IDS521 Project A (Individual Assignment)

Create a sample case how to use data warehousing with MS SQL Server’s SSIS (Integration Services), SSAS (Analysis Services), and SSRS (Reporting Services).

A new database file should include:

- Data analytic purpose data and tables
- Queries for data analytic or SSAS/SSRS results (not limited to but this database should not be transactional data)

Data analytic purpose data and tables

Scenario:

Let's consider a scenario where a university wants to enhance its registration system to analyze student enrollment, instructor performance, and course popularity. They have transactional data stored in their operational database but want to create a data warehousing solution for analytical purposes.

Solution:

We'll create a data warehousing solution with MS SQL Server's SSIS (Integration Services), SSAS (Analysis Services), and SSRS (Reporting Services) for the university's registration system.

Data Warehouse Design:We'll design the following tables for analytical purposes:
Student: Contains information about students enrolled in courses.
Instructor: Stores details about instructors teaching courses.
Course: Holds information about courses offered by the university.
Section: Represents sections of courses offered in different semesters.
Student_Takes: Records the enrollment of students in various course sections.
SSIS (Integration Services):
Use SSIS to extract data from the university's operational database and load it into the data warehouse.
Design SSIS packages to perform ETL operations, extracting data from tables like Student, Instructor, Course, Section, and Student_Takes.
Transform the data as needed, such as aggregating student enrollment numbers by course or semester.
Load the transformed data into the data warehouse tables designed for analytics.
SSAS (Analysis Services):
Utilize SSAS to create a multidimensional model or tabular model based on the data warehouse.
Design dimensions representing attributes like student demographics, course details, instructor information, and time (semester).
Define measures such as student enrollment count, average course rating, and instructor evaluation scores.
Create hierarchies to organize data, such as course categories and student demographics.
Process the SSAS model to populate it with data from the data warehouse.
SSRS (Reporting Services):
Leverage SSRS to create reports and visualizations based on the SSAS model or directly from the data warehouse.
Design reports to analyze student enrollment trends, instructor performance evaluations, course popularity, and student demographics.
Create interactive dashboards to monitor key metrics such as enrollment numbers, course completion rates, and instructor satisfaction scores.
Schedule report delivery to university administrators, faculty, and staff for decision-making purposes.
Implement drill-down and drill-through functionalities for users to explore data in detail, such as viewing enrollment trends by department or analyzing student performance by course.
Sample Queries for Analytical Insights:

Query to calculate the enrollment count by course and semester.
Query to identify courses with the highest enrollment numbers.
Query to analyze student demographics by department and course level.
Query to evaluate instructor performance based on student feedback and course completion rates.
Query to compare course popularity over multiple semesters.
By implementing this data warehousing solution with SSIS, SSAS, and SSRS, the university can gain valuable insights into student enrollment patterns, instructor effectiveness, and course performance, enabling them to make data-driven decisions to improve the quality of education and student experience

create tables script

```
/*

Professor. Michael Choi
Student. Vinay Joneja
IDS 521. Advanced Databases.

project. tool. Microsoft SQL Server Management Studio and Access DB
script. Language SQL - Created using Microsoft SQL Management Studio
create database


Summary. DATABASE. SQL Server
A. create student
B. create instructor
C. create course
D. create section
E. create takes
F. create department

*/


CREATE DATABASE IDS521

USE [ids521]
GO

/****** Table [dbo].[course] ******/



CREATE TABLE [dbo].[course](
[course_id] [varchar](8) NOT NULL,
[title] [varchar](50) NULL,
[dept_name] [varchar](20) NULL,
[credits] [numeric](2, 0) NULL,
[titles] [varchar](255) NULL,
PRIMARY KEY CLUSTERED
(
[course_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

/****** Table [dbo].[department] ******/
CREATE TABLE [dbo].[department](
[deptid] [int] NOT NULL,
[dept_name] [varchar](250) NULL,
[budget] [decimal](10, 2) NULL,
[instructor_id] [int] NULL,
PRIMARY KEY CLUSTERED
(
[deptid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Table [dbo].[instructor] ******/

CREATE TABLE [dbo].[instructor](
[id] [char](5) NULL,
[name_] [varchar](20) NULL,
[dept_name] [varchar](20) NULL,
[salary] [numeric](8, 2) NULL,
[names] [varchar](255) NULL
) ON [PRIMARY]


/****** Table [dbo].[section] ******/
CREATE TABLE [dbo].[section](
[section_name] [varchar](255) NULL,
[building] [varchar](255) NULL,
[class] [varchar](255) NULL,
[instructor_id] [varchar](10) NULL,
[sec_id] [varchar](10) NOT NULL,
[course_id] [varchar](10) NULL,
PRIMARY KEY CLUSTERED
(
[sec_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

/****** Table [dbo].[student] ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[student](
[ID] [varchar](5) NOT NULL,
[name] [varchar](20) NOT NULL,
[dept_name] [varchar](20) NULL,
[tot_cred] [numeric](3, 0) NULL,
PRIMARY KEY CLUSTERED
(
[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

/****** Table [dbo].[takes] ******/

CREATE TABLE [dbo].[takes](
[ID] [varchar](5) NOT NULL,
[course_id] [varchar](8) NULL,
[sec_id] [varchar](8) NULL,
[semester] [varchar](6) NULL,
[year] [numeric](4, 0) NULL,
[grade] [varchar](2) NULL,
[S_ID] [varchar](10) NULL,
PRIMARY KEY CLUSTERED
(
[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

results. sql script. sql package. sql server integration package. sql server reporting package.

```
/******************
Professor. Michael Choi
Student. Vinay Joneja
IDS 521. Advanced Databases.
project.
Tool. Microsoft SQL Server Management Studio and Access DB
Script. Language SQL - Created using Microsoft SQL Management Studio
ETL SCRIPT. for creating datawarehouse
Extract transfer load
Summary. Datawarehouse. SQL Server Integration Package
A. Student taking course
B. Students taking credits
C. Instructor teach classes
D. Instructor teach credits
E. Students taking course.
F. Instructor teaches course

Summary. Reporting. SQL Server Reporting Services
A. Student taking course
B. Students taking credits
C. Instructor teach classes
D. Instructor teach credits
E. Students taking course.
F. Instructor teaches course

*/
```

results. sql script. sql package. sql server integration package. sql server reporting package.

```
-- CREATE DATABASE FOR DATAWAHRE HOUSE
CREATE DATABASE IDS521DataWarehouse;

/* LINKED SERVER USING THE IDS 410. MICROSOFT ACCESS DATABASE. */

USE [master]
GO
EXEC master.dbo.sp_addlinkedserver
@server = N'IDS521',
@srvproduct=N'',
@provider=N'MSDASQL',
@datasrc=N'IDS521',
@provstr=N'MSDASQL'
```

```
GO

/* DATA IMPORT. ACCESS 410 TO SQL SERVER 521 */
USE ids521
INSERT INTO Student(s.[ID], s.[name])
select *
from openquery
(IDS521,
'Select StudentID, StudentFirstName + " " + StudentLastName as Name from student_t where studentid NOT IN ( 1,2,3)'
)


/* QUERY. */
USE ids521
SELECT * FROM STUDENT
SELECT * FROM COURSE
SELECT * FROM INSTRUCTOR
SELECT * FROM SECTION
SELECT * FROM TAKES
SELECT * FROM DEPARTMENT


-- DATA GENERATION. TABLE. TAKES
USE ids521
-- Variables for generating random data
DECLARE @Counter INT = 1;

-- Loop to insert 100 rows with random data
WHILE @Counter <= 100
BEGIN
INSERT INTO [dbo].[takes] ([ID], [course_id], [sec_id], [semester], [year], [grade])
VALUES (
CONCAT('ID', REPLICATE('0', 3 - LEN(@Counter)) + CAST(@Counter AS VARCHAR(3))),
CONCAT('C', RIGHT('0' + CAST(ABS(CHECKSUM(NEWID())) % 1000 AS VARCHAR(4)), 4)),
CONCAT('SEC', RIGHT('0' + CAST(ABS(CHECKSUM(NEWID())) % 1000 AS VARCHAR(4)), 4)),
CASE ABS(CHECKSUM(NEWID())) % 2 WHEN 0 THEN 'Fall' ELSE 'Spring' END,
2020 + ABS(CHECKSUM(NEWID())) % 5,
CASE ABS(CHECKSUM(NEWID())) % 5 WHEN 0 THEN 'A' WHEN 1 THEN 'B' WHEN 2 THEN 'C' WHEN 3 THEN 'D' ELSE 'F' END
);

SET @Counter = @Counter + 1;
END;



-- DATA. GENERATION. TABLE. SECTION

USE ids521
-- Variables for generating random data
DECLARE @CounterA INT = 1;


-- Loop to insert 100 rows with random data
WHILE @CounterA <= 100
BEGIN
INSERT INTO [dbo].[section] ([sec_id], [section_name], [instructor_id], [building], [class])
VALUES (
CONCAT('SEC', RIGHT('0' + CAST(@CounterA AS VARCHAR(7)), 7)),
'Section' + CAST(@CounterA AS VARCHAR(3)),
CONCAT('INST', RIGHT('0' + CAST(ABS(CHECKSUM(NEWID())) % 20 + 1 AS VARCHAR(2)), 10)),
'Building' + CAST(ABS(CHECKSUM(NEWID())) % 10 + 1 AS VARCHAR(2)),
'Class' + CAST(ABS(CHECKSUM(NEWID())) % 10 + 1 AS VARCHAR(2))
);

SET @CounterA = @CounterA + 1;
END;


-- DATA. GENERATION. TAKES

CREATE TABLE #RandomNumbers (
RandomNumber INT
);

DECLARE @Counter INT = 1;
DECLARE @RandomNumber INT;

WHILE @Counter <= 100
BEGIN
-- Generate a random number between 1 and 100
SET @RandomNumber = ABS(CAST(RAND() * 1000 AS INT)) % 100 + 1;

-- Check if the number already exists in the temporary table
IF NOT EXISTS (SELECT 1 FROM #RandomNumbers WHERE RandomNumber = @RandomNumber)
BEGIN
-- Insert the unique number into the temporary table
INSERT INTO #RandomNumbers (RandomNumber) VALUES (@RandomNumber);
SET @Counter = @Counter + 1;
END
END

UPDATE takes
SET S_ID = RandomNumber
FROM #RandomNumbers
WHERE [dbo].[takes].id = CONCAT('ID', RIGHT('0' + CAST(RandomNumber AS VARCHAR(3)), 7));

-- Select the distinct random numbers from the temporary table
SELECT RandomNumber
FROM #RandomNumbers;


-- DATAWAREHOUSE. CREATE TABLE. INSTRUCTOR SECTION.

