Student Database Management System

IFT 530: Advanced Database Management Systems

Systems Proposal Initial

Shiva Kumar Poojari

Hariharan Balakrishnan

Arizona State University

Instructor Name: Dr. Robert Rucker

Spring 2022

TABLE OF CONTENTS

SECTION- 1

INTRODUCTION AND BACKGROUND

A database management system is software that captures and analyzes data through interacting with end users, applications, and the database itself. The database management system software also includes the essential tools for managing the database. Database management systems are used by every organization, especially in the education domain.

A student database management system is automation of manual performance record management which enables the user to assess necessary data at any place and any time through the internet. The student web portal contains a login page where after providing the login details the home page is appeared for the user where it shows important notifications and activities in the college like semester fee payment dates, exam registration, change in exam timetable, workshops or fests to be held, etc. the student can check his attendance a, semester marks and mid-exam marks for every semester which enables them to improve his performance in forthcoming semester. The faculty can also make changes in marks in case of any mistake immediately which eliminates the time-consuming activities like registering a complaint and then faculty approving it then the administration making changes.

Questions:

- What is the current students list in the school ?
- Who are the faculties of a particular school ?
- What are the reports of each student in the school ?
- Which are the subjects taught by the faculty ?
- What are the resources that are available in the library ?
- What is the average grade of students from all subjects ?

List of Entities and Attributes:

Student Management:

Creating student profiles with unlimited custom categories and fields including demographic data, enrollment, attendance, schedule and more, and share academic records with faculties, and administrators.

Courses:

Enroll for academic courses and monitor the progress of students through the degree program with the flexibility to view course timetables and events in the calendar.

Attendance:

Allows teachers to mark attendance in the class and send attendance reports to administrators. Streamline attendance tracking and additionally further upgrade the system to automatically send notifications to parents through SMS, email, and

messaging.

Grades:

Enables students to check their academic track record from time to time and improve their performance and increase their GPA.

Reports:

Shows the results of every semester – the marks obtained in mid sectional exams and grades obtained in semester.

Staff details:

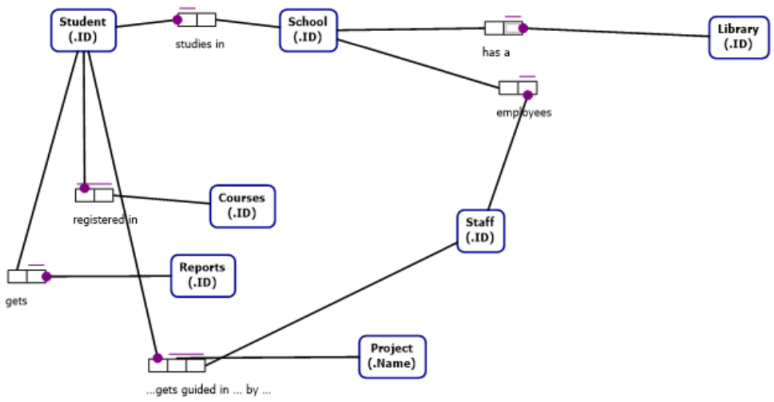
The faculty details include the contact details of each professor like e-mail address, office number and professional details like designation, specialization, qualification (department wise)

Library:

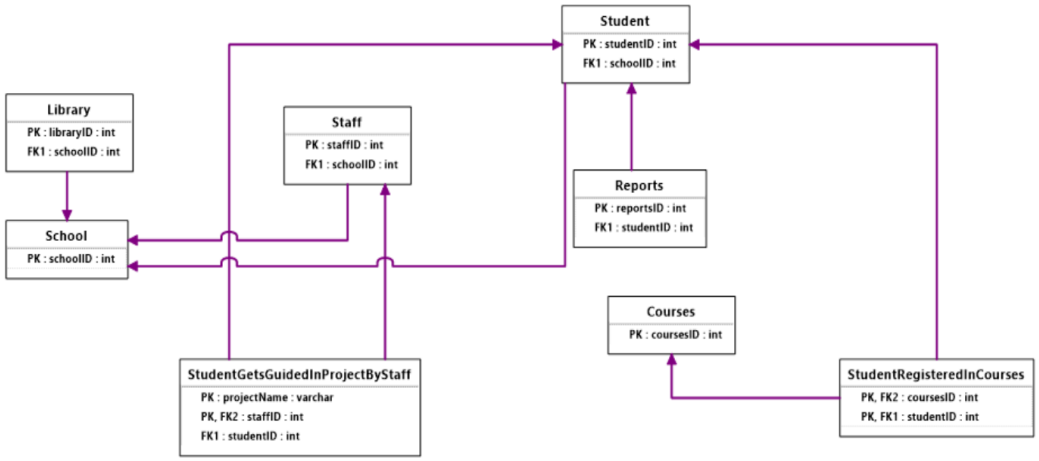
Provides data on the available books and journal papers in the library department wise and details of the library in charge and time of library availability

SECTION 2

ORM DIAGRAM

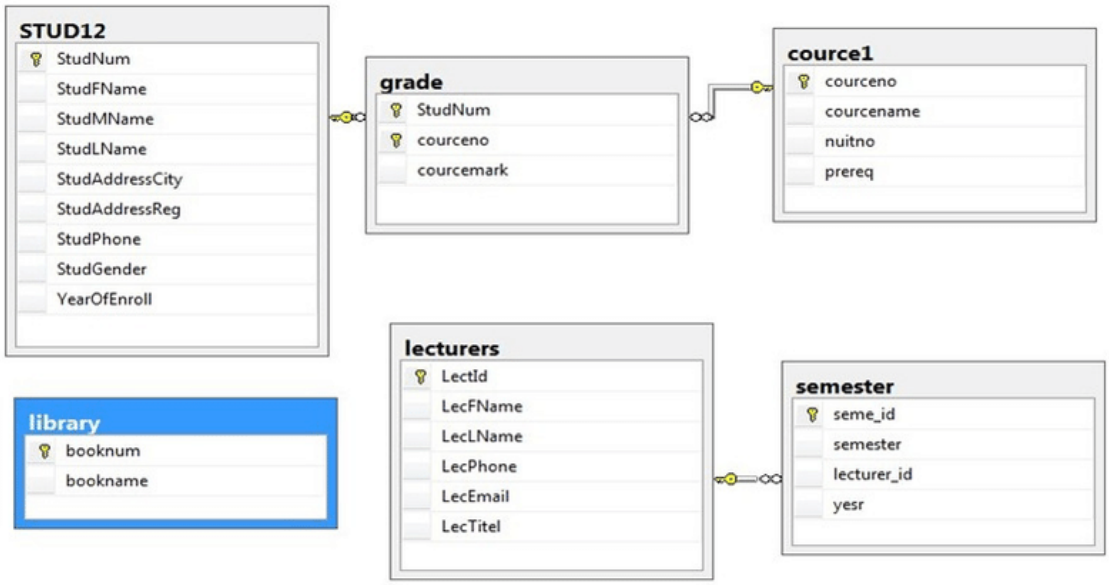


THE RELATIONAL VIEW



SECTION 3

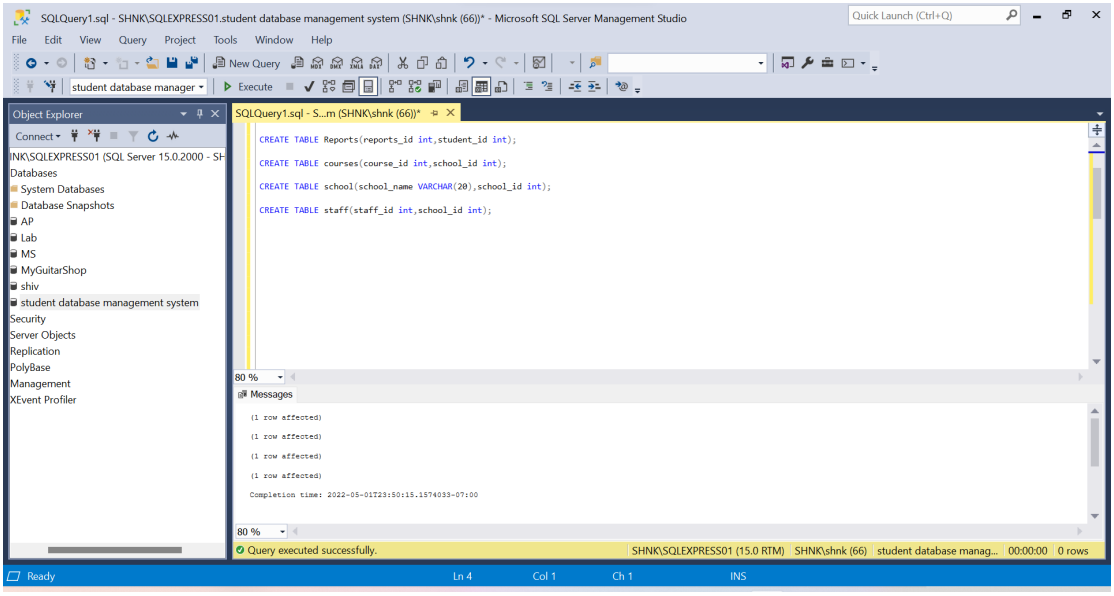
RELATIONAL SCHEMA



SECTION 4

TABLES CREATION

```
CREATE TABLE Student(Student_id int, school_name VARCHAR(20));

CREATE TABLE Library(library_id int, school_name VARCHAR(20));

CREATE TABLE Reports(reports_id int,student_id int);

CREATE TABLE courses(course_id int,school_id int);

CREATE TABLE school(school_name VARCHAR(20),school_id int);

CREATE TABLE staff(staff_id int,school_id int);
```



```
insert into student(student_id,school_name) VALUES(1,'ASU');
insert into student(student_id,school_name) VALUES(3,'CSU');
insert into student(student_id,school_name) VALUES(4,'UMC');
insert into student(student_id,school_name) VALUES(5,'LBHS');

insert into library(library_id,school_name) VALUES(101,'ASU');
insert into library(library_id,school_name)VALUES(102, 'CSU');
insert into library(library_id,school_name)VALUES(103,'UMC');
insert into library(library_id,school_name) VALUES(104,'LBHS');

insert into Reports(reports_id,student_id) VALUES(121,1);
insert into Reports(reports_id,student_id) VALUES(122,3);
insert into Reports(reports_id,student_id) VALUES(123,4);
insert into Reports(reports_id,student_id) VALUES(124,5);

insert into courses(course_id,school_id) VALUES (500,111);
insert into courses(course_id,school_id) VALUES(598,222);
insert into courses(course_id,school_id) VALUES(530,333);
insert into courses(course_id,school_id) VALUES(540,444);

insert into school(school_name,school_id) VALUES('ASU',111);
insert into school(school_name,school_id) VALUES('CSU',222);
```

insert into school(school_name,school_id) VALUES('UMC',333);

