University of Connecticut School of Business

OPIM 5272

Data Management and Business Process Modeling

Project: Optimization Of Electronic Claim Submission

Group 10

Vinay Kiran Reddy Chinnakondu

Likhitha Guggilla

Nimisha Ekka

Contents

EXECUTIVE SUMMARY

Background

The business process for which we desire to create a model is a combination of Healthcare and Insurance. The healthcare industry consistently strives to improve their business processes to ensure both financial sustainability and high-quality patient care. The healthcare sector primarily depends on precise and effective medical billing and claims processing. This project aims to address key issues related to the submission of Electronic Claims and focuses on optimizing critical processes.

Purpose and Focus

Data Management and business process modeling is essential for accuracy, efficiency, compliance, and security of medical billing and claims processing. It not only ensures that healthcare providers receive timely and accurate reimbursements but also contributes to better patient care by reducing administrative burdens and improving financial transparency. Lack of accuracy and efficiency causes delays in claim processing. Our business model focuses on addressing delays in the insurance claim process within the healthcare sector which impacts patients, healthcare providers, and insurance companies. There are other factors that contribute to this challenge, but our main emphasis is on human errors, it can be data entry errors or code assigning errors.

Solution and Conclusion

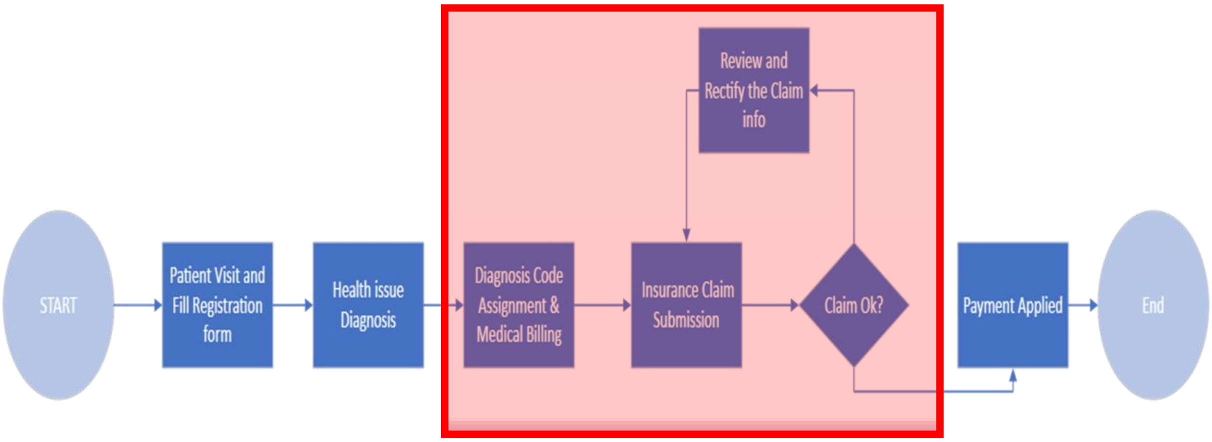
The proposed business model aims to automate the process of assigning codes in the healthcare sector, under the assumption that the insurance procedures are ideal, and all patients possess active life insurance. To proceed with the automation process, it is necessary to build a user interface and implement automation. Additionally, a database is required for the execution of the automation process. We performed database evaluation and implemented appropriate steps to optimize the database. The primary goal is to minimize human errors, hence improving the efficiency of timely and precise reimbursements and patient care. The implementation of the new automated procedure alleviates the administrative costs and addresses the labor-intensive nature of the manual process.

BUSINESS PROCESS ANALYSIS

Business Process Modeling is pivotal in healthcare for streamlining operations, improving patient care, and ensuring compliance. By modeling processes, healthcare institutions can visualize and scrutinize the flow of activities, identify inefficiencies, and develop solutions. In a sector where the accuracy of information and timeliness of service are critical, BPM aids in optimizing resource allocation, reducing wait times, and enhancing the quality of care. Moreover, it supports regulatory compliance and data security, which are vital in handling sensitive patient information. The implementation of effective BPM can lead to significant cost savings and improved patient outcomes in healthcare.

