[Date]

Dheeraj Krishna Banna

[company name]

[Document title]

[Document subtitle]

Contents

[Introduction 2](#_Toc134025820)

[Requirements 2](#_Toc134025821)

[Entities: 3](#_Toc134025822)

[Entities and their attributes: 3](#_Toc134025823)

[ER Diagram: 4](#_Toc134025824)

[Database Design: 5](#_Toc134025825)

[Creating Tables with Columns 5](#_Toc134025826)

[Data Integrity 9](#_Toc134025827)

[Data Dictionary: 9](#_Toc134025828)

[Data generation and Loading: 12](#_Toc134025829)

[Performance Tuning: 13](#_Toc134025830)

[Indexing 13](#_Toc134025831)

[Query Writing: 16](#_Toc134025832)

[Data visualisation: 18](#_Toc134025833)

[Conclusion: 18](#_Toc134025834)

# Introduction

A hospital database is a comprehensive and organized collection of information related to a hospital's operations, patient care, medical procedures, and other healthcare-related activities. The database is designed to manage and store large amounts of data securely, accurately, and efficiently. It serves as a critical tool for healthcare providers to access and manage patient records, medical histories, diagnostic test results, and other relevant information to provide high-quality care.

The hospital database includes various modules, such as patient registration, medical history, appointment scheduling, pharmacy, laboratory, radiology, billing, and more. The database can be accessed by authorized healthcare personnel, including doctors, nurses, administrative staff, and other medical professionals, to retrieve and update patient records, track medical procedures, and monitor the progress of patients' treatment.

With the help of a hospital database, healthcare providers can improve patient care, reduce medical errors, increase efficiency, and better manage resources. It is a critical component of modern healthcare systems and plays a vital role in delivering quality healthcare services.

# Requirements

Claim table: This table gives the information about the people and what they claimed for about their medical condition.

Degree Table: This table gives the information about the doctor’s type and level of education degree. In which degree they have been specialized.

Department Table: This table gives the information about doctors and patients assigned to different type of medical departments like Neurology, Nephrology, Cardiology etc.

Doctor Table: This table gives the information about All the information about doctor like Name, Contact Number, Address, Department etc.

Insurance company Table: This table gives the information about name, Type and level of insurance company and the patient associated with that company.

Nurse Table: This table gives the information about name and degree and the information of the nurses.

Operation Table: This table gives the information about the patient’s name and name of the operation and the doctor performing the operation.

Operation Type Table: This table gives the information about the operation type and the cost of the operation.

Patient Table: This table gives all the information about the patient like Name of the patient and Contact Number, Address Etc.

# Entities:

* Claim
* Degree
* Department
* Doctor
* Insurance\_Company
* Nurse
* Operation
* Operatin\_Type
* Patient

# Entities and their attributes:

* Claim
  + Claim\_Number
  + Company\_Name
  + Patient\_Number
  + Diagnosis
  + Amount
* Degree
  + Doctor\_Number
  + Degree\_type
  + Major
  + University
  + Year\_Grad
* Department
  + Departmennt\_number
  + Department\_Office\_Number
  + Department\_Telephone
* Doctor
  + Department\_Number
  + Doctor\_number
  + Doctor\_FirstName
  + Doctor\_LastName
  + Doctor\_Office\_Number
  + Doctor\_Telephone
  + BackUp\_Doc
* Insurance\_Company
  + Company\_Name
  + Telephone
  + City
  + State
* Nurse
  + Department\_Number
  + Nurse\_Number
  + Nurse\_FirstName
  + Nurse\_LastName
  + Certification
  + Year\_Hired
  + Supervised\_ID
* Operation
  + Patient\_Number
  + Doctor\_Number
  + Operation\_Name
  + Date\_of\_op
  + Start\_Time
  + End\_Time
  + Operating\_Room\_no
* Operatin\_Type
  + Operation\_Name
  + Duration
  + Cost
* Patient
  + Patient\_Number
  + Patient\_FirstName
  + Patient\_LastName
  + House
  + Street
  + City
  + State

# ER Diagram:

# Database Design:

## Creating Tables with Columns

Below is the query we used to create tables and columns. Each table has a primary given

CREATE TABLE Department(

Department\_Number Numeric(3,0) identity(100,1) NOT NULL,

Department\_Office\_Number Numeric(3,0) NOT NULL,

Department\_Telephone Varchar(15) NOT NULL,

CONSTRAINT PK\_Department\_Number PRIMARY KEY (Department\_Number)

);

CREATE TABLE Nurse(

Department\_Number Numeric(3,0) NOT NULL,

Nurse\_Number Numeric(10,0) identity(1000000000,1) NOT NULL,

Nurse\_FirstName Varchar(100) NOT NULL,

Nurse\_LastName Varchar(100) NOT NULL,

Certification Varchar(100) NOT NULL,

Year\_Hired Datetime NOT NULL,

Supervisor\_ID Numeric(10,0),

CONSTRAINT PK\_Nurse\_Number PRIMARY KEY (Nurse\_Number),

CONSTRAINT FK\_Department\_Number FOREIGN KEY (Department\_Number) REFERENCES Department(Department\_Number),

