**Data Engineering**

**(CSL234)**

Lab Practical Report



Faculty name : Deepika Garg Student name – Nishant Rajora

Roll No.: 23csu220

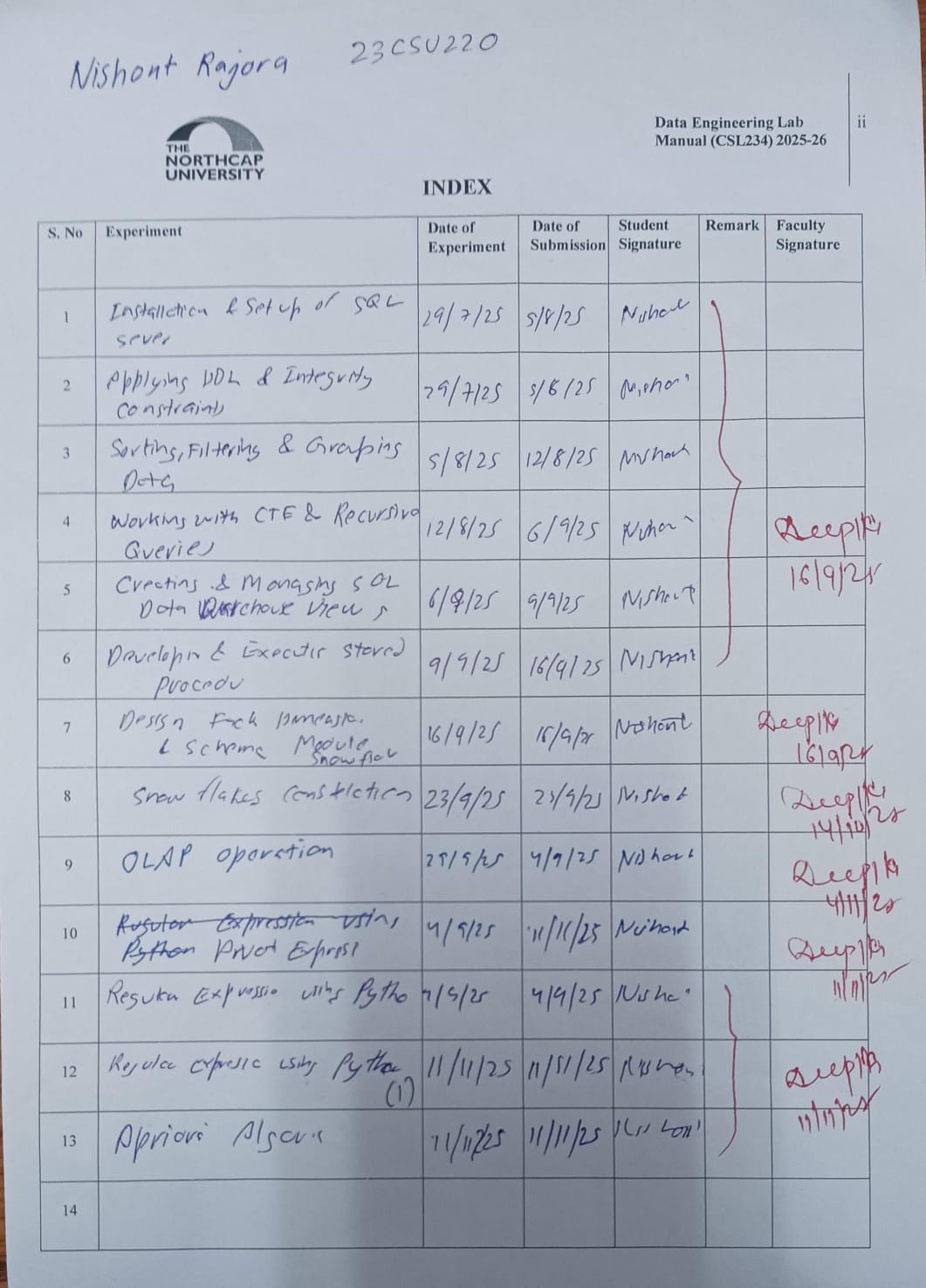
Semester: 5th

Group: DS-A

Department of Computer Science and Engineering

The NorthCap University, Gurugram- 122001, India

Session 2025-26

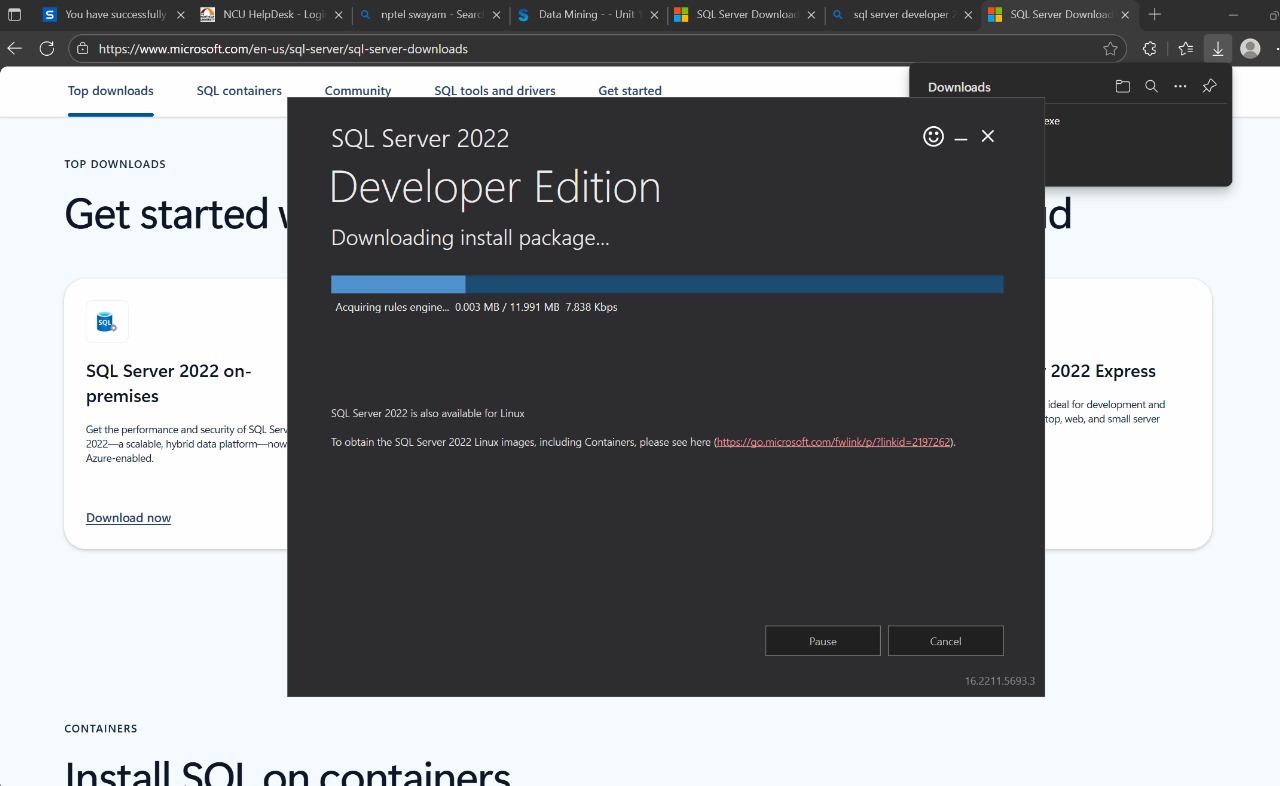
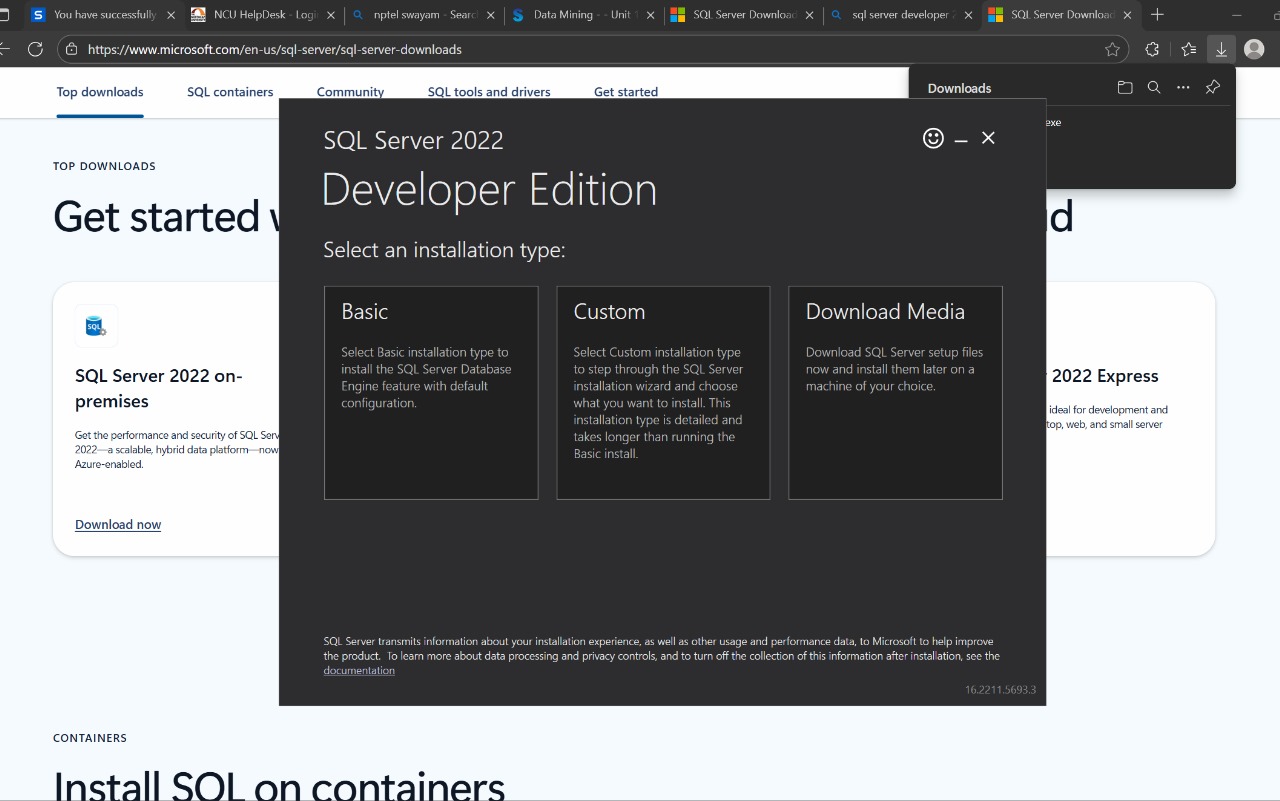


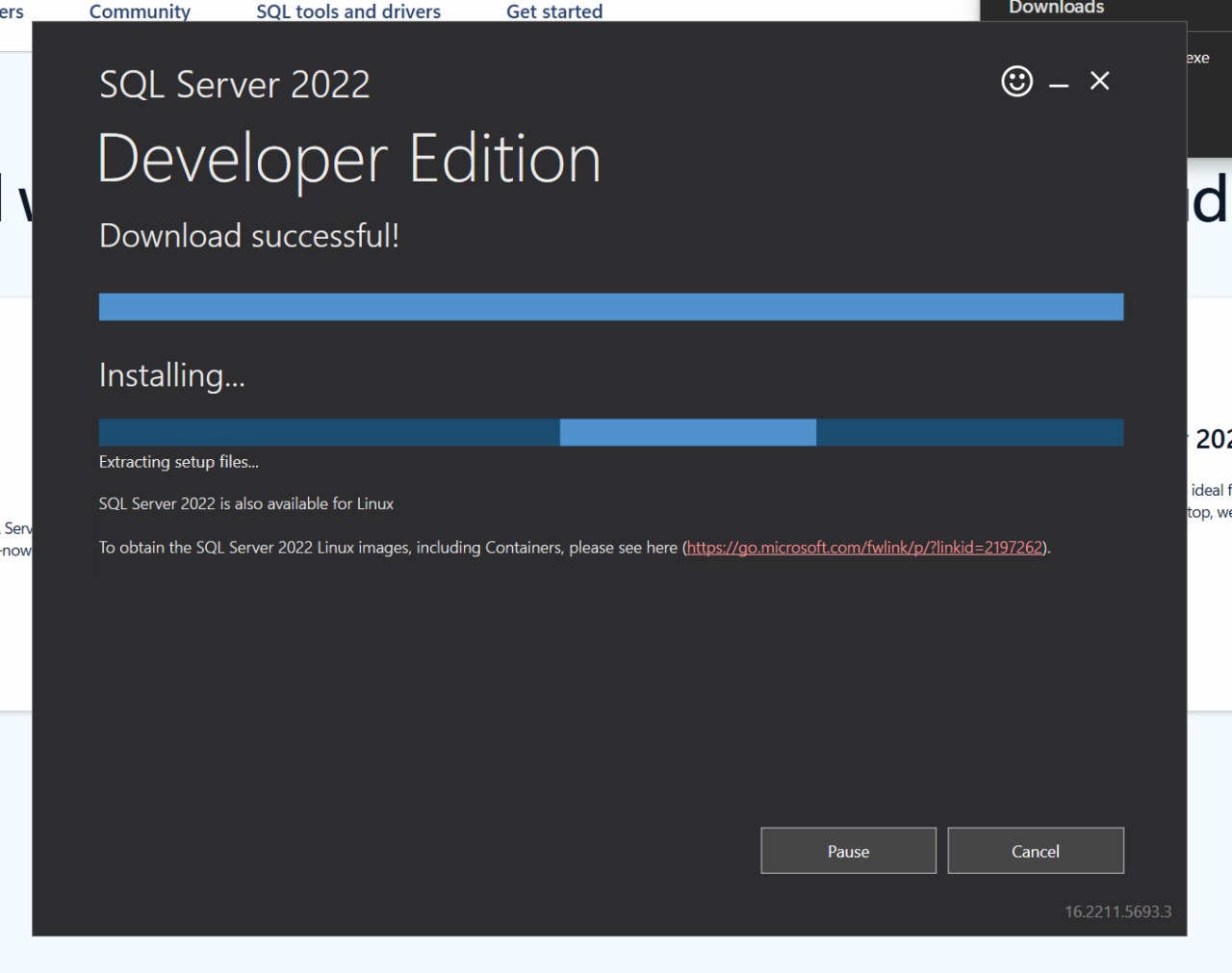
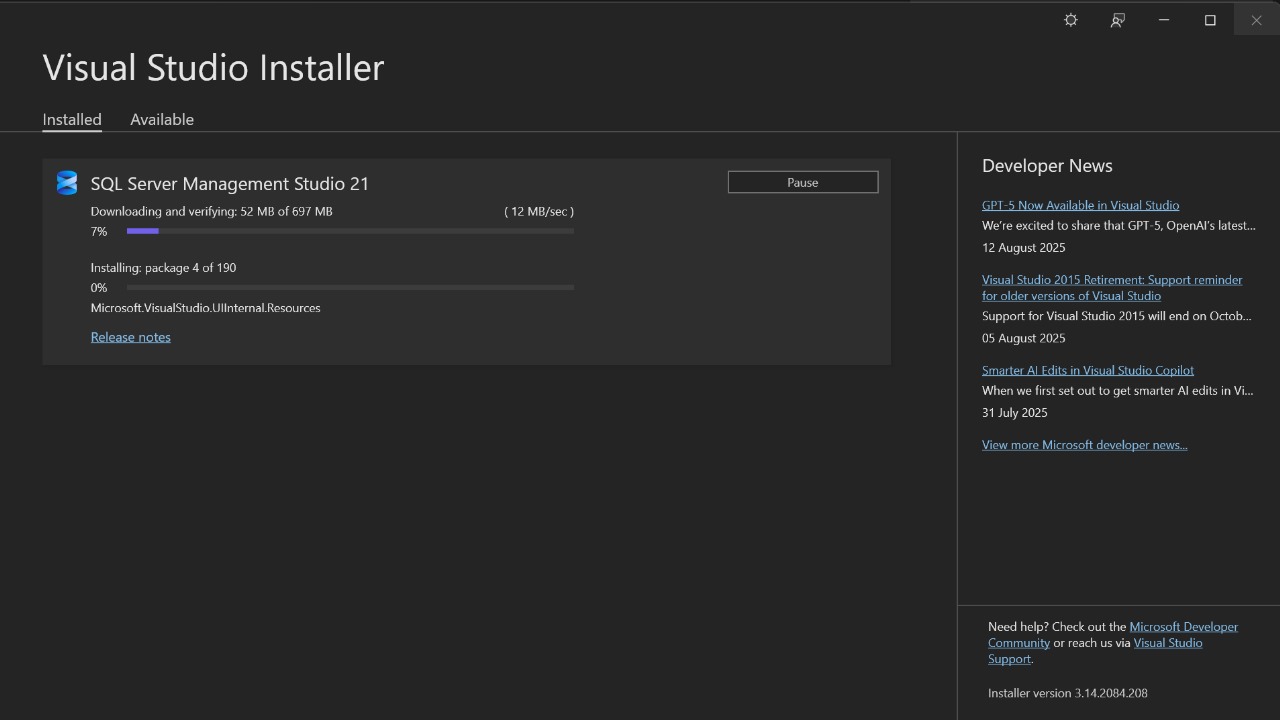
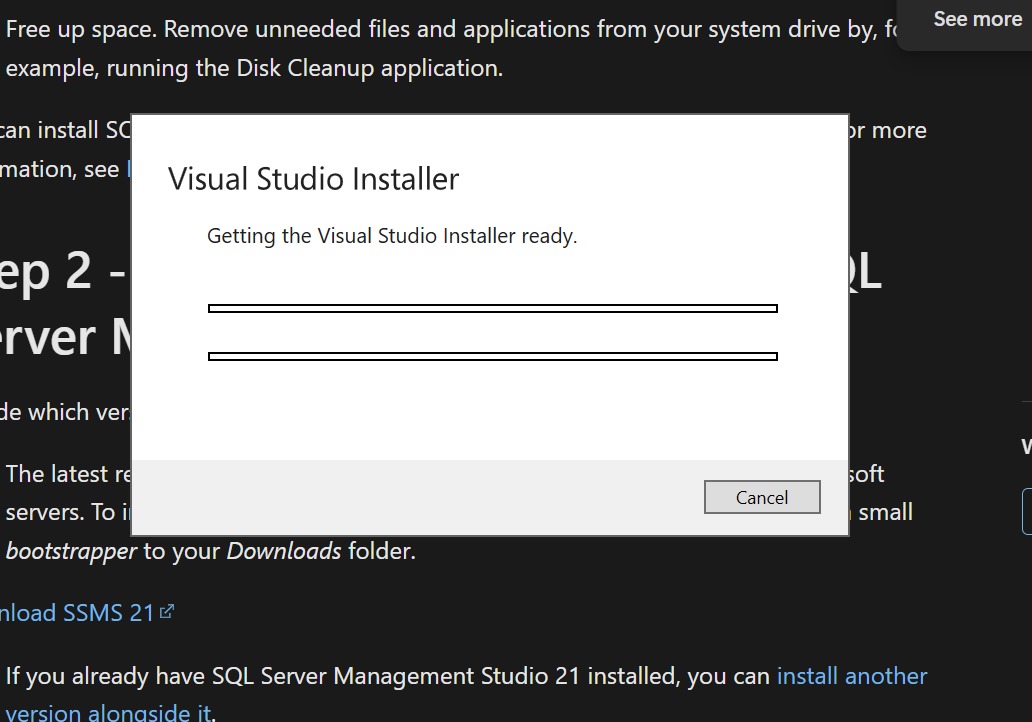
**EXPERIMENT NO. 1**

