***Final Project Report***

*MSIS 618 Database Management – Spring 2023*

*Bindu Sai Sri Yerra*

*02084144*

*May 16th, 2023*

***Health Insurance Database***

*Under guidance*:

***Kui Du***

TABLE OF CONTENTS

[*1.* *Executive Summary* 3](#_Toc135173595)

[2. Project Description 4](#_Toc135173596)

[3. Database Design: ER Diagram 5](#_Toc135173597)

[4. Table Details 6](#_Toc135173598)

[**4.1** Policy\_Holders Table 6](#_Toc135173599)

[**4.2** Insurance\_Plans Table 6](#_Toc135173600)

[**4.3** Healthcare\_Providers Table 7](#_Toc135173601)

[**4.4** Enrollment\_Details Table 7](#_Toc135173602)

[**4.5** Diagnosis\_Details Table 8](#_Toc135173603)

[**4.6** Payment\_Details Table 8](#_Toc135173604)

[**4.7** Claims\_Details Table 8](#_Toc135173605)

[**4.8** Policy\_Details Table 9](#_Toc135173606)

[**4.9** Relations between the tables 9](#_Toc135173607)

[5. Normalization 11](#_Toc135173608)

[5.1 PolicyHolder Table 11](#_Toc135173609)

[5.2 Insurance\_Plans 11](#_Toc135173610)

[5.3 Healthcare\_Providers 11](#_Toc135173611)

[5.4 Enrollment\_Details 12](#_Toc135173612)

[5.5 Payment\_Details 12](#_Toc135173613)

[5.6 Diagnosis\_Details 12](#_Toc135173614)

[5.7 Claims\_Details 12](#_Toc135173615)

[5.8 Insurance\_Plans\_has\_Policy\_Holders 13](#_Toc135173616)

[5.9 Healthcare\_Providers\_has\_Policy\_Holders 13](#_Toc135173617)

[6. Database Implementation – Queries 13](#_Toc135173618)

# *Executive Summary*

The Health Insurance database project intends to provide a thorough and effective means of handling health insurance data. This database system provides a wide range of functions that can greatly boost operations, process efficiency, and decision making.

* The database acts as a single repository for organizing, storing, and retrieving vital information on policy holders, insurance policies, healthcare provider details, claims, payments, and diagnosis. As such, there is no longer a need for distant and inaccurate data sources, making it simple to success, retrieve, and analyze data.
* The database makes it possible to manage policy holder data including personal information, insurance information, and policy information efficiently. This makes it easier to manage patient coverage accurately, track policy benefits and maintain smooth communication between the company and the insured people.
* The database enables the recording and tracking of policy holder claims, linking them to a particular diagnosis and insurance information, which enhances the effectiveness of the claim verification, settlement, and payment procedures while optimizing the claims processing workflow and minimizing manual errors.
* It makes possible effective financial management and analysis by enabling the inclusion of payment records. It offers insights into payment trends, claim reimbursement patterns, and assists in identifying potential cost-saving solutions.
* It makes it easier to handle contact and address information for healthcare providers. This improves referral accuracy, keeps the provider network up-to-date and raises the overall standard of care for our covered customers.

# Project Description

Company: Blue Cross Blue Shield Insurance Company

Business Activity: Claims Processing

Leading provider of health insurance services, Blue Cross Blue Shield Insurance company covers a variety of medical costs for its clients. Processing claims is one of the company’s key business operations. This task entails processing insurance claims for payment of medical expenses that policyholders have submitted efficiently and accurately.