CONSTRAINT FK\_Supervisor FOREIGN KEY (Supervisor\_ID) REFERENCES Nurse(Nurse\_Number)

);

CREATE TABLE Doctor(

Department\_Number Numeric(3,0) NOT NULL,

Doctor\_Number Numeric(10,0)identity(1000000000,1) NOT NULL,

Doctor\_FirstName Varchar(100) NOT NULL,

Doctor\_LastName Varchar(100) NOT NULL,

Doctor\_Office\_Number Numeric(3,0),

Doctor\_Telephone Varchar(15) NOT NULL,

BackUp\_Doc Numeric(10,0)

CONSTRAINT PK\_Doctor\_Number PRIMARY KEY (Doctor\_Number),

CONSTRAINT FK\_Department\_Number\_Doc FOREIGN KEY (Department\_Number) REFERENCES Department(Department\_Number),

CONSTRAINT FK\_BackupDoctor FOREIGN KEY (BackUp\_Doc) REFERENCES Doctor(Doctor\_Number)

);

CREATE TABLE Degree(

Doctor\_Number Numeric(10,0) NOT NULL,

Degree\_Type VarChar(10) NOT NULL,

Major Varchar(60) NOT NULL,

University Varchar(80) NOT NULL,

Year\_Grad date ,

CONSTRAINT PK\_Doctor\_Number\_and\_Degree\_Type PRIMARY KEY (Doctor\_Number,Degree\_Type),

CONSTRAINT FK\_Doctor\_Number FOREIGN KEY (Doctor\_Number) REFERENCES Doctor(Doctor\_Number)

);

CREATE TABLE Patient(

Patient\_Number Numeric(15,0) identity(100000000000000,1) NOT NULL,

Patient\_FirstName Varchar(50) NOT NULL,

Patient\_LastName VarChar(50) NOT NULL,

House VarChar (50) NOT NULL,

Street VarChar (40) NOT NULL,

City VarChar (40) NOT NULL,

State\_ VarChar (40)

CONSTRAINT PK\_Patient\_Number PRIMARY KEY (Patient\_Number)

);

CREATE TABLE Operation\_Type(

Operation\_Name VarChar (100) NOT NULL,

Duration Numeric (2,2) NOT NULL,

Cost Numeric (7,3) NOT NULL,

CONSTRAINT PK\_Operation\_Name PRIMARY KEY (Operation\_Name)

);

CREATE TABLE Operation(

Patient\_Number Numeric (15,0) NOT NULL,

Doctor\_Number Numeric (10,0) NOT NULL,

Operation\_Name Varchar(100) NOT NULL,

Date\_of\_Op Datetime NOT NULL,

Start\_Time Time NOT NULL,

End\_Time Time NOT NULL,

Operating\_Room\_No Numeric(5,0)NOT NULL,

CONSTRAINT PK\_Patient\_Number\_Composite PRIMARY KEY (

Patient\_Number,Doctor\_Number,Operation\_Name,Date\_of\_Op),

CONSTRAINT FK\_Patient\_FK\_ FOREIGN KEY (Patient\_Number) REFERENCES Patient(Patient\_Number),

CONSTRAINT FK\_Doctor\_Number\_\_ FOREIGN KEY (Doctor\_Number) REFERENCES Doctor(Doctor\_Number),

CONSTRAINT FK\_Operation\_Name FOREIGN KEY (Operation\_Name) REFERENCES Operation\_Type(Operation\_Name),

);

CREATE TABLE Insurance\_Company(

Company\_Name VarChar (70),

Telephone VARCHAR (15),

City VarChar (40),

State\_ VarChar (40),

CONSTRAINT PK\_Company\_Name PRIMARY KEY (Company\_Name)

);