|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code:** |
| **Date: 05/08/2025** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |
| **Objective:**  **With the objective of making sure that students have access to SQL Server long before they need to work with it for an assignment; Install SQL server with management studio.** |
| **Outcome:**  **Students will understand how to interact with SQL Server interface** |
| **Problem Statement:**  Install the SQL Server 2017 Developer Edition and SQL Server Mangement Studio (SSMS) from the below link and perform the following mentioned tasks  <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>  **Install the SQL Server**  Walk you through the steps of installing SQL Server 2017 Developer Edition on your computer or local server  **Connect to the SQL Server**  Show you how to connect to the SQL Server using SQL Server Management Studio (SSMS).  **Explore SQL Server Database**  Introduce you to an SQL Server sample database called SAMPLE  **Load the SQL Server Sample database**  Guide you on how to load the Sample Database into the SQL Server for practicing. |
| **Background Study:**  SQL Server is a relational database management system, or RDBMS, developed and marketed by Microsoft. Similar to other RDBMS software, SQL Server is built on top of SQL, a standard programming language for interacting with the relational databases. SQL server is tied to Transact-SQL, or T-SQL, the Microsoft’s implementation of SQL that adds a set of proprietary programming constructs.  SQL Server works exclusively on Windows environment for more than 20 years. In 2016, Microsoft made it available on Linux. SQL Server 2017 became generally available in October 2016 that ran on both Windows and Linux.  **SQL Server Services and Tools**  Microsoft provides both data management and business intelligence (BI) tools and services together with SQL Server.  For data management, SQL Server includes SQL Server Integration Services (SSIS), SQL Server Data Quality Services, and SQL Server Master Data Services. To develop databases, SQL Server provides SQL Server Data tools; and to manage, deploy, and monitor databases SQL Server has SQL Server Management Studio (SSMS).  For data analysis, SQL Server offers SQL Server Analysis Services (SSAS). SQL Server Reporting Services (SSRS) provides reports and visualization of data. The Machine Learning Services technology appeared first in SQL Server 2016 which was renamed from the R Services.  **SQL Server Editions**  SQL Server has four primary editions that have different bundled services and tools. Two editions are available free of charge:  SQL Server Developer edition for use in database development and testing.  SQL Server Expression for small databases with the size up to 10 GB of disk storage capacity.  For larger and more critical applications, SQL Server offers the Enterprise edition that includes all SQL server’s features.  SQL Server Standard Edition has partial feature sets of the Enterprise Edition and limits on the Server regarding the numbers of processor core and memory that can be configured. |
| **Question Bank:** |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**



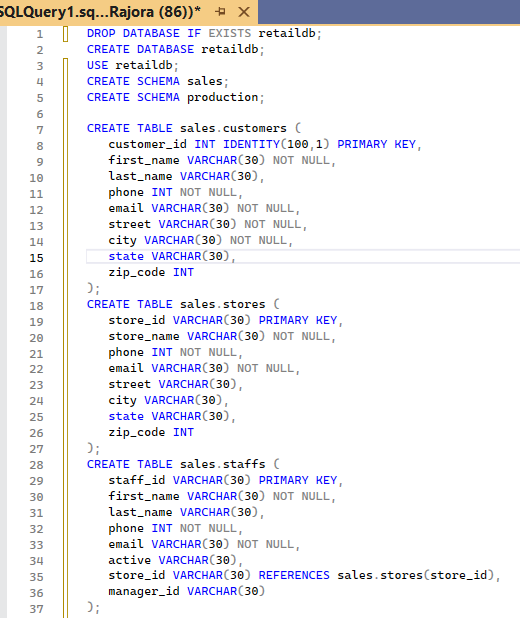
**EXPERIMENT NO. 2**

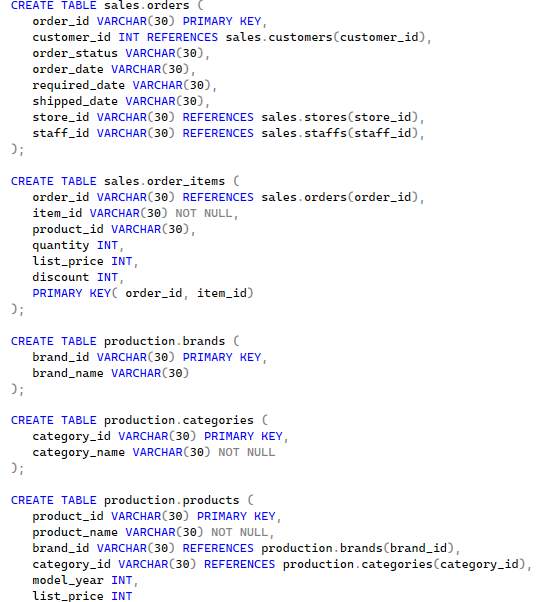
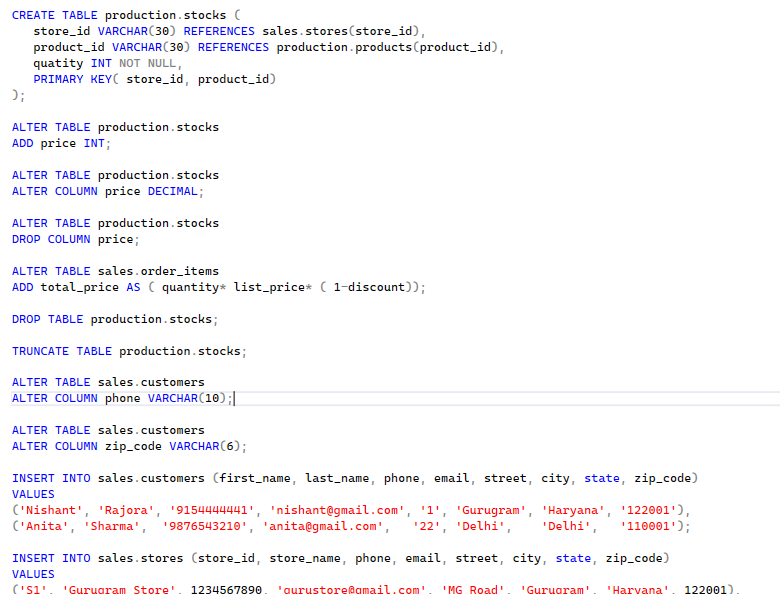
|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code:** |
| **Date: 03/08/2025** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

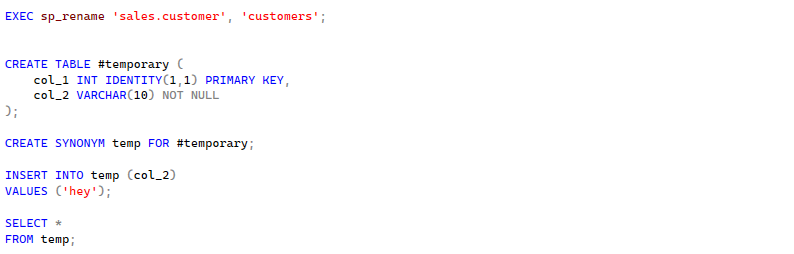
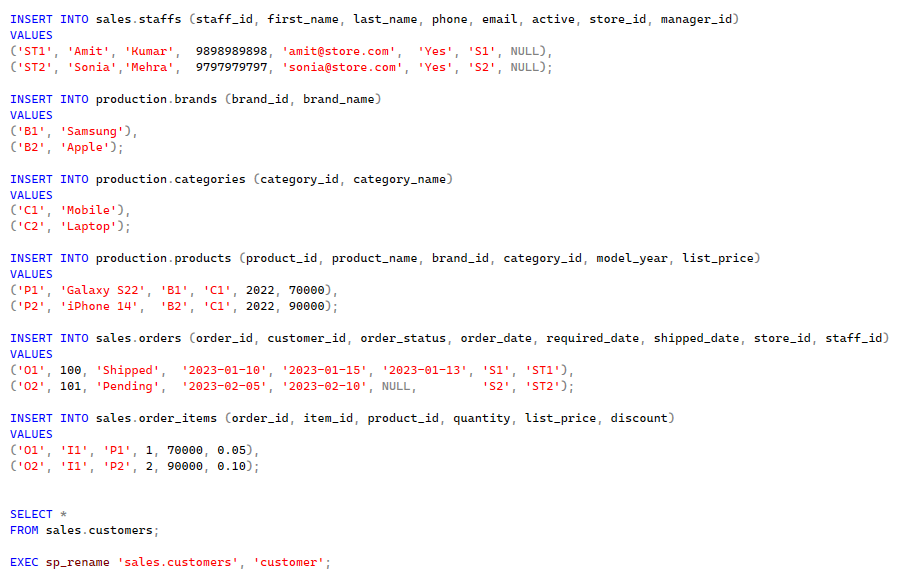
|  |
| --- |
| **Objective:**  To apply SQL integrity constraints as per the DDL statements for SAMPLE database.  The following illustrates the SAMPLE database diagram:  https://www.sqlservertutorial.net/wp-content/uploads/SQL-Server-Sample-Database.png |
| **Outcome:**  The students will understand how a database is created followed by insertion of relevant data.  The students will understand the need of applying various types of integrity constraints such as primary key, foreign key, unique key, NOT NULL, default and CHECK etc |
| **Problem Statement:**  **Perform the basic DDL queries for the sample database.** |
| **Background Study:**  **Background Study**   1. Primary Key Constraint: A column or group of columns in a table which helps us to uniquely identifies every row in that table is called a primary key. This DBMS can't be a duplicate. The same value can't appear more than once in the table.   **Syntax to define a Primary key at column level:**  *column name datatype [CONSTRAINT constraint\_name] PRIMARY KEY*  **Syntax to define a Primary key at table level:**  *[CONSTRAINT constraint\_name] PRIMARY KEY (column\_name1,column\_name2,..)*  Rules for defining Primary key:   * Two rows can't have the same primary key value * It must for every row to have a primary key value. * The primary key field cannot be null. * The value in a primary key column can never be modified or updated if any foreign key refers to that primary key.  1. Foreign Key (Referential integrity constraint): This constraint identifies any column referencing the PRIMARY KEY in another table. It establishes a relationship between two columns in the same table or between different tables. For a column to be defined as a Foreign Key, it should be a defined as a Primary Key in the table which it is referring. One or more columns can be defined as Foreign key.   **Syntax to define a Foreign key at column level:**  *[CONSTRAINT constraint\_name] REFERENCES Referenced\_Table\_name(column\_name)*  **Syntax to define a Foreign key at table level:**  *[CONSTRAINT constraint\_name] FOREIGN KEY(column\_name) REFERENCES referenced\_table\_name(column\_name);*   1. SQL Not Null Constraint : This constraint ensures all rows in the table contain a definite value for the column which is specified as not null. Which means a null value is not allowed.   Syntax to define a Not Null constraint:  *[CONSTRAINT constraint name] NOT NULL* SQL Unique Key: This constraint ensures that a column or a group of columns in each row have a distinct value. A column(s) can have a null value but the values cannot be duplicated. **Syntax to define a Unique key at column level:**  *[CONSTRAINT constraint\_name] UNIQUE*  **Syntax to define a Unique key at table level:**  *[CONSTRAINT constraint\_name] UNIQUE(column\_name)* SQL Check Constraint : This constraint defines a business rule on a column. All the rows must satisfy this rule. The constraint can be applied for a single column or a group of columns. **Syntax to define a Check constraint:**  *[CONSTRAINT constraint\_name] CHECK (condition)* |
| **Question Bank: Perform the following basic DDL queries**   * Create a SAMPLE data base * CREATE SCHEMA : to create the new schema in the databse * CREATE TABLE : a new table to a specific schema of the database * Identity column– how to use the IDENTITY property to create the identity column for a table. * ALTER TABLE ADD column – show you how to add one or more columns to an existing table * ALTER TABLE ALTER COLUMN – show you how to change the definition of existing columns in a table. * ALTER TABLE DROP COLUMN – learn how to drop one or more columns from a table. * Computed columns – how to use the computed columns to resue the calculation logic in multiple queries. * DROP TABLE – show you how to delete tables from the database. * TRUNCATE TABLE – delete all data from a table faster and more efficiently. * SELECT INTO – learn how to create a table and insert data from a query into it. * Rename a table –  walk you through the process of renaming a table to a new one. * Temporary tables – introduce you to the temporary tables for storing temporarily immediate data in stored procedures or database session. * Synonym – explain you the synonym and show you how to create synonyms for database objects. |

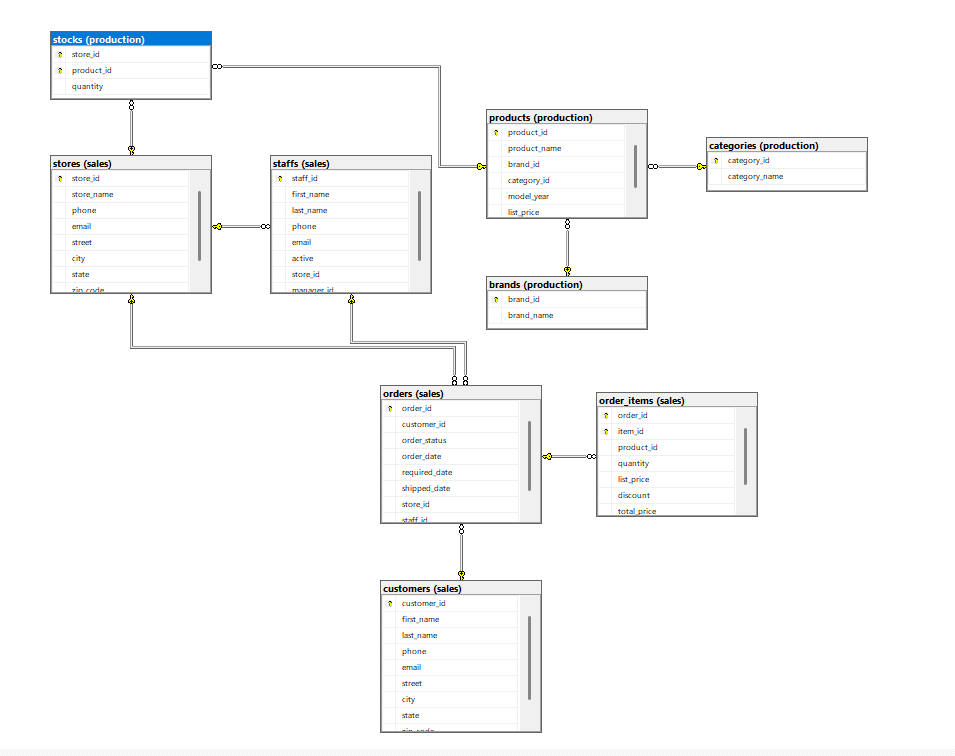
**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

