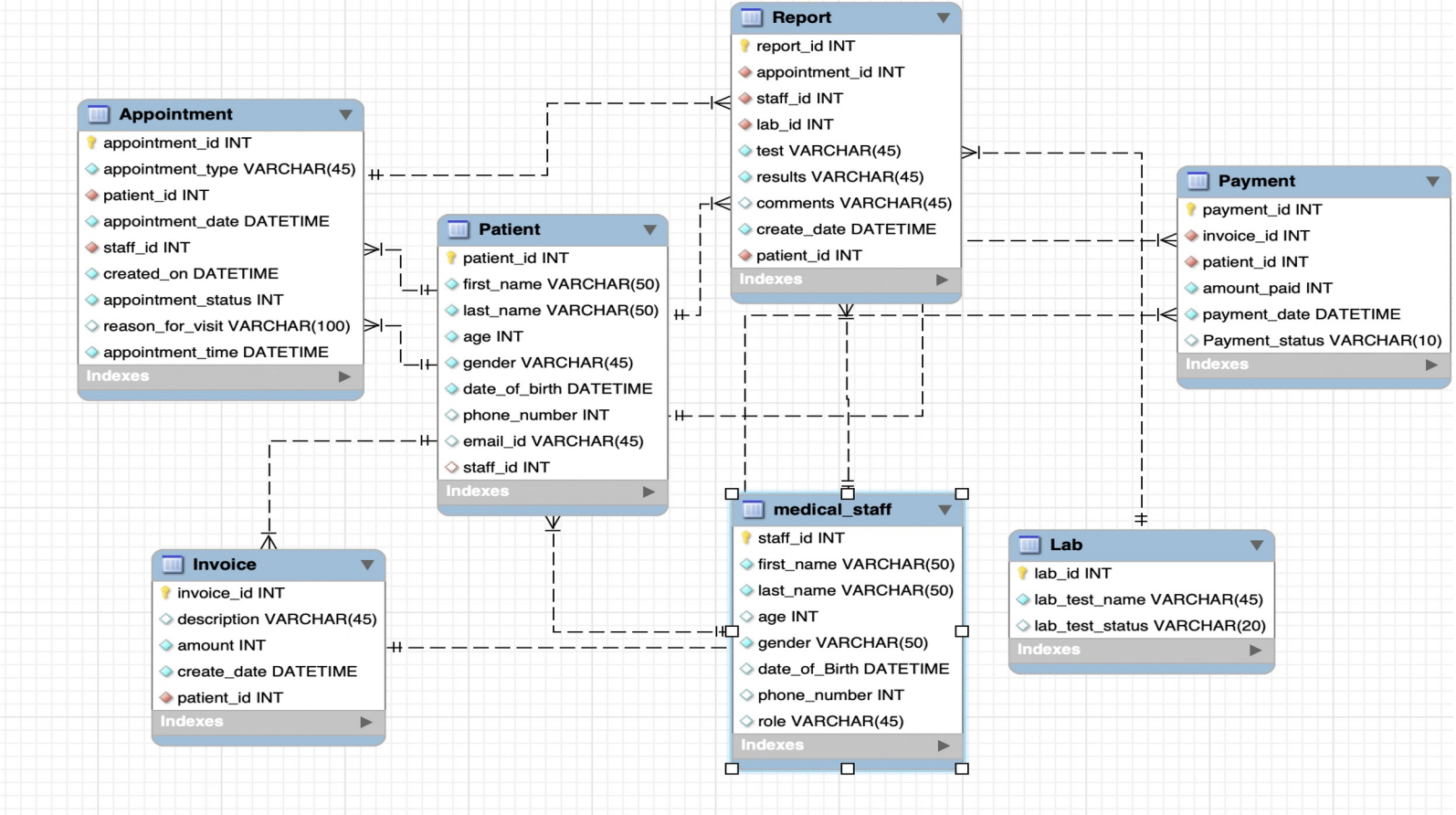

Report
report_id INT
appointment_id INT
staff_id INT
lab_id INT
test VARCHAR(45)
results VARCHAR(45)
comments VARCHAR(45)
create_date DATETIME
patient_id INT
Indexes
Appointment
appointment_id INT
appointment_type VARCHAR(45)
patient_id INT
appointment_date DATETIME
staff_id INT
created_on DATETIME
appointment_status INT
reason_for_visit VARCHAR(100)
appointment_time DATETIME
Indexes
Patient
patient_id INT
first_name VARCHAR(50)
last_name VARCHAR(50)
age INT
gender VARCHAR(45)
date_of_birth DATETIME
phone_number INT
email_id VARCHAR(45)
staff_id INT
Indexes
Payment
payment_id INT
invoice_id INT
patient_id INT
amount_paid INT
payment_date DATETIME
Payment_status VARCHAR(10)
Indexes
Invoice
invoice_id INT
description VARCHAR(45)
amount INT
create_date DATETIME
patient_id INT
Indexes
medical_staff
staff_id INT
first_name VARCHAR(50)
last_name VARCHAR(50)
age INT
gender VARCHAR(50)
date_of_Birth DATETIME
phone_number INT
role VARCHAR(45)
Indexes
Lab
lab_id INT
lab_test_name VARCHAR(45)
lab_test_status VARCHAR(20)
Indexes

Table creation

```
20
21  CREATE TABLE IF NOT EXISTS medical_staff (
22      staff_id INT NOT NULL,
23      first_name VARCHAR(50) NOT NULL,
24      last_name VARCHAR(50) NOT NULL,
25      age INT NOT NULL,
26      gender VARCHAR(50) NOT NULL,
27      date_of_birth DATE,
28      phone_number VARCHAR(45),
29      role VARCHAR(45),
30      reporting_to_id VARCHAR(45),
31      PRIMARY KEY(staff_id)
32      );
33
```

Table creation

```
7
8  CREATE TABLE IF NOT EXISTS patient (
9  patient_id INT NOT NULL,
10 first_name VARCHAR(50) NOT NULL,