CREATE TABLE Claim(

Claim\_Number Int IDENTITY (1,1) NOT NULL,

Company\_Name VarChar (70) NOT NULL,

Patient\_Number Numeric (15,0) NOT NULL,

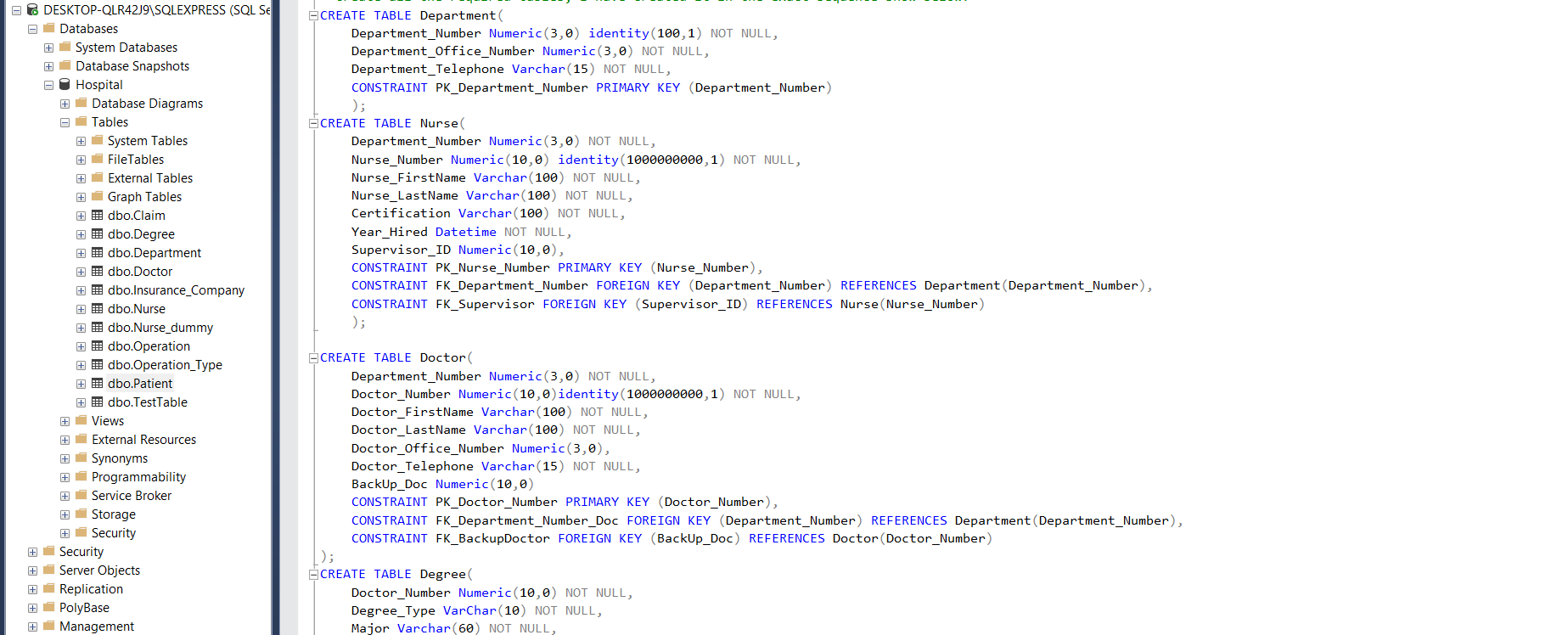
Diagnosis VarChar(80) NOT NULL,

Amount Numeric (7,3),

CONSTRAINT PK\_Claim\_Number PRIMARY KEY (Claim\_Number),

CONSTRAINT FK\_Patient\_Number\_\_ FOREIGN KEY (Patient\_Number) REFERENCES Patient(Patient\_Number),

CONSTRAINT FK\_Company\_Name\_\_ FOREIGN KEY (Company\_Name) REFERENCES Insurance\_Company(Company\_Name)





A new table in the database can be created by using CREATE command and when a new

table is created by utilizing an existing table, the new table will be loaded with the old

table's existing values.

## Data Integrity

Data Integrity refers to the consistency and maintenance of the data through the life cycle of the database. In a database, data integrity can be ensured through the implementation of Integrity Constraints in a table. Integrity constraints help apply business rules to the database tables. The constraints can either be at a column level or a table level. Some of the most common constraints are,

• NOT NULL – Prevents a column from having a NULL value.

• PRIMARY KEY – Uniquely identifies each row or record in table.

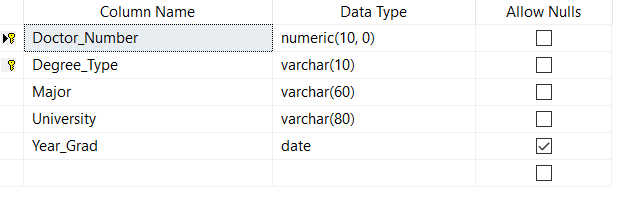
• FOREIGN KEY – Uniquely identifies a column that references a PRIMARY KEY in another table.

• UNIQUE – Prevents a column from having duplicate values.

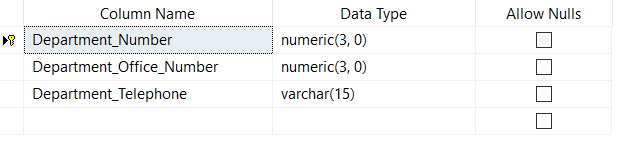
• CHECK – Checks for values that satisfy a specific condition as defined by the user.