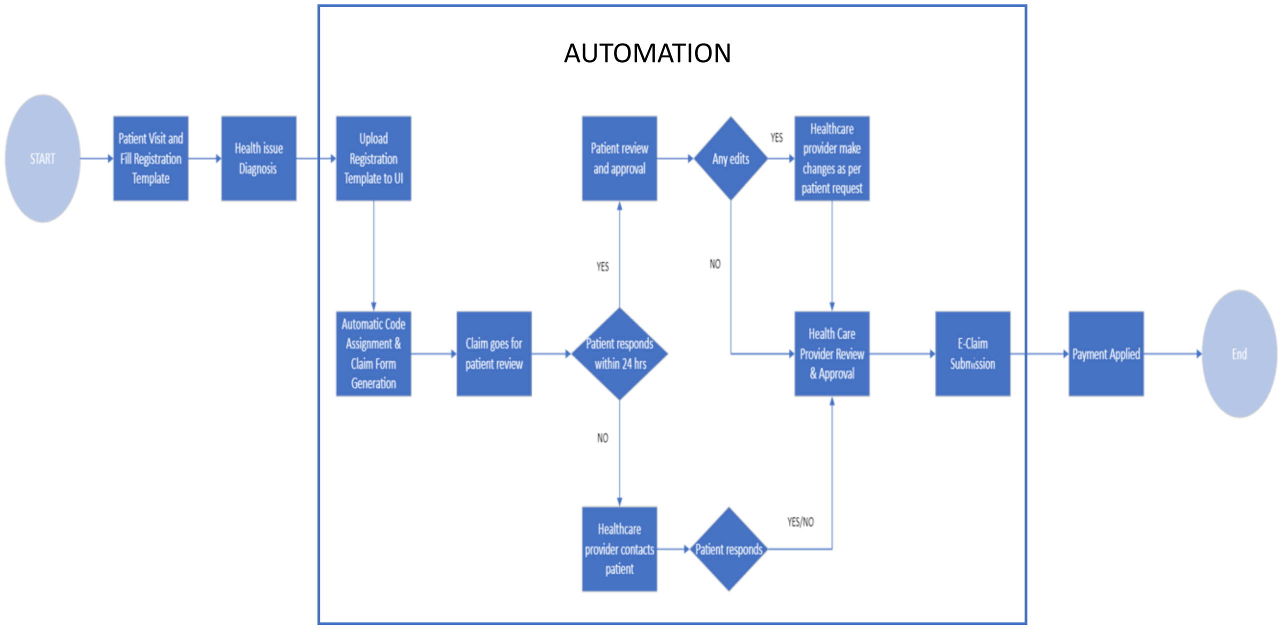
Current Business Process Model

The manual process diagram illustrates the sequential steps involved in the current business model, starting from patient registration, and ending with the application of the final payment. The process commences with the patient's arrival and registration, subsequently leading to the identification and classification of the health condition, allocation of diagnostic codes, and the subsequent invoicing for medical services rendered. Subsequently, the claim is forwarded to the insurance company for a meticulous evaluation to verify the claim's accuracy. Upon approval of the claim, payment is processed; otherwise, it is returned for rectification. The purpose of this cycle is to guarantee accurate processing of claims, although it is susceptible to human errors and inefficiencies. The below flow chart represents the existing business model.


START
Patient Visit and Fill Registration form
Health issue Diagnosis
Diagnosis Code Assignment & Medical Billing
Insurance Claim Submission
Review and Rectify the Claim info
Claim Ok?
Payment Applied
End

Business Process Improvement

The enhanced business process model incorporates automation into the workflow for submitting healthcare claims. Starting with patient registration, the process now encompasses an automated code assignment and claim form generation following the diagnosis. The claim is sent for the patient's review and approval, with a designated timeframe of 24 hours for a response. The healthcare provider addresses any modifications requested by the patient prior to granting final approval. Upon approval, the electronic claim submission is initiated, resulting in the payment process. The primary objective of this streamlined method is to optimize accuracy and quicken the processing of insurance claims. The below flowchart represents the enhanced business process model.