USE IDS521DataWarehouse
-- Create a new table to store the results
CREATE TABLE [dbo].[instructor_section](
-- details from instructor table
[id] [char](5) NOT NULL,
[name_] [varchar](20) NULL,
[dept_name] [varchar](20) NULL,
[salary] [numeric](8, 2) NULL,
[names] [varchar](255) NULL,

-- details from section table

[section_name] [varchar](255) NULL,
[building] [varchar](255) NULL,
[class] [varchar](255) NULL,
[instructor_id] [varchar](10) NULL,
[sec_id] [varchar](10) NOT NULL,
[sec_course_id] [varchar](8) NOT NULL,

-- details from course table
[course_id] [varchar](8) NOT NULL,
[title] [varchar](50) NULL,
[c_dept_name] [varchar](20) NULL,
[credits] [numeric](2, 0) NULL,
[titles] [varchar](255) NULL

PRIMARY KEY CLUSTERED
(
[ID] ASC,
[sec_id] ASC
)
) ON [PRIMARY];
GO


-- DATAWAREHOUSE TABLE. INSTRUCTOR SECTION.
-- DATAWAREHOUSE TABLE. INSTRUCTOR SECTION.
-- DATA SQL SERVER INTEGRATION PACKAGE.
-- DATA ETL IDS21 TO IDS521DATAWAREHOUSE DATABASE


USE IDS521DataWarehouse
--Drop table [dbo].[instructor_section]

USE IDS521DataWarehouse
-- ETL Script
INSERT INTO [dbo].[instructor_section]
(
[id], [name_], [dept_name] , [salary], [names],
[section_name], [building], [class], [instructor_id], [sec_id], [sec_course_id],
[course_id], [title], [c_dept_name], [credits], [titles] )
SELECT
i.[id],i.[name_], i.[dept_name] , i.[salary], i.[names],
sec.[section_name], sec.[building], sec.[class], sec.[instructor_id], sec.[sec_id], sec.[course_id],
c.[course_id], c.[title], c.[dept_name], c.[credits], c.[titles]
FROM
[ids521].[dbo].[instructor] i
JOIN
[ids521].[dbo].[section] sec ON sec.[instructor_id] = i.[ID]
JOIN
[ids521].[dbo].[course] c ON c.[course_id] = sec.[course_id];


-- QUERY. INSTRUCTOR TEACH SECTIONS.

SELECT TOP (1000) [id], [names], [dept_name], [salary], [section_name],
[building], [class], [instructor_id],[sec_id], [sec_course_id],
[course_id], [title], [c_dept_name] , [credits], [titles]
FROM [IDS521DataWarehouse].[dbo].[instructor_section]
ORDER BY ID asc;
```

```
-- REPORTING. INSTRUCTOR TEACH CREDITS

SELECT TOP (1000) [id], [names], SUM( [credits]) AS sumcredits
FROM [IDS521DataWarehouse].[dbo].[instructor_section]
GROUP BY [id], [names]

ORDER BY ID asc;



-- REPORTING. INSTRUCTOR IN DEPARTMENTS.

SELECT TOP (1000) [dept_name], COUNT( [ID]) AS countinstructors

FROM [IDS521DataWarehouse].[dbo].[instructor_section]

GROUP BY [dept_name]

ORDER BY [dept_name] asc;


-- REPORTING. STUDENTS CREDITS DEPARTMENTS.
SELECT TOP (1000) [id] , [name], SUM( [tot_cred]) AS sumcredits
FROM [IDS521DataWarehouse].[dbo].[student_section]
GROUP BY [id], [name]
ORDER BY [id] asc;

-- REPORTING. STUDENTS. DEPARTMENTS.
SELECT TOP (1000) [dept_name], COUNT( [ID]) AS countstudents
FROM [IDS521DataWarehouse].[dbo].[student_section]
GROUP BY [dept_name]
ORDER BY [dept_name] asc;


-- REPORTING. STUDENTS. CLASS. SECTION.

SELECT TOP (1000) [sec_id], COUNT([ID]) AS countstudents

FROM [IDS521DataWarehouse].[dbo].[student_section]

GROUP BY [sec_id]

ORDER BY [sec_id] asc;
```

User Document

A user document file (PPT/doc/PDF format only) includes:

1. How to use your analytic database
2. Screenshots of queries or SSAS/SSRS screenshots
3. Describe your sample data
4. How to use SSIS/SSAS/SSRS (brief summary with your computer screenshots)

How to use your analytic database

Using SQL Server Management Studio (SSMS) is essential for database administrators, developers, and analysts working with Microsoft SQL Server databases. To effectively utilize SSMS, start by launching the application and connecting to your SQL Server instance. Upon connection, you'll have access to a wide range of tools and functionalities to manage databases, execute queries, and perform administrative tasks.

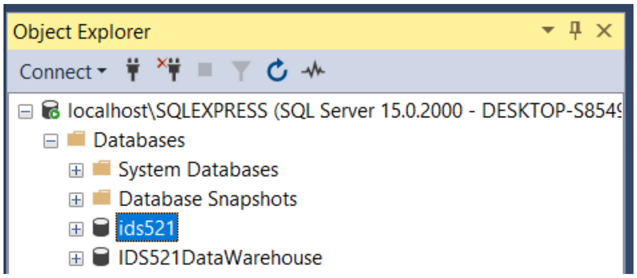
One of the primary features of SSMS is the Object Explorer, which provides a hierarchical view of the SQL Server instance, including databases, tables, views, stored procedures, and more. You can use Object Explorer to navigate through your database schema, view properties of database objects, and perform various management tasks such as creating, modifying, or deleting objects.

SSMS also includes a powerful query editor that allows you to write and execute Transact-SQL (T-SQL) queries against your databases. The query editor provides syntax highlighting, IntelliSense, and debugging capabilities, making it easier to write and debug complex SQL queries.

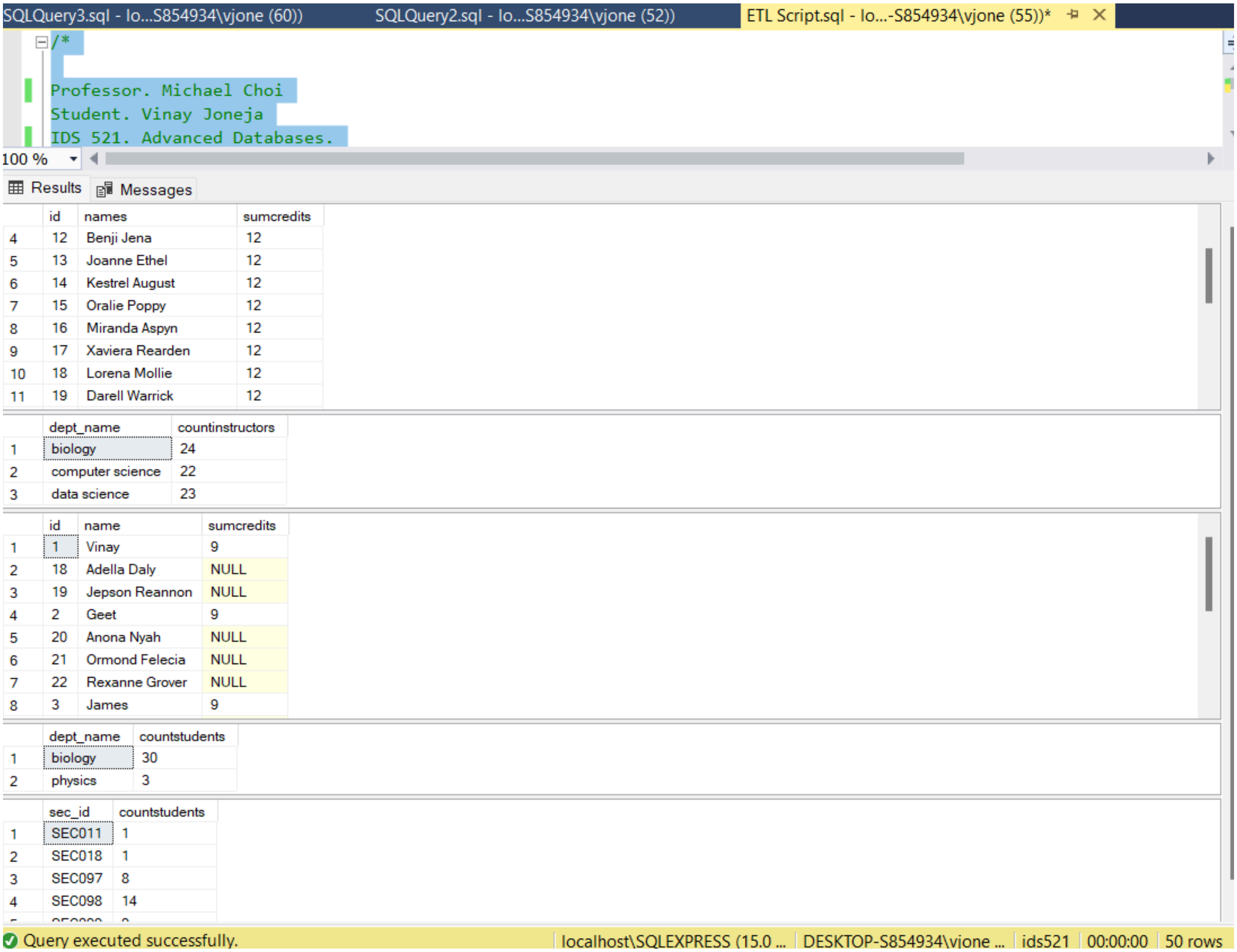
In addition to querying databases, SSMS offers tools for database administration tasks such as creating and managing database backups, configuring security settings, and monitoring database performance. You can also use SSMS to import and export data, generate database diagrams, and perform database maintenance tasks.

Furthermore, SSMS supports integration with source control systems, allowing you to manage database scripts and version control your database schema changes.

Overall, mastering SQL Server Management Studio enables you to efficiently manage and administer SQL Server databases, optimize performance, troubleshoot issues, and ensure the integrity and security of your data. With its comprehensive set of features and user-friendly interface, SSMS is an indispensable tool for anyone working with SQL Server databases.



Screenshots of queries or SSAS/SSRS screenshots



SQLQuery3.sql - lo...S854934\vjone (60))
SQLQuery2.sql - lo...S854934\vjone (52))
ETL Script.sql - lo...-S854934\vjone (55))*
100 %
Results
Messages

```
/*

Professor. Michael Choi
Student. Vinay Joneja
IDS 521. Advanced Databases.
```

	id	names	sumcredits
4	12	Benji Jena	12
5	13	Joanne Ethel	12
6	14	Kestrel August	12
7	15	Oralie Poppy	12
8	16	Miranda Aspyn	12
9	17	Xaviera Rearden	12
10	18	Lorena Mollie	12
11	19	Darell Warrick	12

	dept_name	countinstructors
1	biology	24
2	computer science	22
3	data science	23

	id	name	sumcredits
1	1	Vinay	9
2	18	Adella Daly	NULL
3	19	Jepson Reannon	NULL
4	2	Geet	9
5	20	Anona Nyah	NULL
6	21	Ormond Felecia	NULL
7	22	Rexanne Grover	NULL
8	3	James	9

	dept_name	countstudents
1	biology	30
2	physics	3

	sec_id	countstudents
1	SEC011	1
2	SEC018	1
3	SEC097	8
4	SEC098	14

Query executed successfully. | localhost\SQLEXPRESS (15.0 ... | DESKTOP-S854934\vjone ... | ids521 | 00:00:00 | 50 rows

Describe your sample data

Course Table:
course_id: Unique identifier for the course.
title: Title of the course.
dept_name: Department offering the course.
credits: Number of credits assigned to the course.
titles: Additional title information (potentially redundant with title).
Department Table:
deptid: Unique identifier for the department.
dept_name: Name of the department.
budget: Budget allocated to the department.
instructor_id: ID of the instructor associated with the department.
Instructor Table:
id: Unique identifier for the instructor.
name_: Name of the instructor.
dept_name: Department to which the instructor belongs.
salary: Salary of the instructor.
names: Additional names information.
Section Table:
section_name: Name of the section.
building: Building where the section is held.
class: Class identifier.

instructor_id: ID of the instructor teaching the section.

sec_id: Unique identifier for the section.

course_id: ID of the course associated with the section.

Student Table:

ID: Unique identifier for the student.

name: Name of the student.

dept_name: Department to which the student belongs.

tot_cred: Total credits earned by the student.

Takes Table:

ID: Unique identifier for the enrollment.

course_id: ID of the course taken.

sec_id: ID of the section in which the course is taken.

semester: Semester in which the course is taken.

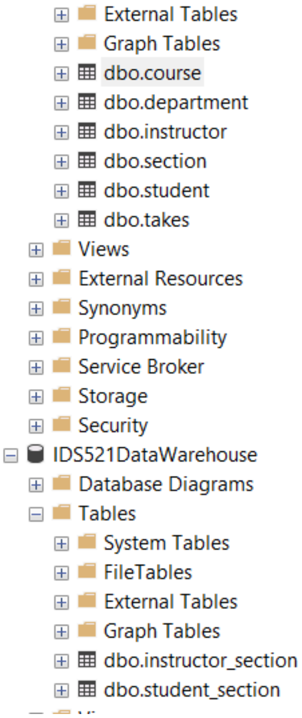
year: Year in which the course is taken.

grade: Grade received in the course.

S_ID: Student ID associated with the enrollment.

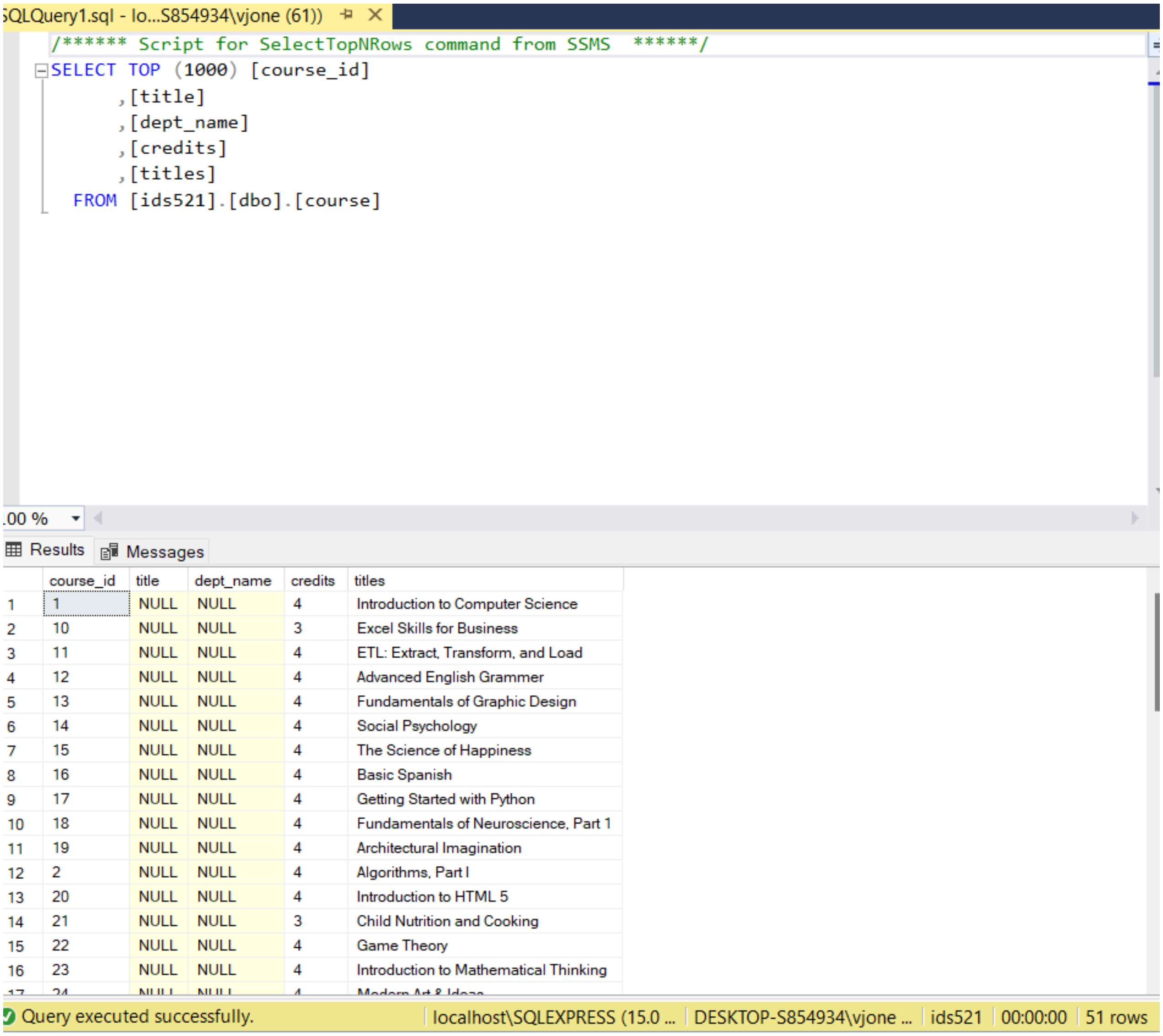
These tables seem to represent a simplified schema for managing courses, departments, instructors, sections, students, and their interactions such as enrollments and teaching assignments.

ids 521 database and datawharehouse structure



```
        ⊞ External Tables
        ⊞ Graph Tables
        ⊞ dbo.course
        ⊞ dbo.department
        ⊞ dbo.instructor
        ⊞ dbo.section
        ⊞ dbo.student
        ⊞ dbo.takes
    ⊞ Views
    ⊞ External Resources
    ⊞ Synonyms
    ⊞ Programmability
    ⊞ Service Broker
    ⊞ Storage
    ⊞ Security
⊟ IDS521DataWarehouse
    ⊞ Database Diagrams
    ⊟ Tables
        ⊞ System Tables
        ⊞ FileTables
        ⊞ External Tables
        ⊞ Graph Tables
        ⊞ dbo.instructor_section
        ⊞ dbo.student_section
```

course

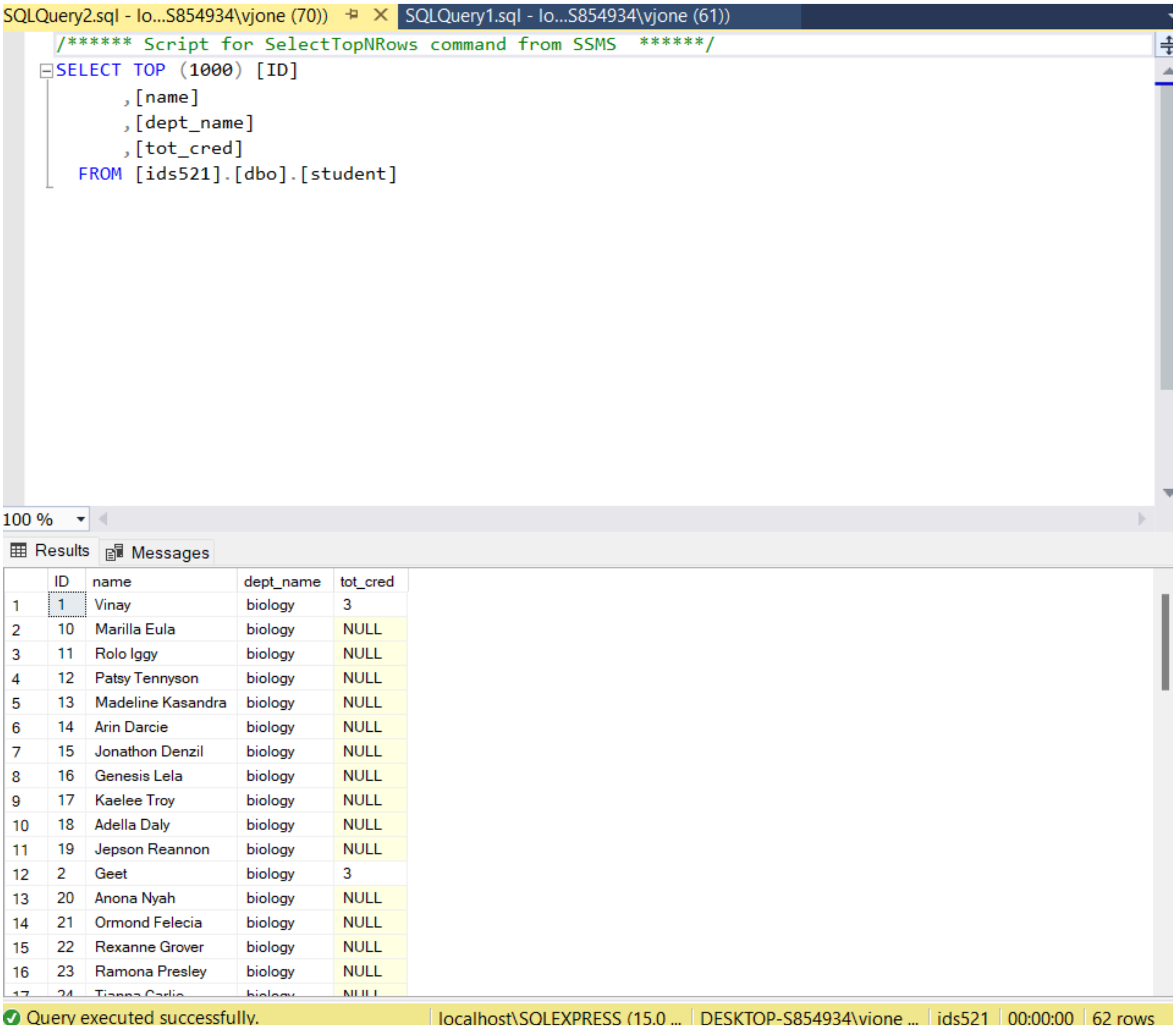


```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [course_id]
      ,[title]
      ,[dept_name]
      ,[credits]
      ,[titles]
  FROM [ids521].[dbo].[course]
```

	course_id	title	dept_name	credits	titles
1	1	NULL	NULL	4	Introduction to Computer Science
2	10	NULL	NULL	3	Excel Skills for Business
3	11	NULL	NULL	4	ETL: Extract, Transform, and Load
4	12	NULL	NULL	4	Advanced English Grammer
5	13	NULL	NULL	4	Fundamentals of Graphic Design
6	14	NULL	NULL	4	Social Psychology
7	15	NULL	NULL	4	The Science of Happiness
8	16	NULL	NULL	4	Basic Spanish
9	17	NULL	NULL	4	Getting Started with Python
10	18	NULL	NULL	4	Fundamentals of Neuroscience, Part 1
11	19	NULL	NULL	4	Architectural Imagination
12	2	NULL	NULL	4	Algorithms, Part I
13	20	NULL	NULL	4	Introduction to HTML 5
14	21	NULL	NULL	3	Child Nutrition and Cooking
15	22	NULL	NULL	4	Game Theory
16	23	NULL	NULL	4	Introduction to Mathematical Thinking
17	24	NULL	NULL	4	Modern Art & Ideas

Query executed successfully. | localhost\SQLEXPRESS (15.0 ... | DESKTOP-S854934\vjone ... | ids521 | 00:00:00 | 51 rows

student

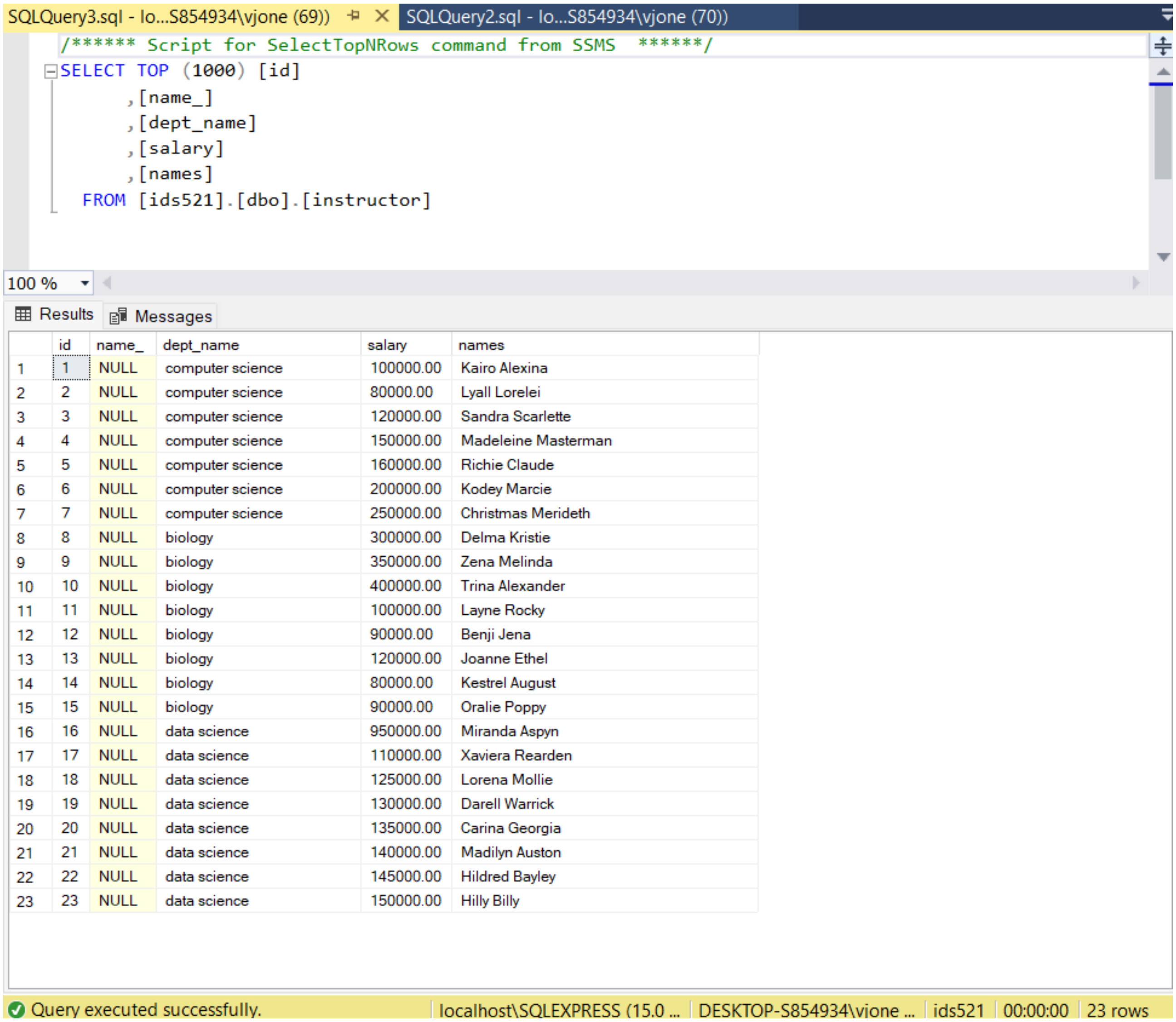

SQLQuery2.sql - lo...S854934\vjone (70))
SQLQuery1.sql - lo...S854934\vjone (61))
100 %
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [ID]
      ,[name]
      ,[dept_name]
      ,[tot_cred]
  FROM [ids521].[dbo].[student]
```

	ID	name	dept_name	tot_cred
1	1	Vinay	biology	3
2	10	Marilla Eula	biology	NULL
3	11	Rolo Iggy	biology	NULL
4	12	Patsy Tennyson	biology	NULL
5	13	Madeline Kasandra	biology	NULL
6	14	Arin Darcie	biology	NULL
7	15	Jonathon Denzil	biology	NULL
8	16	Genesis Lela	biology	NULL
9	17	Kaelee Troy	biology	NULL
10	18	Adella Daly	biology	NULL
11	19	Jepson Reannon	biology	NULL
12	2	Geet	biology	3
13	20	Anona Nyah	biology	NULL
14	21	Ormond Felecia	biology	NULL
15	22	Rexanne Grover	biology	NULL
16	23	Ramona Presley	biology	NULL
17	24	Tianna Carlie	biology	NULL