## Data Dictionary:

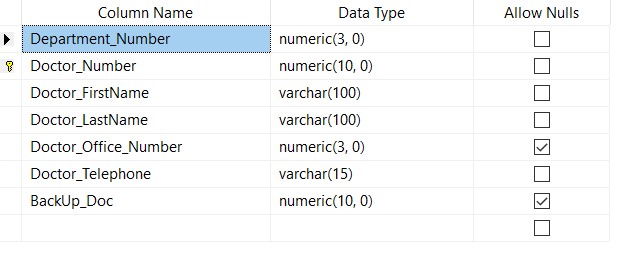
The Data Dictionary for the Doctor Table



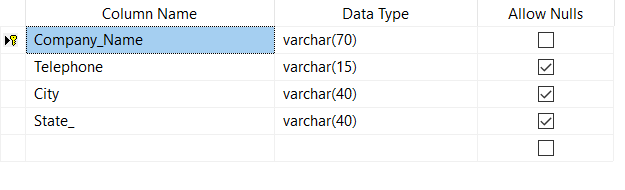
The Data Dictionary for the Department Table



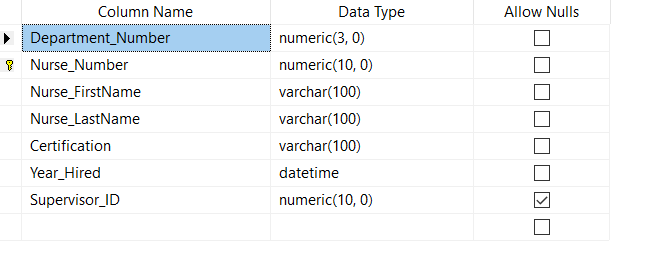
The Data Dictionary for the Doctor Table



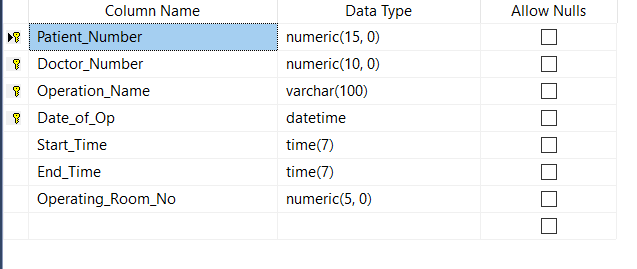
The Data Dictionary for the Insurance\_Company Table



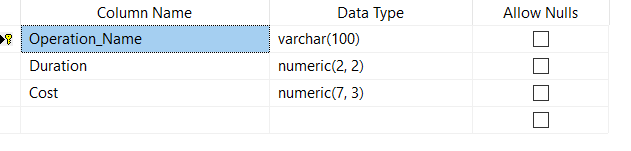
The Data Dictionary for the Nurse Table



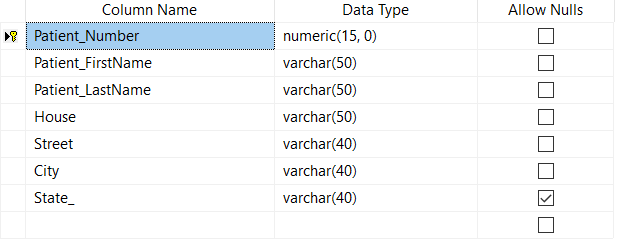
The Data Dictionary for the Operation Table



The Data Dictionary for the Operation\_Type Table



The Data Dictionary for the Patient Table



## Data generation and Loading:

--Insert data in Table Nurse

insert into Nurse values(110,'atif','aslam','Acute care Nurse Practitioner','2013-06-26',1000000010);

--Insert data in TABLE Doctor

insert into Doctor values(104,'Adam','Levigne',330,12343252345,1000000003)

--Insert data in TABLE Degree

insert into Degree values(1000000012,'Med','Dental','Harvard University',2012-03-06);

--Insert data in TABLE Operation\_Type

insert into dbo.Operation\_Type values('Carotid2',.9,34.32);

--Insert data in TABLE Patient

insert into Patient values('Charles','Darwin',34,'jayson str','Miami','Florida');

--Insert data in TABLE Patient

insert into Patient values('Charles','Darwin',34,'jayson str','Miami','Florida');

--Insert data in TABLE Operation

INSERT INTO DBO.Operation VALUES (100000000000006,1000000004,'Dilation and curettage',

'2018-05-19',

'12:12:43','03:09:06',735);

--Insert data in TABLE Insurance\_Company

insert into Insurance\_Company values('LifeInsur',8378298748,'Boston','Massachusets');

--Insert data in TABLE Claim

insert into Claim values ('Highmark Group',100000000000003,'Dental',127.23);

# Performance Tuning:

## Indexing

Table-1: “Nurse” Table in Hospital

Table-2: “Nurse\_Dummy” Table in Hospital

Created a dummy table named Nurse\_Dummy.