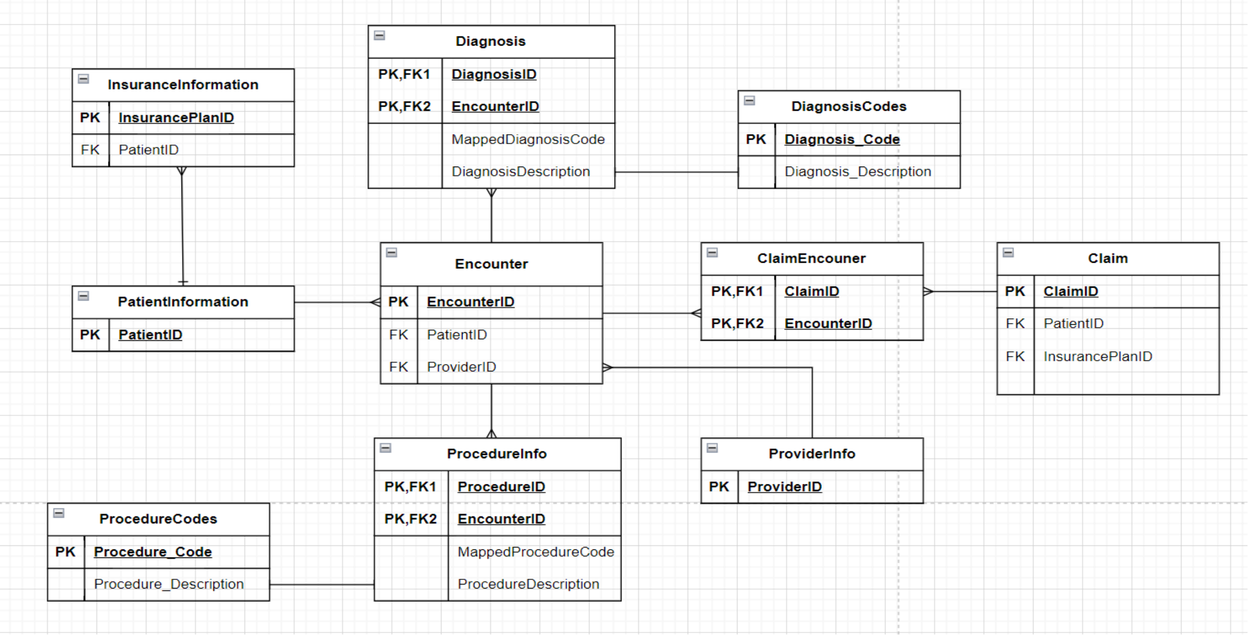

AUTOMATION
START
Patient Visit and Fill Registration Template
Health issue Diagnosis
Upload Registration Template to UI
Automatic Code Assignment & Claim Form Generation
Claim goes for patient review
Patient responds within 24 hrs
YES
NO
Patient review and approval
Any edits
YES
NO
Healthcare provider make changes as per patient request
Healthcare provider contacts patient
Patient responds
YES/NO
Health Care Provider Review & Approval
E-Claim Submission
Payment Applied
End

DATA BASE DESIGN AND IMPLEMENTATION

Data management plays a crucial role in automation processes by ensuring that accurate and relevant data is available to support automated decision-making and operations. In the context of automation, robust data management can streamline workflows, reduce errors, and maintain data integrity. By organizing and maintaining a clean database, automation tools can efficiently retrieve and manipulate necessary information, leading to improved efficiency and effectiveness of automated systems. This is especially important in sectors like healthcare where data accuracy is critical for patient care and safety.

For database implementation, the steps taken were sequential and systematic. The process began with designing an Entity-Relationship (E-R) Model to outline the database structure, followed by the actual creation of the database. After establishing the database, tables were created to organize data into structured formats. Lastly, relevant data was inserted into these tables, laying the groundwork for a functioning database system. This foundational process is critical for supporting data management and the subsequent automation of business processes.

E-R Model



This is the well-structured ER Model that shows the relationship between patient information, healthcare providers, encounters, diagnosis, procedures, and insurance claims.

Database Creation

We created an RDS MariaDB Database and tested the connection using DBeaver. We followed the following steps.

Steps:

1. Created RDS Database Instance

2. Configured RDS for Connection

3. Connected to RDS Database Instance using DBeaver

Tables Creation

To store the necessary data to execute the automation process, we developed a total of 10 tables with the relevant columns. The table names and corresponding columns are listed below.

PatientInformation : To store patient details and featured columns are FirstName, LastName, DateOfBirth, Gender, Address, PhoneNumber, EmailAddress, InsurancePlanID.

InsuranceInformation *:* To store details about the insurance plans associated with patients and featured columns are InsurancePlanID, PatientID, PolicyNumber, PayerName, CoverageSTartDate, CoverageEndDate, CoPaymentAmount, DeductibleAmount, OutOfPocketMaximum.

ProviderInfo **:** To store information related to healthcare providers and featured columns are ProviderID, ProviderName, ProviderType, ProviderNPINumber, TaxIDNumber, ProviderAddress, ContactInformation.

Encounter **:** To store information about each interaction or visit between a patient and a healthcare provider and featured columns are EncounterID, PatientID, ProviderID, DateOfEncounter, TimeOfEncounter, ChiefComplaint, VisitType, BillingStatus.

Claim **:** To store all the relevant details about insurance claims that are submitted following patient encounters with healthcare providers and featured columns are ClaimID, EncounterID, PatientID, InsurancePlanID, DiagnosisCodes, ProcedureCodes, ClaimStatus, ClaimAmount, SubmissionDate, PaymentDate.

