Health Insurance Management System

Database Design Document

Team Name: **Data Pirates**

Team Members:

Saranya Chintalapati (U76825685)

Hareesh Prathipati (U35227907)

Yeswanth Chamarthy (U13939822)

Sudeshana Mullaguru (U17723314)

Naga Poojitha Thaduri (U76975993)

Contents

PROBLEM STATEMENT:

Customers and Admins both can manage their insurance data with the Health Insurance Management System. The project's major goal is to make it easier for customers to comprehend their claims, benefits, bills and treatments by creating a platform where the system can provide them with all the information they need. Customers, policies, claim to process, and premium records can all be kept track of by the insurance firm. This will be accomplished by building a database management system that makes use of various entities and their relationships to create a useful system.

OBJECTIVES:

1. Maintaining the insurance company's data records.
2. Maintaining data records about new and existing policies.
3. Maintaining data records for the agents.
4. To manage data records of corporate clients.
5. To keep track of customer data.
6. To maintain data records of the insurance claim.
7. To manage patients' treatment records.
8. To keep track of the bills that have been created.
9. To develop a user-friendly application.
10. To track insurance policy information and improve efficiency.

PROPOSED SOLUTION:

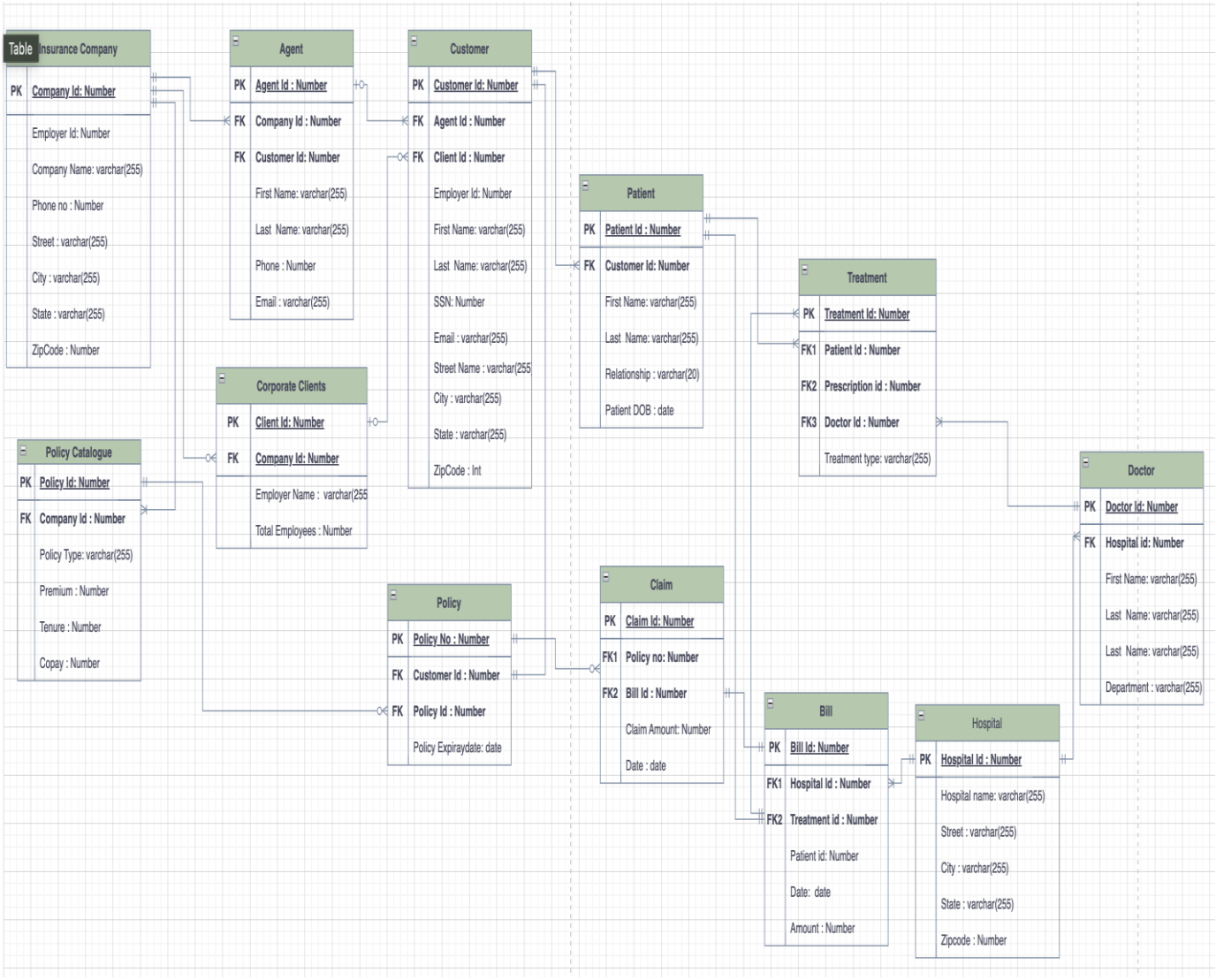
- The Customer can enter their personal information and it will be stored in the **Customer** table.
- The **insurance company** entity helps to provide policy details to the customer.
- Agent table will serve as a middleman between the customer and the Insurance company
- Customer would be able to track their claim status using the **Claim** table.
- The insurance company can keep track of Customers, policies, and claim to process using the tables.
- Corporate Clients can track their employee's insurance status using the **Customer** table.
- The insurance company can keep track of claim-related information like a hospital, doctor, treatment, and patient details using different tables.