****



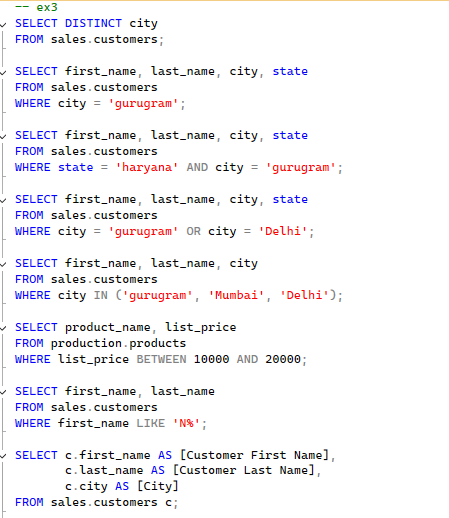


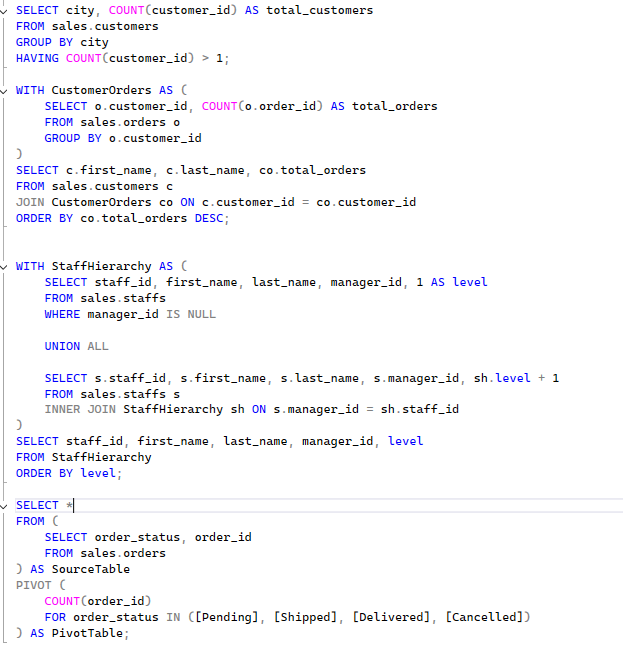
**EXPERIMENT NO. 3**

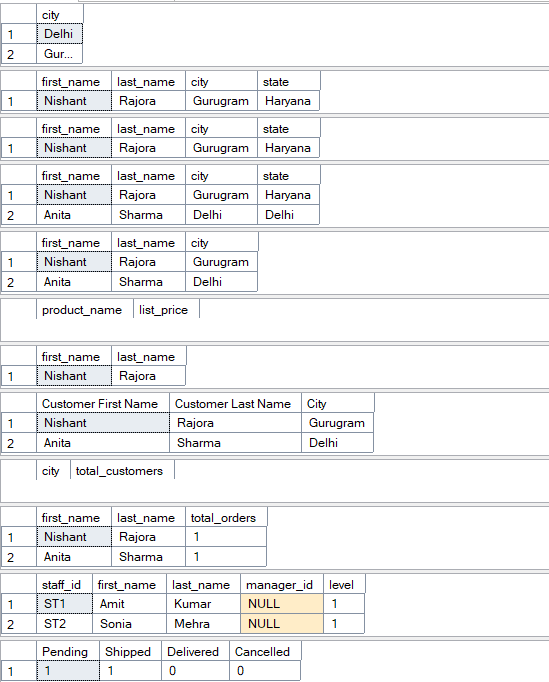
|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code:** |
| **Date: 12/08/2025** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

|  |
| --- |
| **Objective:**  To Learn how to query the data from SQL Server database . |
| **Outcome:**  **Students will understand how to sort, filter, modify and group the data**  **Students will learn how to apply CTE (Common Table Expression) and recursive CTE** |
| **Problem Statement:**  **Perform the following queries for the sample database.**   * Select distinct values in one or more columns of a table. * Filter rows in the output of a query based on one or more conditions. * Combine two Boolean expressions and return true if all expressions are true. * Combine two Boolean expressions and return true if either of conditions is true. * Check whether a value matches any value in a list or a subquery. * Test if a value is between a range of values. * Check if a character string matches a specified pattern. * Show you how to use column aliases to change the heading of the query output and table alias to improve the readability of a query. * Use common table expressions to make complex queries more readable. * Query hierarchical data using recursive CTE. * Convert rows to columns |
| **Background Study:**  Students should be aware how to change the contents of tables in the database. The SQL commands for modifying data such as insert, delete, and update are referred to as data manipulation language (DML).   * INSERT – insert a row into a table * INSERT multiple rows – insert multiple rows into a table using a single INSERT statement * INSERT INTO SELECT – insert data into a table from the result of a query. * UPDATE – change the existing values in a table. * UPDATE JOIN– update values in a table based on values from another table using JOIN clauses. * DELETE– delete one or more rows of a table. * MERGE– walk you through the steps of performing a mixture of insertion, update, and deletion using a single statement. |
| **Question Bank:** |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**





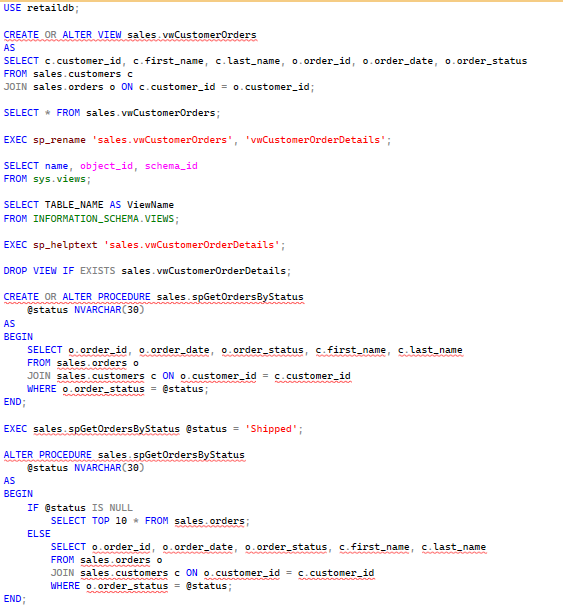
**EXPERIMENT NO. 4**

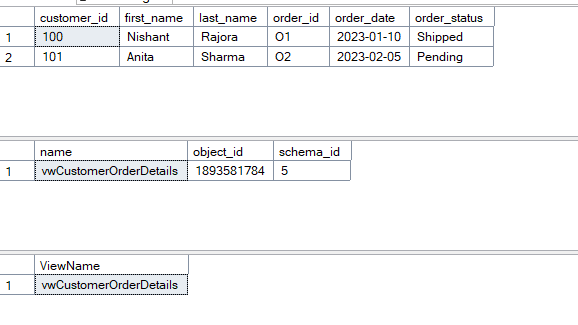
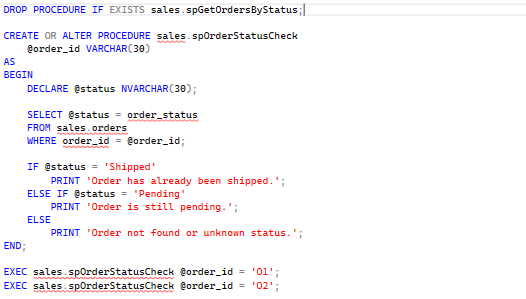
|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code:** |
| **Date: 12/08/2025** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

|  |
| --- |
| **Objective:**  To learn about views and how to manage views  To introduces the stored procedures |
| **Outcome:**  Students will learn to create a new view, removing a view, and updating data of the underlying tables through a view.  **Students will learn how** to develop flexible stored procedures to optimize database access. |
| **Problem Statement:**  **Perform the following queries to Manage views in SQL Server**   * Creating a new view– show you how to create a new view in a SQL Server database. * Renaming a view – learn how to rename a view using the SQL Server Management Studio (SSMS) or Transact-SQL command. * Listing views in SQL Server – discuss the various ways to list all views in a SQL Server Database. * Getting view information – how to get information about a view. * Removing a view – guide you how to use the DROP VIEW statement to remove one or more views from the database. * Create, modify and delete a stored procedure * Make use of control-of-flow statements in stored procedure |
| **Background Study:**  By definition, views do not store data except for indexed views.  A view may consist of columns from multiple tables using joins or just a subset of columns of a single table. This makes views useful for abstracting or hiding complex queries.  The following picture illustrates a view that includes columns from multiple tables:  SQL Server Views Advantages of views Generally speaking, views provide the following advantages: Security You can restrict users to access directly to a table and allow them to access a subset of data via views.  For example, you can allow users to access customer name, phone, email via a view but restrict them to access the bank account and other sensitive information. Simplicity A relational database may have many tables with complex relationships e.g., one-to-one and one-to-many that make it difficult to navigate.  However, you can simplify the complex queries with joins and conditions using a set of views. Consistency Sometimes, you need to write a complex formula or logic in every query.  To make it consistent, you can hide the complex queries logic and calculations in views.  Once views are defined, you can reference the logic from the views rather than rewriting it in separate queries  **Stored Procedure**  SQL Server stored procedures are used to group one or more Transact-SQL statements into logical units. The stored procedure is stored as a named object in the SQL Server Database Server.  When you call a stored procedure for the first time, SQL Server creates an execution plan and stores it in the cache. In the subsequent executions of the stored procedure, SQL Server reuses the plan to execute the stored procedure very fast with reliable performance.  This experiment introduces you to the stored procedures and shows you how to develop flexible stored procedures to optimize database access. |
|  |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**



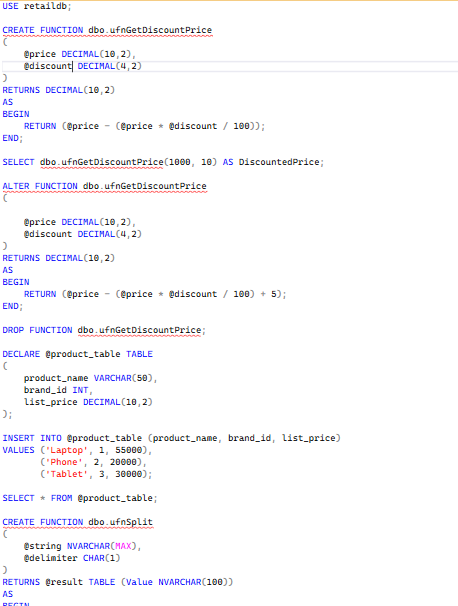


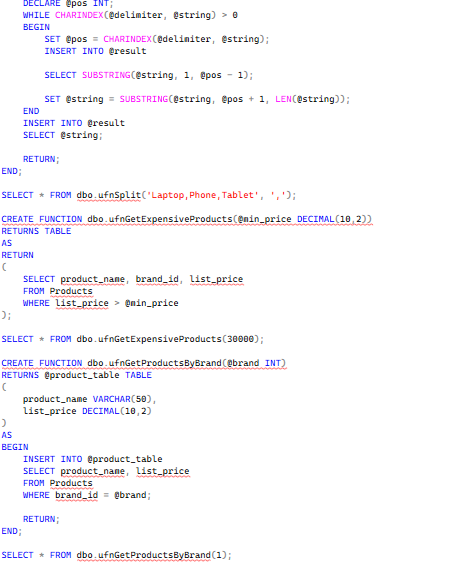
**EXPERIMENT NO. 5**

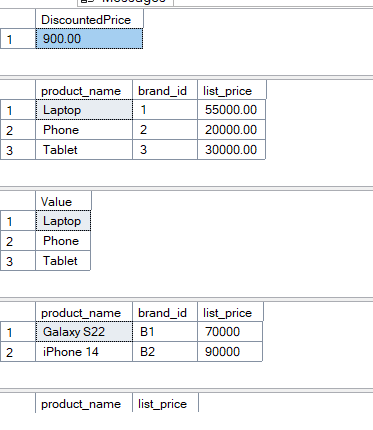
|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code:** |
| **Date: 06/09/2025** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

|  |
| --- |
| **Objective:**  To learn about views and how to manage views  To introduces the stored procedures |
| **Outcome:**  Students will learn to create a new view, removing a view, and updating data of the underlying tables through a view.  **Students will learn how** to develop flexible stored procedures to optimize database access. |
| **Problem Statement:**  **Perform the following queries using user defined function in SQL Server**   * Create, call, modify and remove a scalar function. * Declares a table variable named @product\_table which consists of three columns: product\_name, brand\_id, and list\_price * Insert data into table variable * Querying data from table variables * Create a user-defined function named ufnSplit() that returns a table variable. * Create table valued function and multi-statement table valued function. |
| **Background Study:**  SQL Server user-defined functions include scalar-valued functions which return a single value and table-valued function which return rows of data.  The SQL Server user-defined functions help you simplify your development by encapsulating complex business logic and make them available for reuse in every query.   * User defined scalar function – cover the user-defined scalar functions that allow you to encapsulate complex formula or business logic and reuse them in every query. * Table variables – learn how to use table variables as a return value of user-defined functions. * Table valued functions – introduce you to inline table-valued function and multi-statement table-valued function to develop user-defined functions that return data of table types. * Removing user defined functions – learn how to drop one or more existing user-defined functions from the database. |
| **Question Bank:** |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs  
**

****

****

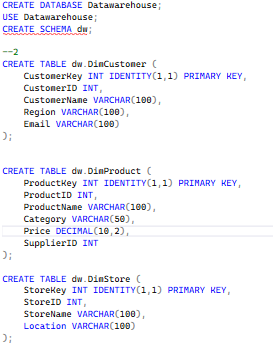
**EXPERIMENT NO. 6**

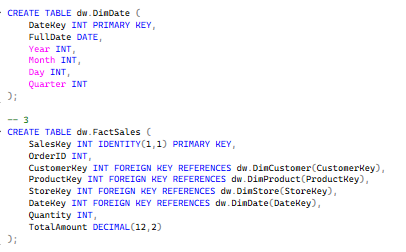
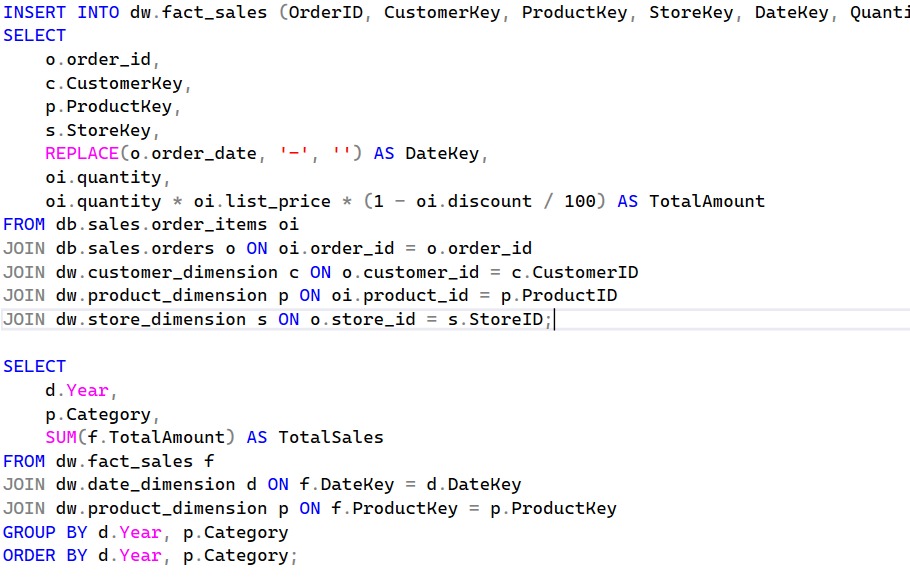
|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code:** |
| **Date: 09/09/2025** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Students will learn to identify facts and dimensions for the given scenario of data warehouse |
| **Outcome:**  Students will learn to create a fact and dimension tables for a sample datawarehouse |
| **Problem Statement:**  **Consider the scenario of sample Data Warehouse:**  Our Data Warehouse is “XBOX Game Pass”. It is similar to Netflix but it is a subscription for video games. This platform allows the user to play video games without purchasing it separately and does not need a powerful system to do so. Just a reliable internet connection is required  To increase our maximum revenue and to ensure the user has the best experience possible. We came up with these queries.  **Query is :**   * How to increase the number of subscriptions and to generate maximum revenue ?   Identify facts and dimensions for the above scenario. |
| **Background Study:**   * A data warehouse is based on a multidimensional data model which views data in the form of a data cube * A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions   + Dimension tables, such as item (item\_name, brand, type), or time(day, week, month, quarter, year)   + Fact table contains measures (such as dollars\_sold) and keys to each of the related dimension tables * In data warehousing literature, an n-D base cube is called a base cuboid. The top most 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid. The lattice of cuboids forms a data cube. |
| **Question Bank:** |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

****

****

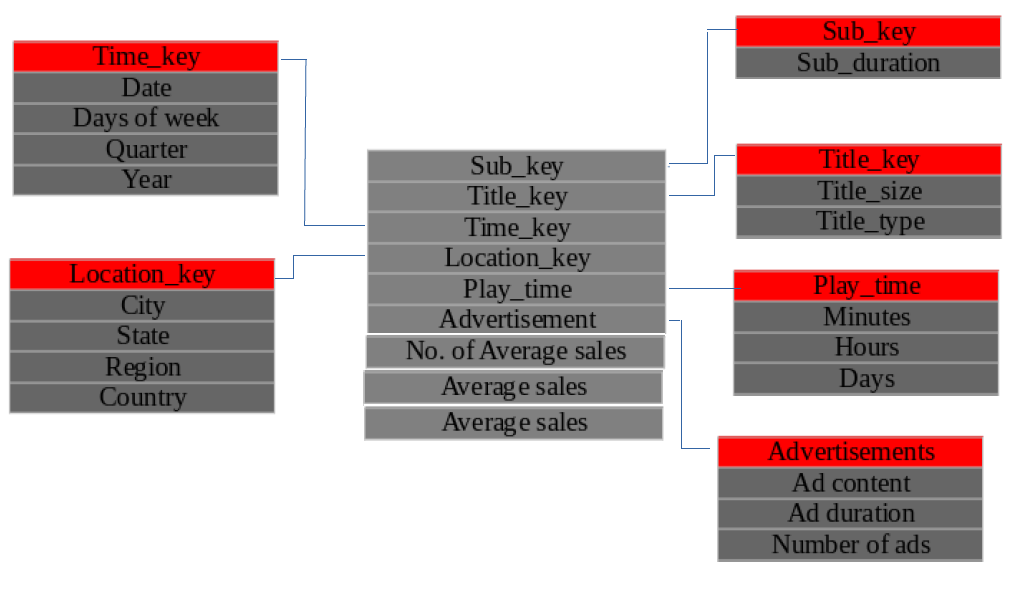
**EXPERIMENT NO. 7**

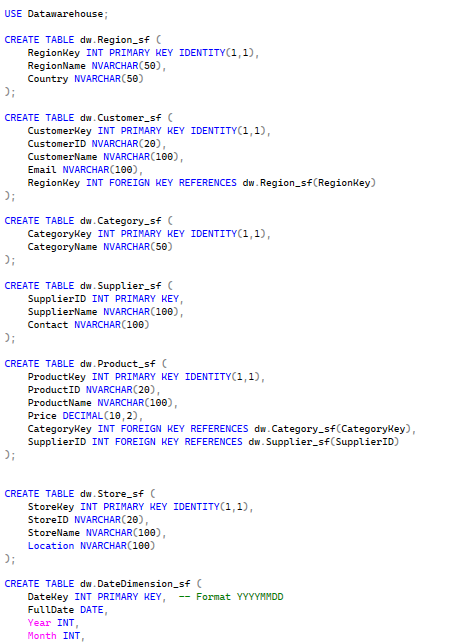
|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code:** |
| **Date: 16/09/2025** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

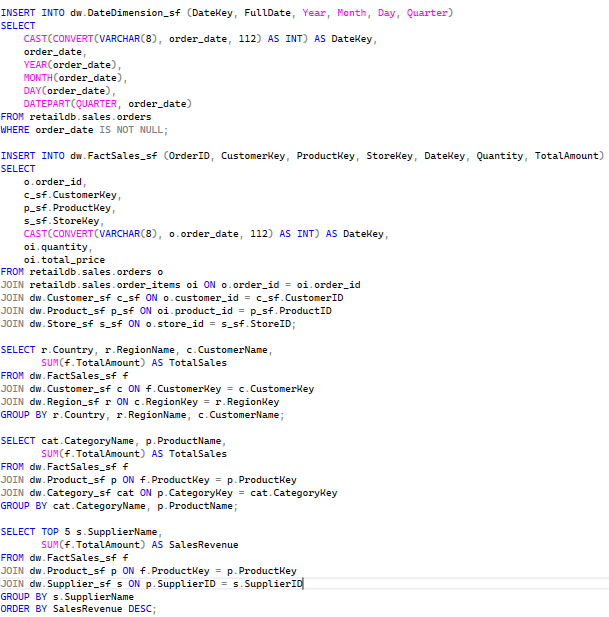
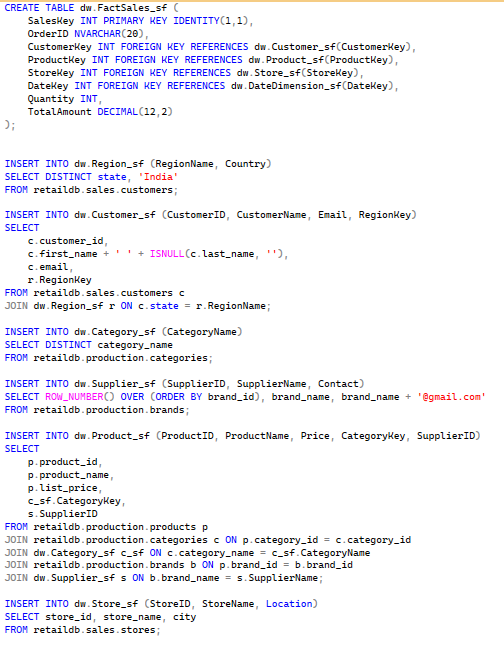
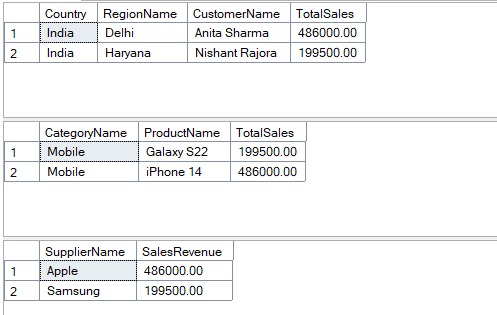
|  |
| --- |
| **Objective:**  Students will learn to create a schema for their datawarehouse |
| **Outcome:**  Students will learn to create star, snowflake and fact constellation schemas as per their problem statement and draw the database diagrams in SQL server. |
| **Problem Statement:**  **Consider the scenario of sample Data Warehouse:**  Our Data Warehouse is “XBOX Game Pass”. It is similar to Netflix but it is a subscription for video games. This platform allows the user to play video games without purchasing it separately and does not need a powerful system to do so. Just a reliable internet connection is required  To increase our maximum revenue and to ensure the user has the best experience possible. We came up with these queries.  **Query is :**   * How to increase the number of subscriptions and to generate maximum revenue ?   **Draw the data base diagram in SQL server** |
| **Background Study:**  Modeling data warehouses: dimensions & measures   * + Star schema: A fact table in the middle connected to a set of dimension tables   + Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake   + Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation |
| **Question Bank:** |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**



****

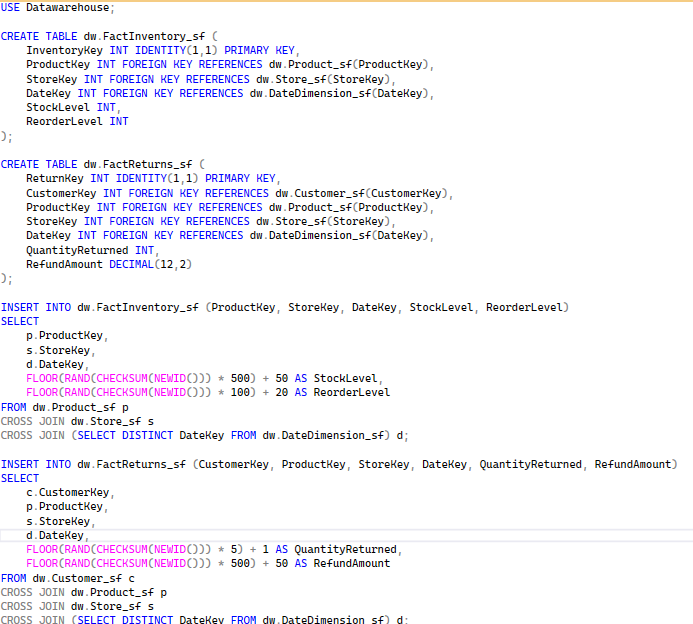
****

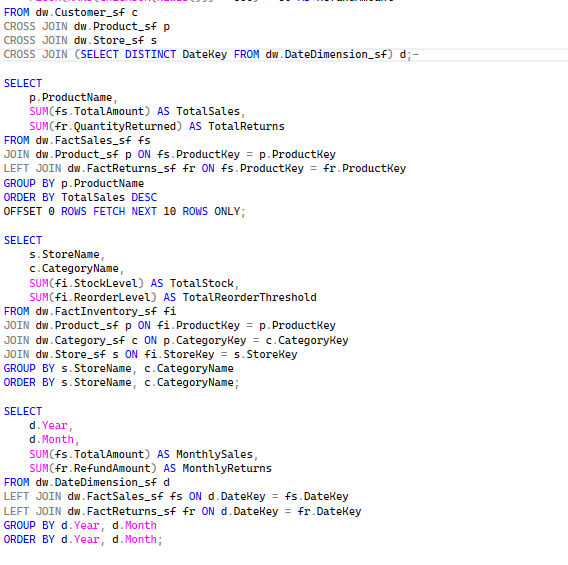
**EXPERIMENT NO. 8**

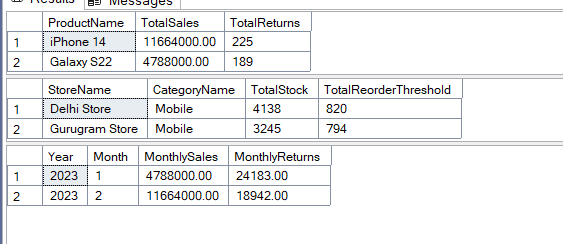
|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code:** |
| **Date: 23/09/2025** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Students will learn to create a schema for their datawarehouse |
| **Outcome:**  Students will learn to create star, snowflake and fact constellation schemas as per their problem statement and draw the database diagrams in SQL server. |
| **Problem Statement:**  **Consider the scenario of sample Data Warehouse:**  Our Data Warehouse is “XBOX Game Pass”. It is similar to Netflix but it is a subscription for video games. This platform allows the user to play video games without purchasing it separately and does not need a powerful system to do so. Just a reliable internet connection is required  To increase our maximum revenue and to ensure the user has the best experience possible. We came up with these queries.  **Query is :**   * How to increase the number of subscriptions and to generate maximum revenue ?   **Draw the data base diagram in SQL server** |
| **Background Study:**  Modeling data warehouses: dimensions & measures   * + Star schema: A fact table in the middle connected to a set of dimension tables   + Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake   + Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation |
| **Question Bank:** |

**Student Work Area**

****

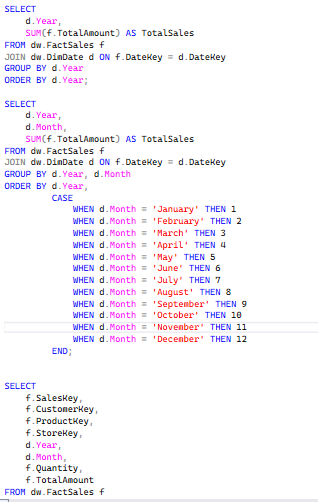
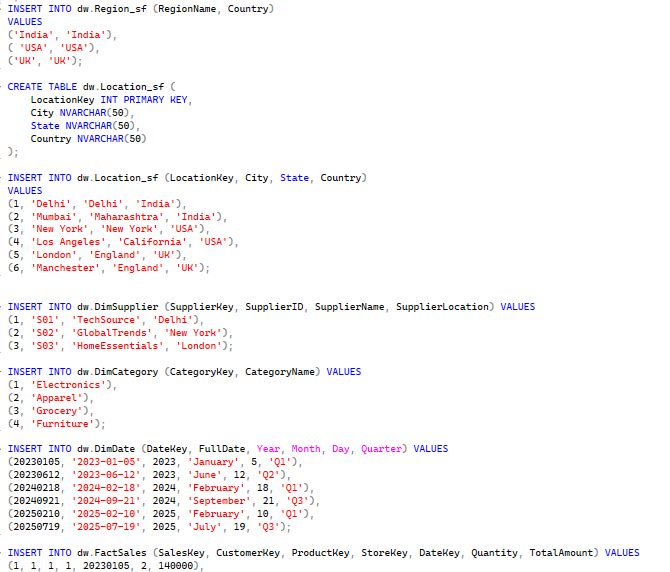
****

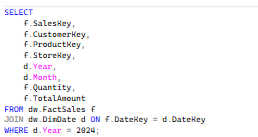
****

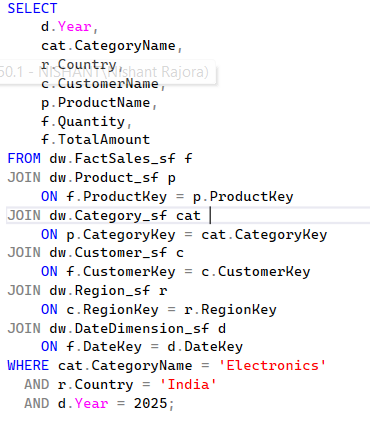
**EXPERIMENT NO. 9**

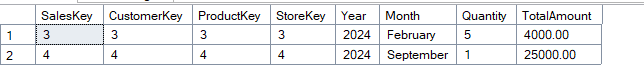
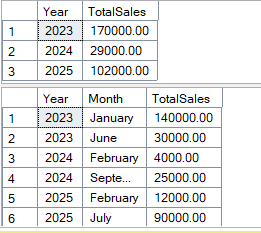
|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code:** |
| **Date: 29/09/2025** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

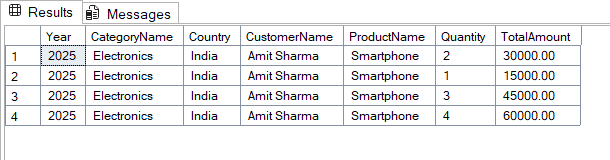
|  |
| --- |
| **Objective:**  Students will learn to create a schema for their datawarehouse |
| **Outcome:**  Students will learn to create star, snowflake and fact constellation schemas as per their problem statement and draw the database diagrams in SQL server. |
| **Problem Statement:**  **Consider the scenario of sample Data Warehouse:**  Our Data Warehouse is “XBOX Game Pass”. It is similar to Netflix but it is a subscription for video games. This platform allows the user to play video games without purchasing it separately and does not need a powerful system to do so. Just a reliable internet connection is required  To increase our maximum revenue and to ensure the user has the best experience possible. We came up with these queries.  **Query is :**   * How to increase the number of subscriptions and to generate maximum revenue ?   **Draw the data base diagram in SQL server** |
| **Background Study:**  Modeling data warehouses: dimensions & measures   * + Star schema: A fact table in the middle connected to a set of dimension tables   + Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake   + Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation |
|  |

****

****

****

****

****

**EXPERIMENT NO. 10**

|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code: 29/09/2025** |
| **Date:** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Design, Create and Process cube by identifying measures and dimensions for Star Schema, for an assigned system by replacing a dimension in the grid, filtering and drilldown using cube browser. |
| **Outcome:**  Students will learn to import data from source, create the corresponding view and create cube. |
| **Problem Statement:**  Implement Pivot operations on data cube |
| **Background Study:**  Star schema resembles a star, one or the dimension tables surround more fact tables. Dimension tables are not normalized - that means even if you have repeating fields such as name or category no extra table is added to remove the redundancy. Star schema also contains the entire dimension hierarchy within a single table. Dimension hierarchy provides a way of aggregating data from the lowest to highest levels within a dimension.  Snowflake schema resembles a snowflake because dimension tables are further normalized or have parent tables. For example, we could extend the product dimension in the dealership warehouse to have a product\_category and product\_subcategory tables. Product categories could include trucks, vans, sport utility vehicles, etc. Product subcategory tables could contain subcategories such as leisure vehicles, recreational vehicles, luxury vehicles, industrial trucks and so forth.  **OLAP Operations**  Since OLAP servers are based on multidimensional view of data, we will discuss OLAP operations in multidimensional data.  **Roll-up**  Roll-up performs aggregation on a data cube in any of the following ways:   * By climbing up a concept hierarchy for a dimension * By dimension reduction * Initially the concept hierarchy was "street < city < province < country". * On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country. * The data is grouped into cities rather than countries. * When roll-up is performed, one or more dimensions from the data cube are removed.   **Drill-down**  Drill-down is the reverse operation of roll-up. It is performed by either of the following ways:   * By stepping down a concept hierarchy for a dimension. * By introducing a new dimension. * Initially the concept hierarchy was "day < month < quarter < year." * On drilling down, the time dimension is descended from the level of quarter to the level of month. * When drill-down is performed, one or more dimensions from the data cube are added. * It navigates the data from less detailed data to highly detailed data.   **Slice**  The slice operation selects one particular dimension from a given cube and provides a new sub-cube.   * It will form a new sub-cube by selecting one or more dimensions.   **Dice**  Dice selects two or more dimensions from a given cube and provides a new sub-cube. The dice operation on the cube based on the following selection criteria involves three dimensions.  **Pivot**  The pivot operation also known as rotation. It rotates the data axes in view in order to provide an alternative presentation of data. |
| Question Bank: |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

The following are the steps for creating cube: 1). Select the analysis services

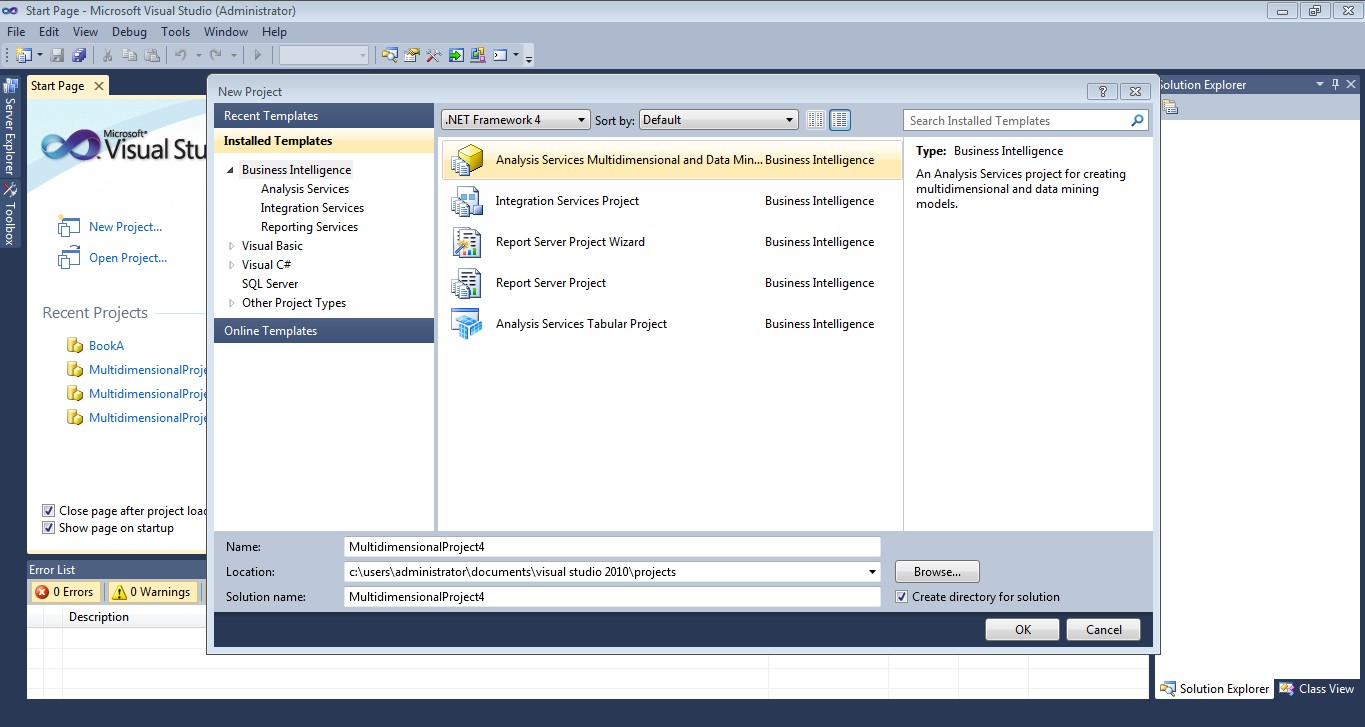
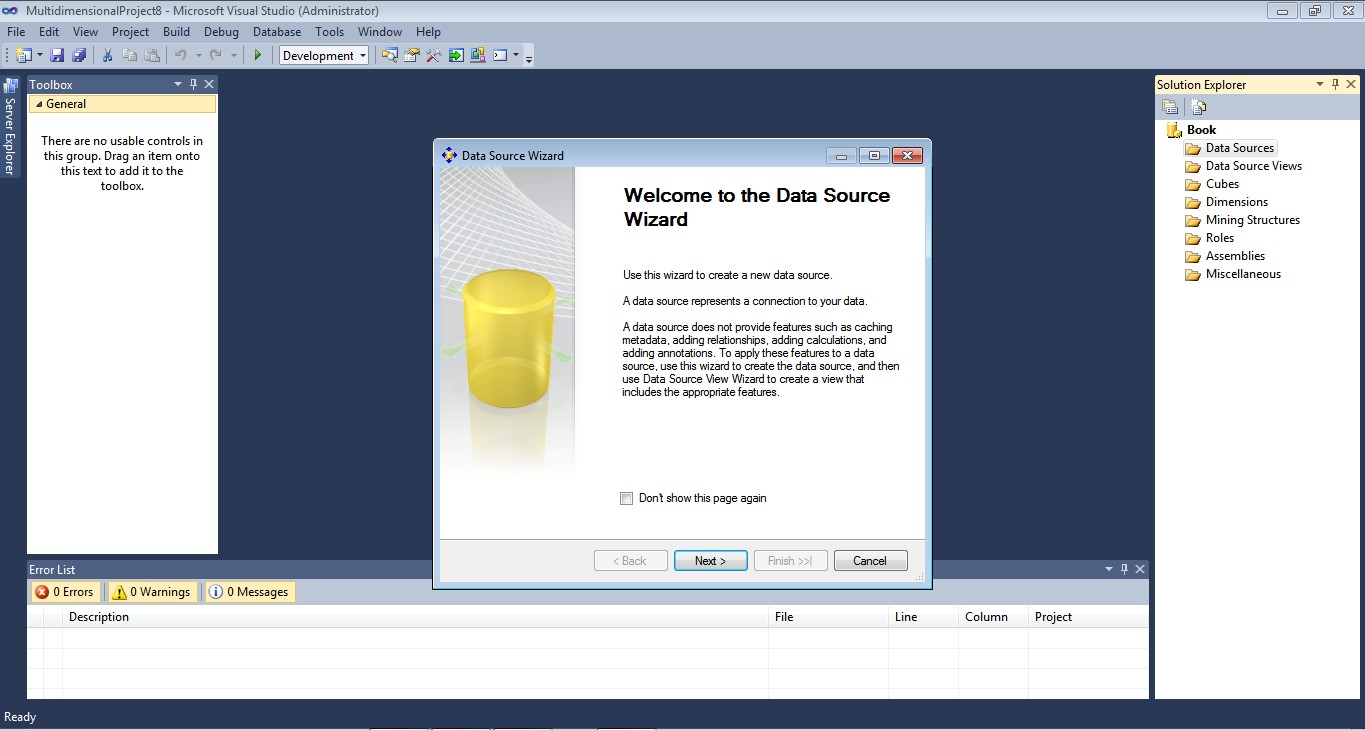


Figure 1: Popup to select type of project

1. Create data source.

Figure 2: Popup to create new Data Source

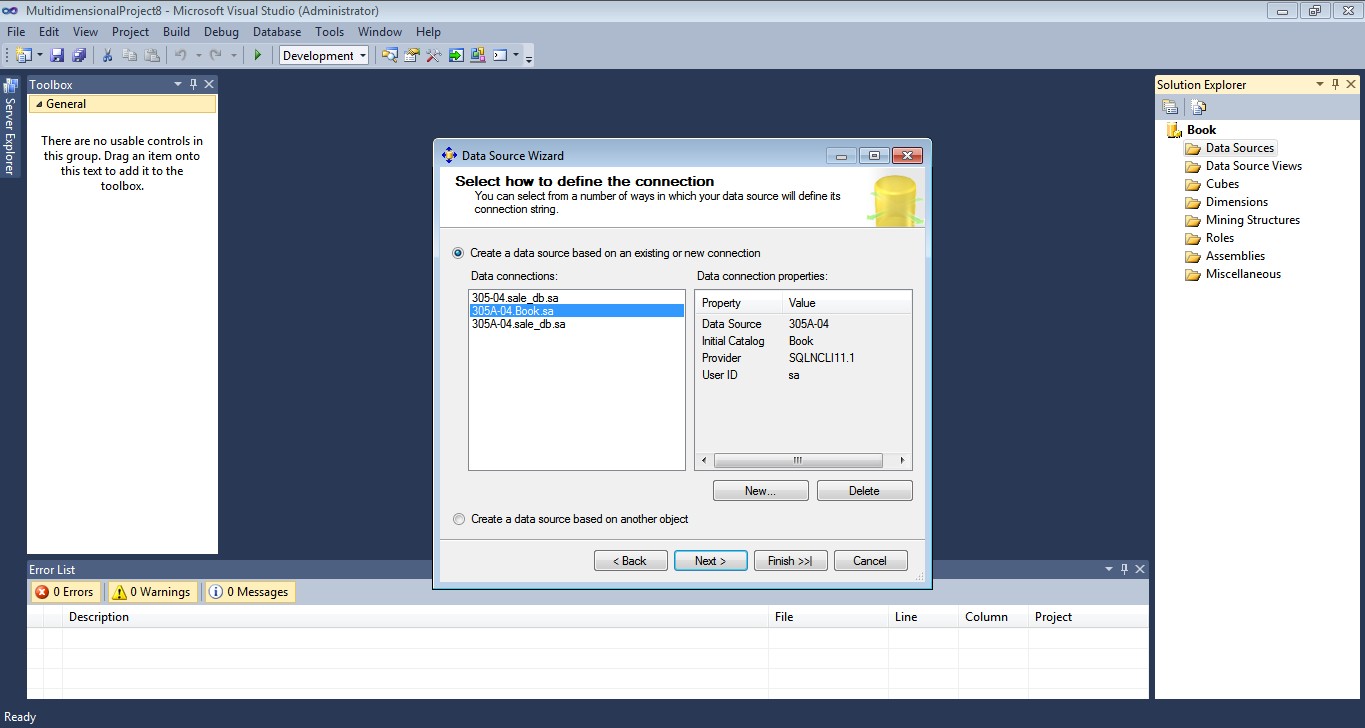


Figure 3: Popup to establish a connection with database

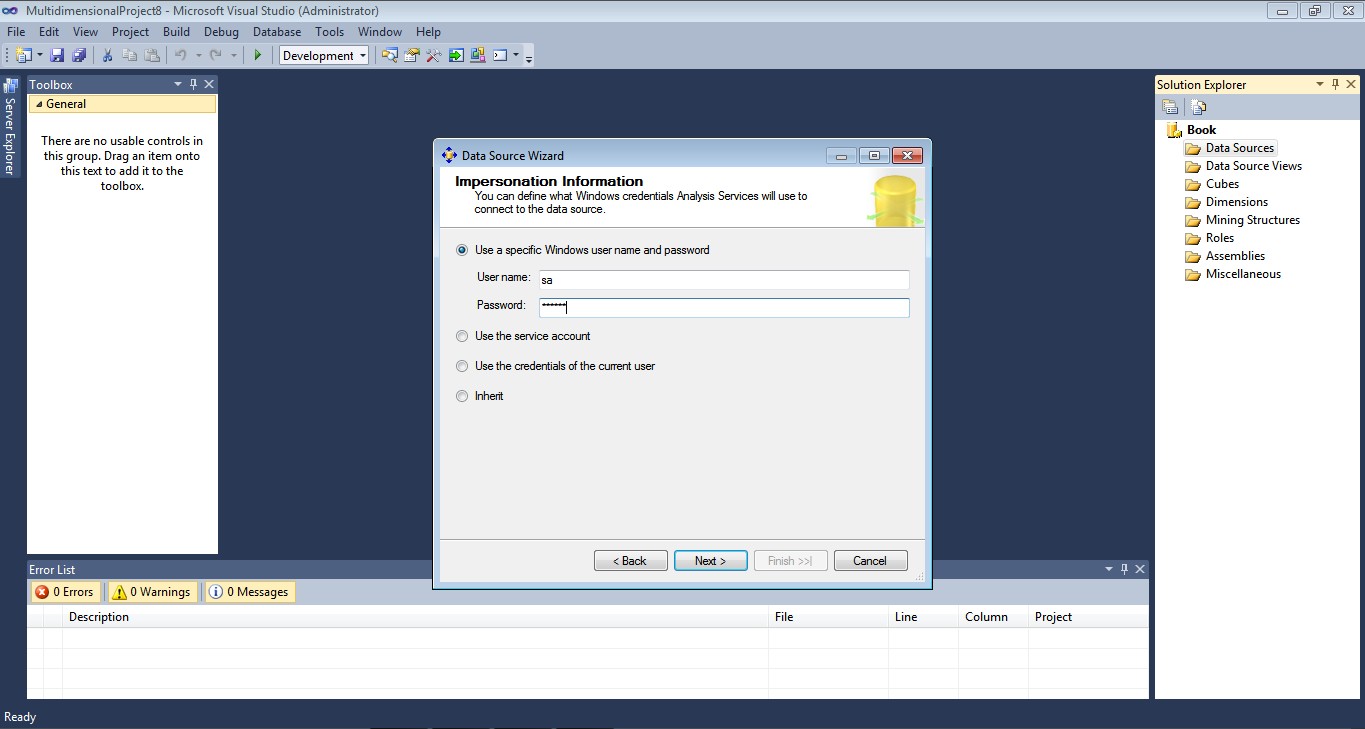


Figure 4: Select authentication method

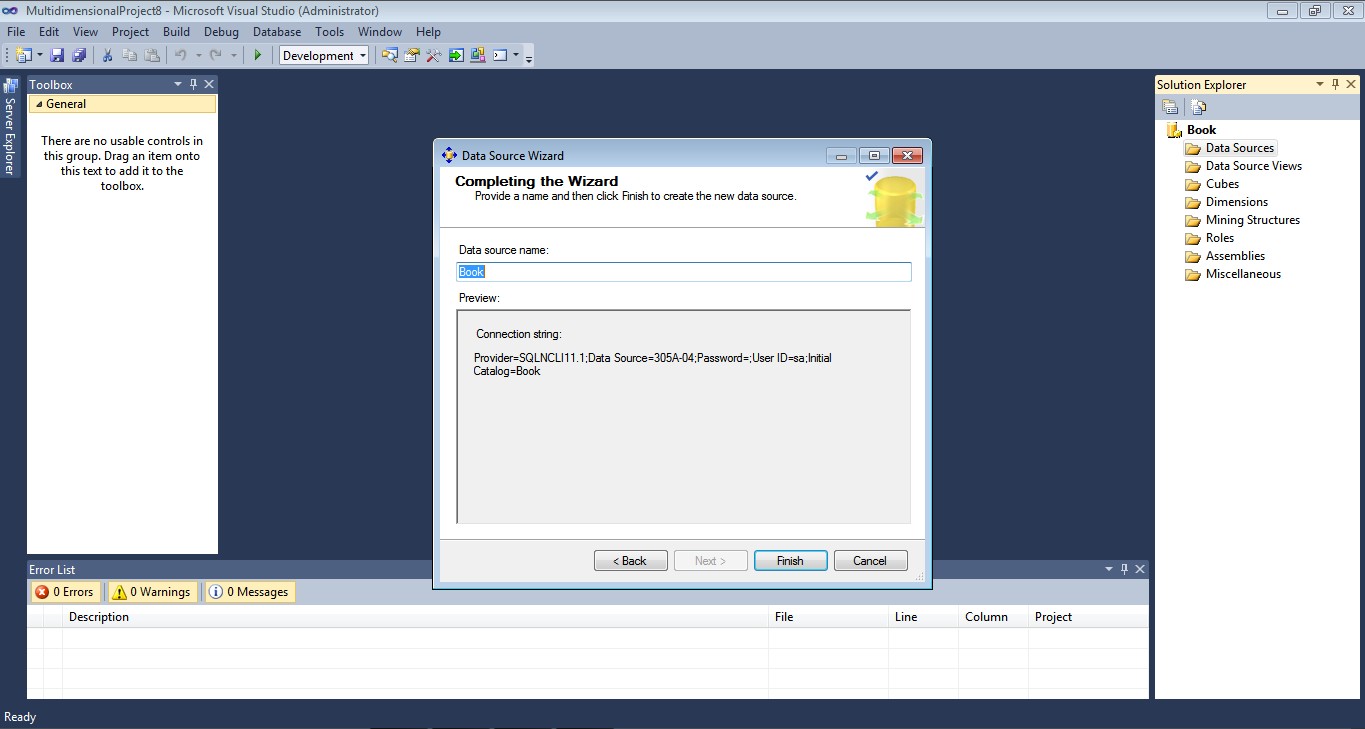


Figure 5: Confirm data source

1. Create data source view.

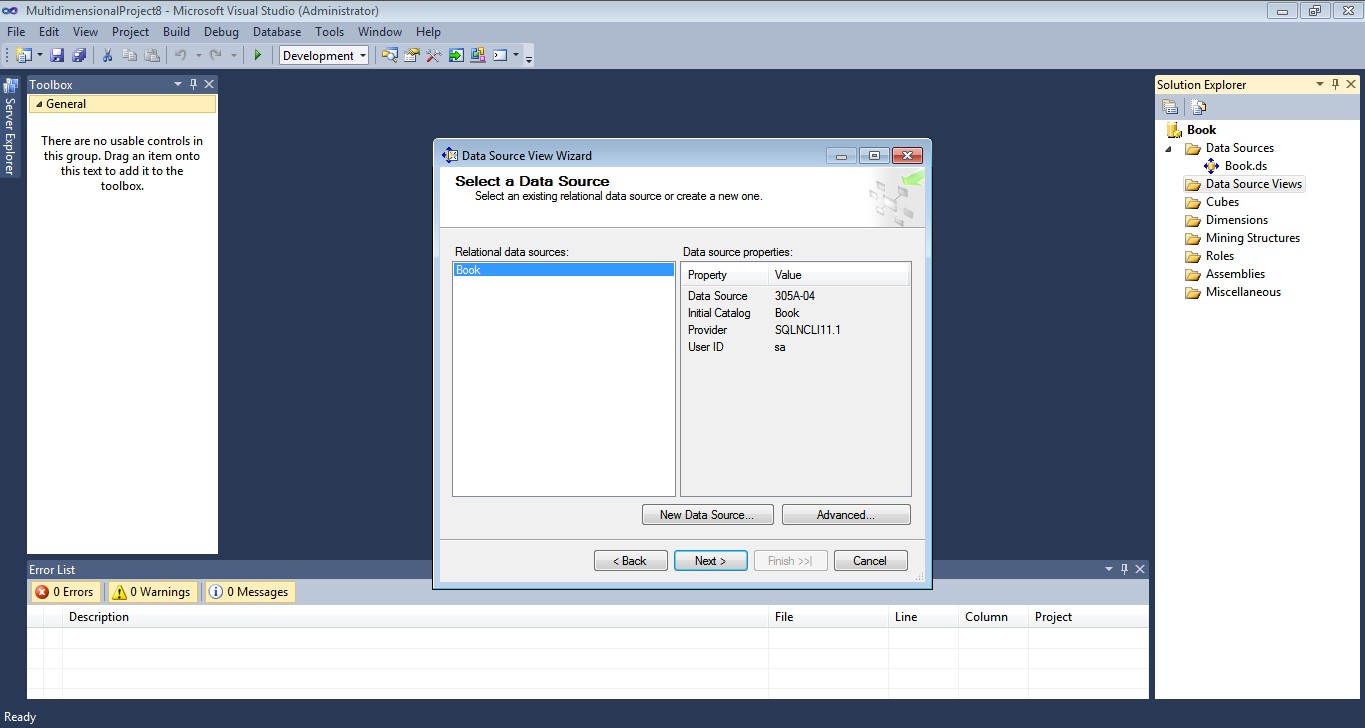


Figure 6: Popup to create data source view

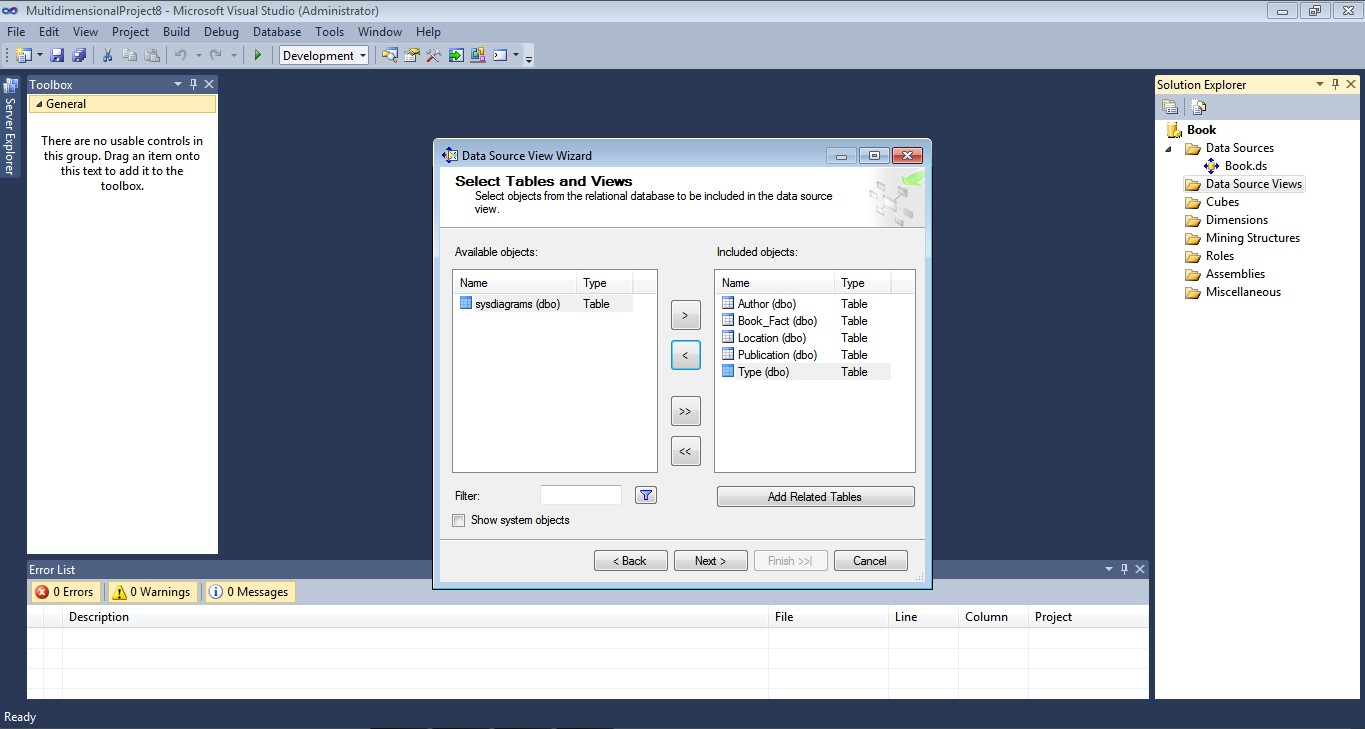


Figure 7: Select tables as per requirement

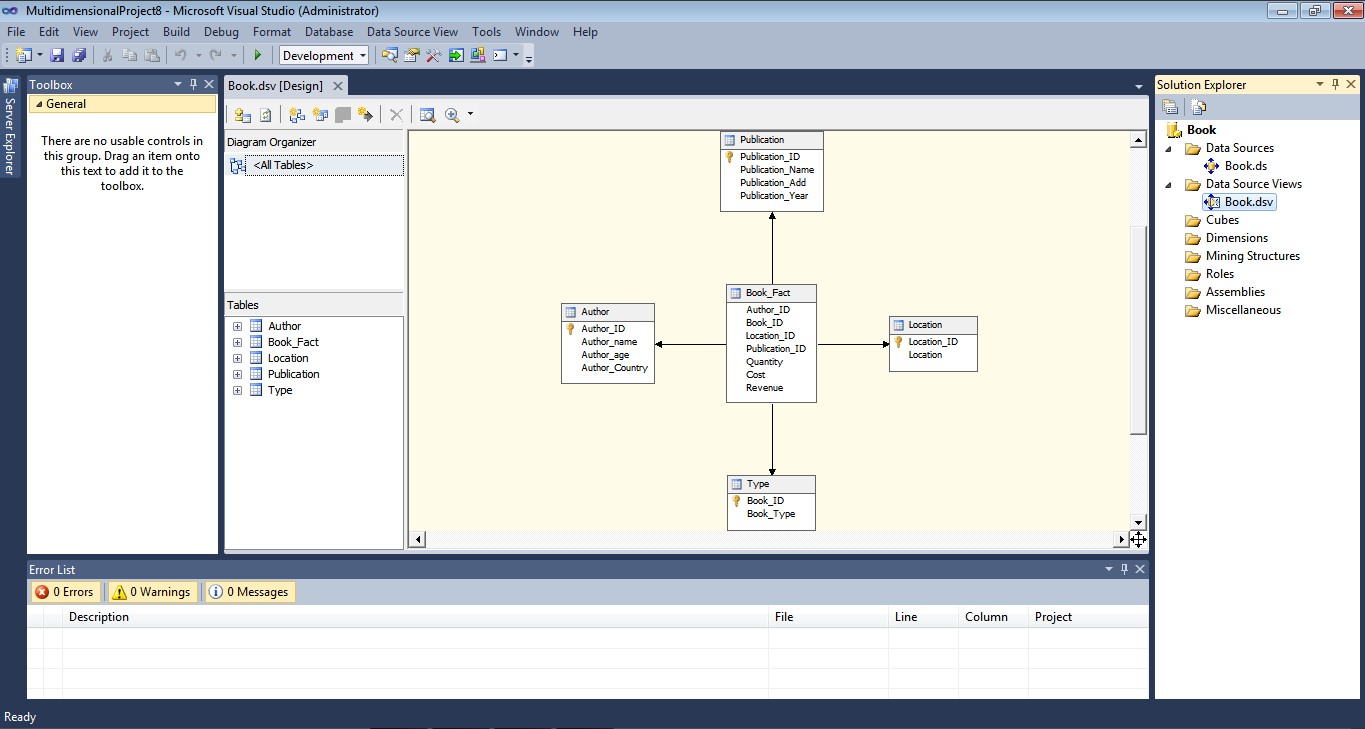


Figure 8: Final Database schema

1. Create Cube.

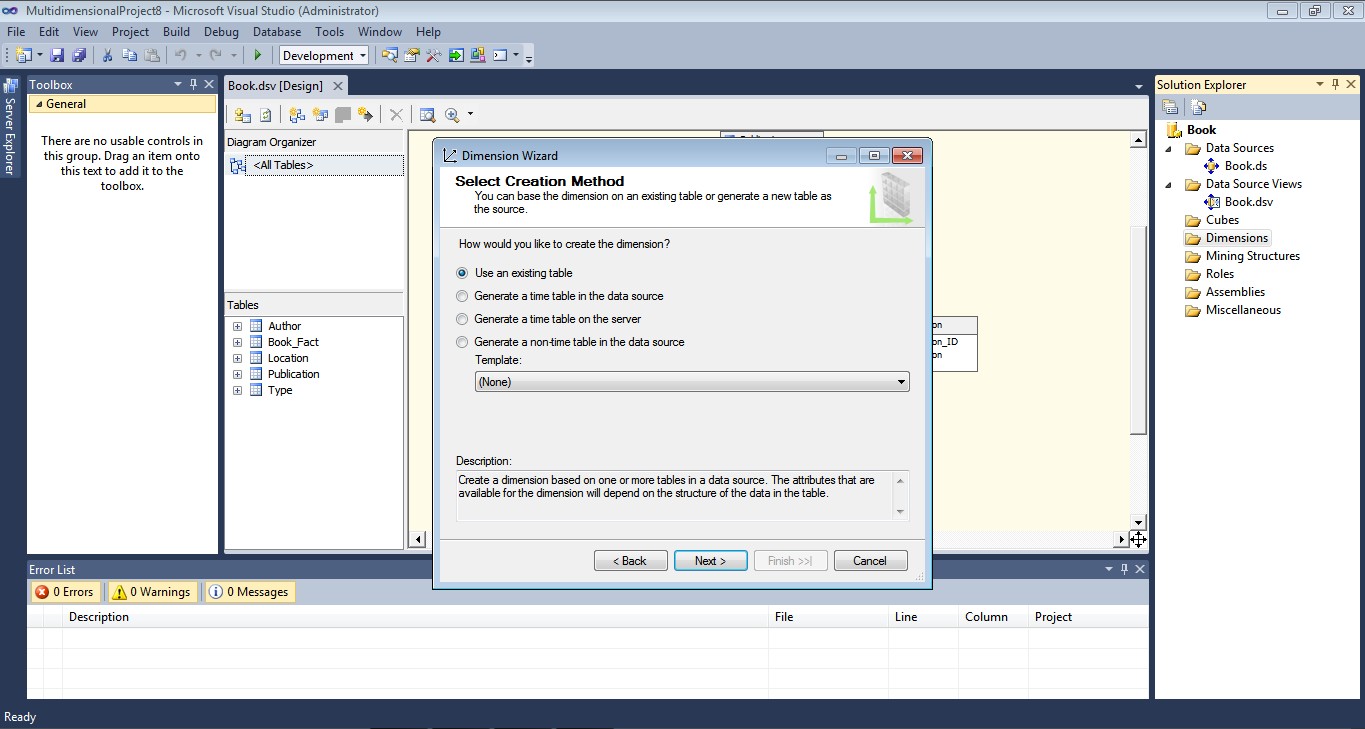


Figure 9: Select to create cube

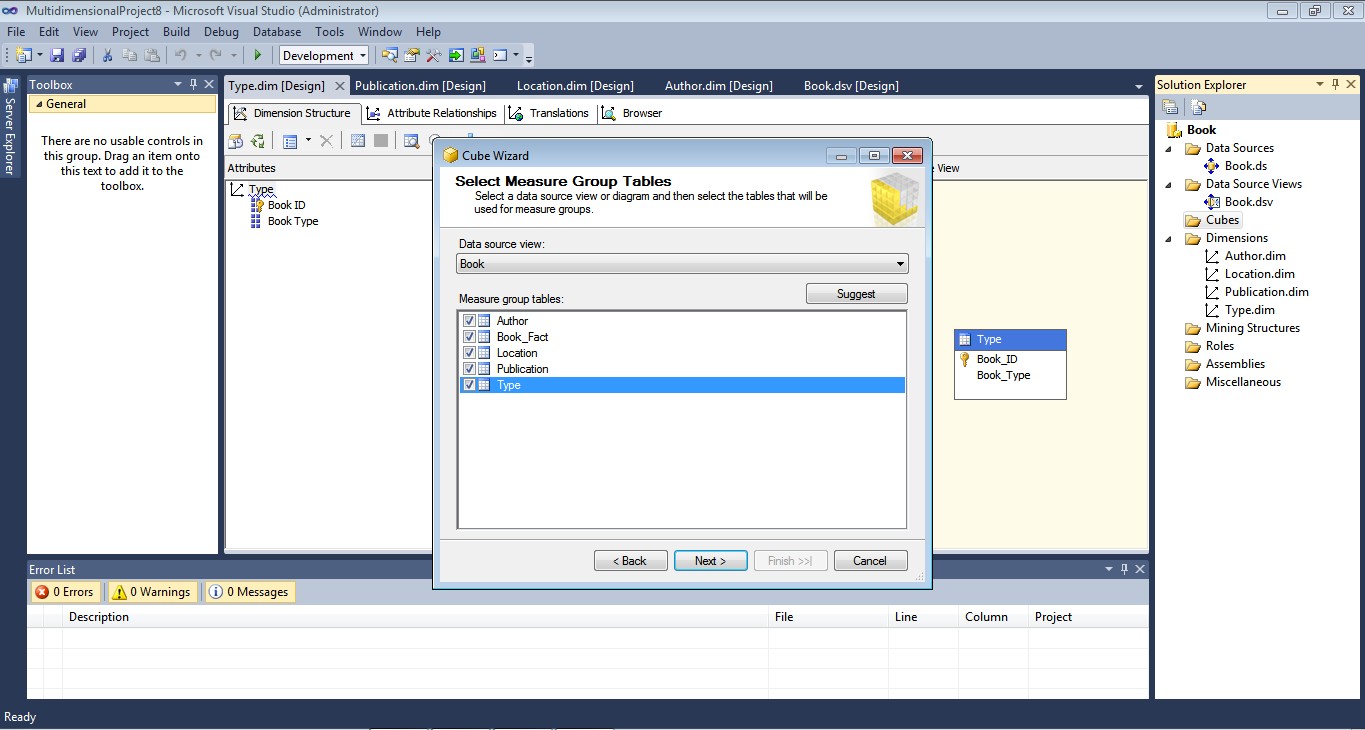


Figure 10: Select measure group tables

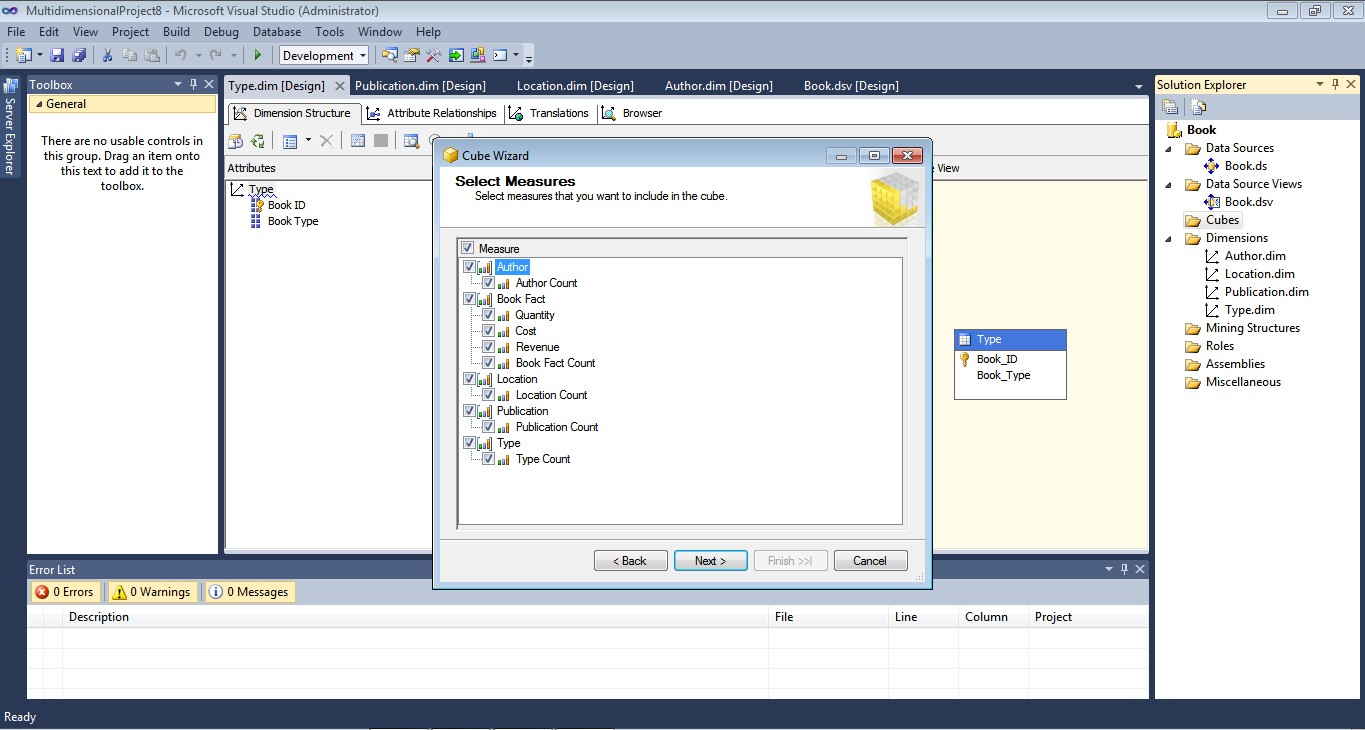
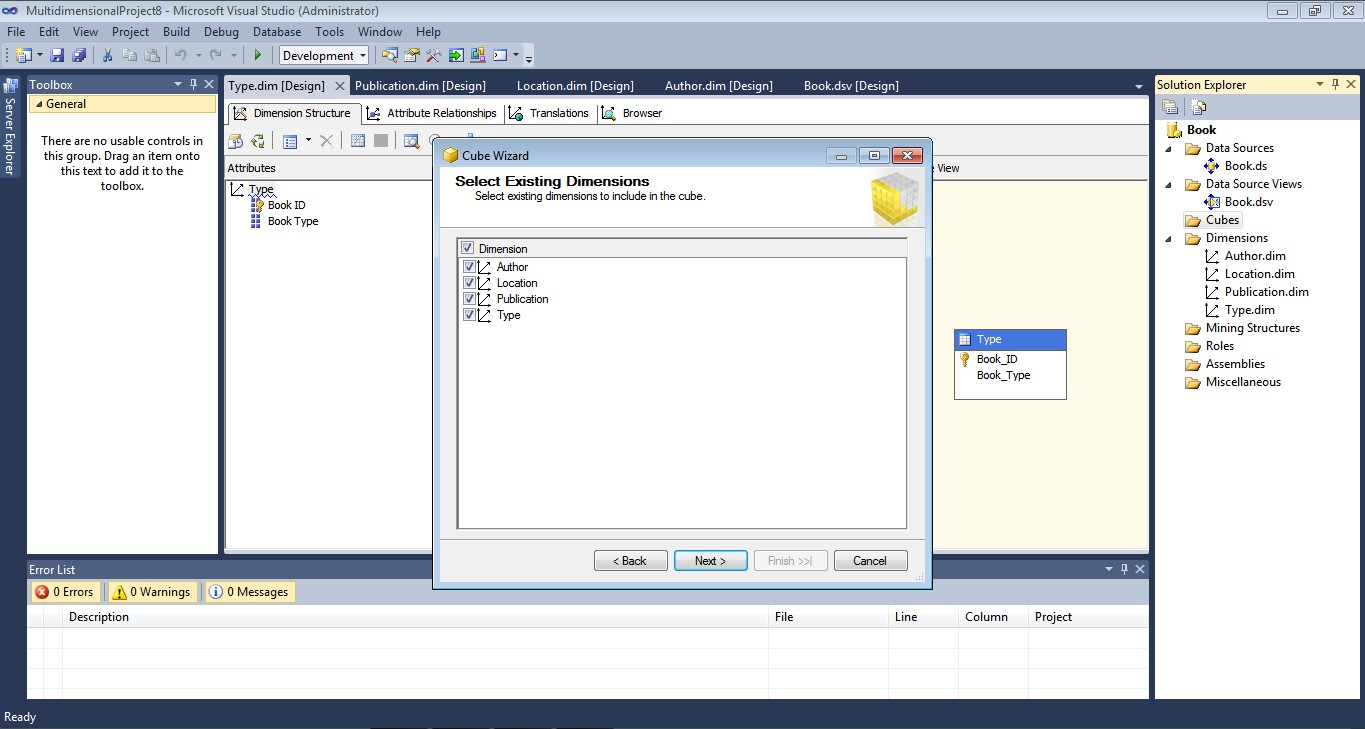


Figure 11: Popup to select measures

Figure 12: Popup to select existing dimensions

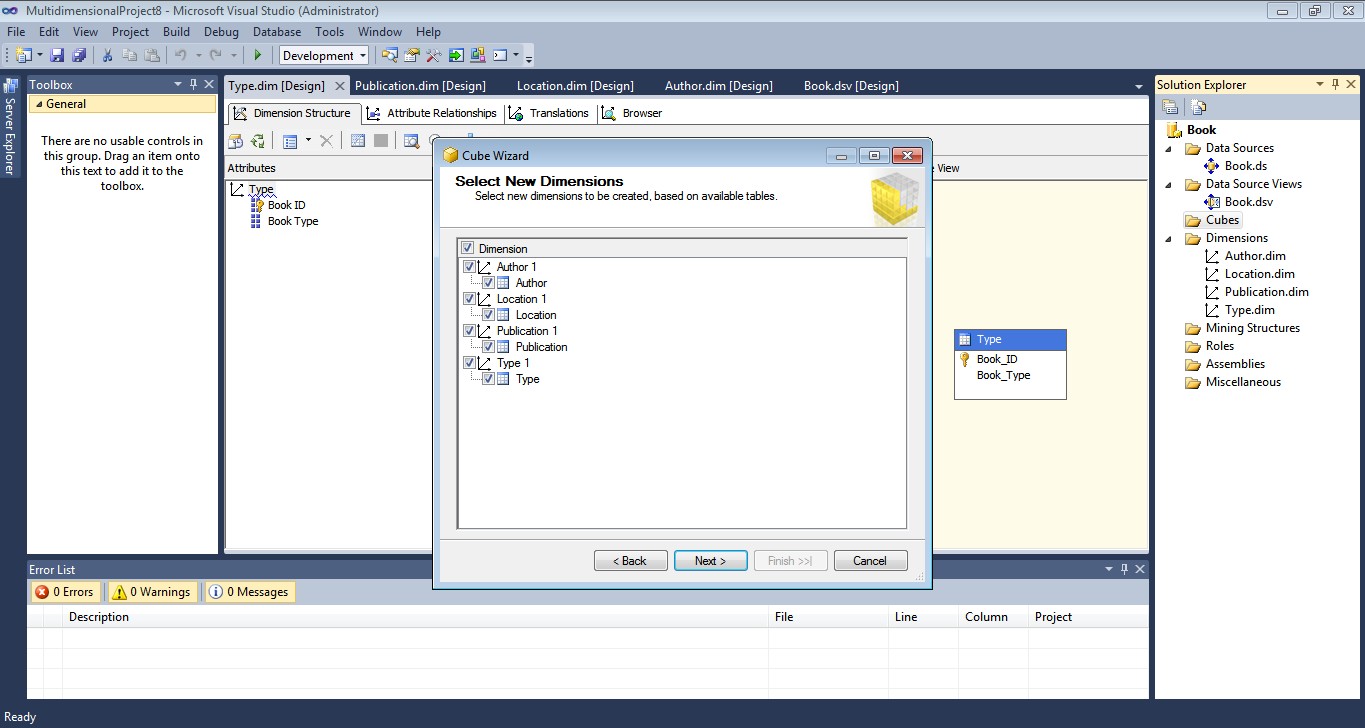


Figure 13: popup to select new dimensions

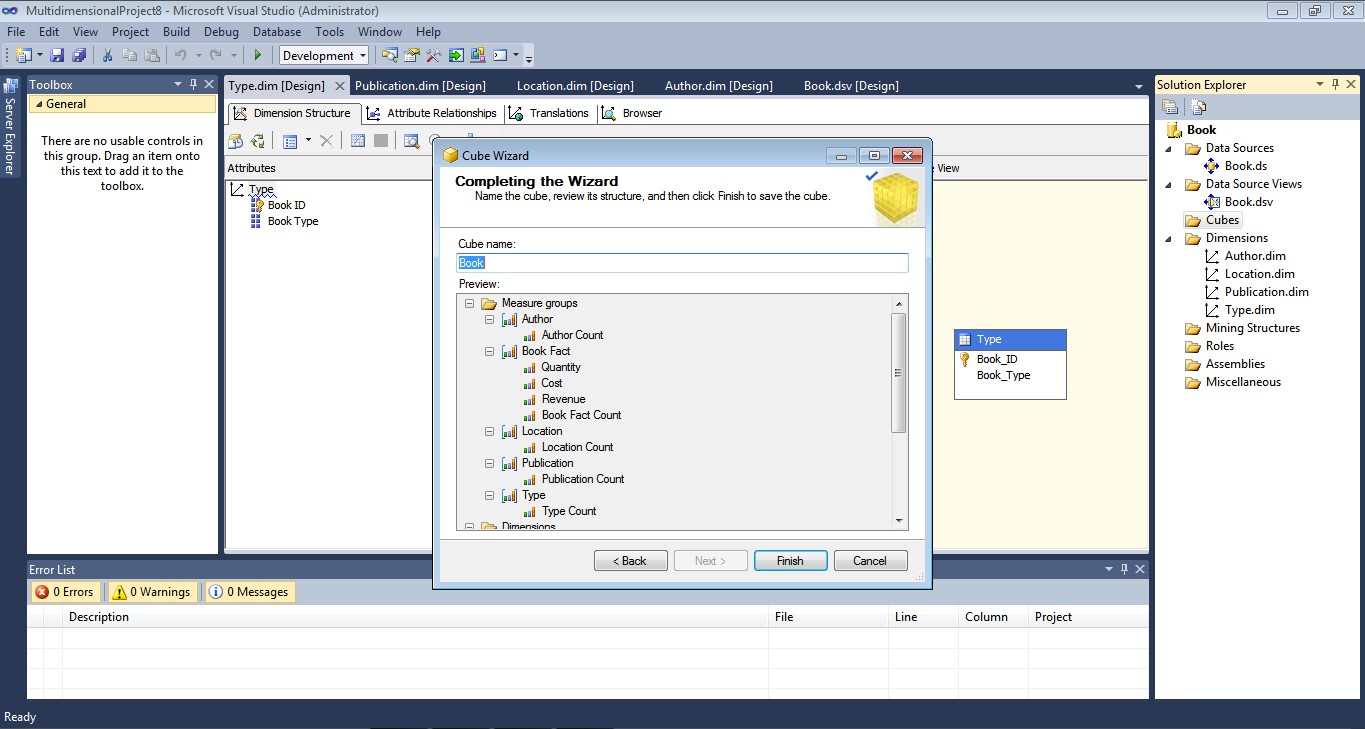


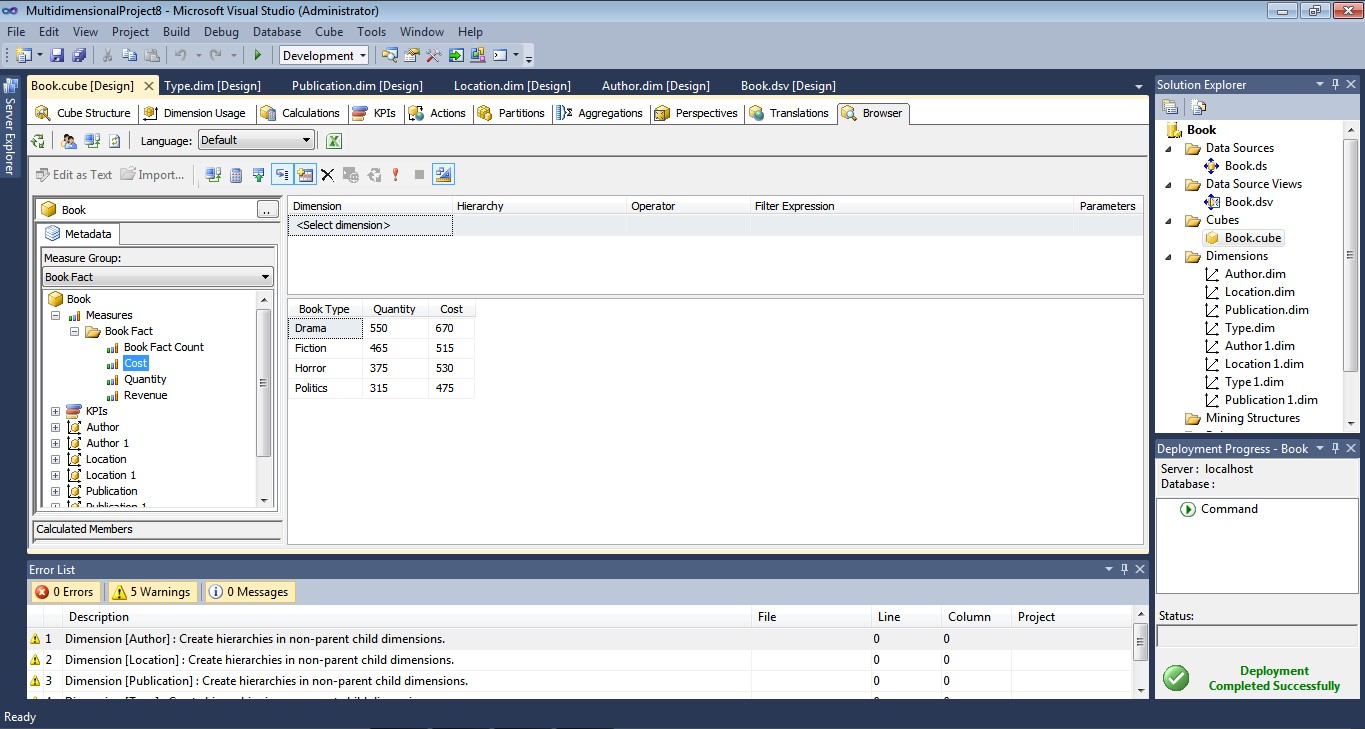
Figure 14: Popup to validate selections



Figure 15: Popup to start processing cube



Figure 16: Successful Process Completion

Figure 17: Result showing #Book type associated with its quantity and cost

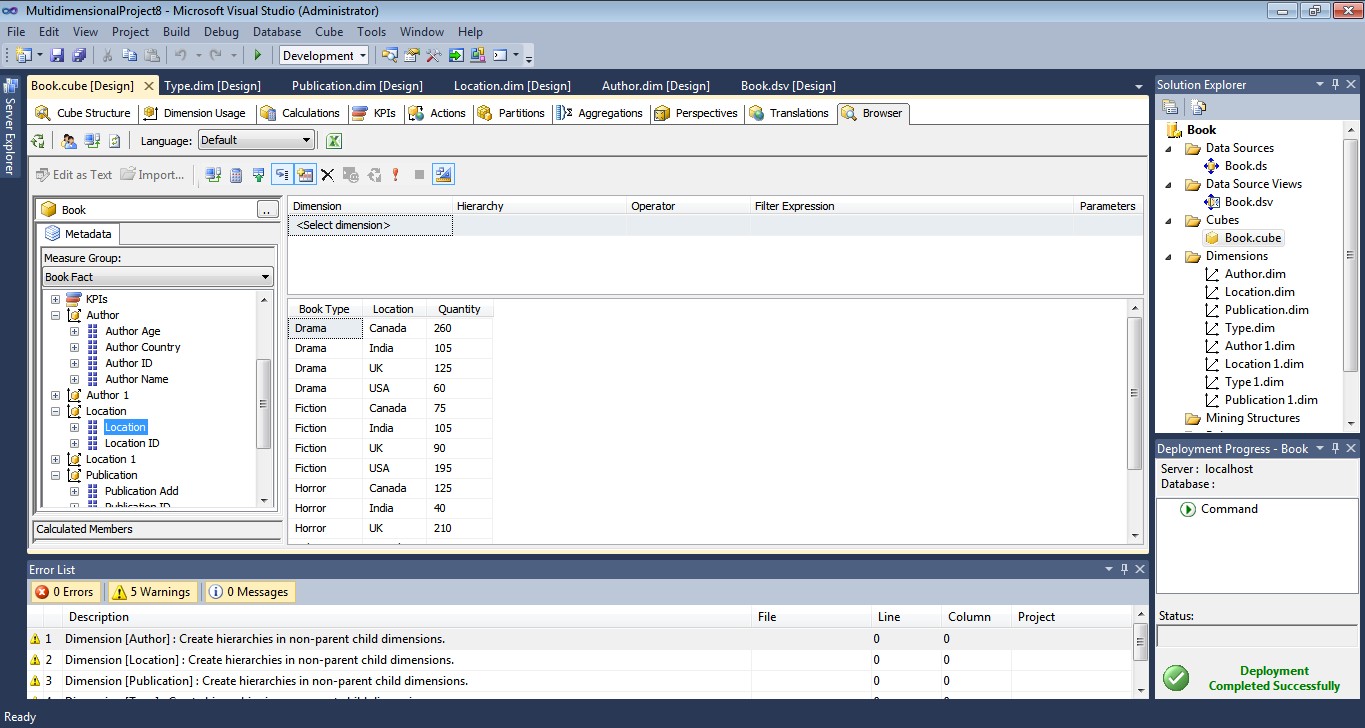
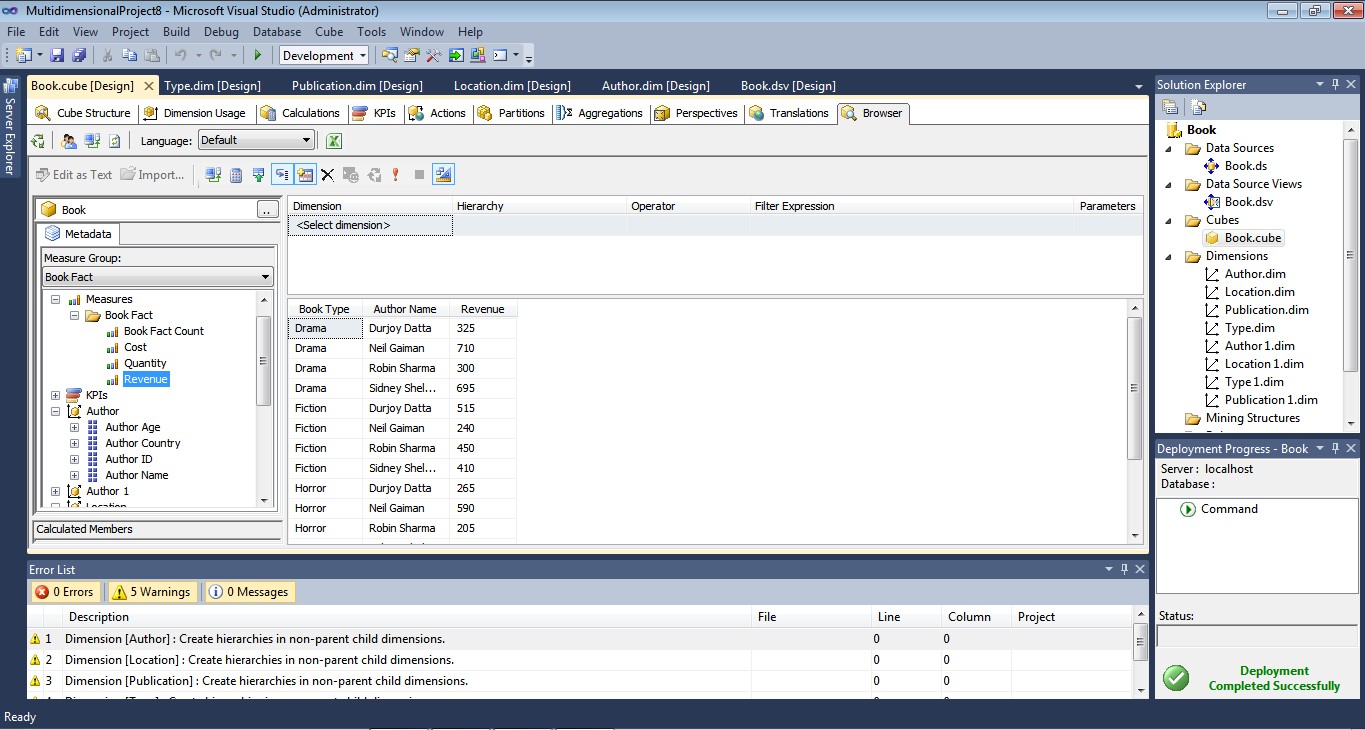
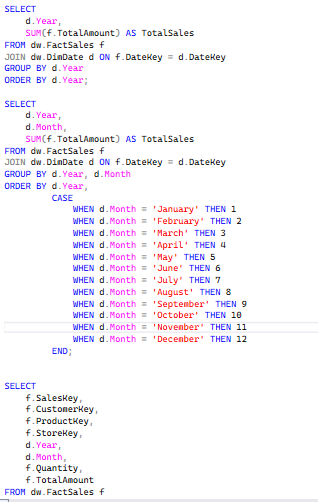
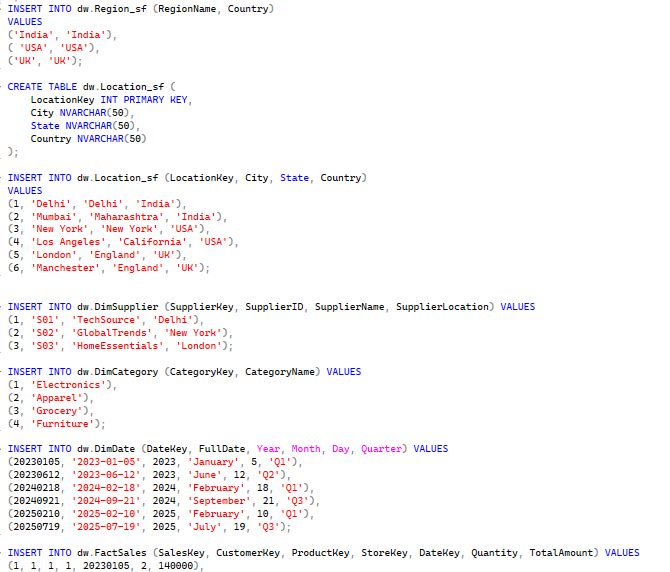
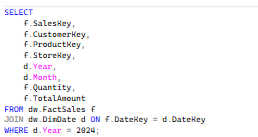
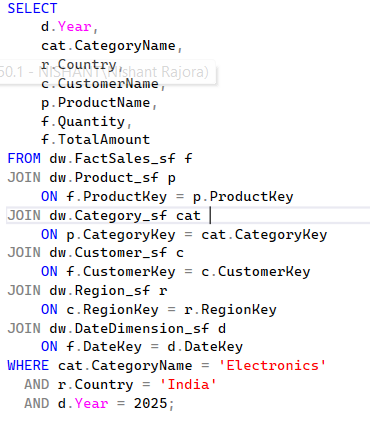


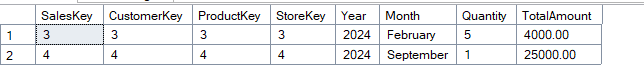
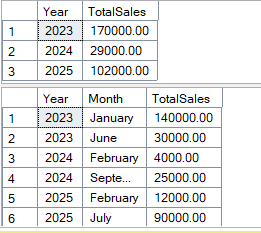
Figure 18: Result showing #Book type and quantity according location

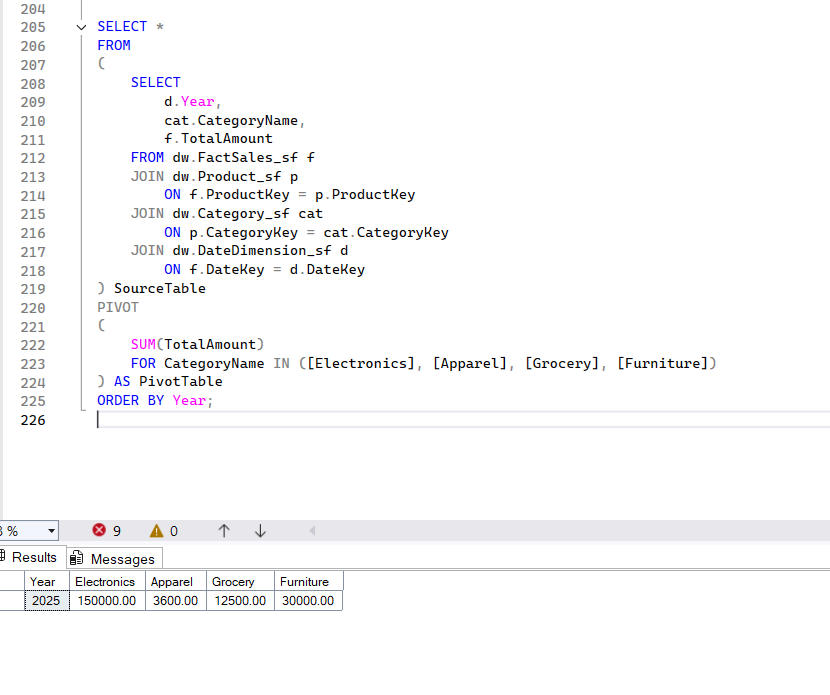
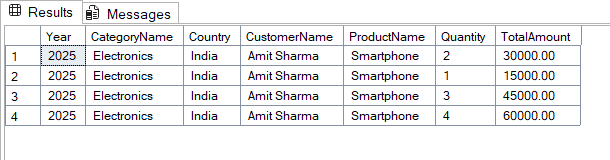


****

****

****

****

****

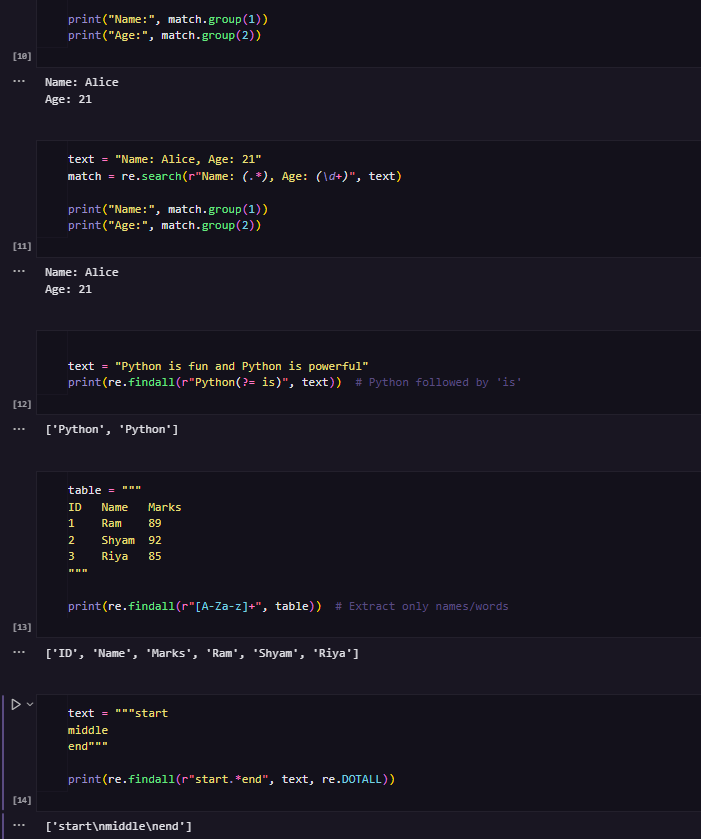
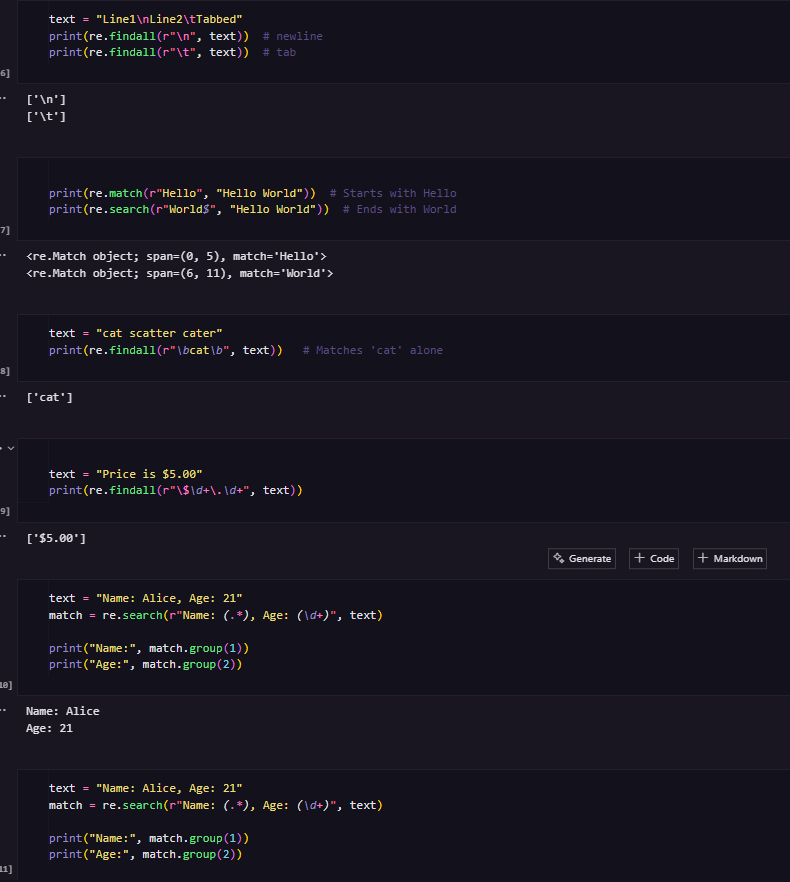
**EXPERIMENT NO. 11**

|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code: 11/11/2025** |
| **Date:** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

|  |
| --- |
| **Objective:**  To implement all regular expression components such as lazy and greedy quantifiers, alternation and grouping, non-printable characters and anchors. |
| **Outcome:**  Students will learn how to implement various regular expression components using Python and online editor. |
| **Problem Statement:**  Perform the following tasks:   * Implementing the following regular expression components: * Matching Fixed Strings, special characters, characters sets * Alternation * Greedy and non-greedy quantifiers * Lazy quantifiers * Grouping * Non-printable characters * Anchors * Boundary matches * Escaping Metacharaters * [Specifying Groups and Fields](https://zone.ni.com/reference/en-XX/help/371714F-01/nirghelp/regular_expressions_components/#group_backref) * [Evaluating Occurrences](https://zone.ni.com/reference/en-XX/help/371714F-01/nirghelp/regular_expressions_components/#occurrences) * Specifying Location, alternatives * [Matching Information from a Table](https://zone.ni.com/reference/en-XX/help/371714F-01/nirghelp/regular_expressions_components/#table) * Capturing Multiple Lines |
| **Background Study:**  Python, Regular expression components, Regex:   * + Regex and Python – A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern. RegEx can be used to check if a string contains the specified search pattern. Python has a built-in package called re, which can be used to work with Regular Expressions.   + Regular expression components- Regular expressions can contain multiple components to accomplish the above given tasks in the problem statement when Requirements Gateway analyzes intermediate files. |
| **Question Bank:** |

**Student Work Area**

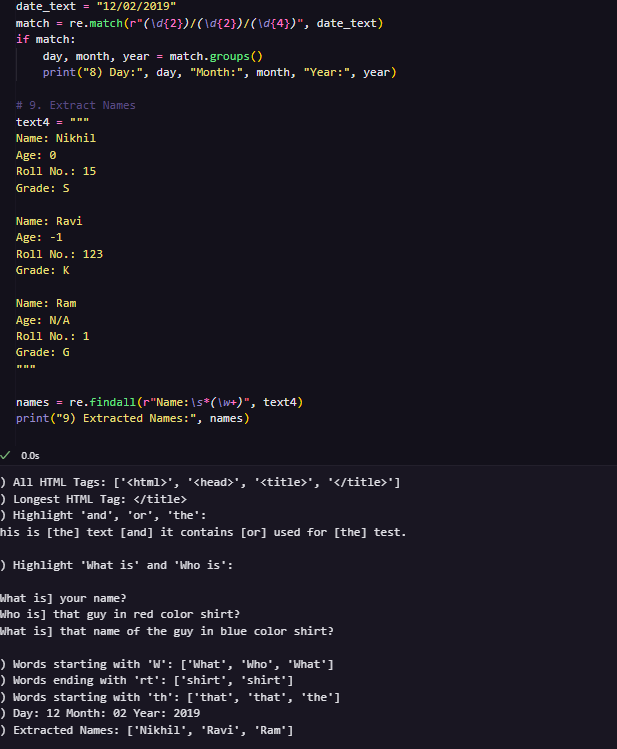
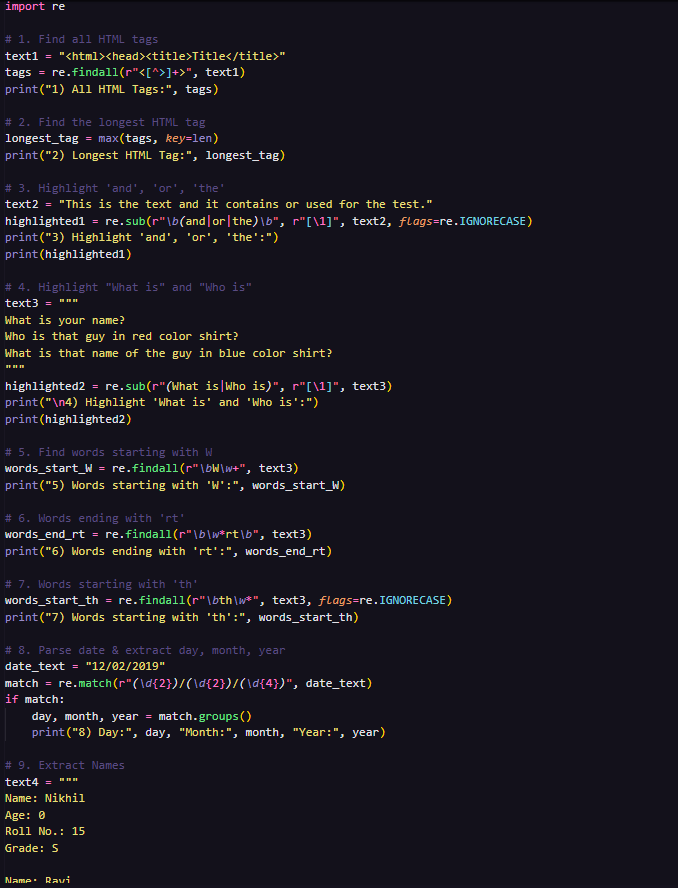
**Algorithm/Flowchart/Code/Sample Outputs**

****

**EXPERIMENT NO. 12**

|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code:** |
| **Date: 11/11/2025** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Students will learn to implement different regular expression components |
| **Outcome:**  Students will learn to implement regular expression components like – greedy and lazy quantifiers, alternation, grouping, non-printable characters and anchors. |
| **Problem Statement:**  **Perform the following queries using regular expression components**   * Write a regular expression to find all html tags in String """<html><head><title>Title</title>""" * Write a regular expression to find the longest html tags in String """<html><head><title>Title</title>""" * Write a regular expression to find and highlight all occurrences of “and”, “or”, “the” in a given text. * Write a regular expression to search and highlight the substrings “What is” and “Who is” in a given text   """  What is your name?  Who is that guy in red color shirt?  What is that name of the guy in blue color shirt?  """   * Find and highlight the text starting with ‘W’ in a text given above. (Output – “What”, “Who” ) * Find the text ending with ‘rt’ in the text given above. * Find the words starting with “th” in the text given above. * Create a regular expression for parsing a date and determine day, month and year from the given text "12/02/2019". * Create a regular expression for finding all the patterns with Name: <some-name> and extract <some-name>. The text is given below:   """  Name: Nikhil  Age: 0  Roll No.: 15  Grade: S  Name: Ravi  Age: -1  Roll No.: 123  Grade: K  Name: Ram  Age: N/A  Roll No.: 1  Grade: G  """ |
| **Background Study:**   * Greedy Behaviour of Quantifiers - The greedy behavior of the quantifiers is applied by default in the quantifiers. A greedy quantifier will try to match as much as possible to have the biggest match result possible. * Non-Greedy Behaviour (or Reluctant/Lazy) of Quantifiers - The **non-greedy** (or **reluctant**) behavior can be requested by adding an extra question mark to the quantifier. A quantifier marked as reluctant will behave like the exact opposite of the greedy ones. They will try to have the smallest match possible like - ??, \*? or +? * Alternation is used to match a single regular expression out of several possible regular expressions. This is accomplished using the pipe symbol |. * Non-printable Characters and Anchors –   ^ - Matches beginning of the string (or line)  $ - Matches end of String (or line)  \b – Matches a word boundary   * Grouping – Frequently you need to obtain more information than just whether the regex pattern matched or not. By placing part of a regular expression inside round brackets or parentheses (, ), you can group that part of the regex pattern together.   **Application of Grouping:**   * Apply a quantifier to the entire group - For example, (ab)+ will match one or more repetitions of ab. * Restrict alternation to part of the regex - For example, my name is ram|sam will match my name is ram and sam whereas my name is (ram|sam) will match my name is ram and my name is sam. * Capture the text matched by group – Groups indicated with (, ) also capture the starting and ending index of the text that they match. Groups can be retrieved by passing an argument to group(), start(), end(), and span() of the Match object. Groups are numbered starting with 0. Group 0 is always present; it captures the whole regex pattern, so all Match object methods have group 0 as their default argument. |

****

**EXPERIMENT NO. 13**

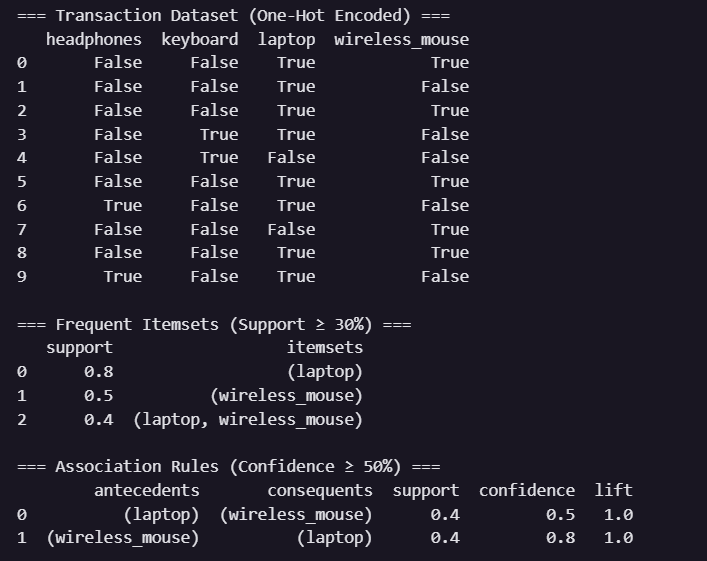
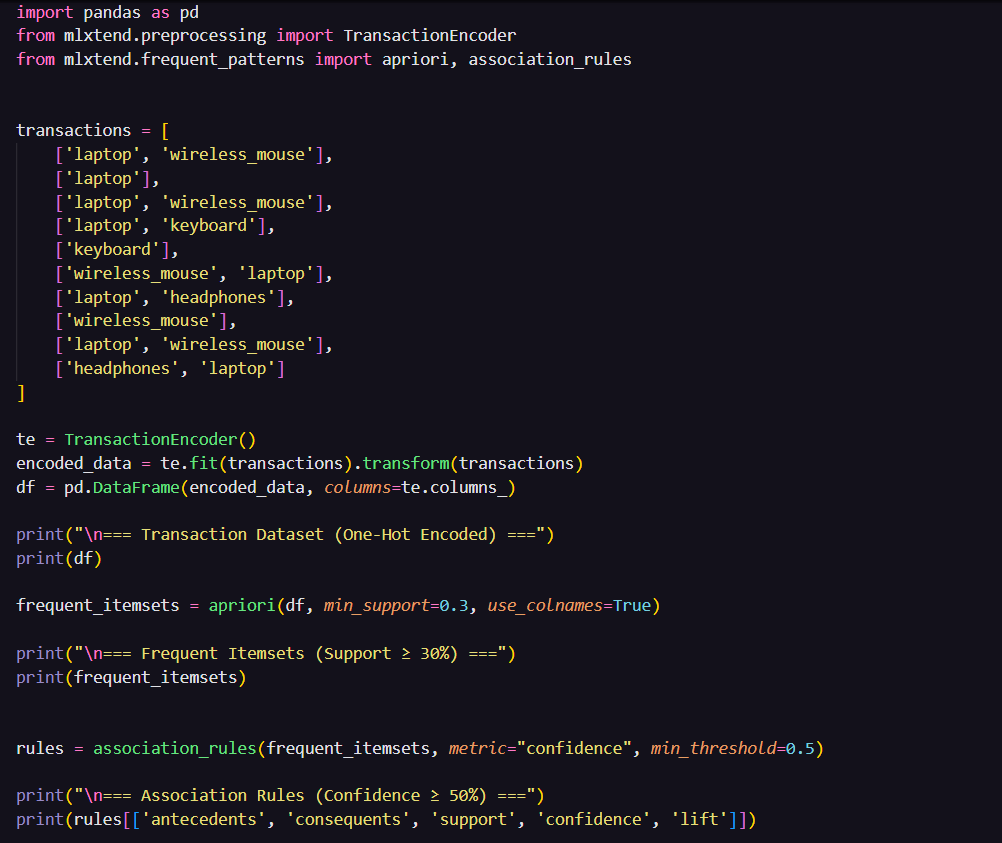
|  |
| --- |
| **Student Name and Roll Number: Nishant Rajora 23csu220** |
| **Semester /Section: 5th DS-A** |
| **Link to Code:** |
| **Date: 11 /11/2025** |
| **Faculty Signature: Deepika Garg** |
| **Marks:** |

|  |
| --- |
| **Objective:**  To implement Apriori algorithm for market basket analysis. |
| **Outcome:**  Students will learn to implement Apriori algorithm for market basket analysis on python. |
| **Problem Statement: Perform the following tasks :**   1. Consider the Groceries data. 2. Import the required libraries 3. Load and explore the data 4. Clean and split the data as per requirements 5. Perform the hot encoding 6. Build the model and analyze the results |
| **Background Study:**  Association rule mining, Apriori algorithm, market basket analysis, GCP   * + Association rule mining – It helps to discover frequent patterns, correlations and associations from data sets found in various kinds of databases. It targets at finding rules to predict the co-occurrences of various items. It is a two-step approach: * Frequent item set generation * Rule generation   + Apriori algorithm using market basket analysis- The Apriori algorithm is used for mining frequent item sets and developing association rules from a transactional database. Two parameters are used: * Support- It refers to frequency of occurrence of various items. * Confidence- It is a conditional probability station how likely an item is purchased when another item is purchased. |
| **Question Bank:** |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

|  |
| --- |
| **Question Bank:** |

****