ClaimEncounter **:** It is an associate entity that relates Claim and Encounter tables and featured columns are ClaimID, EncounterID.

Diagnosis **:** To store detailed information about the medical diagnoses that patients receive during their healthcare encounters and featured columns are DiagnosisID, EncounterID, MappedDiagnosisCode, DiagnosisDescription, DateOfDiagnosis.

ProcedureInfo **:** To store comprehensive details about medical procedures performed during patient encounters and featured columns are ProcedureID, EncounterID, MappedProcedureCode, ProcedureDescription, ProcedureDate, ProcedureFee.

Diagnosiscodes **:** It is a reference table that stores standardized codes for various medical diagnoses, along with their descriptions and featured columns are DiagnosisCode, DiagnosisDescription.

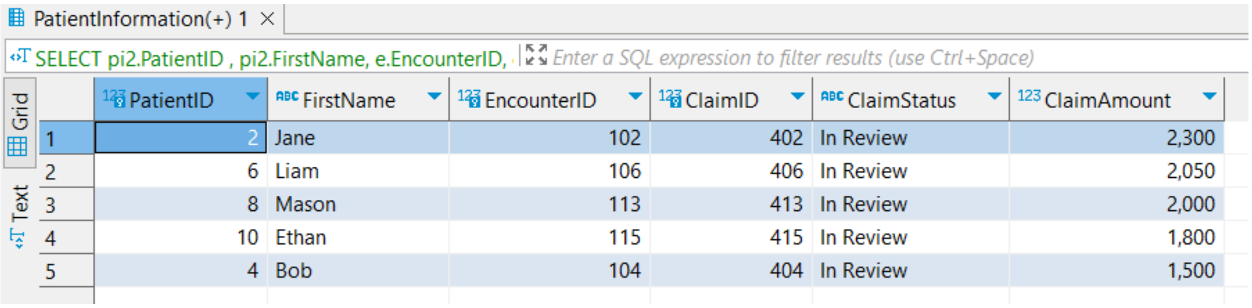
Procedurecodes **:** It is a reference table that stores standardized codes and descriptions for various medical procedures and featured columns are ProcedureCode, ProcedureDescription.

Data Insertion

To test the database and the relationships between the tables, we manually entered 11 records into each table and extracted data for the following question.

Write a sql query to generate report giving patient details whose claim amount is greater than $1000 and status is still in Review?

```
SELECT pi2.PatientID, pi2.FirstName, e.EncounterID, c.ClaimID, c.ClaimStatus,
c.ClaimAmount
FROM PatientInformation pi2
JOIN Encounter e
ON pi2.PatientID =e.PatientID
JOIN ClaimEncounter ce
ON ce.EncounterID = e.EncounterID
JOIN Claim c
ON ce.ClaimID  = c.ClaimID
WHERE c.ClaimAmount > 1000
       AND c.ClaimStatus LIKE 'In Review'
ORDER BY c.ClaimAmount DESC;
```


PatientInformation(+) 1 ×
SELECT pi2.PatientID , pi2.FirstName, e.EncounterID,
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text

	PatientID	FirstName	EncounterID	ClaimID	ClaimStatus	ClaimAmount
1	2	Jane	102	402	In Review	2,300
2	6	Liam	106	406	In Review	2,050
3	8	Mason	113	413	In Review	2,000
4	10	Ethan	115	415	In Review	1,800
5	4	Bob	104	404	In Review	1,500

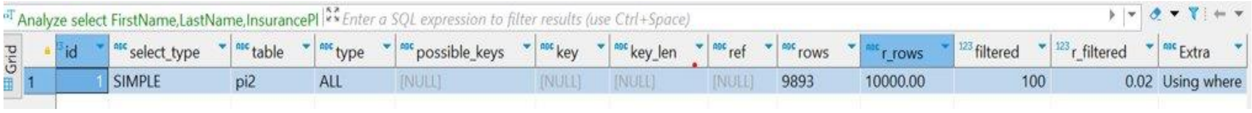
DATABASE PERFORMANCE EVALUATIONS AND IMPLICATIONS

A crucial procedure that evaluates a database system's effectiveness, speed, and dependability is database performance evaluation. The importance of database performance in automation processes is paramount, as databases are often the backbone of automated systems, handling a vast array of data transactions and queries. Database performance evaluation is not a one-time task but a continuous process that involves regular monitoring, tuning, and updating of the database system to meet evolving requirements and to handle growing data volumes efficiently.