BUSINESS RULES:

1. Customers cannot buy insurance directly. He must buy it through an agent or an employer (Here Corporate Client).
2. Only the person buying the insurance can be the patient. You need to buy separate insurance for everyone.
3. The insurance company will have at least one agent.
4. An agent can have multiple customers, but a customer should have only one agent.
5. Claims can be made multiple times for specific policies.
6. Treatment is specific to one patient.
7. Treatment for the patient is provided by the doctor
8. The hospital can have multiple doctors, but a doctor should be associated with only one hospital
9. The policy number should be unique and specific to the customer.

ER DIAGRAM:

ER Diagram stands for Entity Relationship Diagram, also known as ERD, is a diagram that displays the relationship of entity sets stored in a database.



TABLES DESCRIPTION:

INSURANCE COMPANY ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Company ID	NUMBER	PRIMARY KEY
Company Name	VARCHAR2(255)	NOT NULL
Phone _no	NUMBER	UNIQUE KEY, NOT NULL
Street	VARCHAR2(255)	NOT NULL
City	VARCHAR2(255)	NOT NULL
State	VARCHAR2(255)	NOT NULL
Zip code	VARCHAR2(6)	NOT NULL

AGENT ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Agent ID	NUMBER	PRIMARY KEY
Company ID	NUMBER	A Foreign key which REFERENCES Company ID from the INSURANCE COMPANY ENTITY
First Name	VARCHAR2(30)	NOT NULL
Last Name	VARCHAR2(40)	NOT NULL
Phone	NUMBER	UNIQUE KEY, NOT NULL
Email	VARCHAR2(255)	UNIQUE KEY, NOT NULL

POLICY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Policy No	NUMBER	PRIMARY KEY
Customer ID	NUMBER	FOREIGN KEY REFERENCES Customer ID from Customer.
Policy ID	NUMBER	FOREIGN KEY REFERENCES Policy ID from Policy.
Policy expiry date	NUMBER	NOT NULL

CUSTOMER ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Customer ID	NUMBER	PRIMARY KEY
Agent ID	NUMBER	A Foreign key which REFERENCES Agent ID from the Agent Table.
Client ID	NUMBER	FOREIGN KEY REFERENCES Client ID from Client.
First Name	VARCHAR2(30)	NOT NULL
Last Name	VARCHAR2(40)	NOT NULL
SSN	NUMBER	UNIQUE KEY, NOT NULL
Email	VARCHAR2(255)	UNIQUE KEY, NOT NULL
Street Name	VARCHAR2(255)	NOT NULL
City	VARCHAR2(255)	NOT NULL
State	VARCHAR2(255)	NOT NULL
Zip code	VARCHAR2(6)	NOT NULL

CORPORATE CLIENT ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Client ID	NUMBER	PRIMARY KEY
Company ID	NUMBER	FOREIGN KEY REFERENCES Company ID from Insurance Company.
Client Name	VARCHAR2(255)	NOT NULL
Total Employees	NUMBER	NOT NULL

CLAIM ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Claim ID	NUMBER	PRIMARY KEY
Policy No	NUMBER	FOREIGN KEY References Policy Number from Policy
Bill ID	NUMBER	FOREIGN KEY References Bill ID from Bill.
Claim Amount	NUMBER	NOT NULL
Date	DATE	NOT NULL
Status	Varchar2(10)	NOT NULL

HOSPITAL ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Hospital ID	NUMBER	PRIMARY KEY
Hospital Name	VARCHAR2(255)	NOT NULL
Address	VARCHAR2(255)	NOT NULL
City	VARCHAR2(255)	NOT NULL
State	VARCHAR2(255)	NOT NULL
Zip code	VARCHAR2(255)	NOT NULL

POLICY CATALOGUE ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Policy ID	NUMBER	PRIMARY KEY
Company ID	NUMBER	FOREIGN KEY References Company ID from Company.
Premium	VARCHAR2(255)	NOT NULL
Copay	NUMBER	NOT NULL
Tenure	NUMBER	NOT NULL

TREATMENT

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Treatment ID	NUMBER	PRIMARY KEY
Patient ID	NUMBER	FOREIGN KEY References Patient ID from Patient.
Doctor ID	NUMBER	FOREIGN KEY References Doctor ID from Doctor.
Treatment Type	VARCHAR2(255)	NOT NULL

BILL ENTITY

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Bill ID	NUMBER	PRIMARY KEY
Treatment ID	NUMBER	FOREIGN KEY References Treatment ID from Treatment.
Hospital ID	NUMBER	FOREIGN KEY References Hospital ID from Hospital.
Patient ID	NUMBER	NOT NULL
Date	DATE	NOT NULL
Amount	NUMBER	NOT NULL

DOCTOR

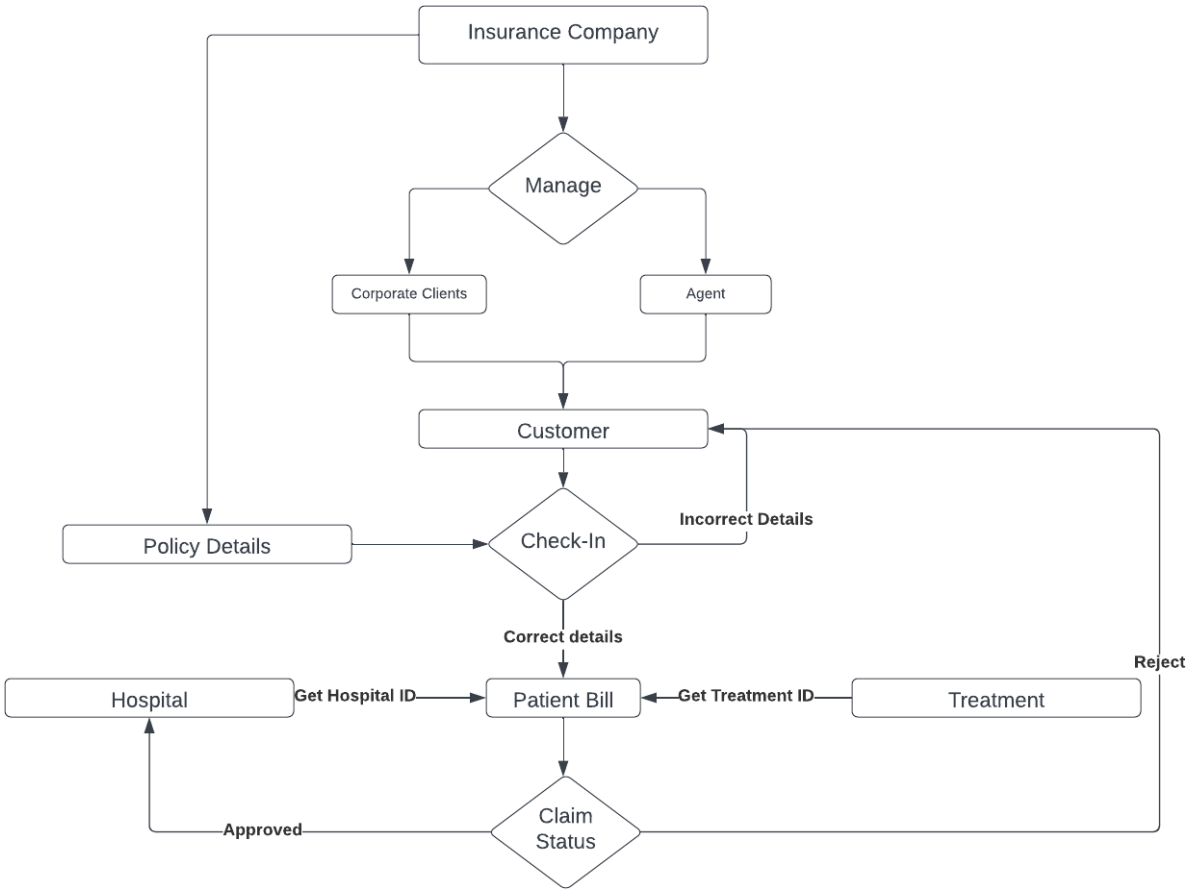
ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Doctor ID	NUMBER	PRIMARY KEY
Hospital ID	NUMBER	FOREIGN KEY References Hospital ID from Hospital.
First Name	VARCHAR2(30)	NOT NULL
Last Name	VARCHAR2(40)	NOT NULL
Department	VARCHAR2(255)	NOT NULL

PATIENT

ATTRIBUTES	DATA TYPE AND SIZE	COMMENTS
Patient ID	NUMBER	PRIMARY KEY
Customer ID	NUMBER	FOREIGN KEY References Customer ID from Customer
First Name	VARCHAR2(30)	NOT NULL
Last Name	VARCHAR2(40)	NOT NULL
Relationship	VARCHAR2(20)	NOT NULL
Patient DOB	Date	NOT NULL

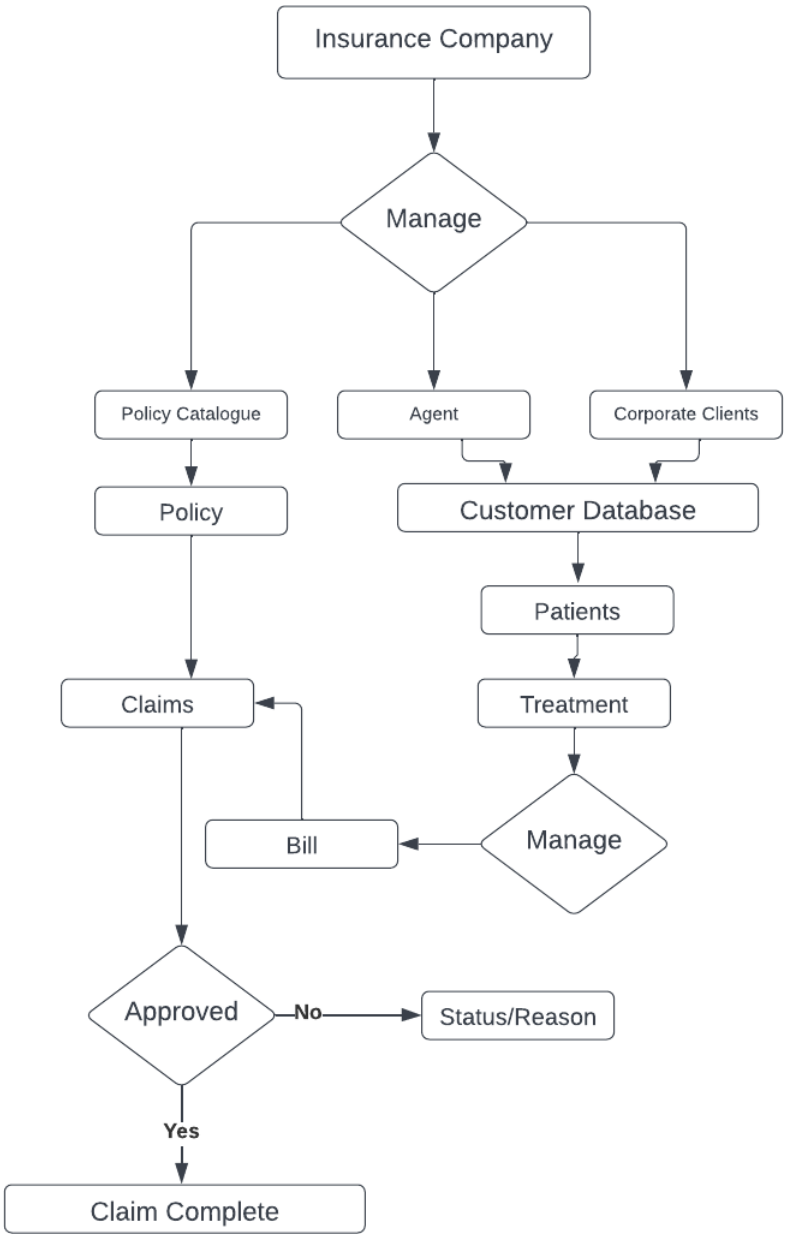
DATA FLOW DIAGRAMS:

1. **Health Insurance Management System**



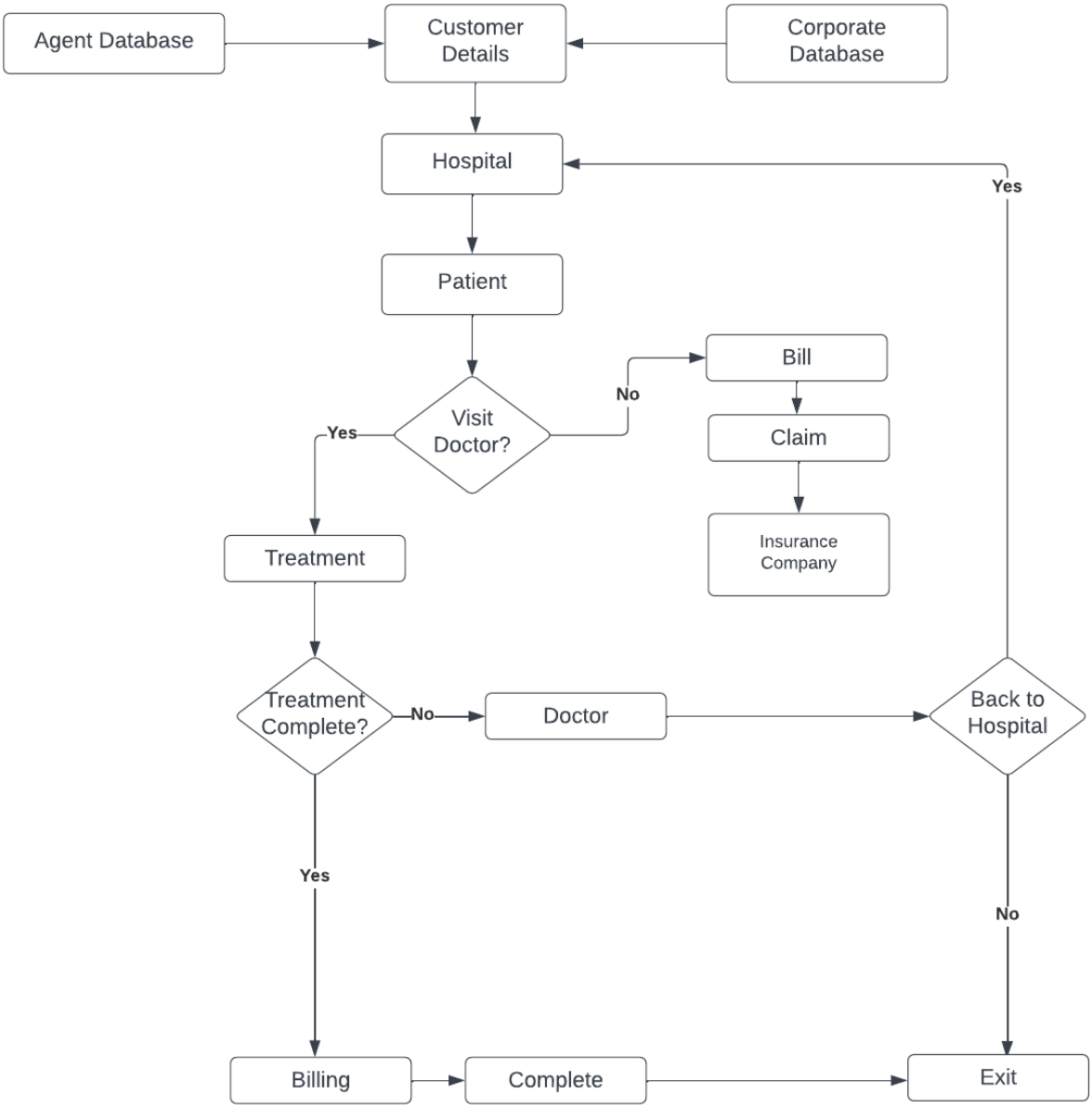
a. **Insurance companies can provide policy policies to an agent or corporate clients.**
b. **Policy claim is to be used to keep track of policy history.**

2. Claim Status



a. **Customers can keep track of the status of the claim.**
b. **Claims can be made for specific policies multiple times.**

3. Customer Registration



a. **Customers can buy insurance either from an agent or through the company he is employed.**
b. **One agent can have multiple customers, but every customer should have only one agent per insurance.**

SECURITY AND ACCESS:

1. **Customer**

    a. The customer can see his policy and all the policy details
    b. Customers can see claim status and claim details
    c. Customers can view the history of the claims
    d. The customer can claim the bill.
    e. Customers can update their customer information.

2. **Admin**

    a. Access to the entire database and metadata
    b. Responsible for the integrity, security, and performance of the database.

3. **Agent**

    a. An agent can see available policies and policy details.
    b. An agent can view customer and customer details registered under the agent.
    c. An agent can update the agent information.

4. **Corporate Client**

    a. Corporate clients can see available policies and policy details.
    b. Corporate clients can view total customers and customer details registered under corporate clients.
    c. Corporate clients can update corporate client information.

VIEWS:

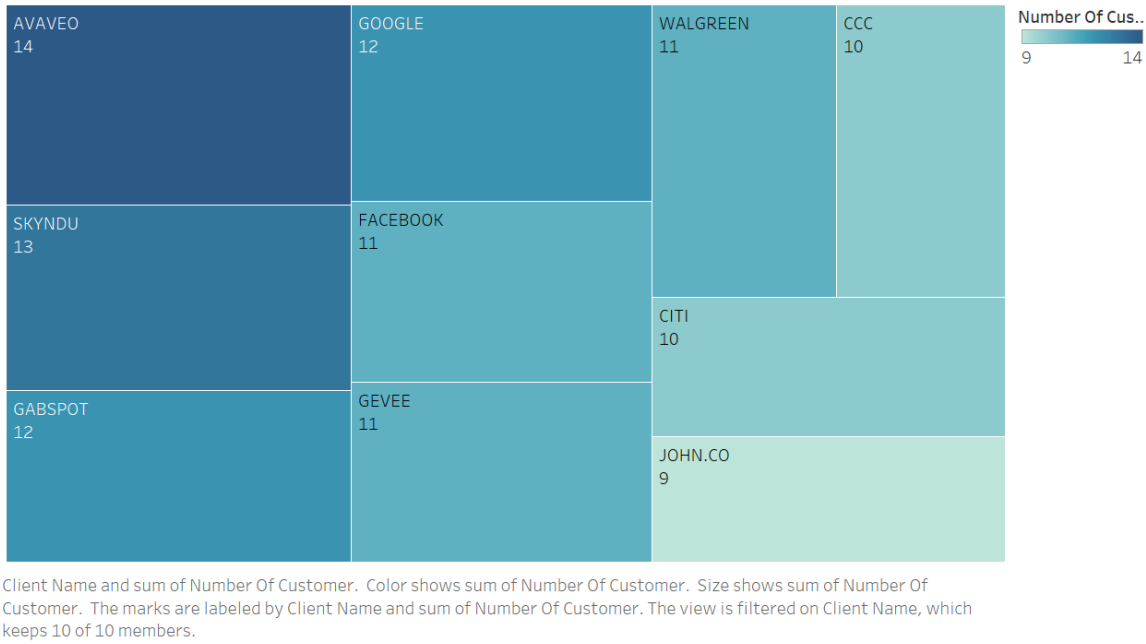
Views in Oracle are virtual tables that don't physically exist. It is stored in the Oracle data dictionary and doesn't keep any data. It can be executed when called.

1. View for Insurance Companies for the total number of customers from Corporate Clients
   a. Track details of Customers from each Corporate Client
   b. Display Corporate Clients with the highest number of insurances purchased.

```
SELECT * FROM CORPORATECLIENTS;
```

Explain Plan | Query Result | SQL | All Rows Fetched: 10 in 0.058 seconds

	CLIENT_ID	CLIENT_NAME	NUMBER_OF_CUSTOMER
1	10	AVAVEO	14
2	9	SKYNDU	13
3	3	GOOGLE	12
4	8	GABSPOT	12
5	7	GEVEE	11
6	4	FACEBOOK	11
7	2	WALGREEN	11
8	5	CCC	10
9	6	CITI	10
10	1	JOHN.CO	9



Client Name and sum of Number Of Customer. Color shows sum of Number Of Customer. Size shows sum of Number Of Customer. The marks are labeled by Client Name and sum of Number Of Customer. The view is filtered on Client Name, which keeps 10 of 10 members.

2. View for the patient per doctor
   a. Display the total count of the patient for the doctor.

Worksheet | Query Builder

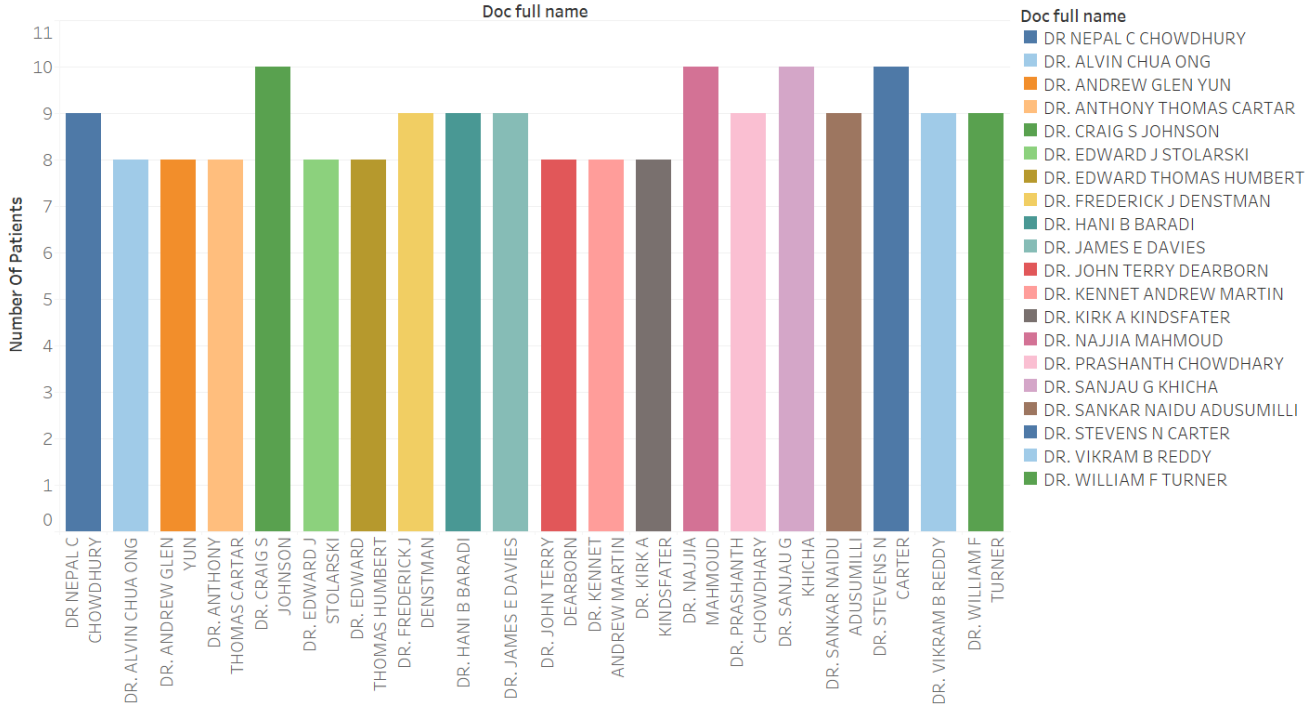
```
Select * from PatientPerDoctor;
```

Query Result

SQL | All Rows Fetched: 25 in 0.059 seconds

	DOCTOR_ID	FIRST_NAME	LAST_NAME	NUMBER_OF_PATIENTS
1	1	DR. HANI B	BARADI	9
2	2	DR. NAJJIA	MAHMOUD	10
3	3	DR. SANKAR NAIDU	ADUSUMILLI	9
4	4	DR. LANCE KYLE	BURNS	8
5	5	DR. VIKRAM B	REDDY	9
6	6	DR. STEVENS N	CARTER	10
7	7	DR. VIRGILIO V	GEORGE	8
8	8	DR. CRAIG S	JOHNSON	10
9	9	DR. FREDERICK J	DENSTMAN	9
10	10	DR. SANJAU G	KHICHA	10

No of patients per doctor



Sum of Number Of Patients for each Doc full name. Color shows details about Doc full name. The view is filtered on Doc full name, which excludes DR. LANCE KYLE BURNS, DR. ROWLAND MARSHALL ROBERSON, DR. SHAMSUDDIN KHWAJA, DR. VIRGILIO V GEORGE and DR. VUONG B NGUYEN.

3. View for Ranking claims for claim amount concerning treatment.
   a. Present the Claim amount for treatment and rank them accordingly.
   b. Show the treatment with the highest claim amount.

```
SELECT * FROM CLAIMRANK;
```

Explain Plan | Query Result

SQL | All Rows Fetched: 134 in 0.126 seconds

	CLAIM_ID	TREATMENT_ID	TREATMENT_TYPE	CLAIM_AMOUNT	STATUS	RANK
1	224	8	KIDNEY DIALISIS	10000	APPROVED	1
2	284	68	TYPE 2 DIABETES	9900	APPROVED	2
3	423	207	STARGARDT DISEASE	9750	APPROVED	3
4	266	50	ERYTHROMELALGIA	9500	APPROVED	4
5	332	116	LYMPHOMA	9200	APPROVED	5
6	379	163	JEUNE SYNDROME	8770	APPROVED	6
7	264	48	LYMPHATIC BIOPSY	8020	APPROVED	7
8	312	96	HEART ATTACK	7900	APPROVED	8
9	412	196	FRACT FLOW RESERVE	7900	APPROVED	8
10	226	10	STOMACHACHE	7900	APPROVED	8
11	381	165	MECKEL SYNDROME	7880	APPROVED	9
12	232	16	NEURO STROKE	7800	APPROVED	10
13	331	115	FISH-EYE DISEASE	7800	APPROVED	10
14	382	166	ROSACEA	7670	APPROVED	11

4. View to display all customer data
   a. Display details of the customer who purchased insurance.

```
SELECT * FROM CUSTOMERDATA;
```

Explain Plan | Query Result

SQL | All Rows Fetched: 216 in 0.241 seconds

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	CITY	POLICY_NO
1	1	JOHN	MICH	BOSTON	1
2	2	JACK	ROY	BOSTON	2
3	3	BOB	HARRIS	HOUSTON	3
4	4	SAHIL	JOSHI	DALLAS	4
5	5	TINA	RAUT	SAN FRANSISCO	5
6	6	SIMRAN	NAGPURKAR	BOSTON	6
7	7	SITA	PATEL	NEW YORK	7
8	8	RAJ	DEV	NEW YORK	8
9	9	SWAROOPA	LAXMI	CHELSI	9
10	10	RAMESH	KUMAR	BRIGHTON	10

5. View to display all agent data
   a. Show all the agent data that is registered under the insurance company
   b. Display the total number of customers who have purchased from the agent

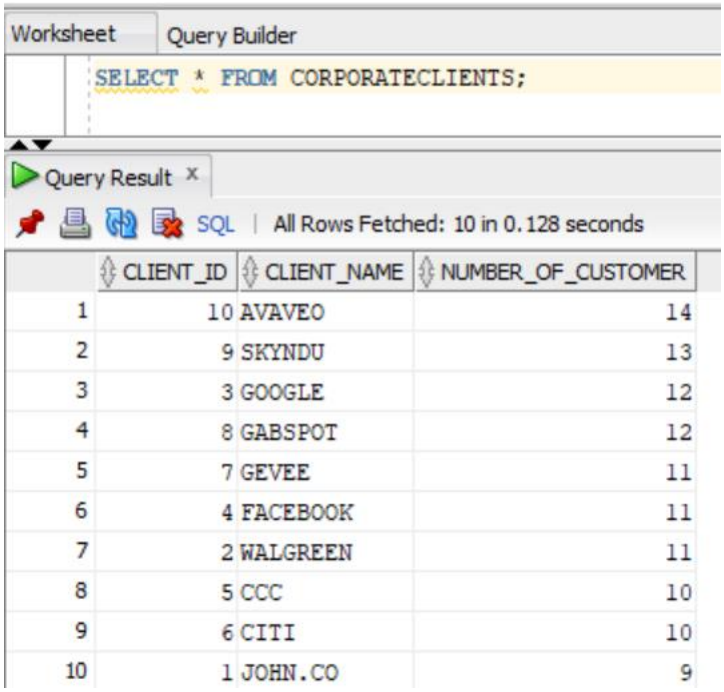
```
SELECT * FROM AGENTDATA;
```

Explain Plan | Query Result

SQL | All Rows Fetched: 8 in 0.036 seconds

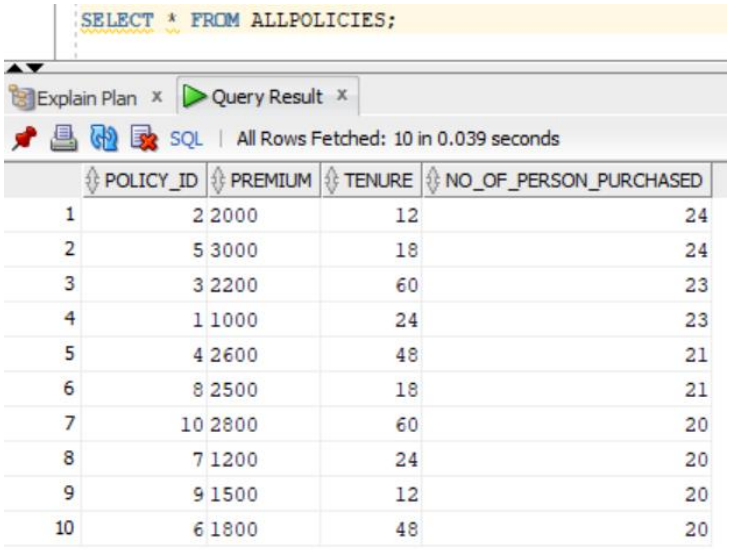
	AGENT_ID	FIRST_NAME	LAST_NAME	NUMBER_OF_CUSTOMER
1	7	SIMONA	MENGO	16
2	4	AURA	MAULKIN	14
3	5	OZZIE	ODY	14
4	3	RUTUJA	GHOGARE	13
5	8	ADRIENA	MATCHELL	13
6	6	POUL	GUERNER	12
7	1	RISHI	FIRKE	11
8	2	ANIL	DESAI	10

6. View to display all corporate client
   a. View all corporate client data that has been registered with the insurance company
   b. Display the total customers who have purchased from corporate clients.



	CLIENT_ID	CLIENT_NAME	NUMBER_OF_CUSTOMER
1	10	AVAVEO	14
2	9	SKYNDU	13
3	3	GOOGLE	12
4	8	GABSPOT	12
5	7	GEVEE	11
6	4	FACEBOOK	11
7	2	WALGREEN	11
8	5	CCC	10
9	6	CITI	10
10	1	JOHN.CO	9

7. View to display all existing policies
   a. Display all the policy and policy details



	POLICY_ID	PREMIUM	TENURE	NO_OF_PERSON_PURCHASED
1	2	2000	12	24
2	5	3000	18	24
3	3	2200	60	23
4	1	1000	24	23
5	4	2600	48	21
6	8	2500	18	21
7	10	2800	60	20
8	7	1200	24	20
9	9	1500	12	20
10	6	1800	48	20

8. View to display the claim status of the policy
   a. Show the status of the claim requested by the customer

```
select * from claimstatus;
```

Query Result

SQL | Fetched 50 rows in 0.047 seconds

	CUSTOMER_FIRST_NAME	CUSTOMER_LAST_NAME	PATIENT_FIRST_NAME	PATIENT_LAST_NAME	CLAIM_AMOUNT	STATUS
1	JARROD	KINLOCK	GIANA	KINLOCK	100	APPROVED
2	PAGE	DULWICH	RICKY	DULWICH	120	REJECTED
3	NOLIE	ADKIN	NOLIE	ADKIN	160	APPROVED
4	ESTELE	PENTONY	BONDIE	PENTONY	200	APPROVED
5	JUSTUS	FAIRBRACE	PETRINA	FAIRBRACE	200	APPROVED
6	GUENDOLEN	BIGGEN	IBRAHIM	BIGGEN	210	APPROVED
7	PEDRO	GRUNDLE	INGLIS	GRUNDLE	250	APPROVED
8	PETEY	TURPIN	NIGEL	TURPIN	300	PENDING
9	ANNABAL	DOBERER	ANNABAL	DOBERER	340	APPROVED
10	JABEZ	DIBDALL	JABEZ	DIBDALL	350	PENDING

Optimizing Query Performance

Index is created on the state column of hospital table and a bitmap index is created on the claim status of claim table. Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

Indexes

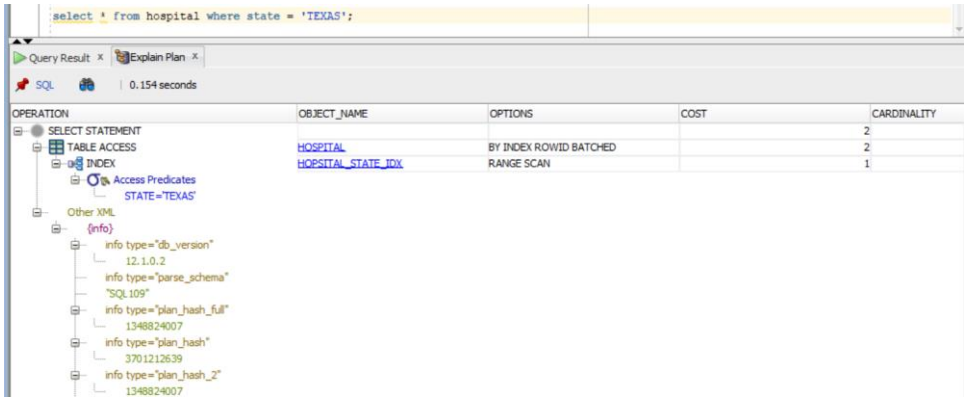
The following indexes are created on our databases.

create index hopsital_state_idx on hospital(state);

create bitmap index claim_status_idx on claim(status);

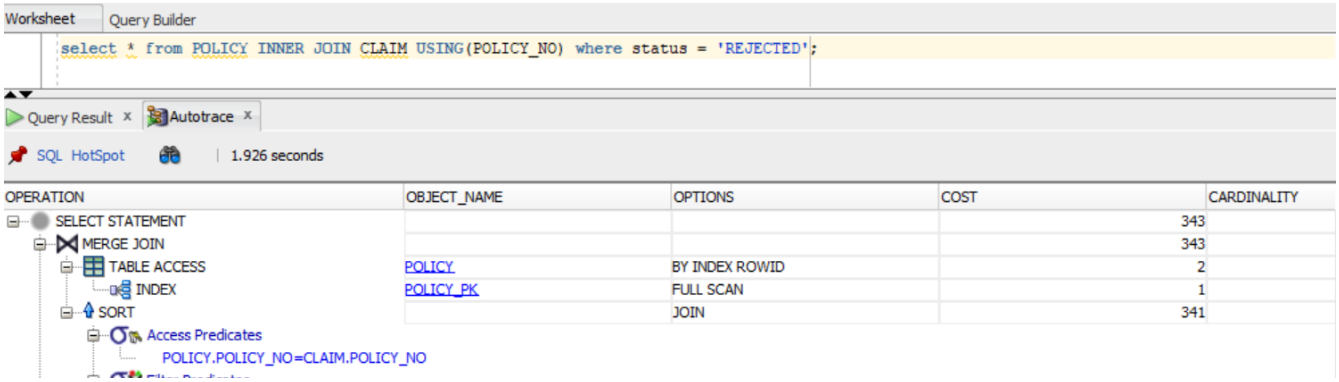
Checking the execution plan post index creation:

select * from hospital where state = 'TEXAS';



Instead of a FULL scan, a RANGE scan is performed on the data and the cost reduced to 2 units. Thus, improving the query performance significantly.

select * from POLICY INNER JOIN CLAIM USING(POLICY_NO) where status = 'REJECTED';



Conclusion:

Thus, the health insurance management system is a comprehensive platform for improving workflow and automating insurance claims. It has all the features required to carry out multiple insurance claims processes.