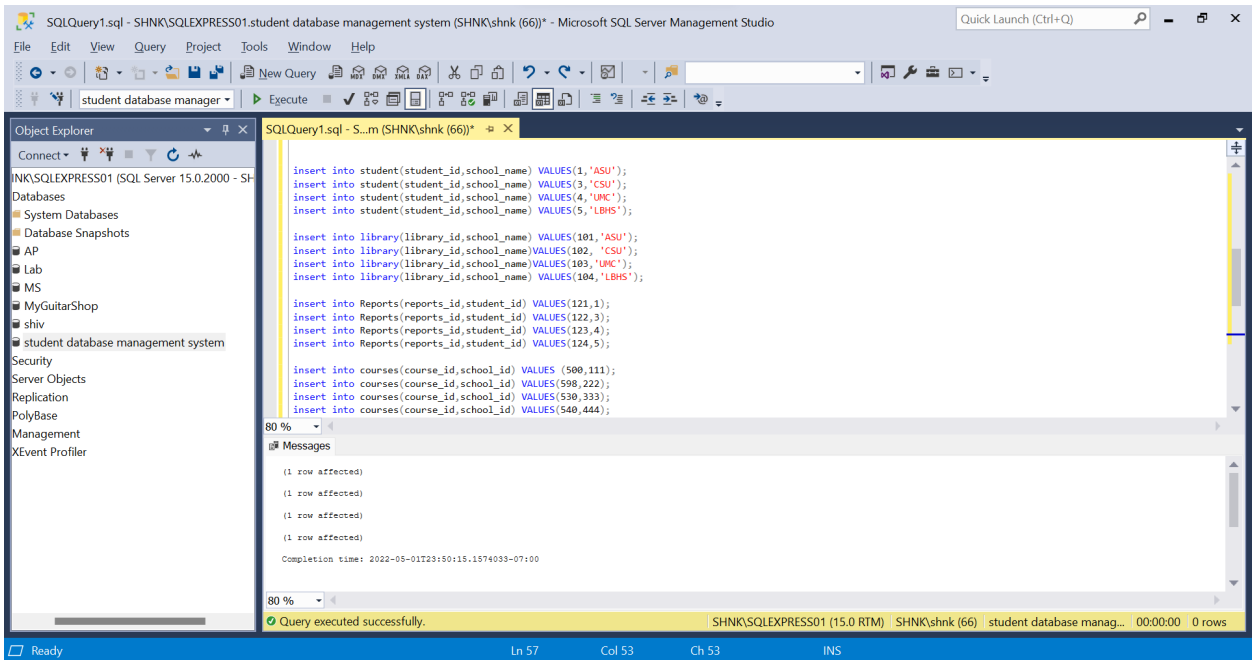
insert into school(school_name,school_id) VALUES('LBHS',444);

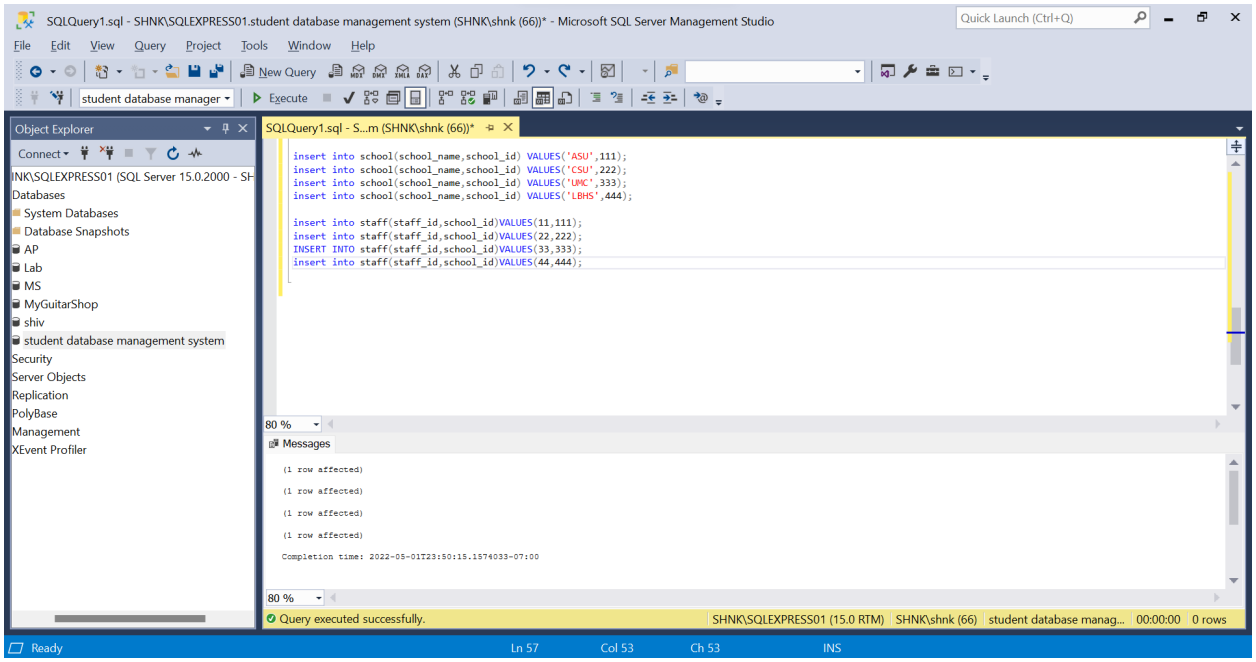
insert into staff(staff_id,school_id)VALUES(11,111);

insert into staff(staff_id,school_id)VALUES(22,222);

INSERT INTO staff(staff_id,school_id)VALUES(33,333);

insert into staff(staff_id,school_id)VALUES(44,444);




SQLQuery1.sql - SHNK\SQLEXPRESS01.student database management system (SHNK\shnk (66))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
student database manager
Execute
Object Explorer
Connect
INK\SQLEXPRESS01 (SQL Server 15.0.2000 - SH
Databases
System Databases
Database Snapshots
AP
Lab
MS
MyGuitarShop
shiv
student database management system
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery1.sql - S...m (SHNK\shnk (66))*
insert into school(school_name,school_id) VALUES('ASU',111);
insert into school(school_name,school_id) VALUES('CSU',222);
insert into school(school_name,school_id) VALUES('UMC',333);
insert into school(school_name,school_id) VALUES('LBHS',444);
insert into staff(staff_id,school_id)VALUES(11,111);
insert into staff(staff_id,school_id)VALUES(22,222);
INSERT INTO staff(staff_id,school_id)VALUES(33,333);
insert into staff(staff_id,school_id)VALUES(44,444);
80 %
Messages
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
Completion time: 2022-05-01T23:50:15.1574033-07:00
80 %
Query executed successfully.
SHNK\SQLEXPRESS01 (15.0 RTM)
SHNK\shnk (66)
student database manag...
00:00:00
0 rows
Ready
Ln 57
Col 53
Ch 53
INS

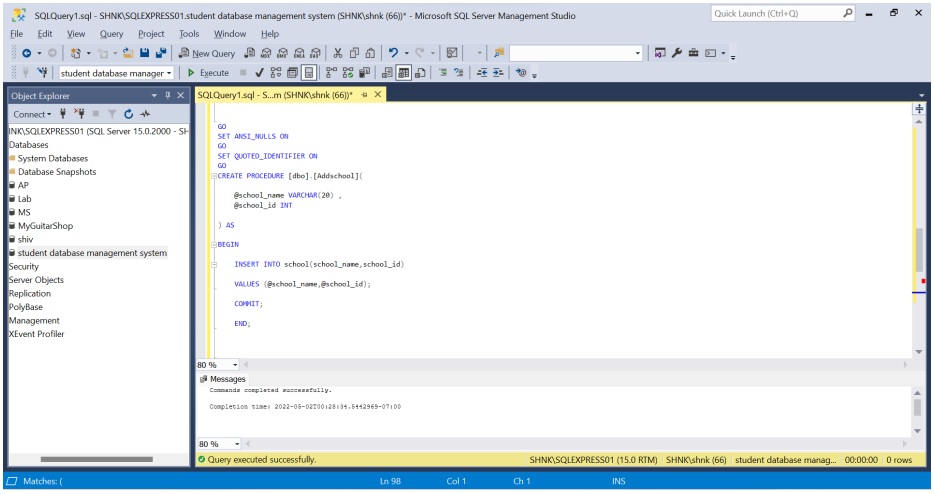
SECTION 5

Additional Constraints

Stored Procedure

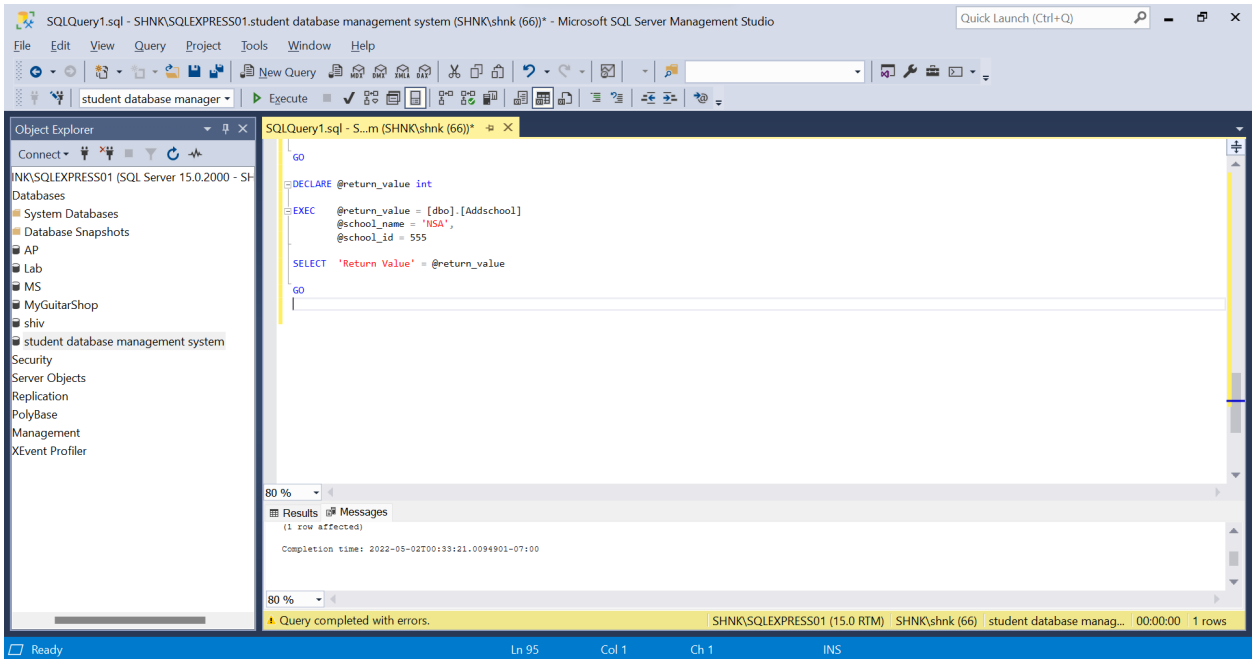
1. Creating a stored procedure for inserting data into school table

```
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Addschool](

    @school_name VARCHAR(20) ,
        @school_id INT

) AS

BEGIN

    INSERT INTO school(school_name,school_id)

    VALUES (@school_name,@school_id);

    COMMIT;

    END;
```



GO

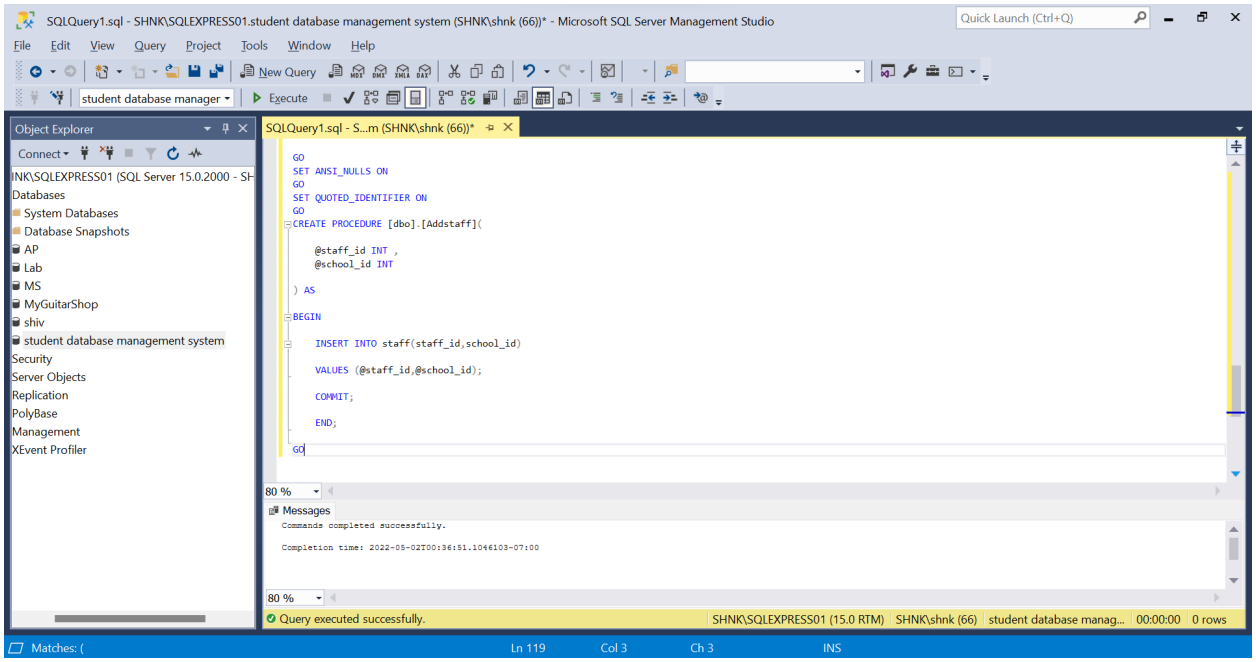
```
DECLARE    @return_value int

EXEC @return_value = [dbo].[Addschool]
                @school_name = 'NSA',
                @school_id = 555

SELECT        'Return Value' = @return_value

GO
```



2. Creating a stored procedure for school and staff.

GO

```
SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[Addstaff](

   @staff_id INT ,

        @school_id INT

) AS

BEGIN

   INSERT INTO staff(staff_id,school_id)

   VALUES (@staff_id,@school_id);

   COMMIT;

   END;

GO
```



GO

DECLARE @return_value int

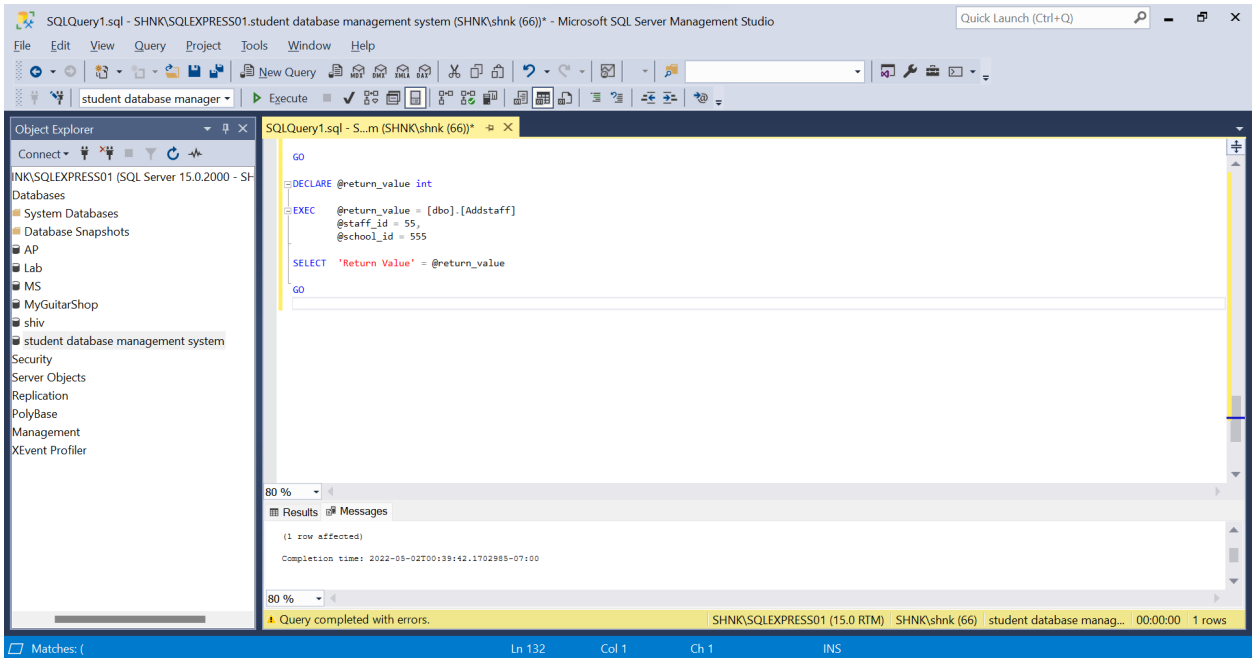
EXEC @return_value = [dbo].[Addstaff]

@staff_id = 55,

@school_id = 555

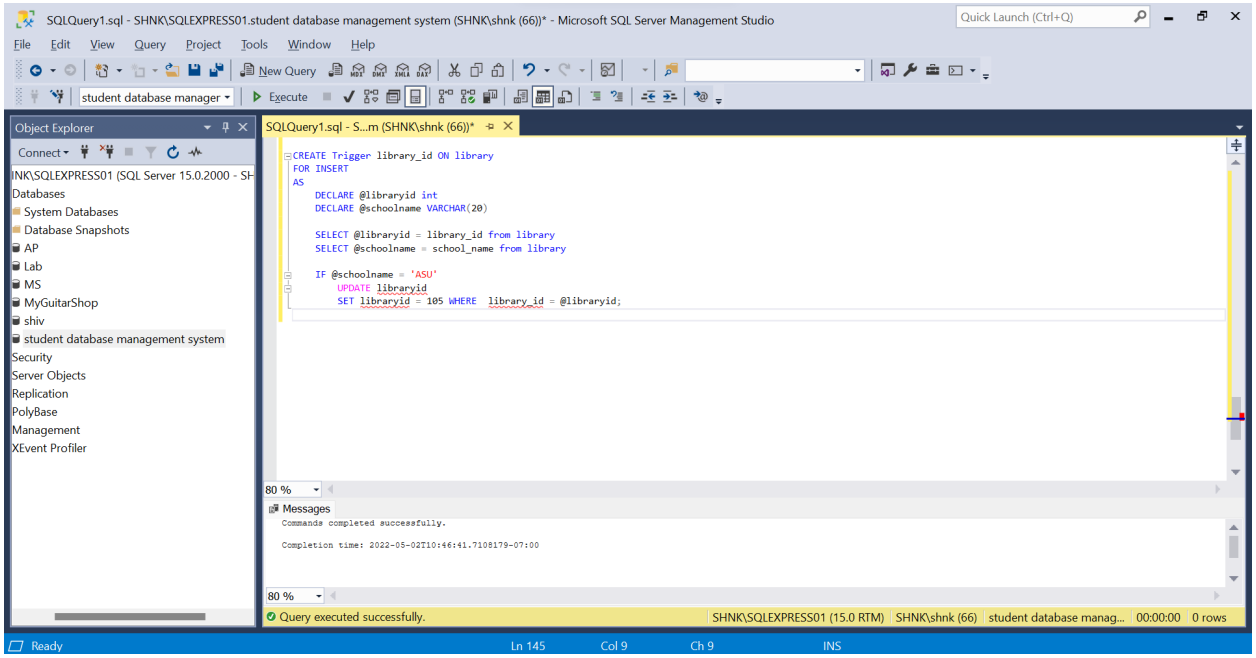
SELECT 'Return Value' = @return_value

GO



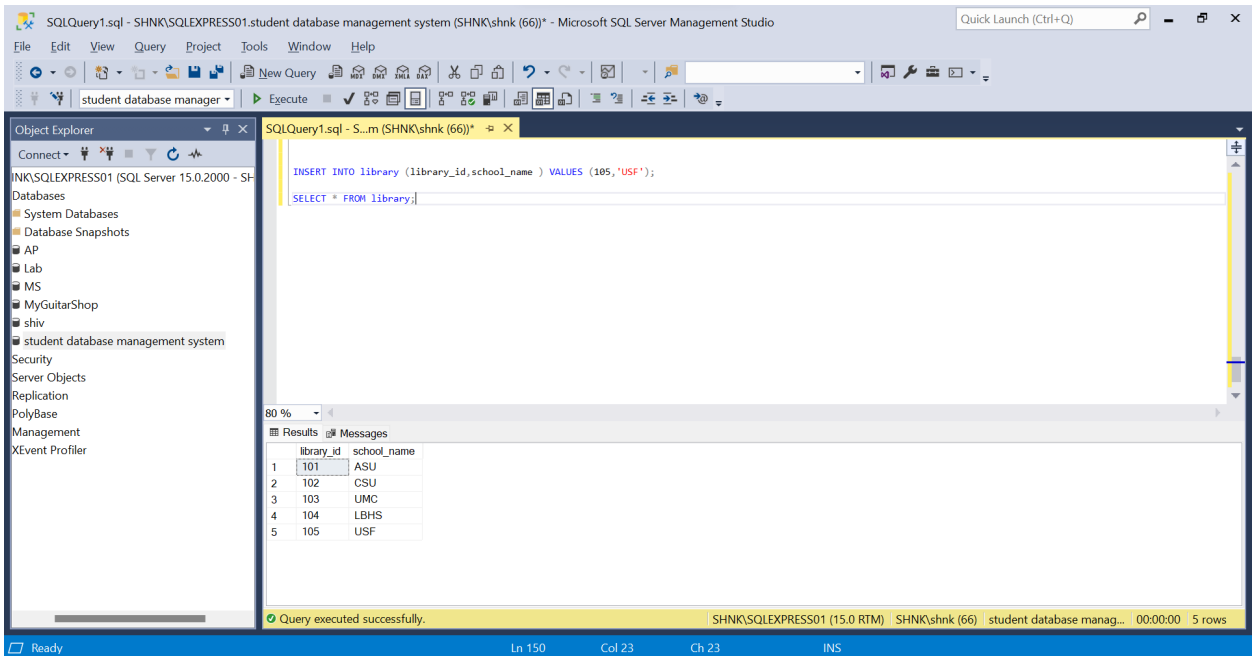
Triggers:

```
CREATE Trigger library_id ON library

FOR INSERT

AS

        DECLARE @libraryid int

        DECLARE @schoolname VARCHAR(20)


        SELECT @libraryid = library_id from library

        SELECT @schoolname = school_name from library


        IF @schoolname = 'ASU'

                UPDATE libraryid

                SET libraryid = 105 WHERE  library_id = @libraryid;
```



INSERT INTO library (library_id,school_name) VALUES (105,'USF');

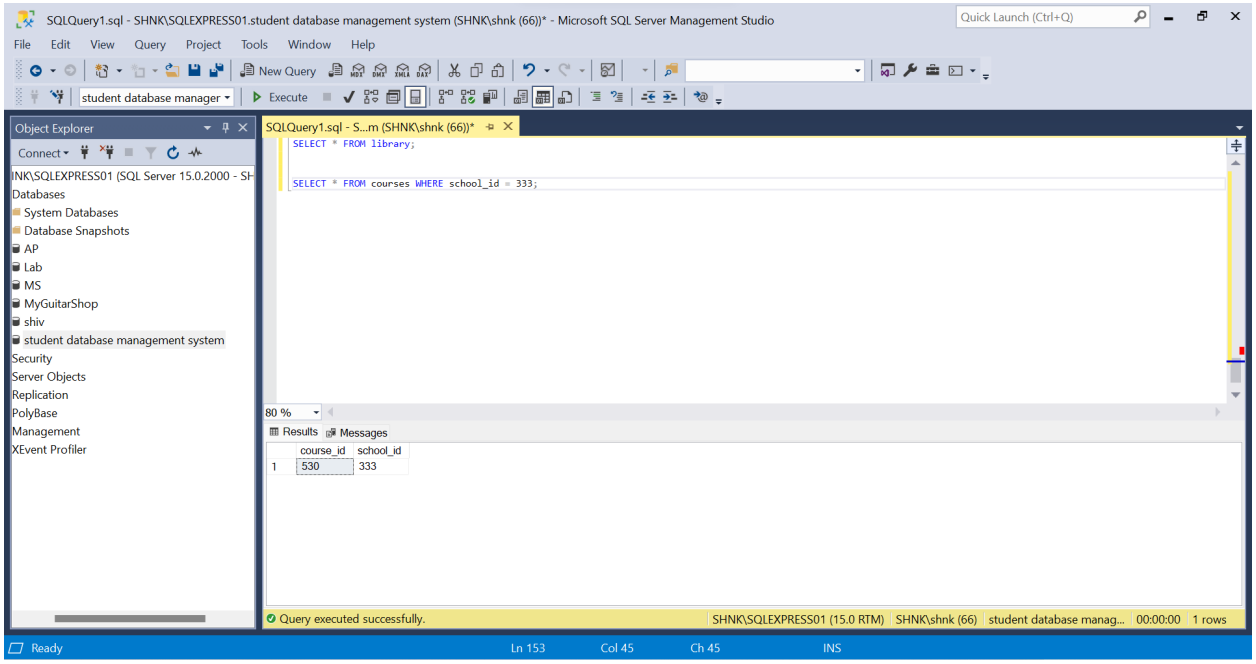
SELECT * FROM library;



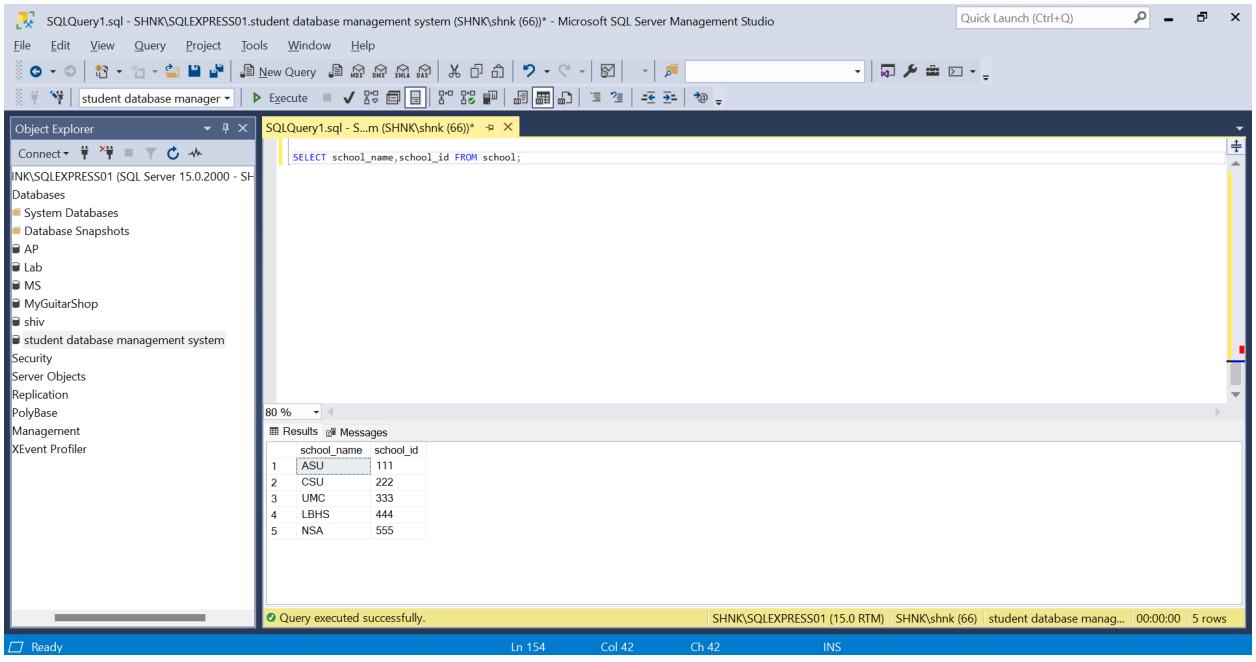
Section - 5.1

Retrieving Queries

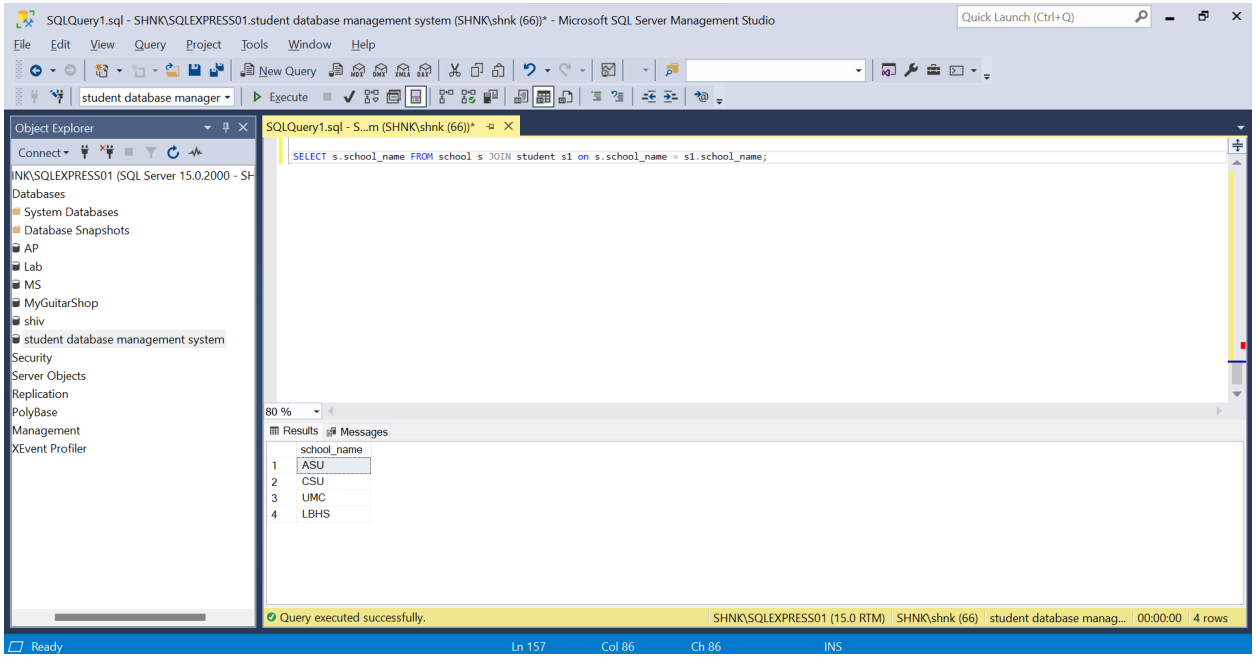
1. SELECT * FROM courses WHERE school_id = 333;



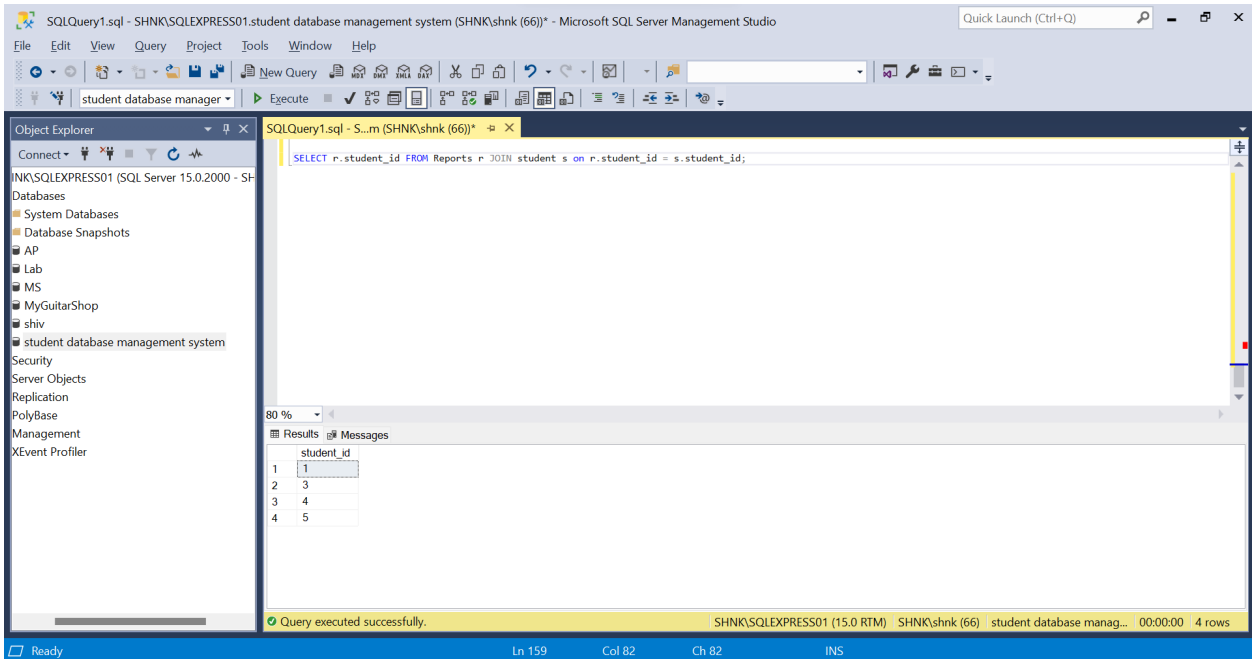
2. SELECT school_name,school_id FROM school;



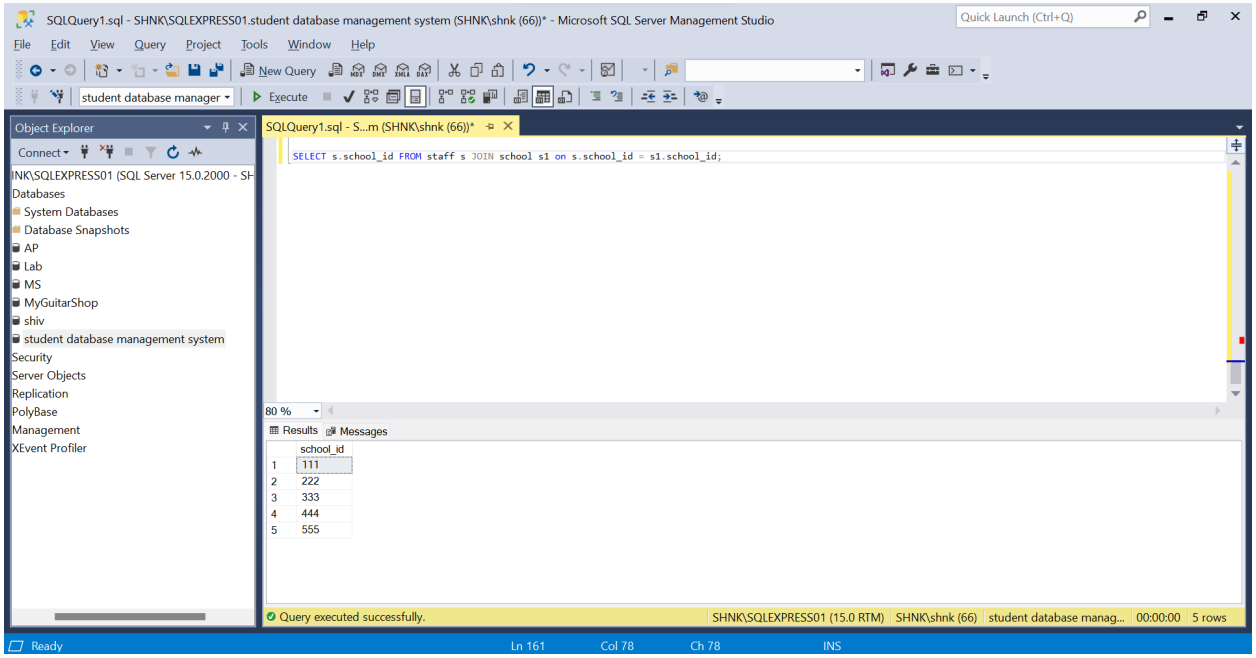
3. SELECT s.school_name FROM school s JOIN student s1 on s.school_name = s1.school_name;



4. SELECT r.student_id FROM Reports r JOIN student s on r.student_id = s.student_id;



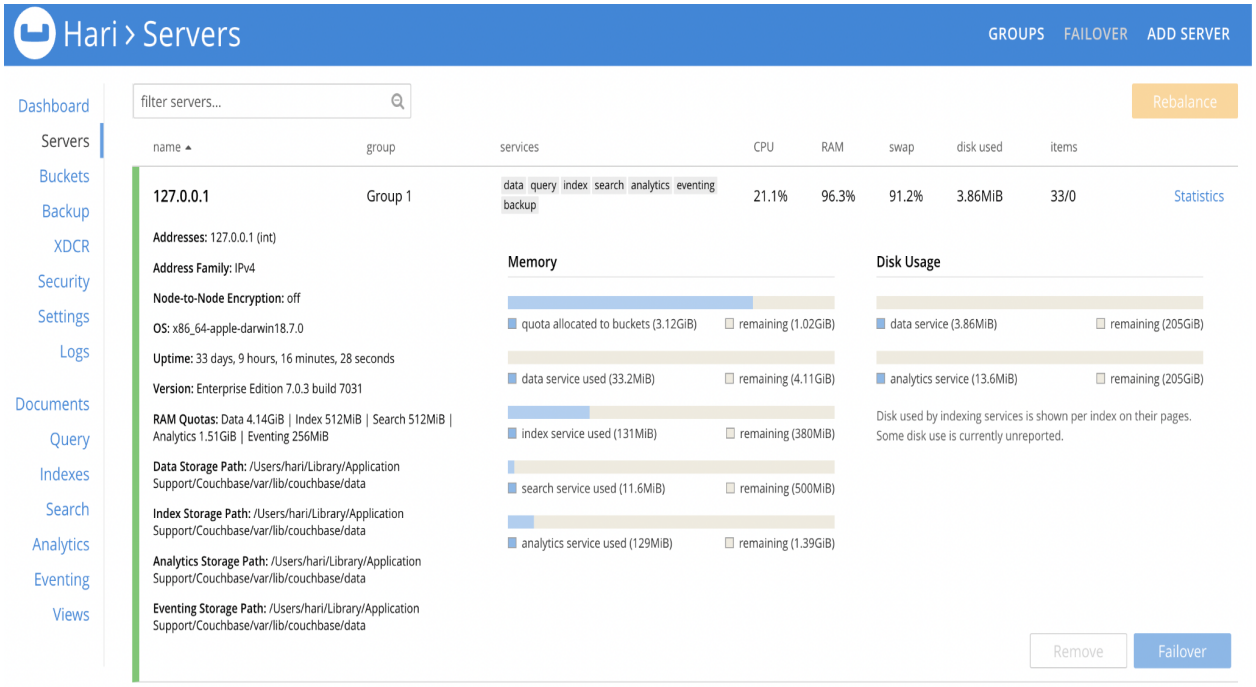
5. SELECT s.school_id FROM staff s JOIN school s1 on s.school_id = s1.school_id;



SECTION 6

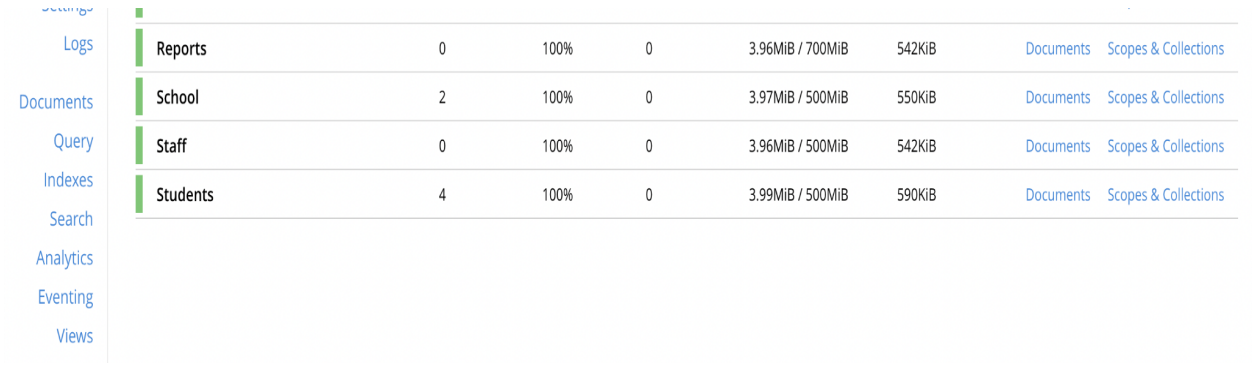
NO SQL

Used couchbase enterprise 7+ edition database to execute queries:



Section 6.1:

Created buckets/ Entities are:



Section 6.2:

Students:

```
insert into  Students(key, value)
values ("101",
  {
    " AdmissionId": "101",
    "RollId": 11,
    "StudentName": "Joe",
    "SchoolName": "IraA Fulton",
    "SchoolID": 1,
    "phoneNumber": "4802284786",
    "DOB": "1998-10-17",
    "emailID": "joe17@gmail.com",
    "Rank": 4,
    "GPA": 3.90
  }),("102",
  {
    " AdmissionId": "102",
    "RollId": 12,
    "StudentName": "Chandler",
    "SchoolName": "W.P Carey",
    "SchoolID": 2,
```

```
   "phoneNumber": "4802284790",
   "DOB": "1998-01-09",
   "emailID": "chandler09@gmail.com",
   "Rank": 3,
   "GPA": 4.00
}),("103",
{
   " AdmissionId": "103",
   "RollId": 13,
   "StudentName": "Mike",
   "SchoolName": "IraA Fulton",
   "SchoolID": 1,
   "phoneNumber": "4752284701",
   "DOB": "1998-10-02",
   "emailID": "mike02@gmail.com",
   "Rank": 2,
   "GPA": 4.11
}),(
"104",
{
   " AdmissionId": "104",
   "RollId": 14,
   "StudentName": "Issac",
```

"SchoolName": "W.P carey",

"SchoolID": 2,

"phoneNumber": "4752284769",

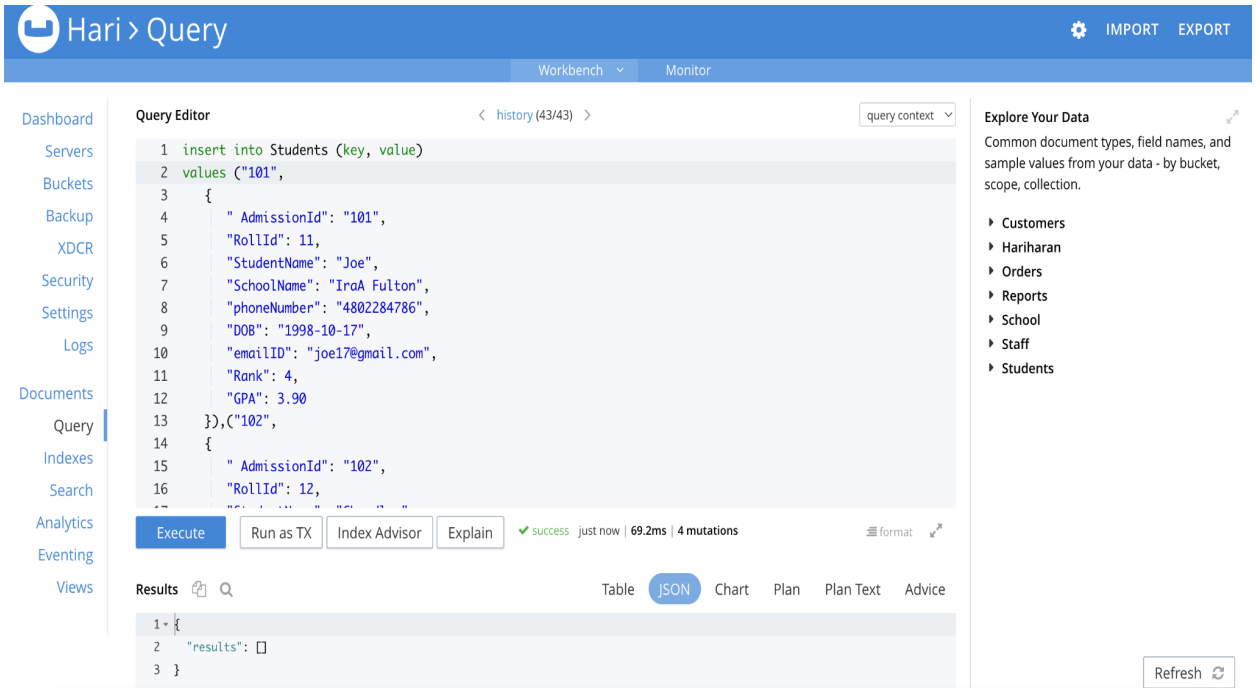
"DOB": "1999-04-04",

"emailID": "issac69@gmail.com",

"Rank": 1,

"GPA": 4.33

})



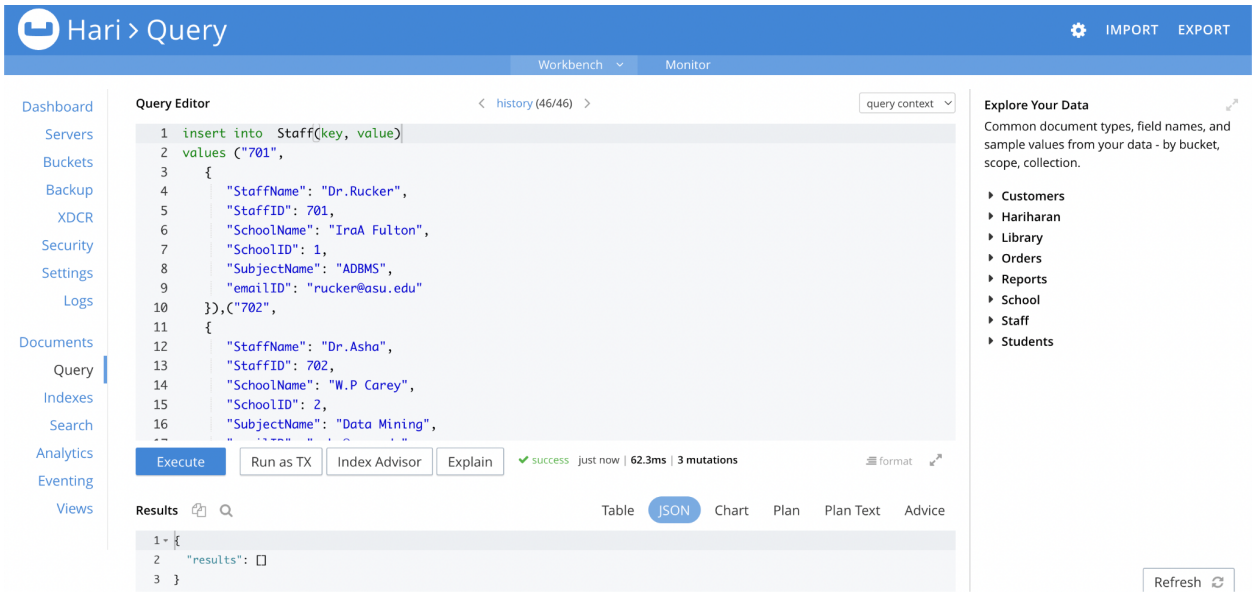
Staff:

insert into Staff(key, value)

values ("701",

```
{
  "StaffName": "Dr.Rucker",
  "StaffID": 701,
  "SchoolName": "IraA Fulton",
  "SchoolID": 1,
  "SubjectName": "ADBMS",
  "emailID": "rucker@asu.edu"
}),("702",
{
  "StaffName": "Dr.Asha",
  "StaffID": 702,
  "SchoolName": "W.P Carey",
  "SchoolID": 2,
  "SubjectName": "Data Mining",
  "emailID": "asha@asu.edu"
}),("703",
{
  "StaffName": "Dr.Atkinson",
  "StaffID": 703,
  "SchoolName": "IraA Fulton",
  "SchoolID": 1,
  "SubjectName": "NLP",
  "emailID": "atkinson@asu.edu"
```

})



School:

insert into School(key, value)

values ("1",

{

"SchoolName": "IraA Fulton",

"SchoolID": 1,

"phoneNumber": "4802280000",

"emailID": "iraafulton@gmail.com"

}),("2",

{

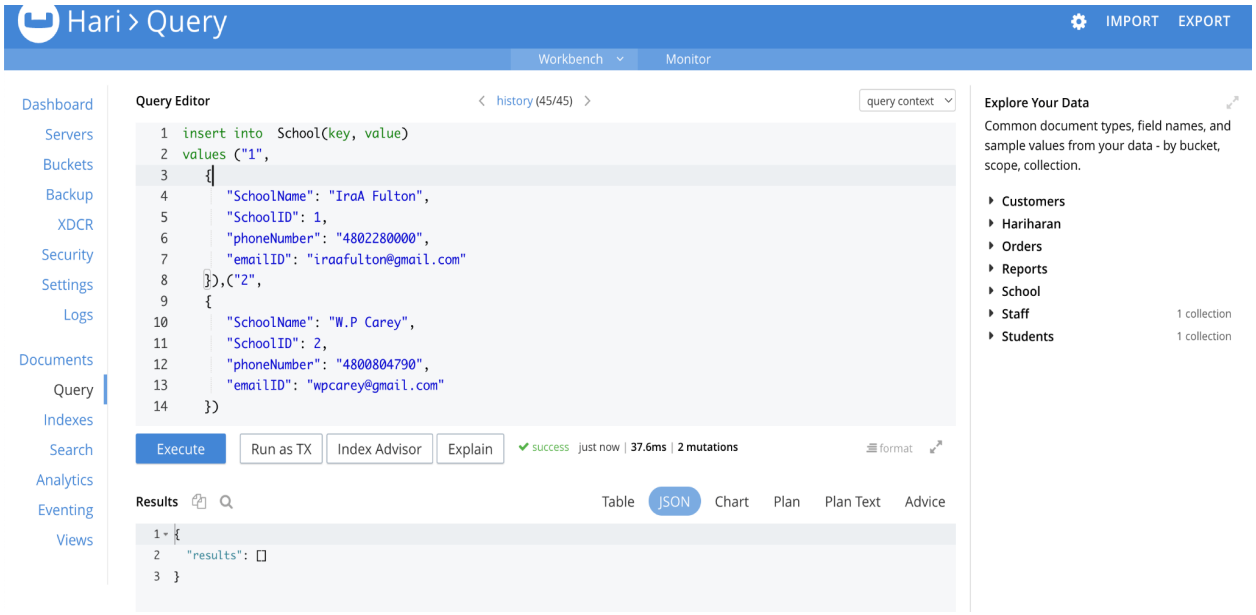
"SchoolName": "W.P Carey",

"SchoolID": 2,

"phoneNumber": "4800804790",

"emailID": "wpcarey@gmail.com"

})



Reports:

insert into Reports(key, value)

values ("81",

{

"StaffName": "Dr.Rucker",

"ReportID": 81,

"StudentName": "Issac",

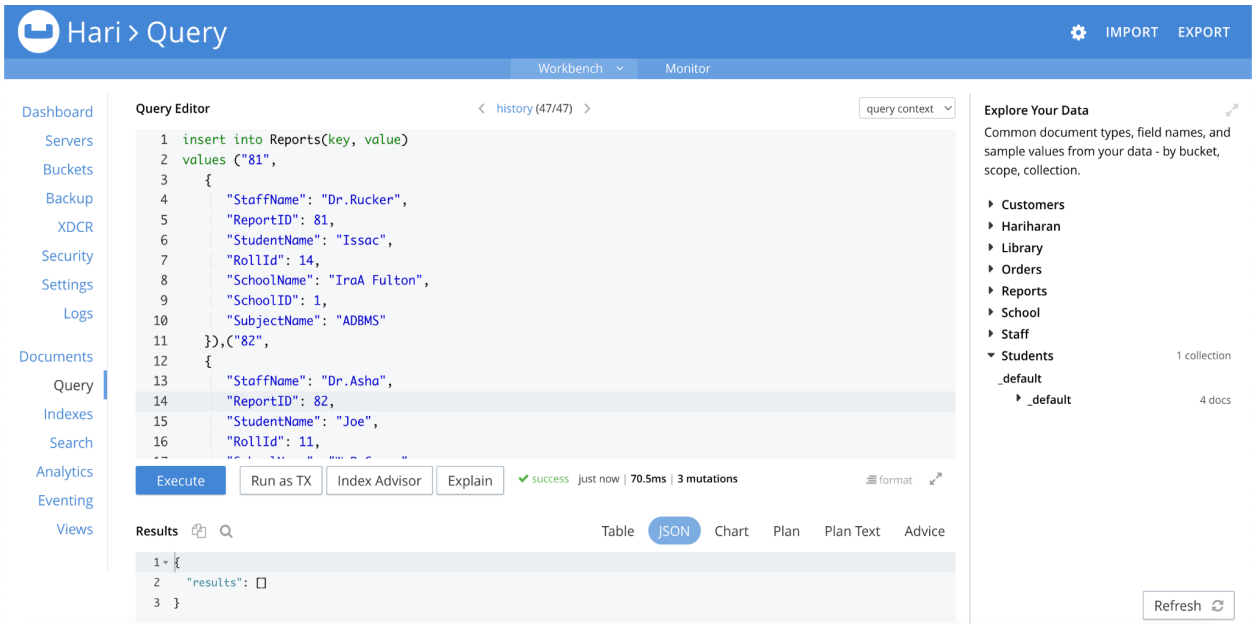
"RollId": 14,

"SchoolName": "IraA Fulton",

"SchoolID": 1,

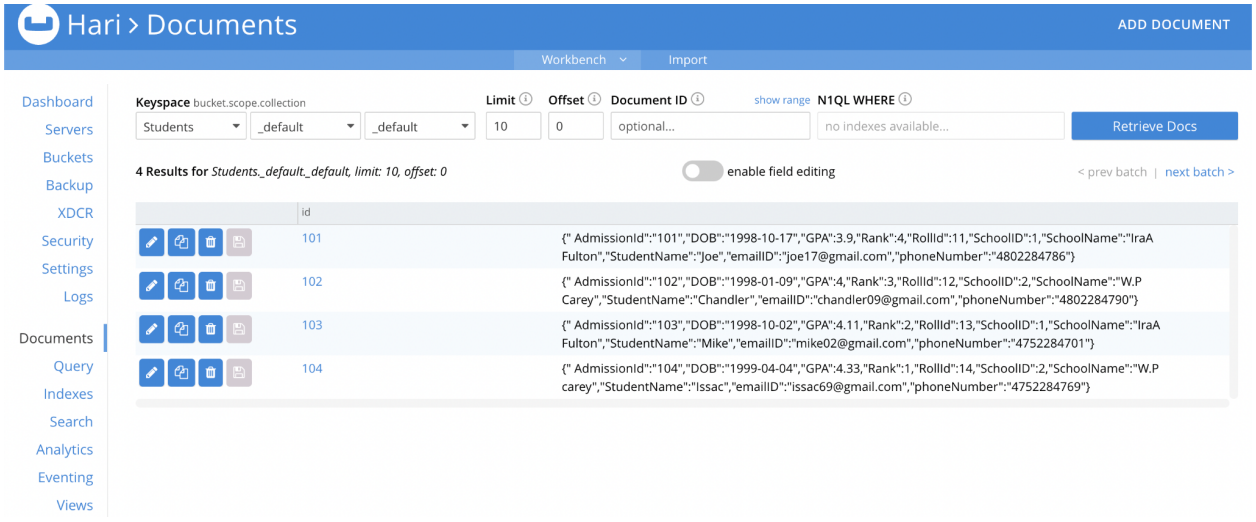
"SubjectName": "ADBMS"

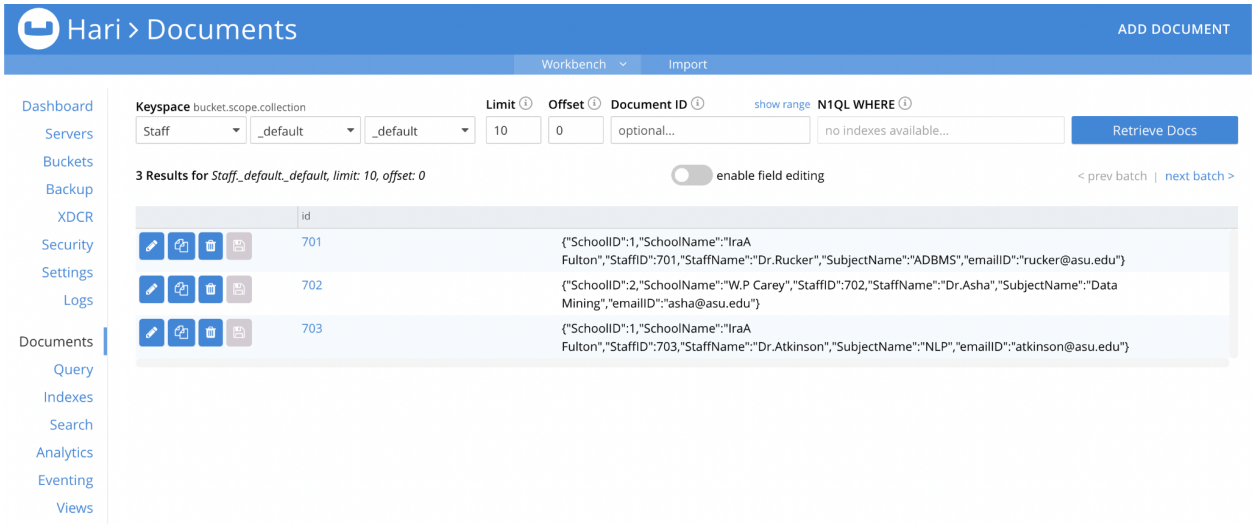
```
}),("82",
{
  "StaffName": "Dr.Asha",
  "ReportID": 82,
  "StudentName": "Joe",
  "RollId": 11,
  "SchoolName": "W.P Carey",
  "SchoolID": 2,
  "SubjectName": "Data Mining"
}),("83",
{
  "StaffName": "Dr.Atkinson",
  "ReportID": 83,
  "StudentName": "Mike",
  "RollId": 13,
  "SchoolName": "IraA Fulton",
  "SchoolID": 1,
  "SubjectName": "NLP"
})
```

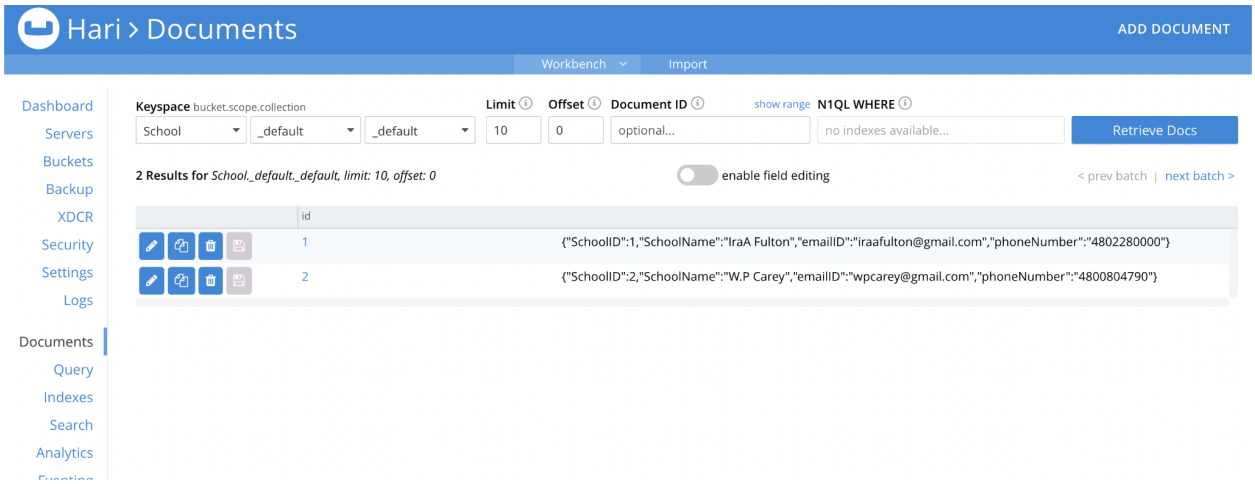


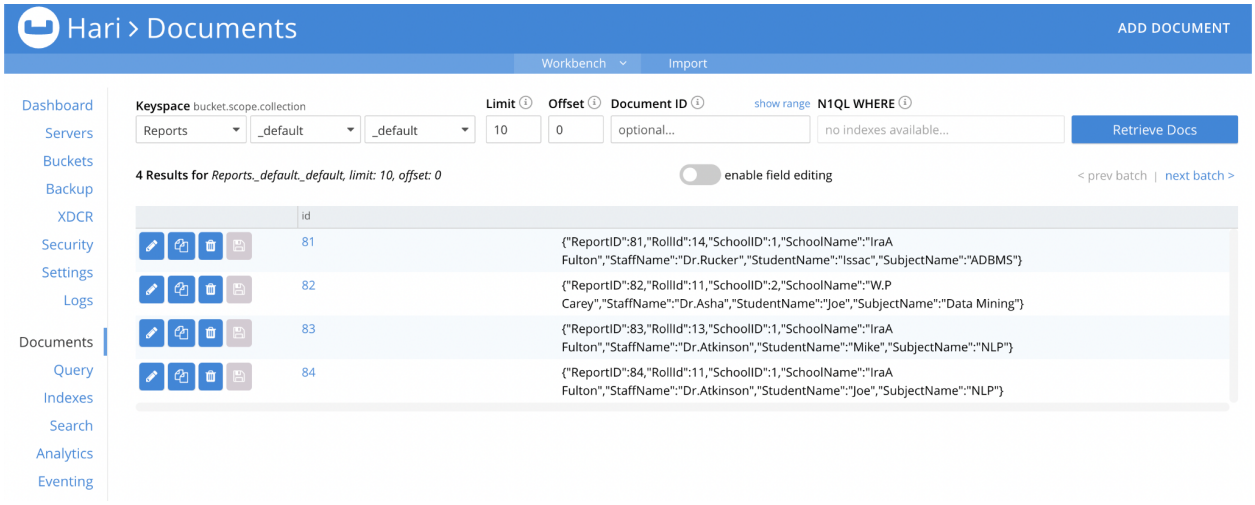
Section 6.3:

JSON Document




Hari > Documents
ADD DOCUMENT
Workbench
Import
Dashboard
Servers
Buckets
Backup
XDCR
Security
Settings
Logs
Documents
Query
Indexes
Search
Analytics
Eventing
Views
Keyspace bucket.scope.collection
Staff
_default
_default
Limit
10
Offset
0
Document ID
optional...
show range
N1QL WHERE
no indexes available...
Retrieve Docs
3 Results for Staff._default._default, limit: 10, offset: 0
enable field editing
< prev batch | next batch >
id
701
{"SchoolID":1,"SchoolName":"IraA Fulton","StaffID":701,"StaffName":"Dr.Rucker","SubjectName":"ADBMS","emailID":"rucker@asu.edu"}
702
{"SchoolID":2,"SchoolName":"W.P Carey","StaffID":702,"StaffName":"Dr.Asha","SubjectName":"Data Mining","emailID":"asha@asu.edu"}
703
{"SchoolID":1,"SchoolName":"IraA Fulton","StaffID":703,"StaffName":"Dr.Atkinson","SubjectName":"NLP","emailID":"atkinson@asu.edu"}


Hari > Documents
ADD DOCUMENT
Workbench
Import
Dashboard
Servers
Buckets
Backup
XDCR
Security
Settings
Logs
Documents
Query
Indexes
Search
Analytics
Keyspace bucket.scope.collection
School
_default
_default
Limit
10
Offset
0
Document ID
optional...
show range
N1QL WHERE
no indexes available...
Retrieve Docs
2 Results for School._default._default, limit: 10, offset: 0
enable field editing
< prev batch | next batch >
id
1
{"SchoolID":1,"SchoolName":"IraA Fulton","emailID":"iraafulton@gmail.com","phoneNumber":"4802280000"}
2
{"SchoolID":2,"SchoolName":"W.P Carey","emailID":"wpcarey@gmail.com","phoneNumber":"4800804790"}

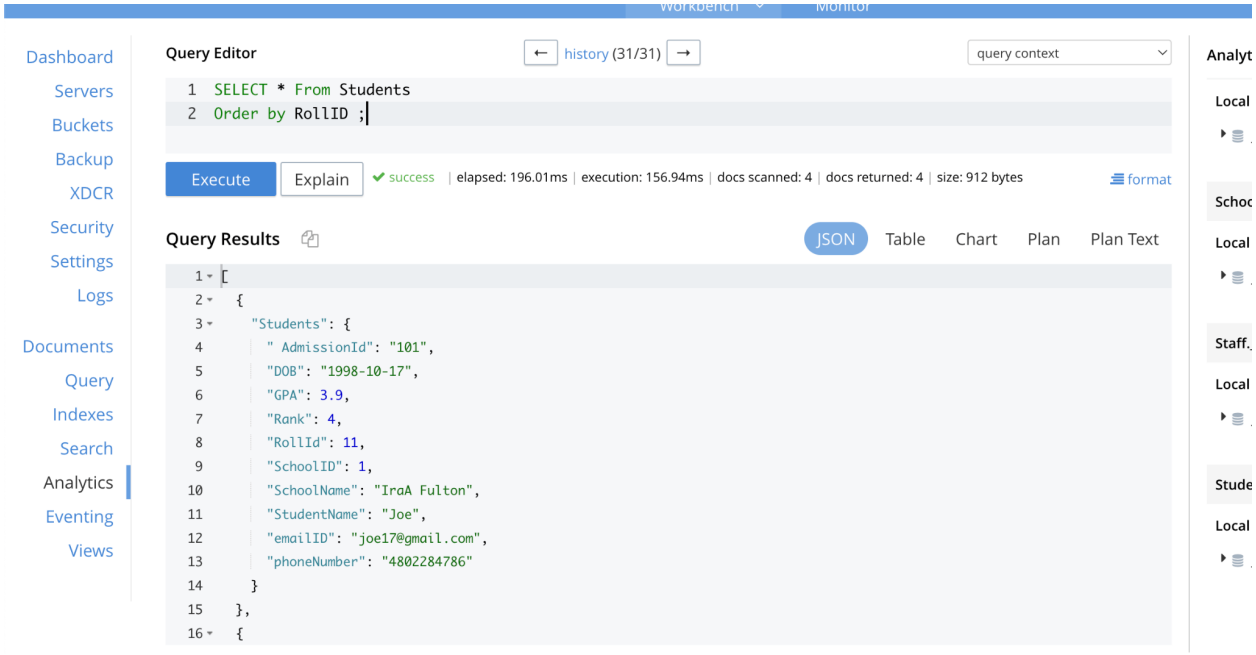

Hari > Documents
ADD DOCUMENT
Workbench
Import
Dashboard
Servers
Buckets
Backup
XDCR
Security
Settings
Logs
Documents
Query
Indexes
Search
Analytics
Eventing
Keyspace bucket.scope.collection
Reports
_default
_default
Limit
10
Offset
0
Document ID
optional...
show range
N1QL WHERE
no indexes available...
Retrieve Docs
4 Results for Reports._default._default, limit: 10, offset: 0
enable field editing
< prev batch | next batch >
id
81
{"ReportID":81,"RollId":14,"SchoolID":1,"SchoolName":"IraA Fulton","StaffName":"Dr.Rucker","StudentName":"Issac","SubjectName":"ADBMS"}
82
{"ReportID":82,"RollId":11,"SchoolID":2,"SchoolName":"W.P Carey","StaffName":"Dr.Asha","StudentName":"Joe","SubjectName":"Data Mining"}
83
{"ReportID":83,"RollId":13,"SchoolID":1,"SchoolName":"IraA Fulton","StaffName":"Dr.Atkinson","StudentName":"Mike","SubjectName":"NLP"}
84
{"ReportID":84,"RollId":11,"SchoolID":1,"SchoolName":"IraA Fulton","StaffName":"Dr.Atkinson","StudentName":"Joe","SubjectName":"NLP"}

Section 6.4:

Query 1: What is the current students list in the schools ?

Displayed list of the student and their information by roll identity number wise from Students bucket..

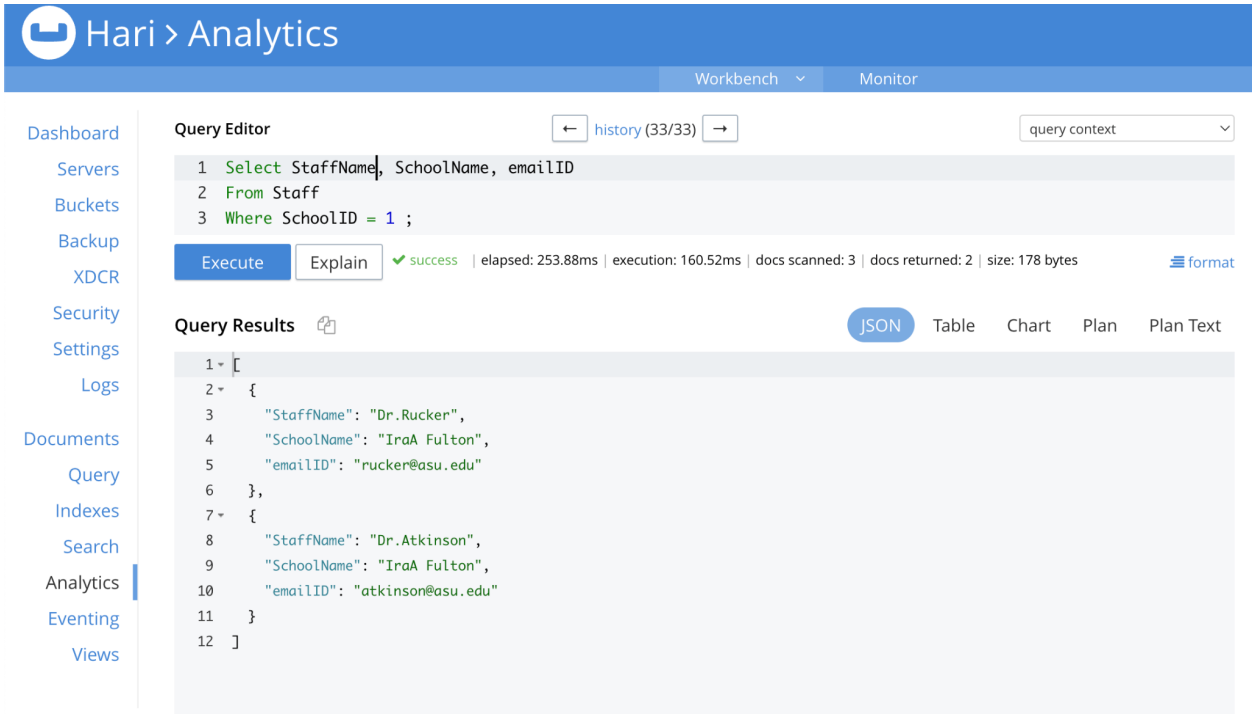
```
SELECT * From Students
Order by RollID ;
```



Query 2: Who are the faculties of a particular school ?

Displayed all the faculties according to their respective school using school ID from staff bucket.

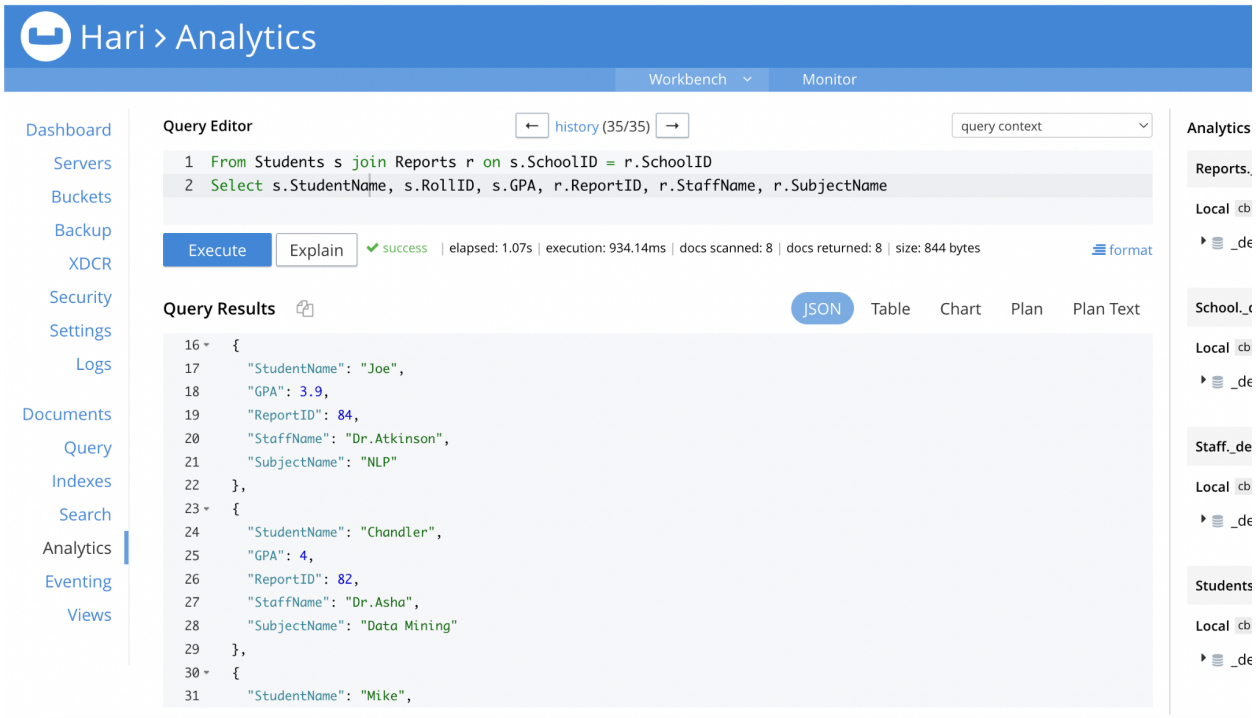
```
Select StaffName, SchoolName, emailID
From Staff
Where SchoolID = 1 ;
```



Query 3: What are the reports of each student in the school ?

Extracted information of students and their reports by joining both buckets using JOIN syntax.

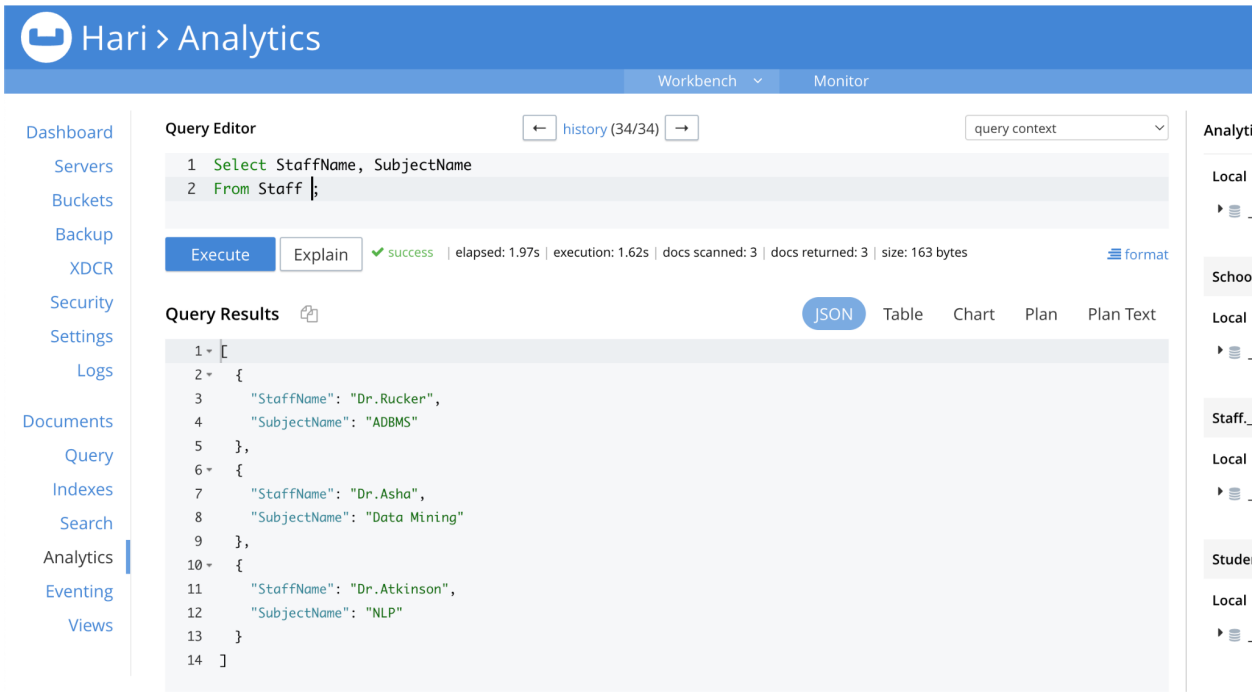
From Students s join Reports r on s.SchoolID = r.SchoolID
Select s.StudentName, s.RollID, s.GPA, r.ReportID, r.StaffName, r.SubjectName



Query 4: Which are the subjects taught by the faculty ?

Extracted information of subjects handled by faculties from the staff bucket.

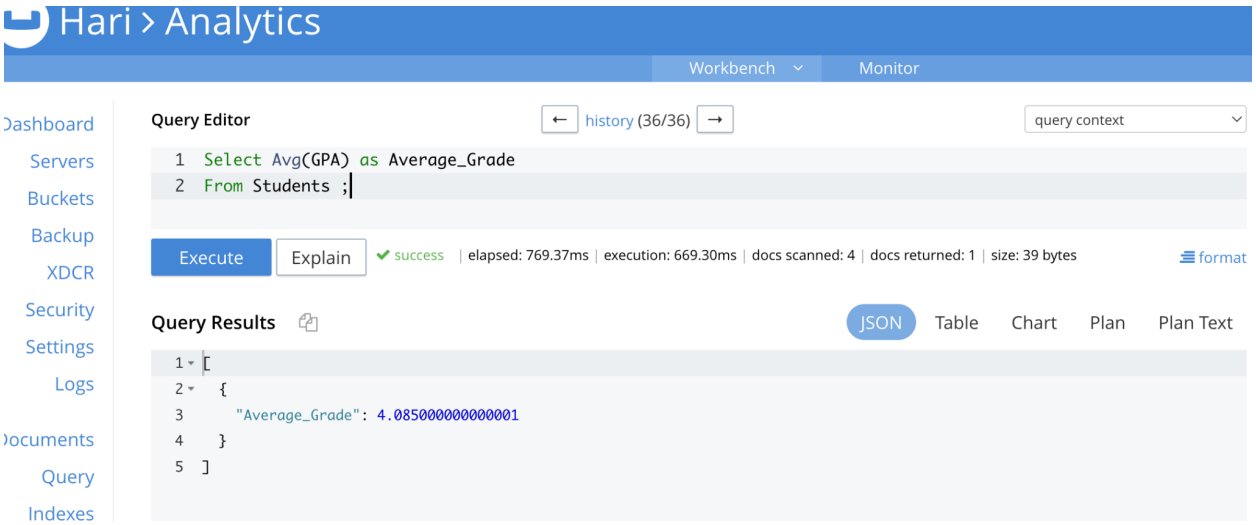
```
Select StaffName, SubjectName
From Staff ;
```



Query 5: What is the average grade of students from subjects ?

Found the average grade points from all students in their enrolled subjects.

```
Select Avg(GPA) as Average_Grade
From Students ;
```



SECTION 7:

SUMMARY

Our student management system aids in the administration, information, and scheduling in schools. A vast amount of data can be generated and used by a school system. This information must be provided to students, faculty, and parents in a suitable manner. A student management system aids schools in storing, managing, and disseminating this data.

Student management systems serve educational institutions in a variety of ways, the most important of which is centralized data administration and accessibility. Teachers will be able to input, maintain, and access student data more simply. Parents and guardians will have a better understanding of how their children perform in class. Compliance and other regulatory obligations at the state level are likewise significantly easier to meet. Instead of papers, an automated student database management system can be used.

CONCLUSION

To sum up, working on this project taught us how to design a database structure. We can and should comprehend a wide range of DB skills. The goal of this project was to learn and understand how to convert an ORM diagram into a Relation schema that can then be converted into tables and databases.

We were able to learn as well as implement schema-oriented SQL by contributing to this project. We also learned what attributes are required for a successful SQL query. SQL's stored procedures, triggers, and functions can help you access critical information from a complex data

structure.

In Couchbase, we operated NoSQL by writing queries in our loaded data, writing SQL++ queries for buckets where the analytics service is active, extending the ORM with nested documents, and creating JSON documents for entities where all tasks were completed.

REFERENCES:

1. TEXTBOOK : Murach's SQL Server 2016 for Developers, by Bryan Syverson and Joel Murach, June 2016.
2. WEBSITE : SQL++ For SQL Users by Don Chamberlin, September 2018 .
   https://www.couchbase.com/analytics-data