To evaluate the performance of the database, we imported 10,000 records into the database and analyzed a few queries and extracted data.

Query-1

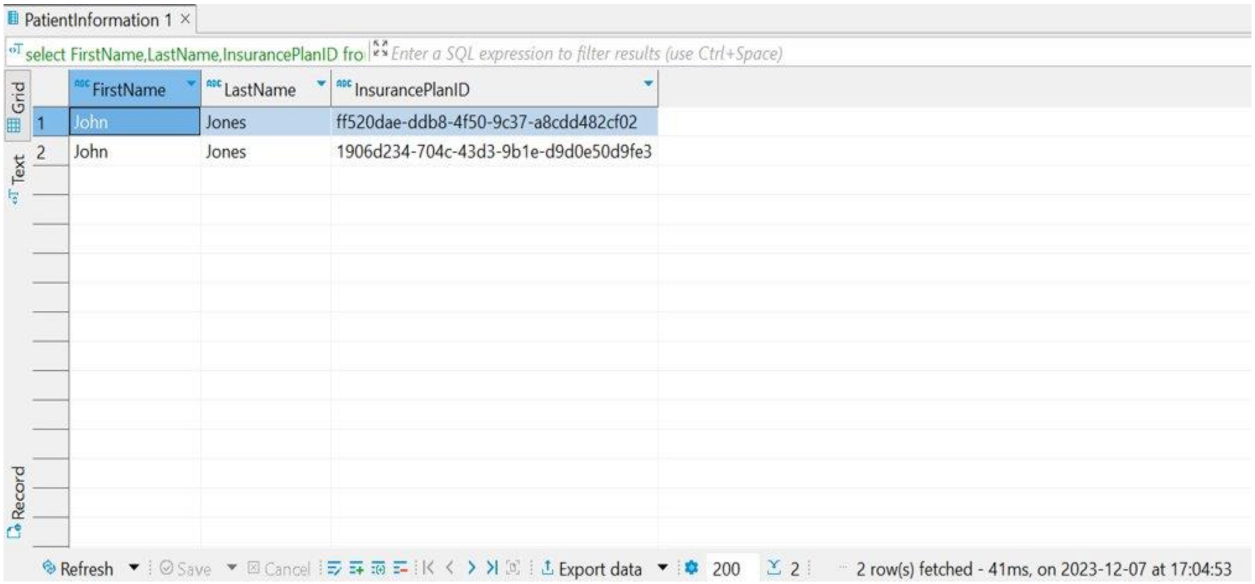
```
ANALYZE SELECT FirstName,LastName,InsurancePlanID
FROM PatientInformation pi2
WHERE FirstName LIKE "John";
```



Analyze select FirstName,LastName,InsurancePl

	id	select_type	table	type	possible_keys	key	key_len	ref	rows	r_rows	filtered	r_filtered	Extra
1	1	SIMPLE	pi2	ALL	[NULL]	[NULL]	[NULL]	[NULL]	9893	10000.00	100	0.02	Using where

We analyzed the above query, and it undergoes full table scan of 10,000 tables which took around 41ms to extract the report. The result is mentioned below.



PatientInformation 1 ×

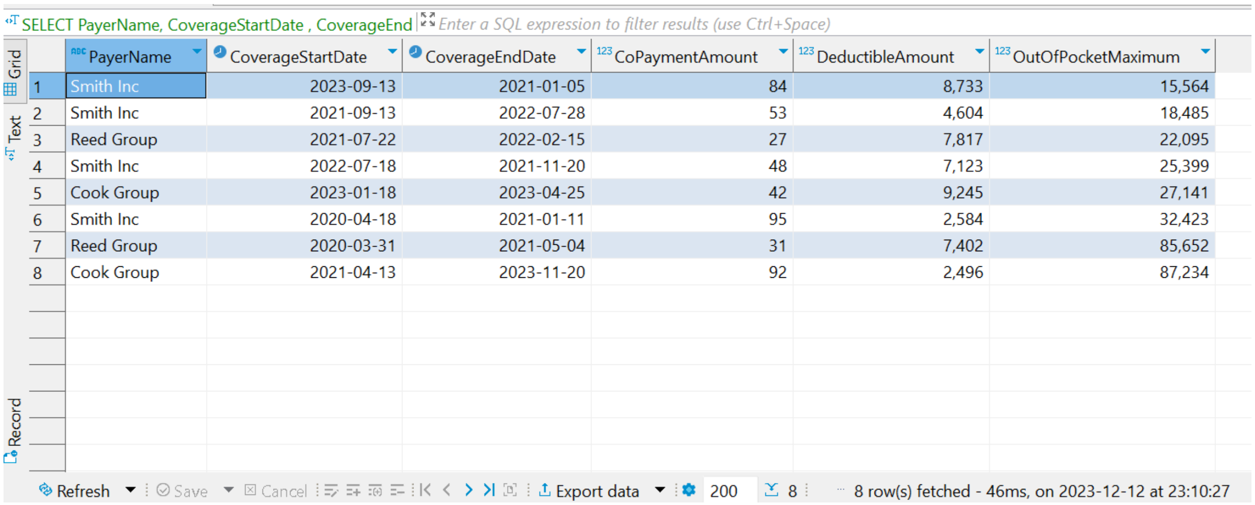
select FirstName,LastName,InsurancePlanID fro | Enter a SQL expression to filter results (use Ctrl+Space)