Query: CREATE TABLE Nurse\_dummy (

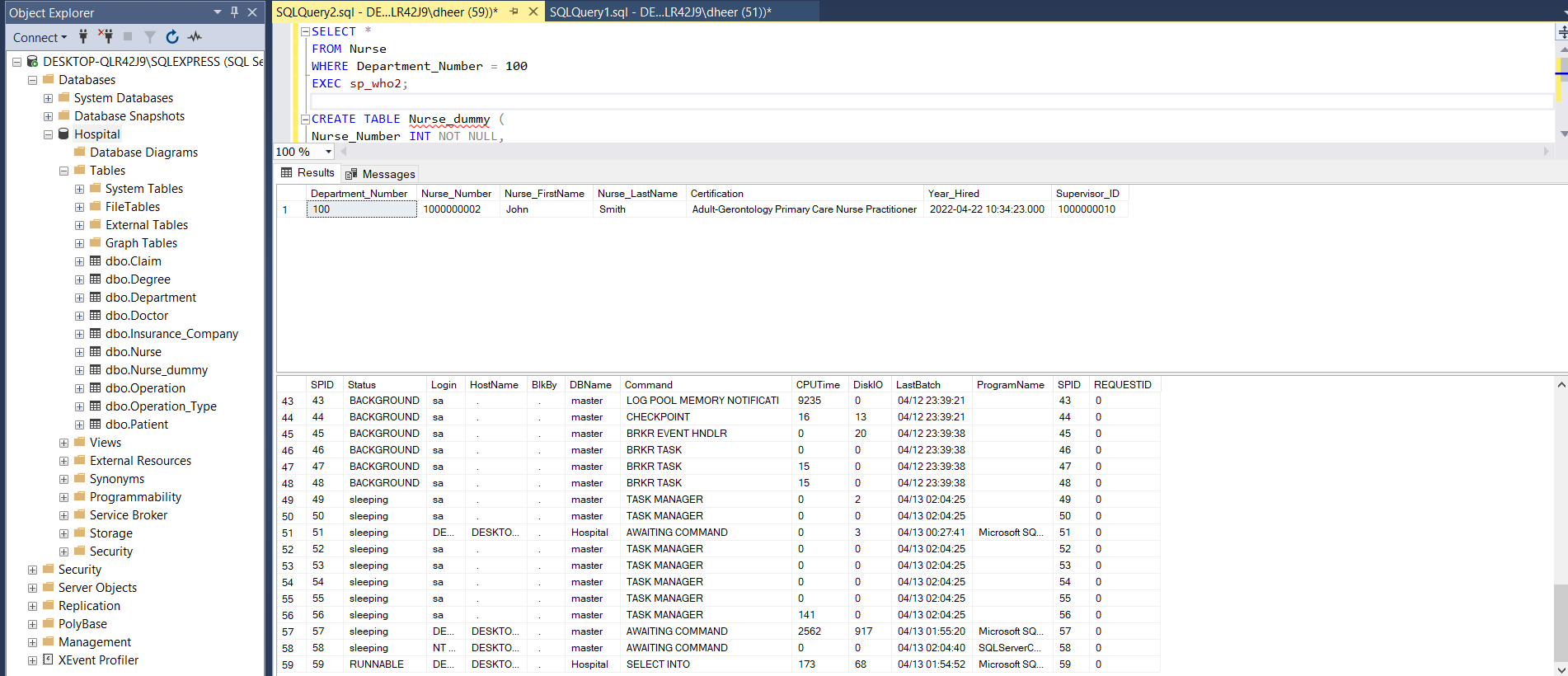
Nurse\_Number INT NOT NULL,

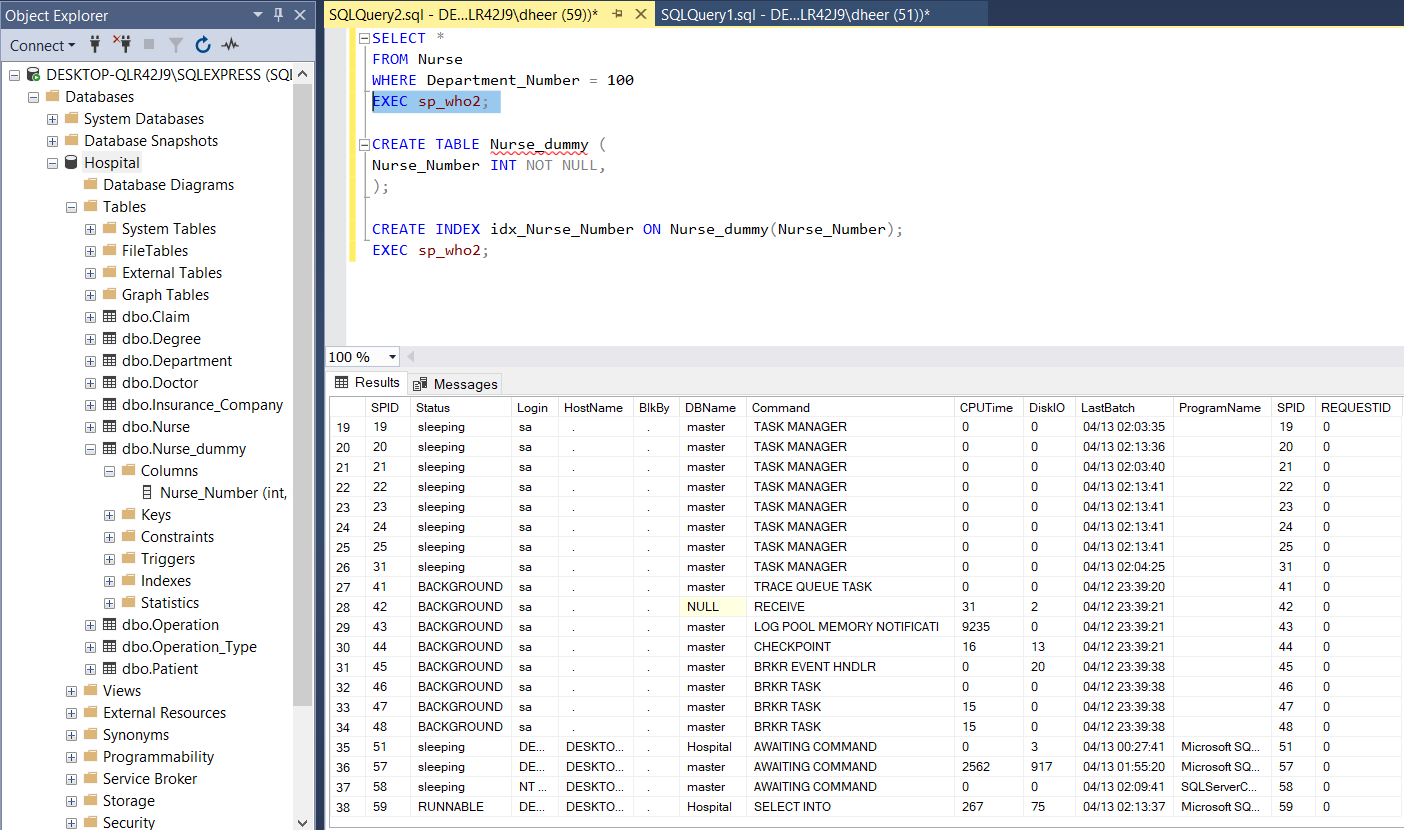
);