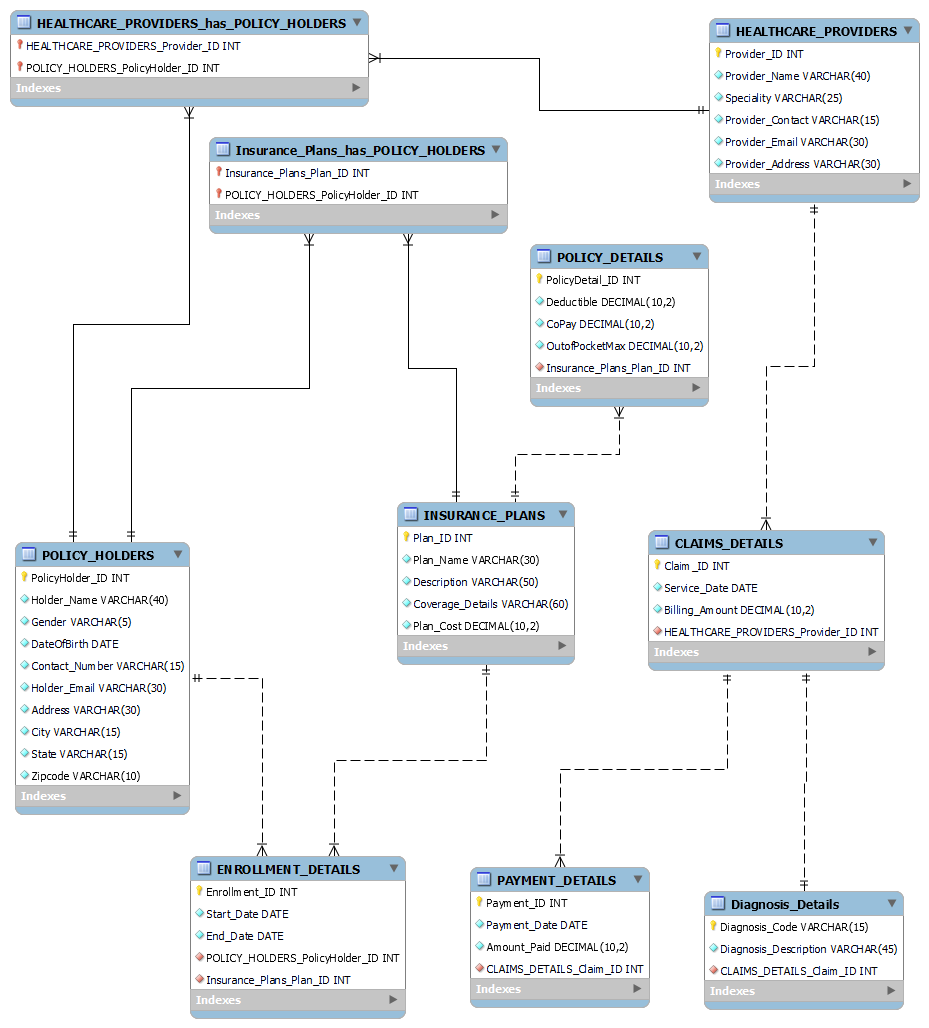
The existing manual claim processing procedure is tedious, prone to mistakes, and deficient in effective data management.

This database project’s goal is to create a thorough claims processing system that will help the business streamline and automate the claims handling procedure. The targeted users of the database system include claims processors, insurance agents, healthcare providers and policy holders.

Business Requirements:

* Efficient Claims Tracking: The database system should make it simple for claim processors to keep track of each claim’s status from submission through settlement, maintaining timely processing and minimizing delays.
* Accurate Claims Validation: To reduce errors and stop fraudulent claims, the system needs to provide a process for validating claim information, such as policyholder information, treatment information, and healthcare provider credentials.
* Seamless Communication: The system should enable speedy resolution of questions and clarifications by facilitating seamless communication and collaboration between claims processors, insurance agents, and healthcare providers.
* Enhanced Customer Service: By allowing policy holders to check the status of their claims online, providing transparency and decreasing customer enquiries, the system should enhance the general customer experience.

# Database Design: ER Diagram



**Figure 1: ER Diagram for Health Insurance**

# Table Details

## Policy\_Holders Table

This table includes the personal information of the policy holders.