Query executed successfully. | localhost\SQLEXPRESS (15.0 ... | DESKTOP-S854934\vjone ... | ids521 | 00:00:00 | 62 rows

instructor

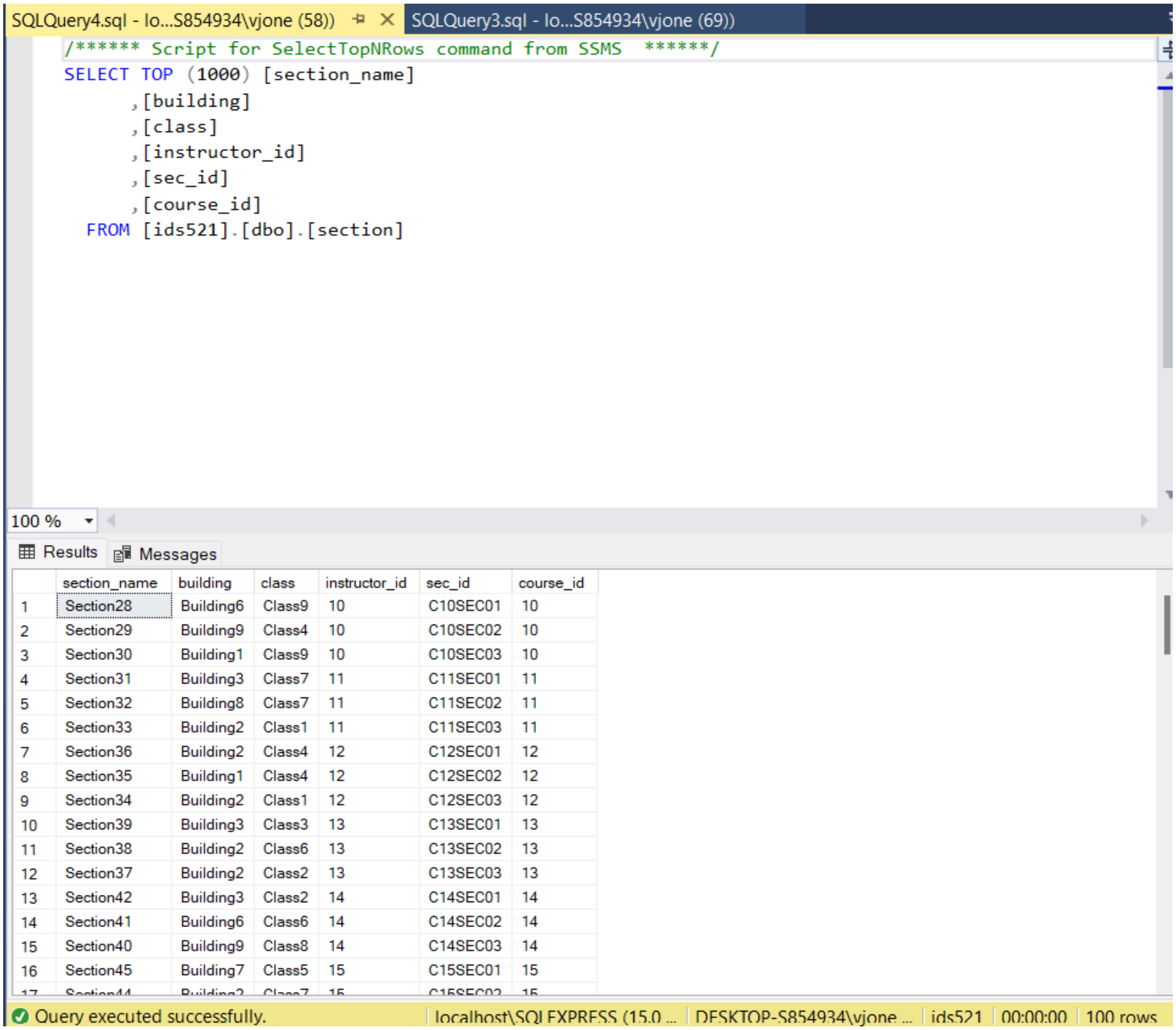

SQLQuery3.sql - lo...S854934\vjone (69))
SQLQuery2.sql - lo...S854934\vjone (70))
100 %
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [id]
      ,[name_]
      ,[dept_name]
      ,[salary]
      ,[names]
  FROM [ids521].[dbo].[instructor]
```

	id	name_	dept_name	salary	names
1	1	NULL	computer science	100000.00	Kairo Alexina
2	2	NULL	computer science	80000.00	Lyall Lorelei
3	3	NULL	computer science	120000.00	Sandra Scarlette
4	4	NULL	computer science	150000.00	Madeleine Masterman
5	5	NULL	computer science	160000.00	Richie Claude
6	6	NULL	computer science	200000.00	Kodey Marcie
7	7	NULL	computer science	250000.00	Christmas Merideth
8	8	NULL	biology	300000.00	Delma Kristie
9	9	NULL	biology	350000.00	Zena Melinda
10	10	NULL	biology	400000.00	Trina Alexander
11	11	NULL	biology	100000.00	Layne Rocky
12	12	NULL	biology	90000.00	Benji Jena
13	13	NULL	biology	120000.00	Joanne Ethel
14	14	NULL	biology	80000.00	Kestrel August
15	15	NULL	biology	90000.00	Oralie Poppy
16	16	NULL	data science	950000.00	Miranda Aspyn
17	17	NULL	data science	110000.00	Xaviera Rearden
18	18	NULL	data science	125000.00	Lorena Mollie
19	19	NULL	data science	130000.00	Darell Warrick
20	20	NULL	data science	135000.00	Carina Georgia
21	21	NULL	data science	140000.00	Madilyn Auston
22	22	NULL	data science	145000.00	Hildred Bayley
23	23	NULL	data science	150000.00	Hilly Billy

Query executed successfully. | localhost\SQLEXPRESS (15.0 ... | DESKTOP-S854934\vjone ... | ids521 | 00:00:00 | 23 rows

section



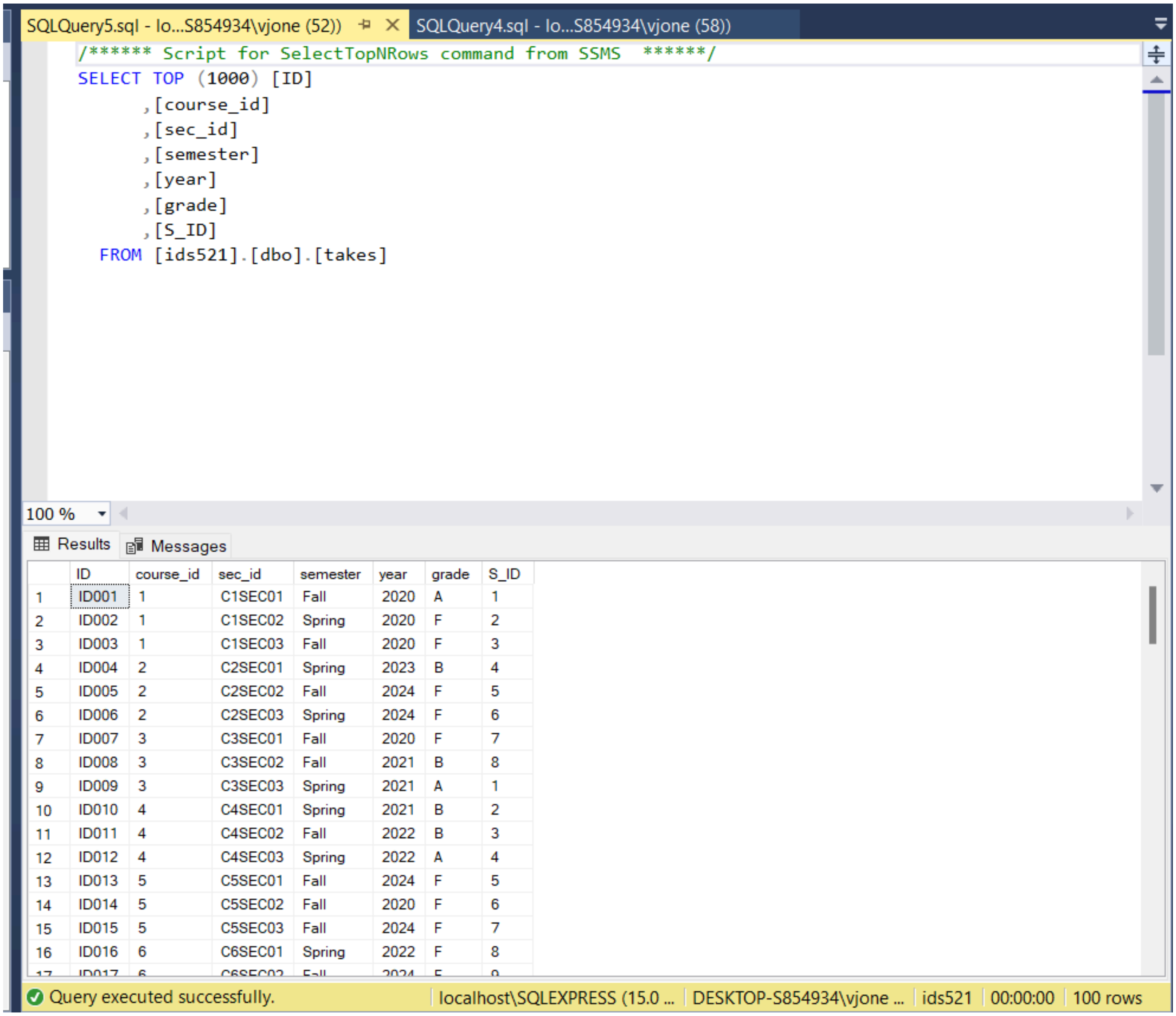
SQLQuery4.sql - lo...S854934\vjone (58))
SQLQuery3.sql - lo...S854934\vjone (69))
100 %
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [section_name]
      ,[building]
      ,[class]
      ,[instructor_id]
      ,[sec_id]
      ,[course_id]
  FROM [ids521].[dbo].[section]
```

	section_name	building	class	instructor_id	sec_id	course_id
1	Section28	Building6	Class9	10	C10SEC01	10
2	Section29	Building9	Class4	10	C10SEC02	10
3	Section30	Building1	Class9	10	C10SEC03	10
4	Section31	Building3	Class7	11	C11SEC01	11
5	Section32	Building8	Class7	11	C11SEC02	11
6	Section33	Building2	Class1	11	C11SEC03	11
7	Section36	Building2	Class4	12	C12SEC01	12
8	Section35	Building1	Class4	12	C12SEC02	12
9	Section34	Building2	Class1	12	C12SEC03	12
10	Section39	Building3	Class3	13	C13SEC01	13
11	Section38	Building2	Class6	13	C13SEC02	13
12	Section37	Building2	Class2	13	C13SEC03	13
13	Section42	Building3	Class2	14	C14SEC01	14
14	Section41	Building6	Class6	14	C14SEC02	14
15	Section40	Building9	Class8	14	C14SEC03	14
16	Section45	Building7	Class5	15	C15SEC01	15
17	Section44	Building2	Class7	15	C15SEC02	15

Query executed successfully. | localhost\SQLEXPRESS (15.0 ... | DESKTOP-S854934\vjone ... | ids521 | 00:00:00 | 100 rows

student takes course section

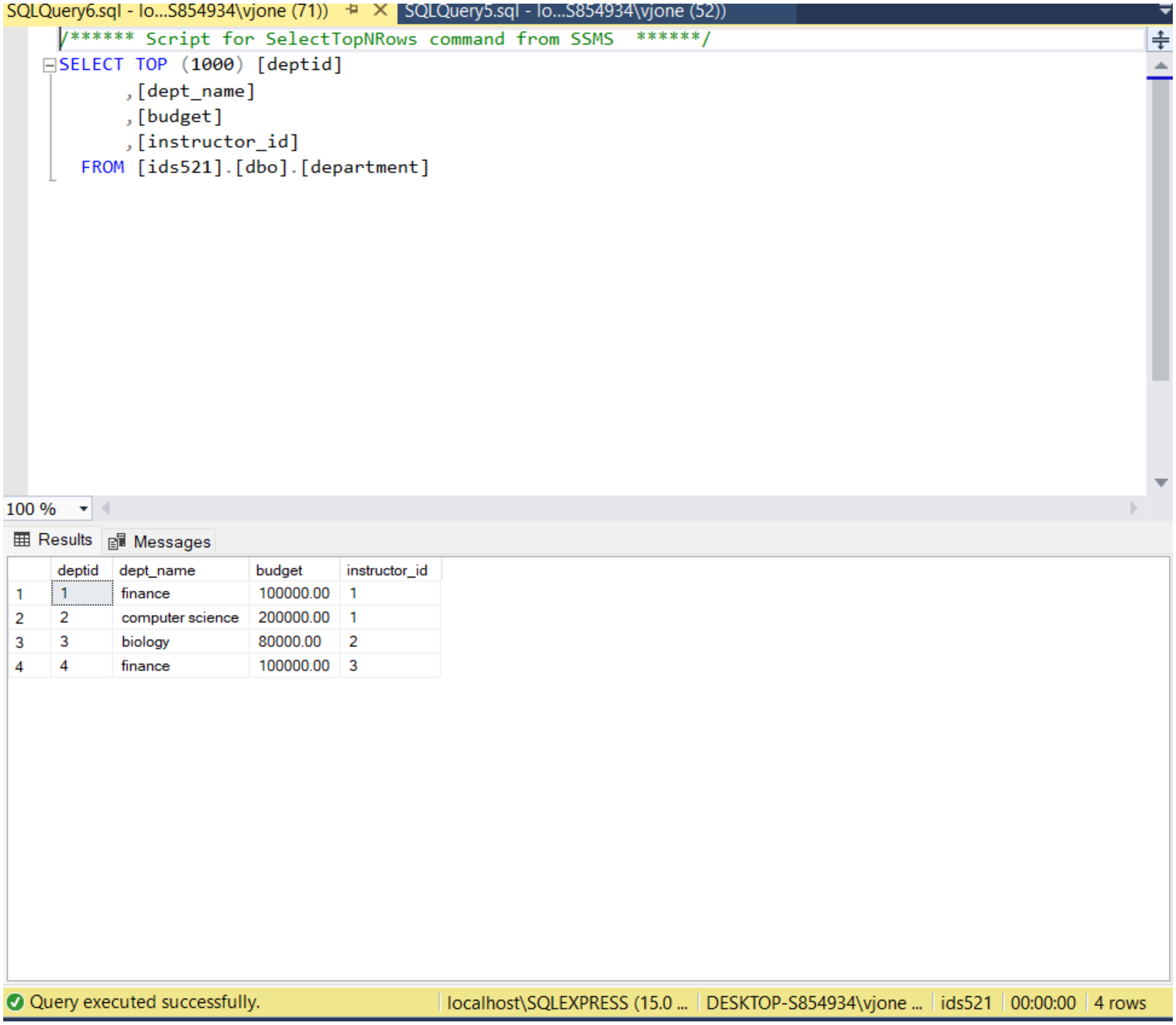

SQLQuery5.sql - lo...S854934\vjone (52))
SQLQuery4.sql - lo...S854934\vjone (58))
100 %
Results
Messages

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [ID]
      ,[course_id]
      ,[sec_id]
      ,[semester]
      ,[year]
      ,[grade]
      ,[S_ID]
  FROM [ids521].[dbo].[takes]
```

	ID	course_id	sec_id	semester	year	grade	S_ID
1	ID001	1	C1SEC01	Fall	2020	A	1
2	ID002	1	C1SEC02	Spring	2020	F	2
3	ID003	1	C1SEC03	Fall	2020	F	3
4	ID004	2	C2SEC01	Spring	2023	B	4
5	ID005	2	C2SEC02	Fall	2024	F	5
6	ID006	2	C2SEC03	Spring	2024	F	6
7	ID007	3	C3SEC01	Fall	2020	F	7
8	ID008	3	C3SEC02	Fall	2021	B	8
9	ID009	3	C3SEC03	Spring	2021	A	1
10	ID010	4	C4SEC01	Spring	2021	B	2
11	ID011	4	C4SEC02	Fall	2022	B	3
12	ID012	4	C4SEC03	Spring	2022	A	4
13	ID013	5	C5SEC01	Fall	2024	F	5
14	ID014	5	C5SEC02	Fall	2020	F	6
15	ID015	5	C5SEC03	Fall	2024	F	7
16	ID016	6	C6SEC01	Spring	2022	F	8
17	ID017	6	C6SEC02	Fall	2024	F	9

Query executed successfully. | localhost\SQLEXPRESS (15.0 ... | DESKTOP-S854934\vjone ... | ids521 | 00:00:00 | 100 rows

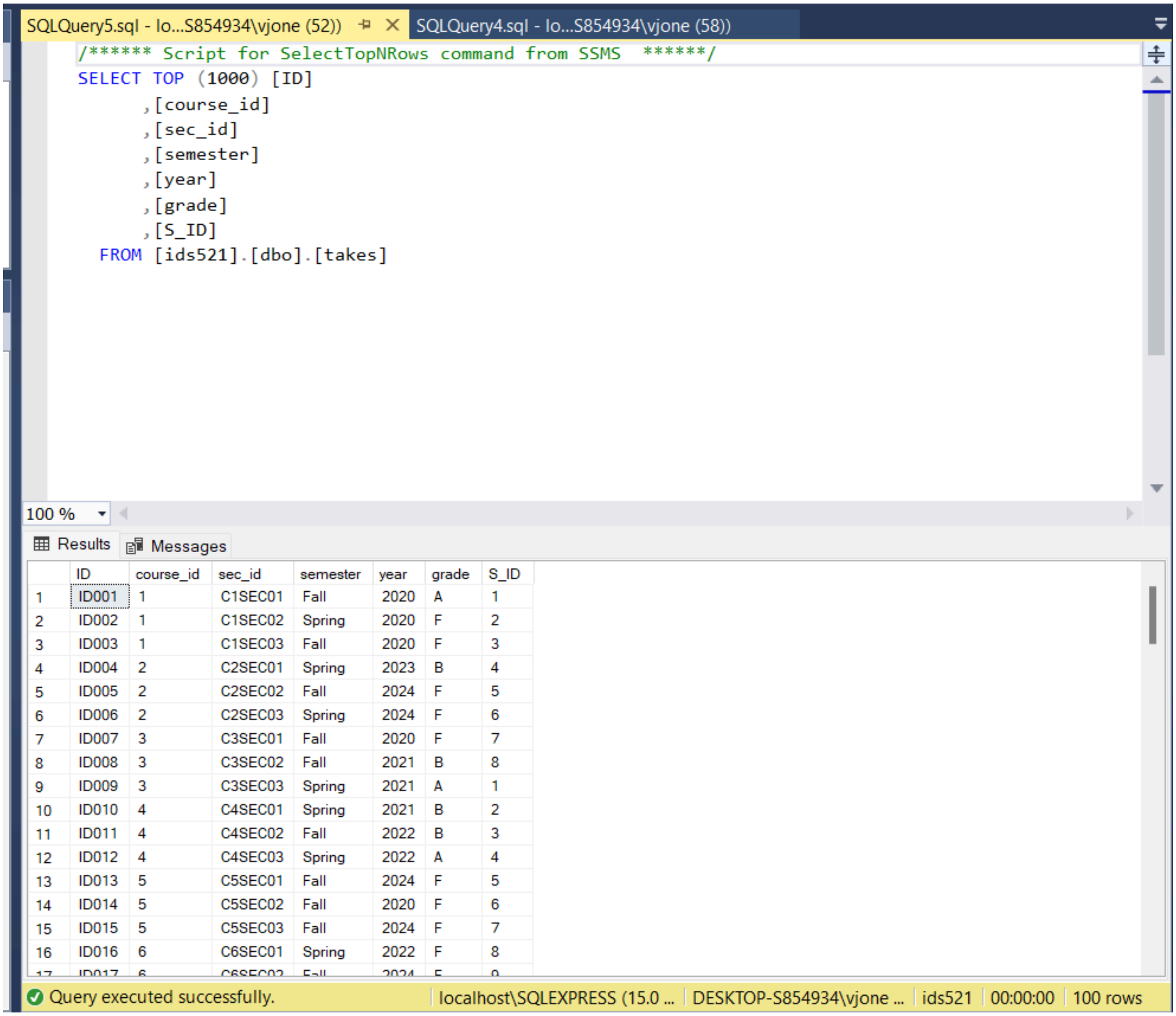
department


SQLQuery6.sql - lo...S854934\vjone (71))
SQLQuery5.sql - lo...S854934\vjone (52))
100 %
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [deptid]
      ,[dept_name]
      ,[budget]
      ,[instructor_id]
  FROM [ids521].[dbo].[department]
```

	deptid	dept_name	budget	instructor_id
1	1	finance	100000.00	1
2	2	computer science	200000.00	1
3	3	biology	80000.00	2
4	4	finance	100000.00	3

Query executed successfully. | localhost\SQLEXPRESS (15.0 ... | DESKTOP-S854934\vjone ... | ids521 | 00:00:00 | 4 rows

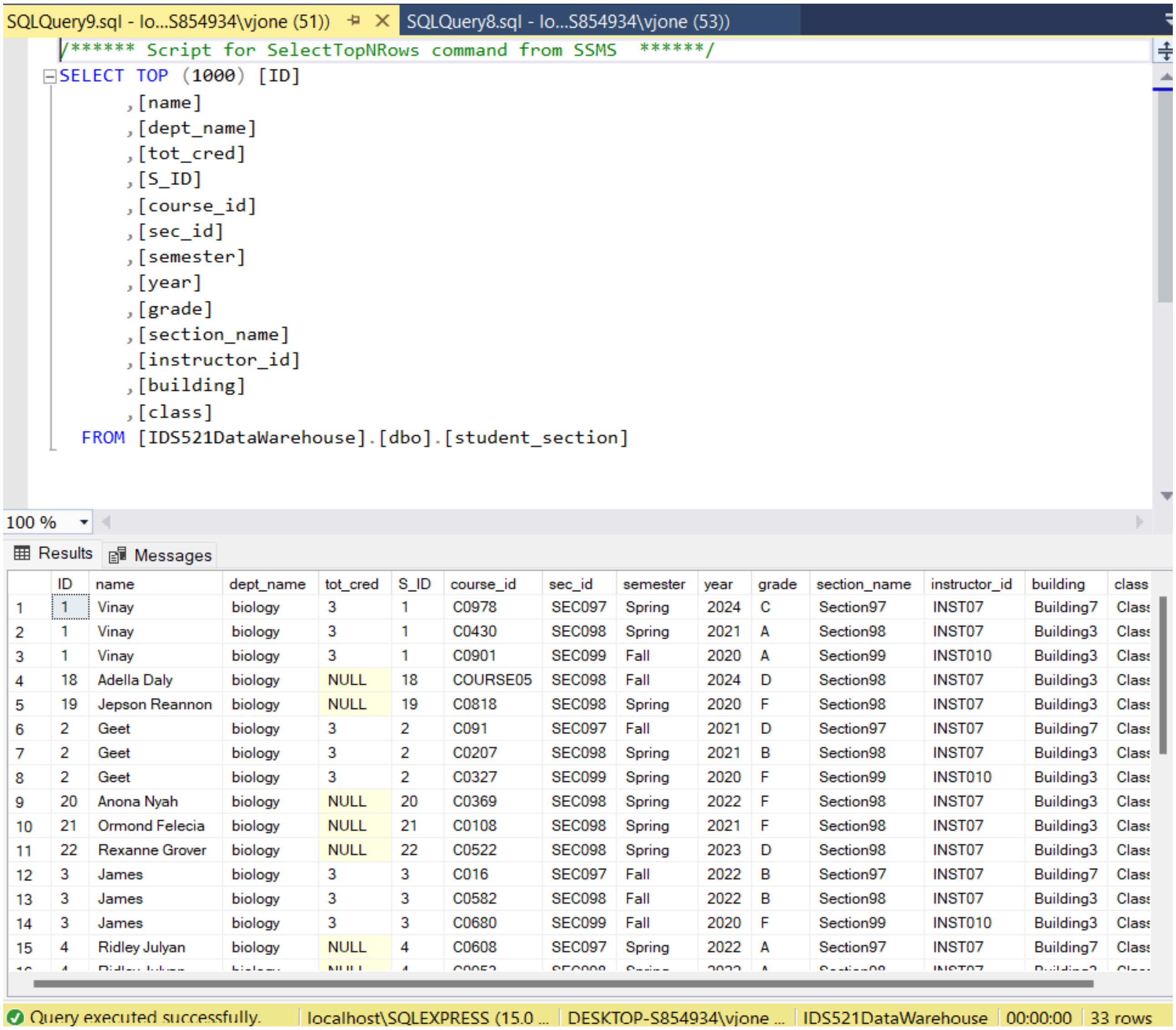

SQLQuery5.sql - lo...S854934\vjone (52))
SQLQuery4.sql - lo...S854934\vjone (58))
100 %
Results
Messages

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [ID]
      ,[course_id]
      ,[sec_id]
      ,[semester]
      ,[year]
      ,[grade]
      ,[S_ID]
  FROM [ids521].[dbo].[takes]
```

	ID	course_id	sec_id	semester	year	grade	S_ID
1	ID001	1	C1SEC01	Fall	2020	A	1
2	ID002	1	C1SEC02	Spring	2020	F	2
3	ID003	1	C1SEC03	Fall	2020	F	3
4	ID004	2	C2SEC01	Spring	2023	B	4
5	ID005	2	C2SEC02	Fall	2024	F	5
6	ID006	2	C2SEC03	Spring	2024	F	6
7	ID007	3	C3SEC01	Fall	2020	F	7
8	ID008	3	C3SEC02	Fall	2021	B	8
9	ID009	3	C3SEC03	Spring	2021	A	1
10	ID010	4	C4SEC01	Spring	2021	B	2
11	ID011	4	C4SEC02	Fall	2022	B	3
12	ID012	4	C4SEC03	Spring	2022	A	4
13	ID013	5	C5SEC01	Fall	2024	F	5
14	ID014	5	C5SEC02	Fall	2020	F	6
15	ID015	5	C5SEC03	Fall	2024	F	7
16	ID016	6	C6SEC01	Spring	2022	F	8
17	ID017	6	C6SEC02	Fall	2024	F	9

Query executed successfully. | localhost\SQLEXPRESS (15.0 ... | DESKTOP-S854934\vjone ... | ids521 | 00:00:00 | 100 rows

datawarehouse student section

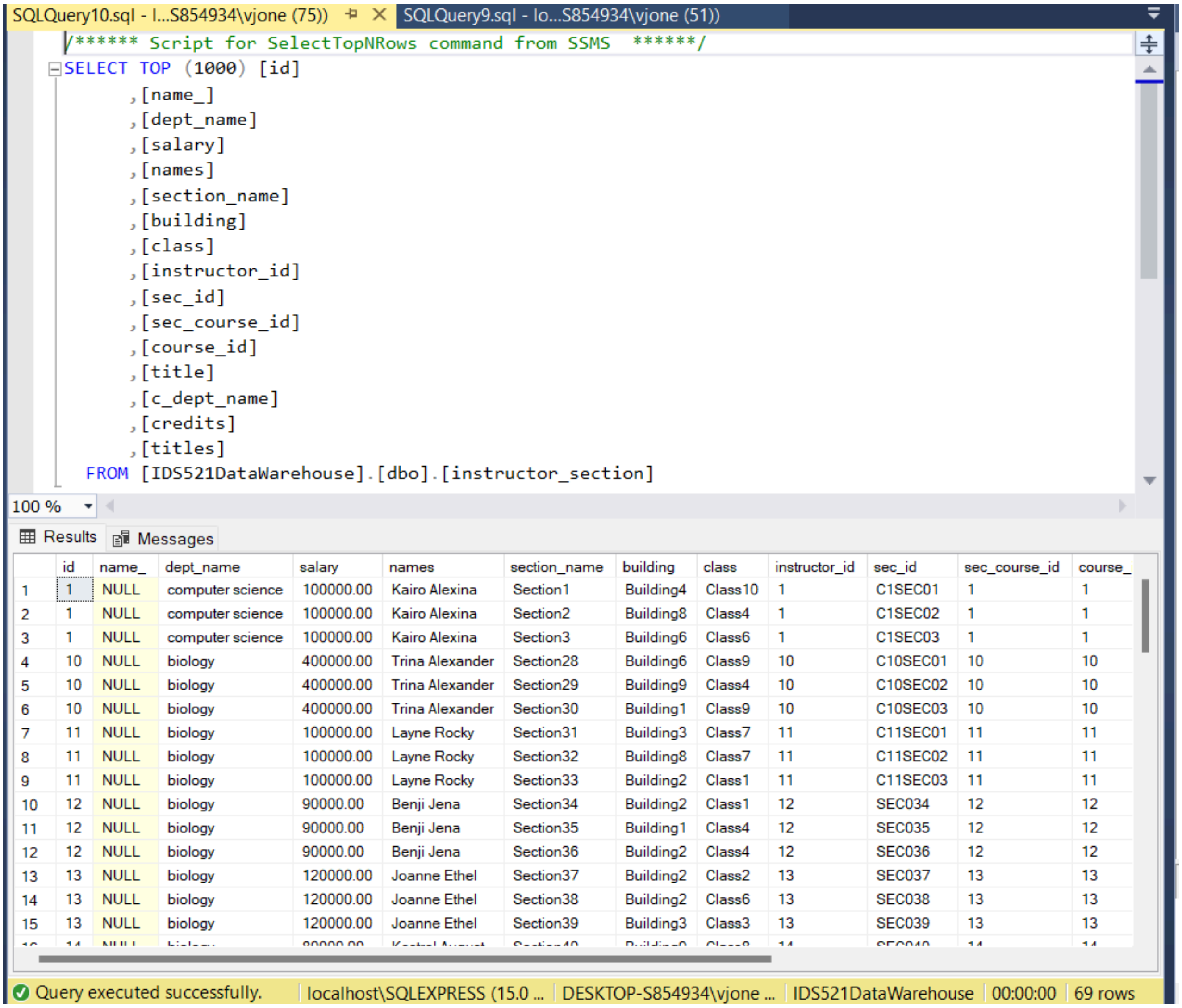

SQLQuery9.sql - lo...S854934\vjone (51))
SQLQuery8.sql - lo...S854934\vjone (53))

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [ID]
      ,[name]
      ,[dept_name]
      ,[tot_cred]
      ,[S_ID]
      ,[course_id]
      ,[sec_id]
      ,[semester]
      ,[year]
      ,[grade]
      ,[section_name]
      ,[instructor_id]
      ,[building]
      ,[class]
  FROM [IDS521DataWarehouse].[dbo].[student_section]
```

	ID	name	dept_name	tot_cred	S_ID	course_id	sec_id	semester	year	grade	section_name	instructor_id	building	class
1	1	Vinay	biology	3	1	C0978	SEC097	Spring	2024	C	Section97	INST07	Building7	Class
2	1	Vinay	biology	3	1	C0430	SEC098	Spring	2021	A	Section98	INST07	Building3	Class
3	1	Vinay	biology	3	1	C0901	SEC099	Fall	2020	A	Section99	INST010	Building3	Class
4	18	Adella Daly	biology	NULL	18	COURSE05	SEC098	Fall	2024	D	Section98	INST07	Building3	Class
5	19	Jepson Reannon	biology	NULL	19	C0818	SEC098	Spring	2020	F	Section98	INST07	Building3	Class
6	2	Geet	biology	3	2	C091	SEC097	Fall	2021	D	Section97	INST07	Building7	Class
7	2	Geet	biology	3	2	C0207	SEC098	Spring	2021	B	Section98	INST07	Building3	Class
8	2	Geet	biology	3	2	C0327	SEC099	Spring	2020	F	Section99	INST010	Building3	Class
9	20	Anona Nyah	biology	NULL	20	C0369	SEC098	Spring	2022	F	Section98	INST07	Building3	Class
10	21	Ormond Felecia	biology	NULL	21	C0108	SEC098	Spring	2021	F	Section98	INST07	Building3	Class
11	22	Rexanne Grover	biology	NULL	22	C0522	SEC098	Spring	2023	D	Section98	INST07	Building3	Class
12	3	James	biology	3	3	C016	SEC097	Fall	2022	B	Section97	INST07	Building7	Class
13	3	James	biology	3	3	C0582	SEC098	Fall	2022	B	Section98	INST07	Building3	Class
14	3	James	biology	3	3	C0680	SEC099	Fall	2020	F	Section99	INST010	Building3	Class
15	4	Ridley Julyan	biology	NULL	4	C0608	SEC097	Spring	2022	A	Section97	INST07	Building7	Class

Query executed successfully. | localhost\SQLEXPRESS (15.0 ... | DESKTOP-S854934\vjone ... | IDS521DataWarehouse | 00:00:00 | 33 rows

datawarehouse instructor section



SQLQuery10.sql - l...S854934\vjone (75)) SQLQuery9.sql - lo...S854934\vjone (51))

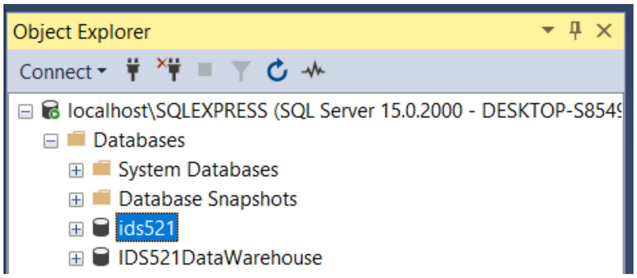
```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [id]
      ,[name_]
      ,[dept_name]
      ,[salary]
      ,[names]
      ,[section_name]
      ,[building]
      ,[class]
      ,[instructor_id]
      ,[sec_id]
      ,[sec_course_id]
      ,[course_id]
      ,[title]
      ,[c_dept_name]
      ,[credits]
      ,[titles]
  FROM [IDS521DataWarehouse].[dbo].[instructor_section]
```

	id	name_	dept_name	salary	names	section_name	building	class	instructor_id	sec_id	sec_course_id	course_
1	1	NULL	computer science	100000.00	Kairo Alexina	Section1	Building4	Class10	1	C1SEC01	1	1
2	1	NULL	computer science	100000.00	Kairo Alexina	Section2	Building8	Class4	1	C1SEC02	1	1
3	1	NULL	computer science	100000.00	Kairo Alexina	Section3	Building6	Class6	1	C1SEC03	1	1
4	10	NULL	biology	400000.00	Trina Alexander	Section28	Building6	Class9	10	C10SEC01	10	10
5	10	NULL	biology	400000.00	Trina Alexander	Section29	Building9	Class4	10	C10SEC02	10	10
6	10	NULL	biology	400000.00	Trina Alexander	Section30	Building1	Class9	10	C10SEC03	10	10
7	11	NULL	biology	100000.00	Layne Rocky	Section31	Building3	Class7	11	C11SEC01	11	11
8	11	NULL	biology	100000.00	Layne Rocky	Section32	Building8	Class7	11	C11SEC02	11	11
9	11	NULL	biology	100000.00	Layne Rocky	Section33	Building2	Class1	11	C11SEC03	11	11
10	12	NULL	biology	90000.00	Benji Jena	Section34	Building2	Class1	12	SEC034	12	12
11	12	NULL	biology	90000.00	Benji Jena	Section35	Building1	Class4	12	SEC035	12	12
12	12	NULL	biology	90000.00	Benji Jena	Section36	Building2	Class4	12	SEC036	12	12
13	13	NULL	biology	120000.00	Joanne Ethel	Section37	Building2	Class2	13	SEC037	13	13
14	13	NULL	biology	120000.00	Joanne Ethel	Section38	Building2	Class6	13	SEC038	13	13
15	13	NULL	biology	120000.00	Joanne Ethel	Section39	Building3	Class3	13	SEC039	13	13
16	14	NULL	biology	80000.00	Kestrel August	Section40	Building9	Class8	14	SEC040	14	14

Query executed successfully. | localhost\SQLEXPRESS (15.0 ... | DESKTOP-S854934\vjone ... | IDS521DataWarehouse | 00:00:00 | 69 rows

How to use SSIS/SSAS/SSRS (brief summary with your computer screenshots)

brief summary of how to use SSIS (SQL Server Integration Services), SSAS (SQL Server Analysis Services), and SSRS (SQL Server Reporting Services):



- localhost\SQLEXPRESS (SQL Server 15.0.2000 - DESKTOP-S854
  - Databases
    - System Databases
    - Database Snapshots
    - ids521
    - IDS521DataWarehouse

SSIS (SQL Server Integration Services):
Open SQL Server Data Tools (SSDT) or SQL Server Management Studio (SSMS) and create a new Integration Services project.
Within the project, create packages to extract, transform, and load (ETL) data from various sources into your SQL Server database.
Use the SSIS Toolbox to drag and drop tasks such as data flow tasks, control flow tasks, and other transformations onto the design surface.
Configure each task by double-clicking on it and setting properties in the editor.
Connect tasks together using precedence constraints to define the flow of data and control logic.
Test and debug your packages locally before deploying them to the SSIS Catalog or SQL Server instance for execution.
Schedule package execution using SQL Server Agent or another scheduling tool.

SSAS (SQL Server Analysis Services):
Launch SQL Server Data Tools (SSDT) or SQL Server Management Studio (SSMS) and create a new Analysis Services project.
Design your multidimensional or tabular model by defining dimensions, measures, hierarchies, and relationships.
Deploy your model to an Analysis Services instance.
Process your model to populate it with data from your data source.
Use SQL Server Management Studio or Excel with Power Pivot to connect to your Analysis Services database and create reports, pivot tables, and data visualizations.
Monitor and optimize your Analysis Services database for performance and scalability.
SSRS (SQL Server Reporting Services):
Open SQL Server Data Tools (SSDT) or SQL Server Report Builder to create a new Reporting Services project or report.
Design your report layout by adding data regions (tables, matrices, charts, etc.), text boxes, images, and other elements onto the report canvas.
Define data sources and datasets to retrieve data for your report.
Write or generate SQL queries to populate your datasets with the required data.
Customize the appearance and formatting of your report elements using properties and expressions.
Preview your report to ensure it looks as expected.
Deploy your report to a Report Server instance for access by users.
Schedule report execution, manage subscriptions, and monitor report usage and performance.

Goal of this project

Experience and learning of data analytic DB/Data warehouse applications/tools.

Submission Requirements

- Submission to Blackboard only (do not compress files into one)
- User documentation (PDF, MS Word, or PPT file format only)
- Database or data file

Minimum length: 20 pages or slides

No email submission (include attempt to replace an incorrect file submission) in any case.

No late submission is accepted (submission link will be unavailable after Blackboard deadline).

Note

If your computer is not MS Windows so if you cannot use MS SSIS/SSAS/SSRS, ask your TA for permission to use other MacBook compatible data warehouses tools/applications.