Create an Index on “Nurse\_Number” in the new table “Nurse\_Dummy”

CREATE INDEX idx\_Nurse\_Number ON Nurse\_dummy(Nurse\_Number);

TO Get the CPUTime we used a query called EXEC sp\_who2;

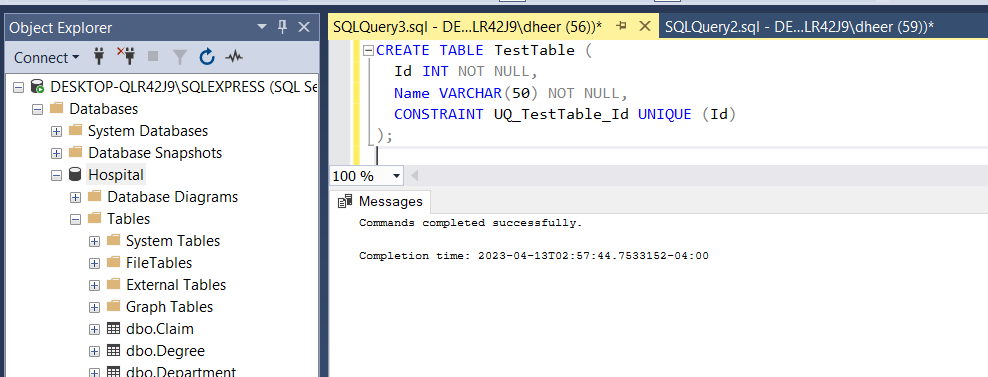




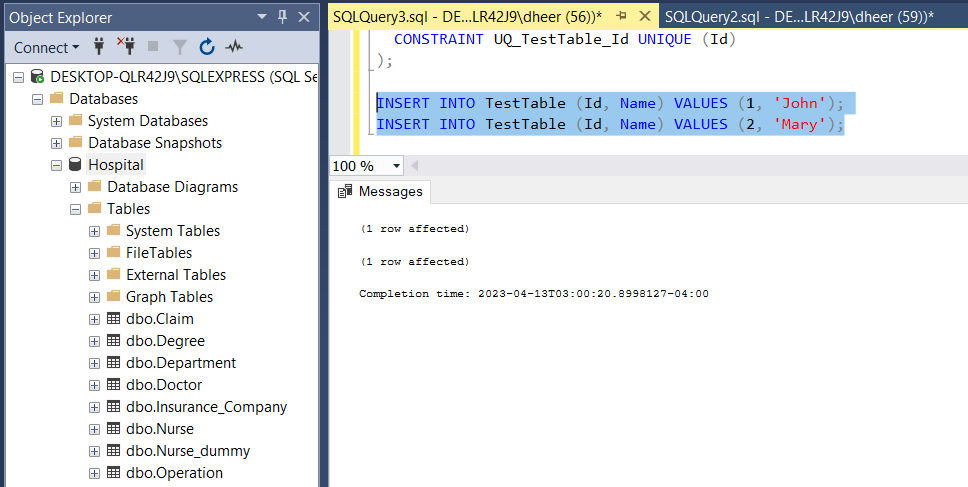
From the above, we see that when index is created on “Nurse\_Number” column, for a query that requires accessing “Nurse\_Number” column, the cost (% CPU) for the 2nd picture which is less than 1st picture for the same query for the table without index.