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN NAME | DATA TYPE | KEY/ CONSTRAINTS | REMARKS |
| PolicyHolder\_ID | INT | Primary Key, Not Null, Unique | Unique identifier of each policy holder |
| Holder\_Name | VARCHAR (40) | Not Null | Name of the policy holder |
| Gender | VARCHAR (5) | Not Null | Gender of the policy holder |
| DateofBirth | DATE | Not Null | Date of birth of the policy holder |
| Contact\_Number | VARCHAR (15) | Not Null | Contact number of the policy Holder |
| Holder\_Email | VARCHAR (30) | Not Null | Email of the policy holder |
| Address | VARCHAR (30) | Not Null | Address of the policy holder |
| City | VARCHAR (15) | Not Null | City where the policy holder resides |
| State | VARCHAR (15) | Not Null | State where the policy holder resides |
| Zipcode | VARCHAR (10) | Not Null | Zipcode of the policy holder’s location. |

## Insurance\_Plans Table

This table stores information about the plan details of insurance.

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN NAME | DATATYPE | KEY/  CONSTRAINTS | REMARKS |
| Plan\_ID | INT | Primary Key, Not Null, Unique | Unique identifier for each insurance plan |
| Plan\_Name | VARCHAR (30) | Not Null | Name of the Insurance plan |
| Description | VARCHAR (50) | Not Null | Description of the insurance plan |
| Coverage\_Details | VARCHAR (60) | Not Null | Details of the coverage provided by the insurance plan, |
| Plan\_Cost | DECIMAL (10,2) | Not Null | Cost of the insurance plan |

## Healthcare\_Providers Table

This table includes information about the healthcare providers and their specialty.

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN NAME | DATA TYPE | KEY/CONSTRAINTS | REMARKS |
| Provider\_ID | INT | Primary Key, Not Null, Unique | Unique identifier of the provider |
| Provider\_Name | VARCHAR (40) | Not Null | Name of the provider |
| Specialty | VARCHAR (25) | Not Null | Specialty of the provider |
| Provider\_Contact | VARCHAR (15) | Not Null | Contact details of the provider |
| Provider\_Email | VARCHAR (30) | Not Null | Email of the provider |
| Provider\_Address | VARCHAR (30) | Not Null | Address of the Provider |

## Enrollment\_Details Table

This table stores information about the validity of the insurance plan such as start and end date.

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN NAME | DATA TYPE | KEY/CONTARINTS | REMARKS |
| Enrollment\_ID | INT | Primary Key, Not Null, Unique | Unique identifier of the enrollment |
| Start\_Date | DATE | Not Null | Start Date of the plan |
| End\_Date | DATE | Not Null | End Date of the plan |

## Diagnosis\_Details Table

This table stores information about the diagnosis code and description.

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN NAME | DATA TYPE | KEY/CONSTRAINTS | REMARKS |
| Diagnosis\_Code | VARCHAR (15) | Primary Key, Not Null | Unique code of the diagnosis |
| Diagnosis\_Description | VARCHAR (45) | Not Null | Description of the Diagnosis |

## Payment\_Details Table

This table includes the information about the payment details.

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN NAME | DATA TYPE | KEY/CONSTRAINTS | REMARKS |
| Payment\_ID | INT | Primary Key, Not Null, Unique | Unique identifier of each payment |
| Payment\_Date | DATE | Not Null | Date of the payment |
| Amount\_Paid | DECIMAL (10,2) | Not Null | Amount paid for the claim |

## Claims\_Details Table

This table stores the information about the claims made by the policy holders.

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN NAME | DATA TYPE | KEY/CONSTRAINTS | REMARKS |
| Claim\_ID | INT | Primary Key, Not Null, Unique | Unique Identifier of each claim |
| Service\_Date | DATE | Not Null | Date on which claim was made |
| Billing\_Amount | DECIMAL (10,2) | Not Null | Billed amount for the medical services claimed. |

## Policy\_Details Table

This table includes details of the policy.

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN NAME | DATA TYPE | KEY/CONSTRAINTS | REMARKS |
| PolicyDetail\_ID | INT | Primary Key, Not Null, Unique | Unique identifier for each policy detail |
| Deductible | DECIMAL (10,2) | Not Null | Deductible amount associated with the policy |
| CoPay | DECIMAL (10,2) | Not Null | Co-payment amount associated with the policy |
| OutofPocketMax | DECIMAL (10,2) | Not Null | Maximum out-of -pocket expense limit associated with the policy |

## Relations between the tables

1. PolicyHolders: Insurance Table 🡺 Many to Many (M:M)

Each policyholder must be associated with at least one insurance plan or more, which indicates every policyholder in the database has an active insurance plan.

Maximum cardinality: M, Minimum cardinality: 1

Each insurance plan must be associated with at least one policyholder or more, which indicates all policy holders are covered by at least one insurance plan.

Maximum cardinality: M, Minimum cardinality: 1

1. PolicyHolders: Enrollment 🡺 One to Many (1:M)

Each policy holder must be associated with at least 1 or more enrollments in the enrollment table.

Maximum cardinality: M, Minimum cardinality: 1

Each enrollment is associated with only one policy holder.

Maximum cardinality: 1, Minimum cardinality: 1

1. Enrollment: Insurance Plan 🡺 Many to One (M:1)

Each Insurance plan can have one or multiple enrollments.

Maximum cardinality: M, Minimum cardinality: 1

Each enrollment is associated with only one insurance plan.

Maximum cardinality: 1, Minimum cardinality: 1

1. Healthcare Provider: PolicyHolder 🡺 Many to Many (M:M)

Each healthcare provider can serve 1 or multiple patients.

Maximum cardinality: M, Minimum cardinality: 1

Each policy holder can be associated with at least one or multiple healthcare providers.

Maximum cardinality: M, Minimum cardinality: 1

1. Policy\_Details: Insurance\_Plan 🡺 Many to One (M:1)

Each Insurance plan can have 1 or more policy details/record.

Maximum cardinality: M, Minimum cardinality: 1

Each Policy details is associated with only one insurance policy.

Maximum cardinality: 1, Minimum cardinality: 1

1. Claims\_Details: Payment\_Details 🡺 One to Many (1:M)

Each claim can have 1 or more payments which can be paid partially or installments.

Maximum cardinality: M, Minimum cardinality: 1

Each payment is linked to only one claim.

Maximum cardinality: 1, Minimum cardinality: 1

1. Claim\_Details: Diagnosis\_Details 🡺 One to One (1:1)

Each claim must have a corresponding diagnosis.

Maximum cardinality: 1, Minimum cardinality: 1

Each claim is associated with a single diagnosis.

Maximum cardinality: 1, Minimum cardinality: 1

1. Healthcare\_Providers: Claim\_Details 🡺 One to Many (1:M)

Each healthcare provider can be associated with one or more claims.

Maximum cardinality: M, Minimum cardinality: 1

Each claim must have a valid healthcare provider with it, which is linked to one healthcare provider.

Maximum cardinality: 1, Minimum cardinality: 1

# Normalization

## PolicyHolder Table

(PolicyHolder\_ID, Holder\_Name, Gender, DateOfBirth, Conatct\_Number, Hodler\_Email, Address, City, State, Zipcode)

STEP 1: Identify Functional Dependency

PolicyHolder\_ID 🡺 Holder\_Name, Gender, DateOfBirth, Conatct\_Number, Hodler\_Email, Address, City, State, Zipcode)

STEP 2: Primary Key: PolicyHolder\_ID

Dependencies: Holder\_Name, Gender, DateOfBirth, Conatct\_Number, Hodler\_Email, Address, City, State, Zipcode

STEP 3: This is in BCNF.

## Insurance\_Plans

(Plan\_ID, Plan\_Name, Description, Coverage\_Details, Plan\_Cost)

STEP 1: Identify Functional Dependency

Plan\_ID 🡺 Plan\_Name, Description, Coverage\_Details, Plan\_Cost

STEP 2: Primary Key: Plan\_ID

Dependencies: Plan\_Name, Description, Coverage\_Details, Plan\_Cost

STEP 3: This is in BCNF.

## Healthcare\_Providers

(Provider\_ID, Provider\_Name, Specialty, Provider\_Contact, Provider\_Email, Provider\_Address)

STEP 1: Identify Functional Dependency

Provider\_ID 🡺 Provider\_Name, Specialty, Provider\_Contact, Provider\_Email, Provider\_Address

STEP 2: Primary Key: Provider\_ID

Dependencies: Provider\_Name, Specialty, Provider\_Contact, Provider\_Email, Provider\_Address

STEP 3: This is in BCNF

## Enrollment\_Details

(Enrollment\_ID, Start\_Date, End\_Date)

STEP 1: Identify Functional Dependency

Enrollment\_ID 🡺 Start\_Date, End\_Date

STEP 2: Primary Key: Enrollment\_ID

Dependencies: Start\_Date, End\_Date

Foreign Keys: PolicyHolder\_ID Reference PolicyHolders Table, Plan\_ID Reference Insurance\_Plan Table

STEP 3: This is in BCNF

## Payment\_Details

(Payment\_ID, Payment\_Date, Amount\_Paid)

STEP 1: Identify Functional Dependency

Payment\_ID 🡺 Payment\_Date, Amount\_Paid

STEP 2: Primary Key: Payment\_ID

Dependencies: Payment\_Date, Amount\_Paid

Foreign Keys: Claim\_ID Reference Claims\_Details Table

STEP 3: This is in BCNF

## Diagnosis\_Details

(Diagnosis\_Code, Diagnosis Description)

STEP 1: Identify Functional Dependency

Diagnosis\_Code 🡺 Diagnosis Description

STEP 2: Primary Key: Diagnosis\_Code

Dependencies: Diagnosis Description

STEP 3: This is in BCNF

## Claims\_Details

(Claim\_ID, Service\_Date, Billing\_Amount)

STEP 1: Identify Functional Dependency

Claim\_ID 🡺 Service\_Date, Billing\_Amount

STEP 2: Primary Key: Claim\_ID

Dependencies: Service\_Date, Billing\_Amount

Foreign Keys: Provider\_ID Reference Healthcare\_Providers Table, Diagnosis\_Code Reference Diagnosis\_Details Table

STEP 3: This is in BCNF

## Insurance\_Plans\_has\_Policy\_Holders

(Insurance\_Plans\_Plan\_ID, Policy\_Holders\_PolicyHolder\_ID)

Composite Primary Key: Insurance\_Plans\_Plan\_ID, Policy\_Holders\_PolicyHolder\_ID

Foreign Keys: Plan\_ID reference Insurance\_Plan Table, PolicyHolder\_ID reference Policy\_Holders Table

## Healthcare\_Providers\_has\_Policy\_Holders

(Healthcare\_Provider\_Provider\_ID, Policy\_Holders\_PolicyHolder\_ID)

Composite Primary Key: Healthcare\_Provider\_Provider\_ID, Policy\_Holders\_PolicyHolder\_ID

Foreign keys: Provider\_ID reference Healthcare\_Provider Table, Policy\_Holder\_ID reference Policy Holders Table

# Database Implementation – Queries

/\* Query: 1 - Count of Policy Holders Enrolled in Each Insurance Plan - We can assess the popularity and demand of various plans and use the information to decide which plans to offer and how to modify them. \*/

SELECT IP.Plan\_ID, IP.Plan\_Name, COUNT(ED.POLICY\_HOLDERS\_PolicyHolder\_ID) AS PolicyHolderCount

FROM INSURANCE\_PLANS IP

LEFT JOIN ENROLLMENT\_DETAILS ED ON IP.Plan\_ID = ED.Insurance\_Plans\_Plan\_ID

GROUP BY IP.Plan\_ID, IP.Plan\_Name;

/\*Query 2: Average Billing amount for each healthcare provider - Analysis of the performance of various healthcare providers is possible by computing the average billing amount for each provider. Higher average billing amounts from providers may be a sign that they are providingmore expensive services or handling more complicated situations. \*/

SELECT hp.Provider\_ID, hp.Provider\_Name, AVG(cd.Billing\_Amount) AS AverageBillingAmount

FROM HEALTHCARE\_PROVIDERS hp

JOIN CLAIMS\_DETAILS cd ON hp.Provider\_ID = cd.HEALTHCARE\_PROVIDERS\_Provider\_ID

GROUP BY hp.Provider\_ID, hp.Provider\_Name;

/\* Query 3: Retrieve the policy details of all policyholders along with their enrollment start and end dates - You can get a thorough understanding of the policyholders and the facts of their individual policies. The coverage, expenses, and advantages of various insurance policies can be analyzed and understood using this data. \*/

SELECT ph.Holder\_Name, pd.Deductible, pd.CoPay, pd.OutofPocketMax, ed.Start\_Date, ed.End\_Date

FROM POLICY\_HOLDERS ph

JOIN ENROLLMENT\_DETAILS ed ON ph.PolicyHolder\_ID = ed.POLICY\_HOLDERS\_PolicyHolder\_ID

JOIN POLICY\_DETAILS pd ON ed.Insurance\_Plans\_Plan\_ID = pd.Insurance\_Plans\_Plan\_ID;

/\*Query - 4: Retrieve the policyholder names who have made payments exceeding the average billing amount -

This data can be used by insurance companies or healthcare organizations to examine how their policyholders' payment patterns. It can reveal information about a policyholder's profitability, reveal high-paying clients, or signal potential irregularities in payment patterns. T \*/

SELECT ph.Holder\_Name

FROM POLICY\_HOLDERS ph

JOIN ENROLLMENT\_DETAILS ed ON ph.PolicyHolder\_ID = ed.POLICY\_HOLDERS\_PolicyHolder\_ID

JOIN CLAIMS\_DETAILS cd ON ed.POLICY\_HOLDERS\_PolicyHolder\_ID = cd.HEALTHCARE\_PROVIDERS\_Provider\_ID

JOIN PAYMENT\_DETAILS pd ON cd.Claim\_ID = pd.CLAIMS\_DETAILS\_Claim\_ID

GROUP BY ph.Holder\_Name

HAVING SUM(pd.Amount\_Paid) > (SELECT AVG(cd.Billing\_Amount) FROM CLAIMS\_DETAILS cd);

/\*Query 5 - the total billing amount for each policyholder's claims -

Insurance companies or healthcare providers can learn more about the expenses related to specific policyholders by obtaining the total billing amount for each policyholder. \*/

SELECT ph.Holder\_Name, SUM(cd.Billing\_Amount) AS Total\_Billing\_Amount

FROM POLICY\_HOLDERS ph

JOIN ENROLLMENT\_DETAILS ed ON ph.PolicyHolder\_ID = ed.POLICY\_HOLDERS\_PolicyHolder\_ID

JOIN CLAIMS\_DETAILS cd ON ed.POLICY\_HOLDERS\_PolicyHolder\_ID = cd.HEALTHCARE\_PROVIDERS\_Provider\_ID

GROUP BY ph.Holder\_Name;

/\* Query 6 - Retrieve the provider names and their corresponding policyholders' names for each claim -

Insurance firms can manage and improve their provider networks by looking at the connections between providers and policyholders. They can spot coverage gaps, spots where there are many or few providers, and decide with knowledge whether to add or remove providers from the network. \*/

SELECT hp.Provider\_Name, ph.Holder\_Name

FROM HEALTHCARE\_PROVIDERS hp

JOIN CLAIMS\_DETAILS cd ON hp.Provider\_ID = cd.HEALTHCARE\_PROVIDERS\_Provider\_ID

JOIN ENROLLMENT\_DETAILS ed ON cd.HEALTHCARE\_PROVIDERS\_Provider\_ID = ed.POLICY\_HOLDERS\_PolicyHolder\_ID

JOIN POLICY\_HOLDERS ph ON ed.POLICY\_HOLDERS\_PolicyHolder\_ID = ph.PolicyHolder\_ID;