11 last_name VARCHAR(50) NOT NULL,
12 age INT NOT NULL ,
13 gender VARCHAR(45) NOT NULL,
14 date_of_birth DATE NOT NULL,
15 phone_number VARCHAR(45),
16 email_id VARCHAR(45),
17 staff_id INT NOT NULL,
18 PRIMARY KEY(patient_id),
19 FOREIGN KEY(staff_id) REFERENCES medical_staff(staff_id)
20  );
21
```

Table creation

```
35
36  CREATE TABLE IF NOT EXISTS appointment (
37      appointment_id INT NOT NULL,
38      patient_id INT NOT NULL,
39      appointment_type VARCHAR(15) NOT NULL,
40      appointment_date DATE NOT NULL,
41      appointment_time TIME NOT NULL,
42      created_on DATE NOT NULL,
43      reason_for_visit VARCHAR(100),
44      appointment_status VARCHAR(45) NOT NULL,
45      staff_id INT NOT NULL,
46      PRIMARY KEY(appointment_id),
47      FOREIGN KEY(patient_id) REFERENCES patient(patient_id),
48      FOREIGN KEY(staff_id) REFERENCES medical_staff(staff_id)
49      );
50
```

Table creation

```
51
52  CREATE TABLE IF NOT EXISTS lab(
53      lab_id INT NOT NULL,
54      immunization_test_name VARCHAR(45) NOT NULL,
55      patient_id INT NOT NULL,
56      blood_work_status VARCHAR(20),
57      PRIMARY KEY(lab_id),
58      FOREIGN KEY(patient_id) REFERENCES patient(patient_id)
59      );
60
```

```
74
75  CREATE TABLE IF NOT EXISTS invoice(
76      invoice_id INT NOT NULL,
77      patient_id INT NOT NULL,
78      description VARCHAR(45),
79      amount INT NOT NULL,
80      create_date DATETIME,
81      invoice_status VARCHAR(45),
82      PRIMARY KEY(invoice_id),
83      FOREIGN KEY(patient_id) REFERENCES patient(patient_id)
84      );
85
```

Table creation

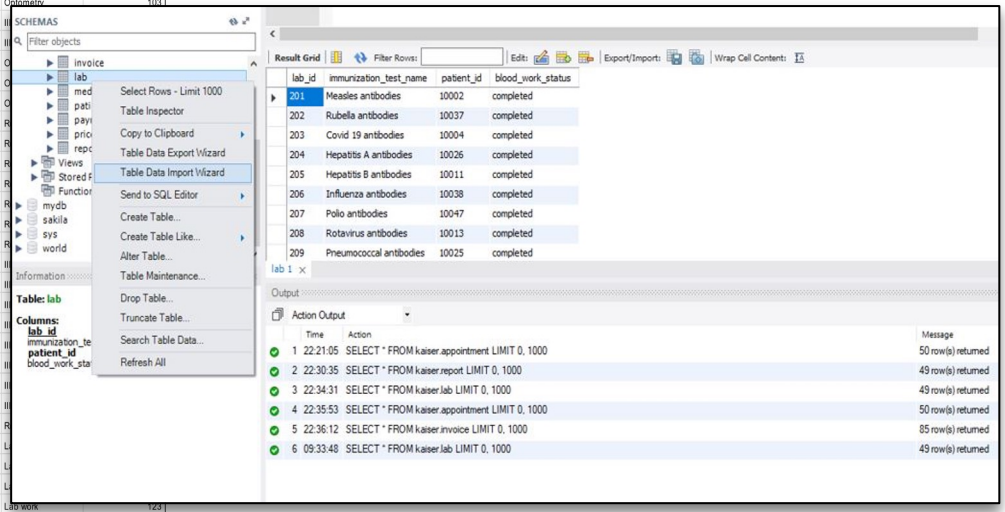
```
60
61  CREATE TABLE IF NOT EXISTS report(
62      report_id INT NOT NULL,
63      staff_id INT NOT NULL,
64      lab_id INT NOT NULL,
65      immunization_test_name VARCHAR(45),
66      create_date DATE NOT NULL,
67      patient_id INT NOT NULL,
68      results VARCHAR(45),
69      PRIMARY KEY(report_id),
70      FOREIGN KEY(staff_id) REFERENCES medical_staff(staff_id),
71      FOREIGN KEY(lab_id) REFERENCES lab(lab_id),
72      FOREIGN KEY(patient_id) REFERENCES patient(patient_id)
73      );
74
```

Table creation

```
85
86  CREATE TABLE IF NOT EXISTS payment(
87      payment_id INT NOT NULL,
88      patient_id INT NOT NULL,
89      invoice_id INT NOT NULL,
90      payment_status VARCHAR(10),
91      payment_date DATETIME,
92      amount_paid INT,
93      PRIMARY KEY(payment_id),
94      FOREIGN KEY(invoice_id) REFERENCES invoice(invoice_id),
95      FOREIGN KEY(patient_id) REFERENCES patient(patient_id)
96      );
97
```

Importing data : CSV File

appointment_id	patient_id	appointment_type	appointment_date	appointment_time	created_on	appointment_status	reason_for_visit	staff_id
1	10002	Telephone	2021-10-22	12:09:24	2021-10-15	0	Lab work	101
2	10007	Telephone	2021-10-25	18:34:07	2021-10-18	1	Routine checkup	102
3	10014	Telephone	2021-10-29	18:48:36	2021-10-22	0	Optometry	103
4	10003	In-person	2021-10-05	13:57:16	2021-10-04	1	Il	
5	10023	In-person	2021-11-08	14:46:59	2021-10-07	0	Il	
6	10031	In-person	2021-11-09	14:52:03	2021-10-29	1	C	
7	10038	In-person	2021-11-11	12:45:01	2021-11-01	0	C	
8	10021	Telephone	2021-11-15	20:31:15	2021-11-02	1	C	
9	10023	Telephone	2021-11-17	21:45:59	2021-11-03	0	R	
10	10003	Telephone	2021-11-19	16:15:18	2021-11-05	1	R	
11	10020	Telephone	2021-11-23	19:31:52	2021-11-09	0	R	
12	10040	In-person	2021-11-25	20:25:45	2021-11-22	1	R	
13	10002	In-person	2021-11-26	20:35:38	2021-11-25	0	R	
14	10022	In-person	2021-11-29	15:08:38	2021-11-12	1	R	
15	10006	In-person	2021-12-01	14:32:48	2021-11-29	0	R	
16	10024	In-person	2021-11-03	14:33:22	2021-10-01	1	Il	
17	10032	In-person	2021-11-07	20:19:24	2021-10-01	0	Il	
18	10019	In-person	2021-12-10	17:24:33	2021-10-01	1	Il	
19	10036	In-person	2021-12-13	12:02:22	2021-10-01	0	Il	
20	10001	Telephone	2021-12-21	17:50:40	2021-10-01	1	Il	
21	10001	Telephone	2021-12-24	21:17:05	2021-10-01	0	Il	
22	10003	Telephone	2021-12-27	13:38:26	2021-10-01	1	Il	
23	10034	Telephone	2021-12-28	20:27:31	2021-10-01	0	Il	
24	10008	Telephone	2021-12-29	21:17:19	2021-10-01	1	R	
25	10033	Telephone	2021-12-30	19:47:57	2021-10-01	0	L	
26	10024	Telephone	2021-10-15	13:28:24	2021-10-15	1	L	
27	10012	Telephone	2021-10-18	13:50:45	2021-10-18	0	L	
28	10025	Telephone	2021-10-22	17:40:13	2021-10-22	1	Lab work	123
29	10012	Telephone	2021-10-26	18:34:28	2021-10-04	0	Lab work	129
30	10026	Telephone	2021-10-28	18:38:22	2021-10-07	1	Lab work	130
31	10011	Telephone	2021-10-29	13:54:41	2021-10-29	0	Lab work	131
32	10016	Telephone	2021-11-01	17:23:27	2021-10-01	1	Lab work	132
33	10006	Telephone	2021-11-02	20:58:05	2021-10-01	0	Lab work	133
34	10011	Telephone	2021-11-03	21:38:49	2021-10-01	1	Lab work	134



Without Joins:

```
# List of patient ids who have taken an appointment and the reason for visit is routine checkup
SELECT patient_id as 'List of patient IDs' from appointment
where reason_for_visit = 'Routine checkup';
```

List of patient IDs
10007
10023
10003
10020
10040
10002
10022
10006
10008

With joins:

```
Use Kaiser;

# 1 Details of patients where Influenza antibodies were not found(covered inner join and where clause)

select patient.first_name, patient.last_name, patient.gender,
patient.age, patient.phone_number, patient.email_id, report.results, report.immunization_test_name
from patient
inner join report on patient.patient_id = report.patient_id
where immunization_test_name = 'Influenza antibodies' AND results = 'Negative';
```

first_name	last_name	gender	age	phone_number	email_id	results	immunization_test_na...
Noah	Jones	M	66	5234127000	noah@gmail.com	Negative	Influenza antibodies
Liam	Olivia	M	61	5231256789	liam@gmail.com	Negative	Influenza antibodies
Liam	Gonzalez	M	31	5422456665	liam@gmail.com	Negative	Influenza antibodies

```sql
# 2 List patientid, first_name, last_name and total amount paid by each patient.(Using inner join and group by)
Select patient.patient_id, patient.first_name, patient.last_name, Sum(invoice.amount) AS Amount
from patient
inner join invoice on patient.patient_id = invoice.patient_id
Group by patient.patient_id;
```

patient_id	first_name	last_name	Amount
10001	Johnson	Smith	800
10002	Bob	Jones	1100
10003	Ben	Robert	800
10010	Joe	Madison	500
10011	Alice	Bryan	500
10021	Henry	Williams	1300
10039	Chau	Nguyen	1300
10027	Ava	Lee	1500
10028	Sophia	Ralph	1500
10007	Teresa	Austin	1500
10009	Sophia	Roy	1000
10032	Siya	Hall	1500
10034	Michael	Campbell	1000
10036	Rizwaan	Torres	1100
10019	Emma	Smith	800

```sql
# 3 List patient details who have scheduled an appointment and order by patient name.
#Also include details of patients with no appointments( Using Left outer join and order by)
Select patient.first_name, patient.last_name,
appointment.appointment_id, appointment.appointment_type, appointment.appointment_date,
 appointment.appointment_time, appointment.reason_for_visit
 from patient left outer join appointment on
 patient.patient_id = appointment.patient_id
 ORDER BY patient.first_name;
```

first_name	last_name	appointment_id	appointment_ty...	appointment_date	appointment_ti...	reason_for_visit
Alex	Rivera	25	Telephone	2021-12-30	19:47:57	Lab work
Alexander	Johnson	11	Telephone	2021-11-23	19:31:52	Routine checkup
Alexander	Johnson	55	Telephone	2021-12-31	21:23:46	Lab work
Alexander	Johnson	77	In-person	2021-10-29	20:25:45	Lab work
Alice	Bryan	31	Telephone	2021-10-29	13:54:41	Lab work
Alice	Bryan	34	Telephone	2021-11-03	21:38:49	Lab work
Alice	Bryan	35	Telephone	2021-11-05	13:13:05	Lab work
Alice	Bryan	62	In-person	2021-12-31	13:13:05	Lab work
Arti	Patil	12	In-person	2021-11-25	20:25:45	Routine checkup
Arti	Patil	63	In-person	2021-12-31	13:07:06	Lab work
Arun	Patel	NULL	NULL	NULL	NULL	NULL
Ava	Lee	49	Telephone	2021-12-28	18:04:36	Illness/other co...
Ben	Robert	4	In-person	2021-10-05	13:57:16	Illness/other co...
Ben	Robert	10	Telephone	2021-11-19	16:15:18	Routine checkup
Ben	Robert	22	Telephone	2021-12-27	13:38:26	Illness/other co...

```sql
# 4 List out all the staff members and their reporting head names (Using self join concept)
Select S1.staff_id, CONCAT(S1.first_name," ",S1.last_name) as Staff_Name,
CONCAT(S2.first_name," ",S2.last_name) as Reporting_Head
from medical_staff S1
inner join medical_staff S2
on S2.staff_id = S1.reporting_to_id;
```

staff_id	Staff_Name	Reporting_Head
101	Ralph Leonard	Kaylen Lawson
102	Charity Rice	Kaylen Lawson
103	Haleigh Fitzgerald	Rob Thomas
104	Zoie Harvey	Richard Williams
105	Kaylen Lawson	Richard Williams
106	Angel Neal	Richard Williams
107	Quentin Bowman	Richard Williams
108	Braeden Koch	Katie Johnson
109	Hanna Villarreal	Katie Johnson
110	Quentino Mcgee	Hanna Villarreal
111	Donovan Townsend	Hanna Villarreal
112	Dennis Burnett	Angel Neal
113	Rachel Green	Katie Adams

Advanced SQL concept application

View creation

```
CREATE OR replace VIEW `Patient_lab_report` AS
SELECT patient.patient_id, patient.first_name, patient.last_name, patient.date_of_birth,
patient.phone_number, patient.email_id, patient.gender, lab.immunization_test_name,
 lab.blood_work_status, report.results, report.create_date
 from patient inner join lab on patient.patient_id = lab.patient_id
 inner join report on lab.patient_id = report.patient_id;
```

Query using view

```
64
65      # Count of males and females who have Covid 19 antibodies
66      Select count(patient_id), gender from patient_lab_report
67      where immunization_test_name = 'Covid 19 antibodies'
68      Group by gender;
69
```

100% 13:72

Result Grid | Filter Rows: Search | Export:

count(patient_...	gender
5	M
3	F

Trigger creation

```
DELIMITER $$
CREATE TRIGGER update_invoice
AFTER UPDATE ON payment
FOR EACH ROW
BEGIN
    UPDATE invoice SET invoice_status='completed'
    where NEW.invoice_id=invoice.invoice_id and new.payment_status='completed';
END;
 $$
 DELIMITER ;
```

Fetching information from invoice & payment table before update

```
37  select * from payment;
38
```

Result Grid | Filter Rows: | Edit: | Export/Import:

payment_id	patient_id	invoice_id	payment_status	payment_date	amount_paid
5001	10001	201	pending	2021-04-13	300
5002	10002				
5003	10003				
5004	10004				
5005	10005				

```
37  select * from invoice;
38
```

Result Grid | Filter Rows: | Edit: | Export/Import:

invoice_id	patient_id	description	amount	create_date	invoice_status
201	10001	Consultation fee	300	2021-01-12 00:00:00	pending
202	10001	Lab test fee	500	2021-01-13 00:00:00	pending
203	10002	Consultation fee	300	2021-01-14 00:00:00	pending
204	10002	Lab test fee	500	2021-01-15 00:00:00	completed

Trigger is triggered

```
35  update payment set payment_status='completed' where payment_id=5001;
36
37  select * from payment;
38
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

payment_id	patient_id	invoice_id	payment_status	payment_date	amount_paid
5001	10001	201	completed	2021-04-13	300
5002	10002	202	pending	2021-04-14	500
5003	1000				
5004	1000				

```
35  update payment set payment_status='completed' where payment_id=5001;
36
37  select * from invoice;
38
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

invoice_id	patient_id	description	amount	create_date	invoice_status
201	10001	Consultation fee	300	2021-01-12 00:00:00	completed
202	10001	Lab test fee	500	2021-01-13 00:00:00	pending
203	10002	Consultation fee	300	2021-01-14 00:00:00	pending
204	10002	Lab test fee	500	2021-01-15 00:00:00	completed
205	10002	Consultation fee	300	2021-01-16 00:00:00	completed
206	10003	Lab test fee	500	2021-01-17 00:00:00	completed

Procedure creation

```
DELIMITER //
CREATE PROCEDURE ViewRecords( test_name varchar(30) )
BEGIN
select p.patient_id, p.first_name, p.last_name,p.gender,p.date_of_birth,r.report_id,r.immunization_test_name,r.r
from report r
JOIN patient p ON
p.patient_id=r.patient_id AND
r.immunization_test_name= test_name;
END// DELIMITER ;
```

Calling procedure

```
93
94    CALL ViewRecords('Covid 19 antibodies');
95
96
97
```

100% 1:102

Result Grid | Filter Rows: Search | Export:

patient_id	first_name	last_name	gender	date_of_bir...	report_id	immunization_test_na...	results
10004	Arun	Patel	M	1955-12-12	1003	Covid 19 antibodies	Positive
10015	Philip	Rose	M	1981-09-16	1017	Covid 19 antibodies	Positive
10034	Michael	Campbell	M	1980-07-11	1022	Covid 19 antibodies	Positive
10021	Henry	Williams	M	1977-09-10	1027	Covid 19 antibodies	Positive

Result 60

Visualizations

Tableau Dashboard

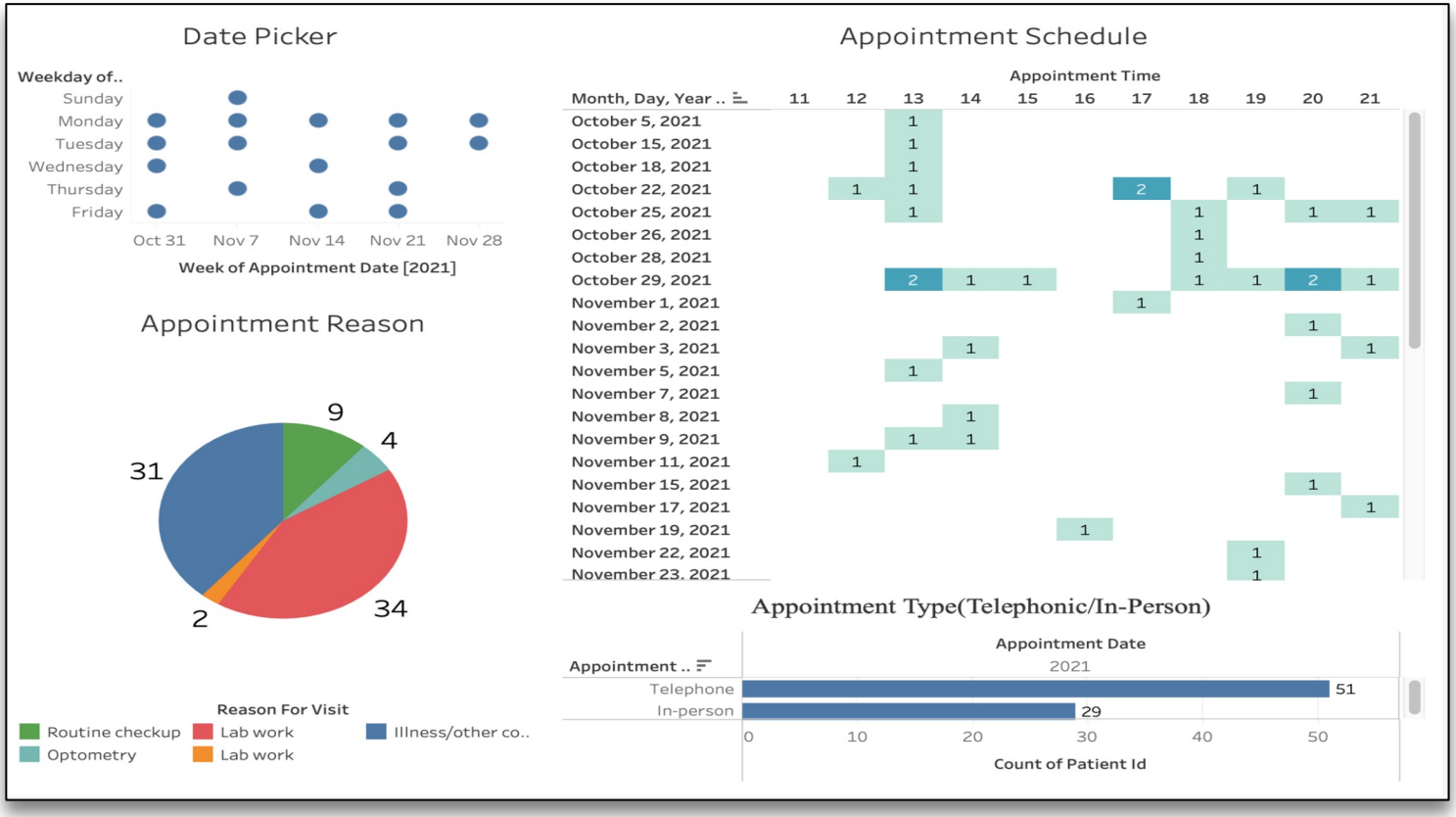
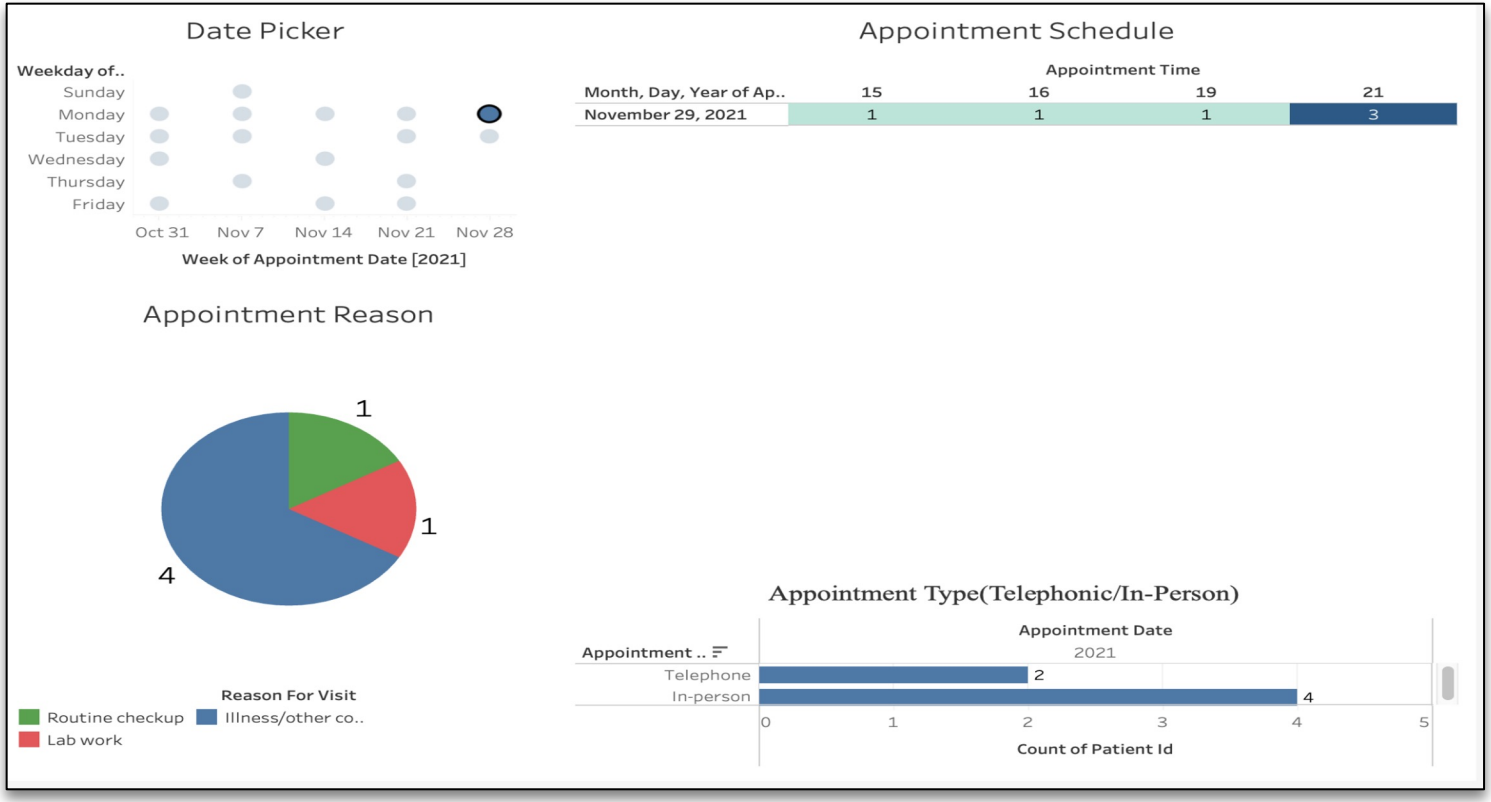
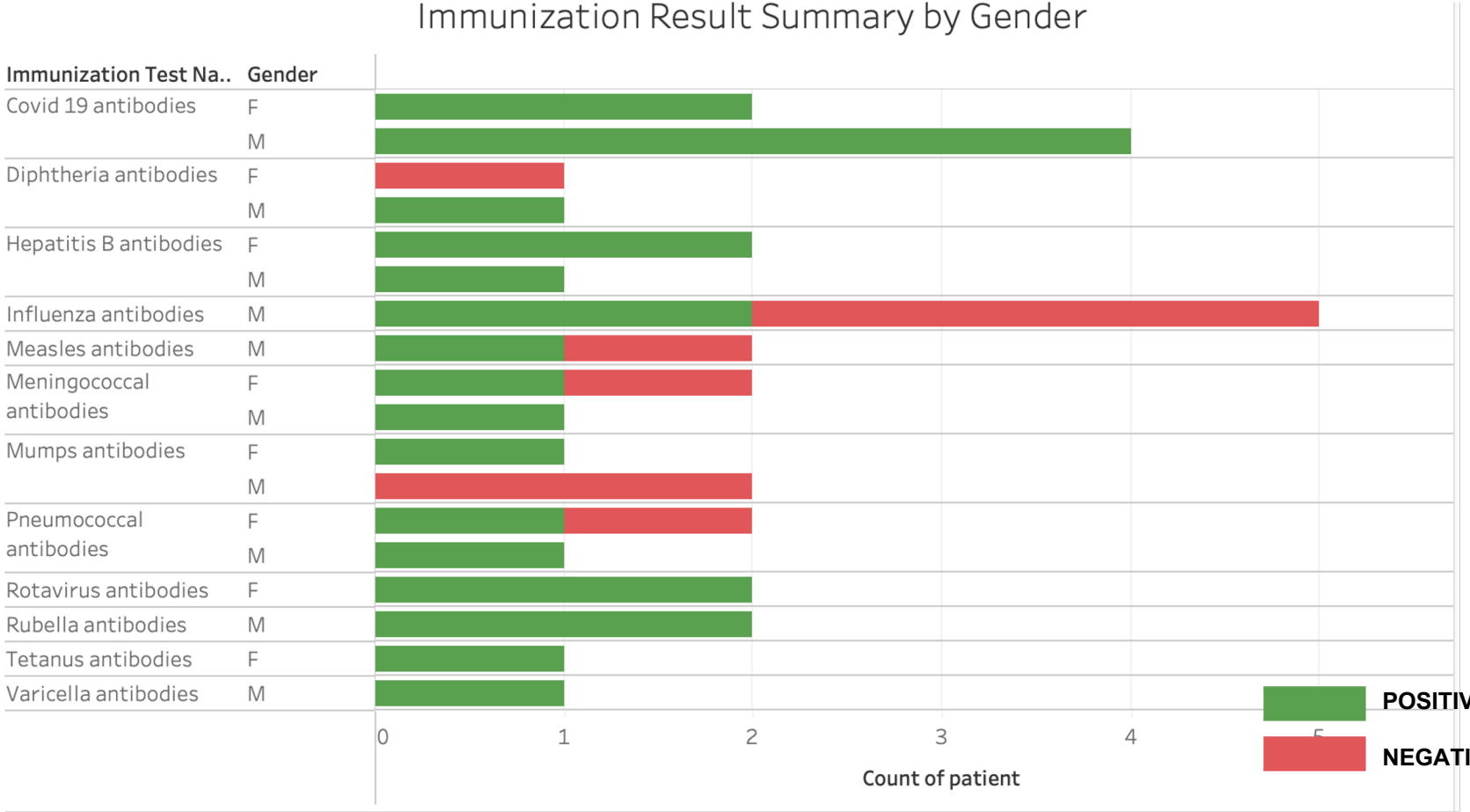


Tableau Dashboard - (Date Filter)



Immunization Result Summary by Gender



Conclusion

- This project helped us to apply the theoretical knowledge of DBMS in a real world scenario.
- We learned and applied following concepts of SQL in our project:
  - Swimlane Diagram, ER Diagram, Table Creation, SQL Joins, Views, Triggers, Procedures.
- If we were to work on this project again or extend this further:
  - Work on real world and large dataset
  - Extend our schema to include data related to location(address), departments, insurance provider, medical history.

THANK YOU