For the next to find the performance we have created a test table and inserted some sample data into so that we can be able to check the performance before adding a primary key and below are the some of the results.

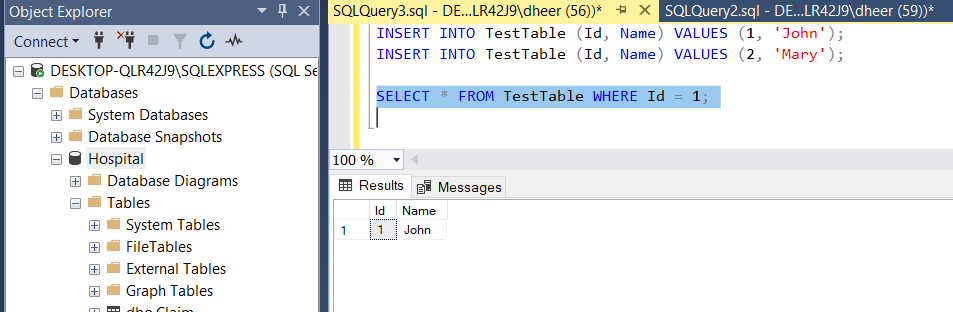
1. Created a table with a unique constraint:



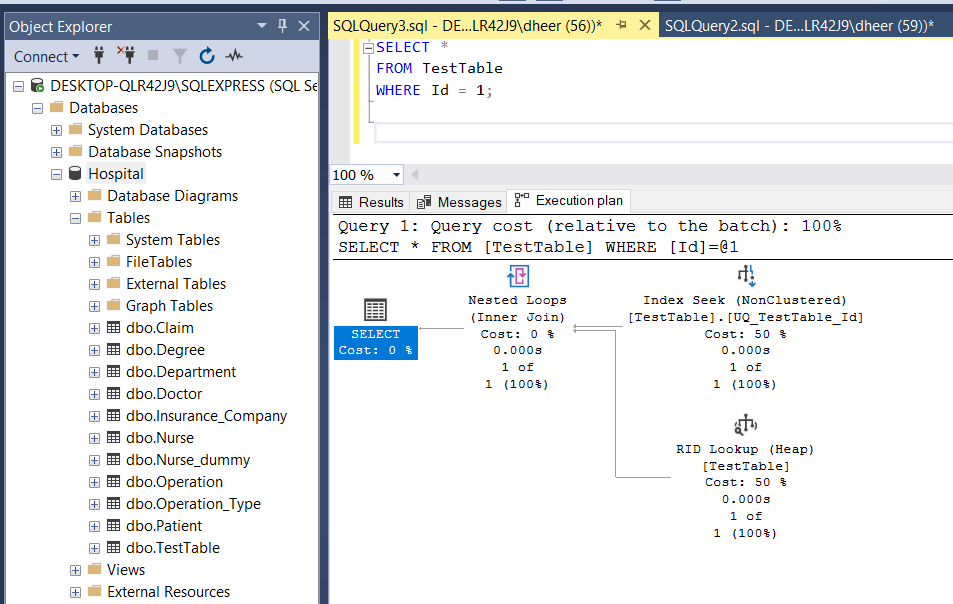
1. Inserted some sample data:



1. Run a SELECT query and examine the execution plan

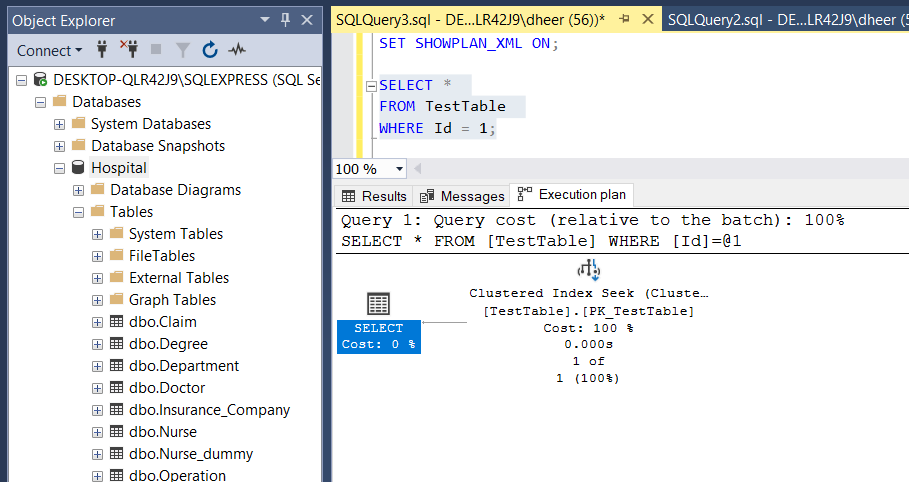


1. Before creating the primary key



1. After adding a primary key constraint to the table using a query:

ALTER TABLE TestTable ADD CONSTRAINT PK\_TestTable PRIMARY KEY CLUSTERED (Id);



Thus, we can observe that TestTable\_Id with no primary key added has more cost when compared to TestTable\_Id with Primary key. So, the preferred one is option with primary key.