	FirstName	LastName	InsurancePlanID
1	John	Jones	ff520dae-ddb8-4f50-9c37-a8cdd482cf02
2	John	Jones	1906d234-704c-43d3-9b1e-d9d0e50d9fe3

2 row(s) fetched - 41ms, on 2023-12-07 at 17:04:53

Query - 2

```
SELECT PayerName, CoverageStartDate , CoverageEndDate , CoPaymentAmount ,
DeductibleAmount , OutOfPocketMaximum
FROM InsuranceInformation ii
JOIN PatientInformation pi2
ON ii.PatientID = pi2.PatientID
WHERE PayerName IN ("Home PLC","Cook Group","SMith Inc","Reed Group")
      AND OutOfPocketMaximum > 15000
ORDER BY OutOfPocketMaximum ;
```



SELECT PayerName, CoverageStartDate , CoverageEnd | Enter a SQL expression to filter results (use Ctrl+Space)

	PayerName	CoverageStartDate	CoverageEndDate	CoPaymentAmount	DeductibleAmount	OutOfPocketMaximum
1	Smith Inc	2023-09-13	2021-01-05	84	8,733	15,564
2	Smith Inc	2021-09-13	2022-07-28	53	4,604	18,485
3	Reed Group	2021-07-22	2022-02-15	27	7,817	22,095
4	Smith Inc	2022-07-18	2021-11-20	48	7,123	25,399
5	Cook Group	2023-01-18	2023-04-25	42	9,245	27,141
6	Smith Inc	2020-04-18	2021-01-11	95	2,584	32,423
7	Reed Group	2020-03-31	2021-05-04	31	7,402	85,652
8	Cook Group	2021-04-13	2023-11-20	92	2,496	87,234

8 row(s) fetched - 46ms, on 2023-12-12 at 23:10:27

In this query, it scanned around 10,000 records to get 8 records and took 46ms long to generate the report.

Query - 3

```
ANALYZE SELECT pi2.ProviderID ,pi2.ProviderName ,pi2.ProviderType,
e.DateOfEncounter, e.TimeOfEncounter , e.BillingStatus
FROM ProviderInfo pi2
JOIN Encounter e
ON e.ProviderID =pi2.ProviderID
WHERE ProviderName IN ('Abbott Group','Acosta, Gomez and Bowen','Martin-
Zimmerman')
     AND BillingStatus = 'Not Billed'
```

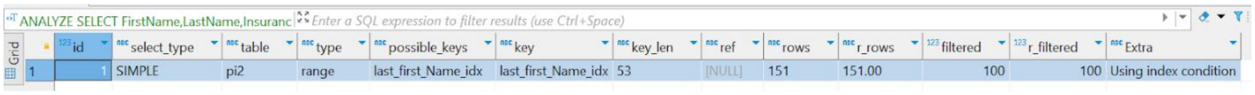
The above query took around 51ms and underwent full table scan of 10K records to generate 3 records which is quite long.

DATABASE PERFORMANCE OPTIMIZATION

Database optimization refers to a range of strategies aimed at improving the performance and efficiency of a database system. It's a crucial aspect of database management, particularly important for systems handling large volumes of data or complex queries. There are many methods in optimizing the database, but we choose indexing. We created appropriate indexes and executed the above queries to show the difference of efficiency of the database.

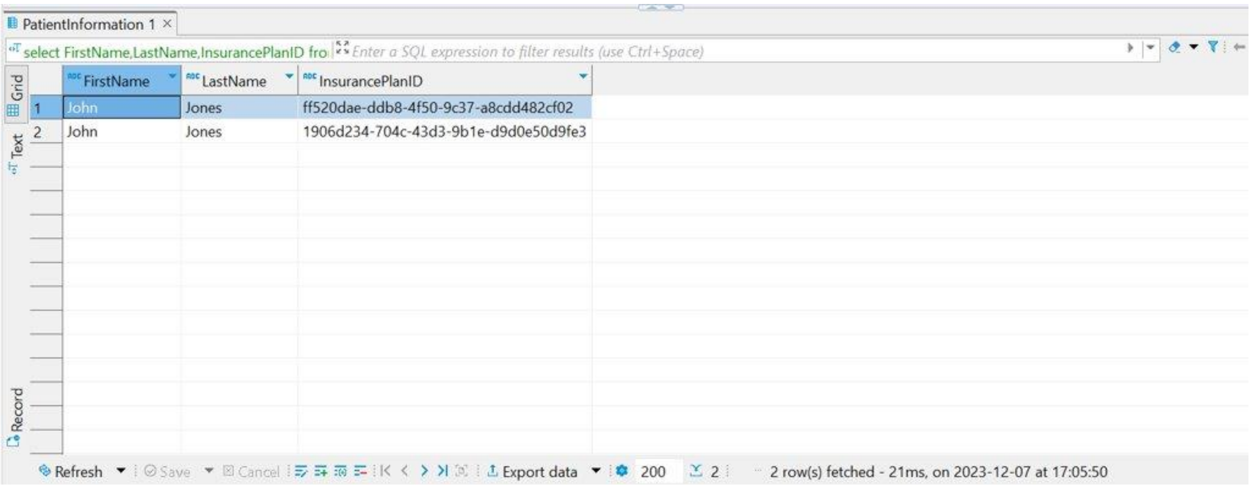
Query-1

```
CREATE INDEX last_first_Name_idx ON PatientInformation(FirstName,LastName);
```



ANALYZE SELECT FirstName,LastName,Insuranc

id	select_type	table	type	possible_keys	key	key_len	ref	rows	r_rows	filtered	r_filtered	Extra
1	SIMPLE	pi2	range	last_first_Name_idx	last_first_Name_idx	53	[NULL]	151	151.00	100	100	Using index condition



PatientInformation 1

select FirstName,LastName,InsurancePlanID fro

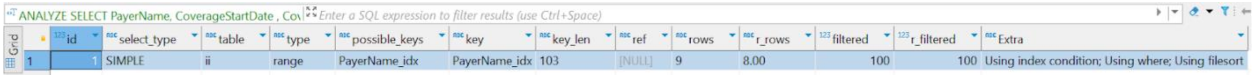
	FirstName	LastName	InsurancePlanID
1	John	Jones	ff520dae-ddb8-4f50-9c37-a8cdd482cf02
2	John	Jones	1906d234-704c-43d3-9b1e-d9d0e50d9fe3

2 row(s) fetched - 21ms, on 2023-12-07 at 17:05:50

After creating index on FirstName and LastName, we can observe that just 151 records were scanned to generate the report and it took around 21ms which shows the optimization of the database.

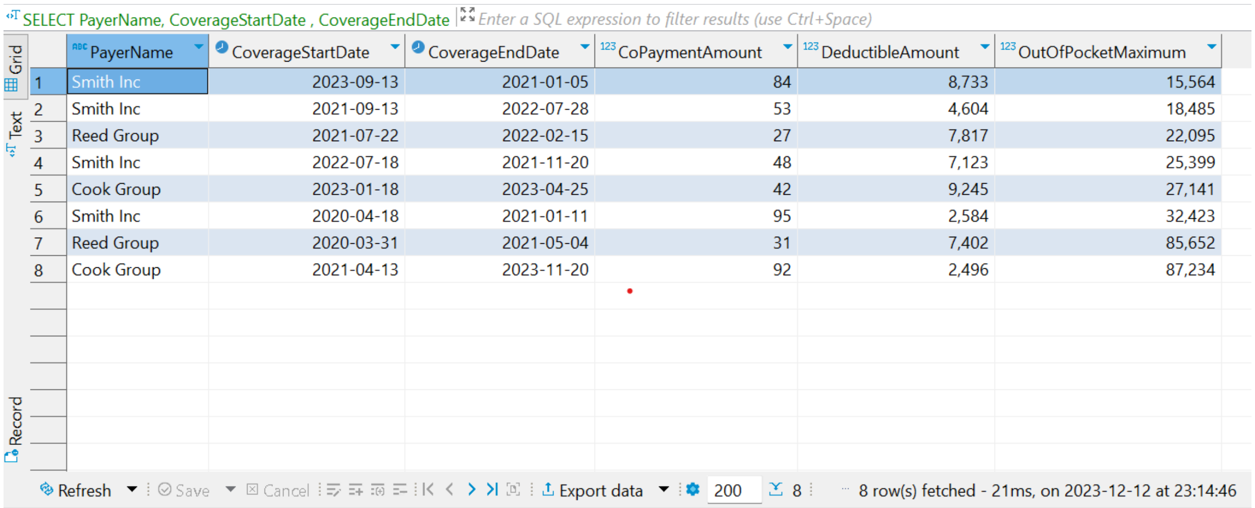
Query – 2

```
CREATE INDEX PayerName_idx ON InsuranceInformation(PayerName);
```



ANALYZE SELECT PayerName, CoverageStartDate , Cov

	id	select_type	table	type	possible_keys	key	key_len	ref	rows	r_rows	filtered	r_filtered	Extra
1	1	SIMPLE	ii	range	PayerName_idx	PayerName_idx	103	[NULL]	9	8.00	100	100	Using index condition; Using where; Using filesort



SELECT PayerName, CoverageStartDate , CoverageEndDate

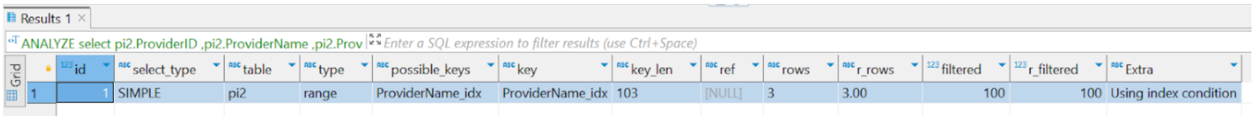
	PayerName	CoverageStartDate	CoverageEndDate	CoPaymentAmount	DeductibleAmount	OutOfPocketMaximum
1	Smith Inc	2023-09-13	2021-01-05	84	8,733	15,564
2	Smith Inc	2021-09-13	2022-07-28	53	4,604	18,485
3	Reed Group	2021-07-22	2022-02-15	27	7,817	22,095
4	Smith Inc	2022-07-18	2021-11-20	48	7,123	25,399
5	Cook Group	2023-01-18	2023-04-25	42	9,245	27,141
6	Smith Inc	2020-04-18	2021-01-11	95	2,584	32,423
7	Reed Group	2020-03-31	2021-05-04	31	7,402	85,652
8	Cook Group	2021-04-13	2023-11-20	92	2,496	87,234

8 row(s) fetched - 21ms, on 2023-12-12 at 23:14:46

After creating index on PayerName, we can observe that just 8 records were scanned to generate the report and it took around 21ms

Query – 3

```
CREATE INDEX ProviderName_idx ON ProviderInfo(ProviderName);
```



Results 1

ANALYZE select pi2.ProviderID ,pi2.ProviderName ,pi2.Prov

	id	select_type	table	type	possible_keys	key	key_len	ref	rows	r_rows	filtered	r_filtered	Extra
1	1	SIMPLE	pi2	range	ProviderName_idx	ProviderName_idx	103	[NULL]	3	3.00	100	100	Using index condition

After creating an index on ProviderName, we can observe that just 3 records were scanned to generate the report and it took around 19ms.

After executing these queries before and after indexing we can clearly observe the difference in execution time which proves the enhancement in efficiency and optimization of the database.

SOURCES

Medical Billing and Claims Processing:

Alice Anne Andress. (2019). Healthcare Billing and Management. Jones & Bartlett Learning. This book provides comprehensive insights into healthcare billing and management, emphasizing the importance of accurate billing practices.

Automation in Healthcare:

Edward H. Shortliffe, and James J. Cimino (Eds.). (2020). Biomedical Informatics: Computer Applications in Health Care and Biomedicine. Springer. This textbook discusses the application of informatics inhealthcare, including automation of various healthcare processes.

Business Process Modeling in Healthcare:

Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A. Reijers. (2018). Fundamentals of Business Process Management. Springer. Although not healthcare-specific, this book is a key resource in understanding business process modeling and improvement.

Database Design and Implementation in Healthcare:

Thomas M. Connolly, and Carolyn E. Begg. (2014). Database Systems: A Practical Approach to

Design, Implementation, and Management. Pearson. This textbook offers a thorough overview of database design and implementation, relevant to the healthcare sector's data management needs.

Database Performance Evaluation and Optimization:

Sam S. Lightstone, Toby J. Teorey, and Tom Nadeau. (2011). Physical Database Design: The Database Professional's Guide to Exploiting Indexes, Views, Storage, and More. Morgan Kaufmann. It provides practical strategies for database performance optimization, relevant for automated systems in healthcare.