# Query Writing:

**Total number of doctors studied at Harvard University**

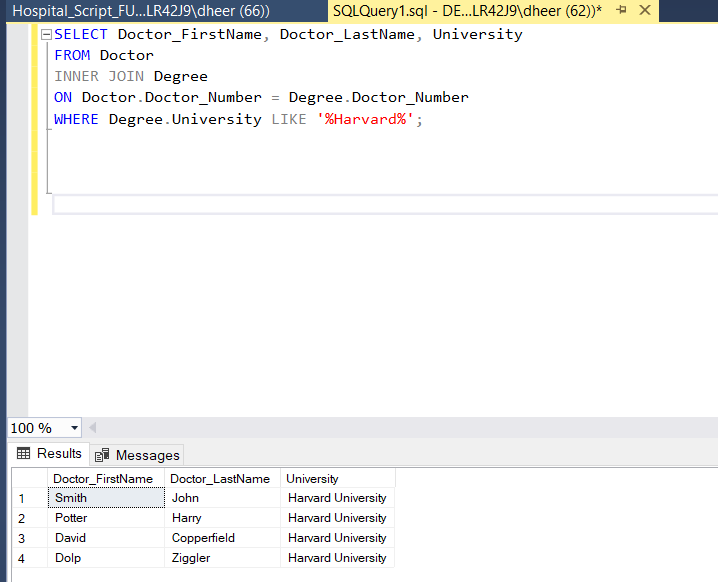
SELECT Doctor\_FirstName, Doctor\_LastName, University

FROM Doctor

INNER JOIN Degree

ON Doctor.Doctor\_Number = Degree.Doctor\_Number

WHERE Degree.University LIKE '%Harvard%';



**Patients who have operation Dilation and curettage:**

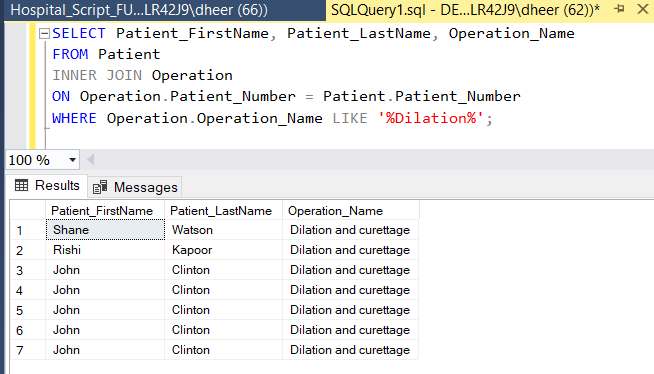
SELECT Patient\_FirstName, Patient\_LastName, Operation\_Name

FROM Patient

INNER JOIN Operation

ON Operation.Patient\_Number = Patient.Patient\_Number

WHERE Operation.Operation\_Name LIKE '%Dilation%';

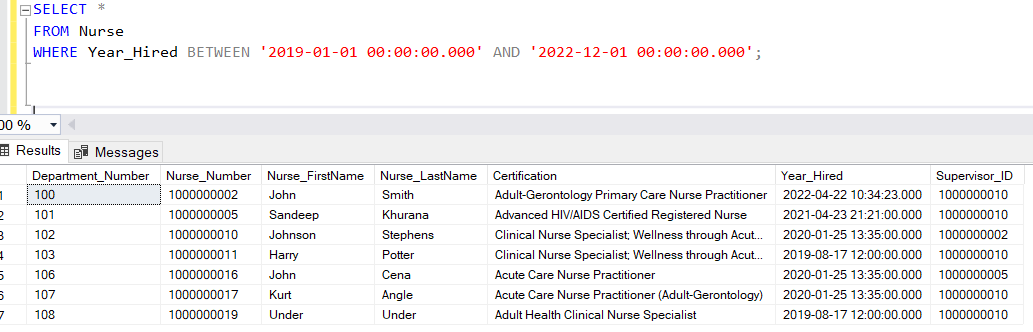


**Nurses who joined after 2020:**

SELECT \*

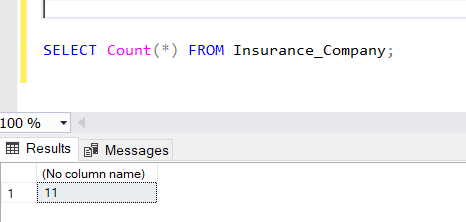
FROM Nurse

WHERE Year\_Hired BETWEEN '2019-01-01 00:00:00.000' AND '2022-12-01 00:00:00.000';



**Number of Insurance company operating:**

SELECT Count(\*) FROM Insurance\_Company;

****

# Data visualisation:

# Conclusion: