SLIIT

Discover Your Future

Database management systems (IT 2040)

Lecture 07 – Database Utilities

Lecture content

- Transferring data between different sources
- Backup and restore of data
- Jobs and job schedules
- Database maintenance plans

Learning outcomes

- At the end of this week, students should be able to
  - Transfer data between a data source and a database
  - Create jobs in SQL server
  - Develop a simple database maintenance plan

Database utilities

- Once the database is developed, you may need to perform several tasks on the created tables and stored data.
- SQL server provides many types of utilities for performing commonly used tasks.
- These include,
  - Data Transferring
  - Creating and managing jobs
  - Backup tools & etc.

data transferring

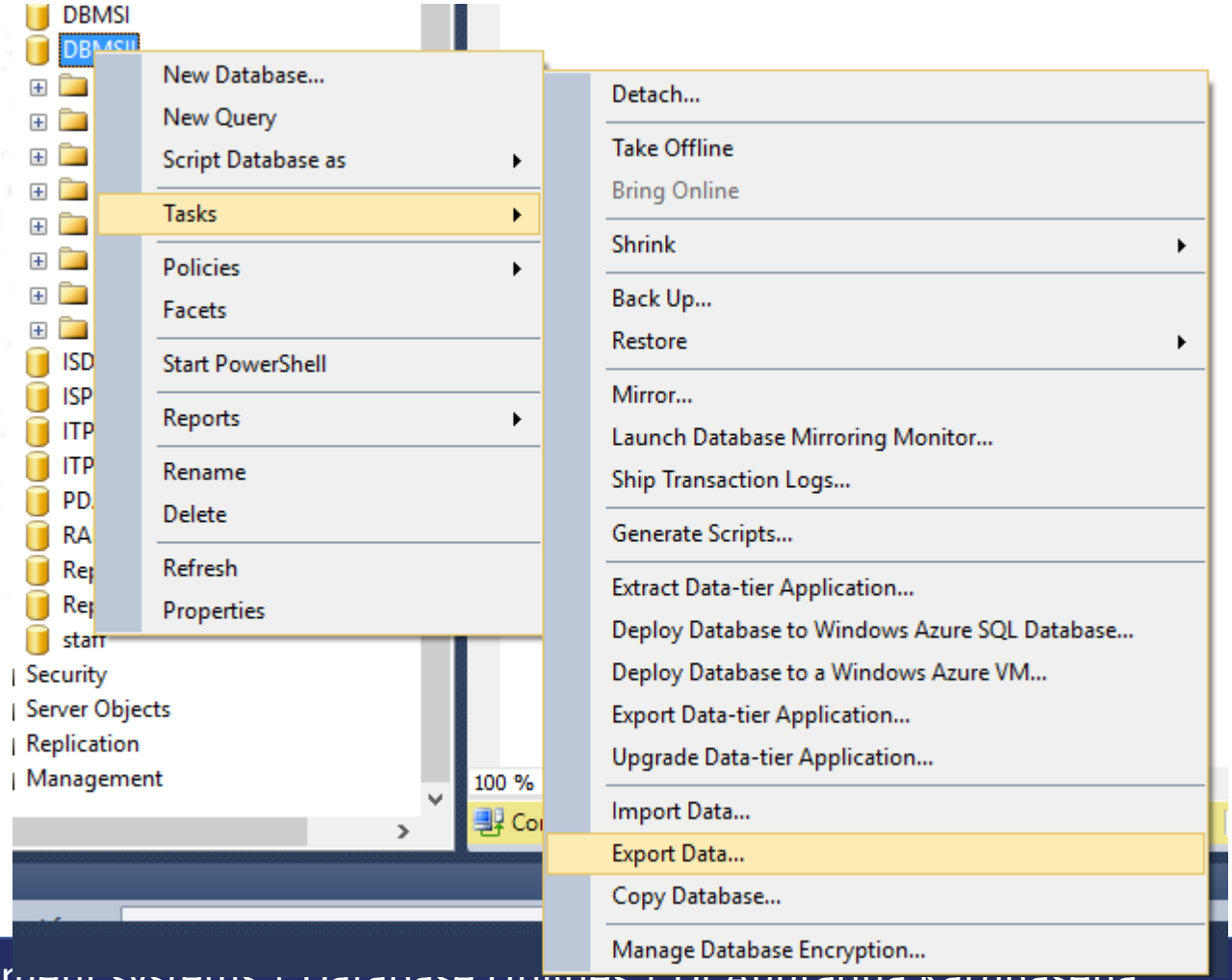
- Some times it is required for transferring data between two sources such as between tables and servers.
- SQL server offers a number of options to facilitate the above as follows :
  - SQL Server Integration Services (SSIS)
  - Using the SQL Server Import and Export Wizard
  - Using BCP to Import and Export Data
  - BULK INSERT
  - SELECT INTO command

SQL sever import and export wizard

- The SQL sever import and export wizard provides a quick way to move data and perform very light transformations of data.
- The wizard is available in all editions of SQL Server except the Local Database edition and Express.
- Next few slides of the lecture shows how data in an Excell file could be exported to a SQL server table in 5 steps.

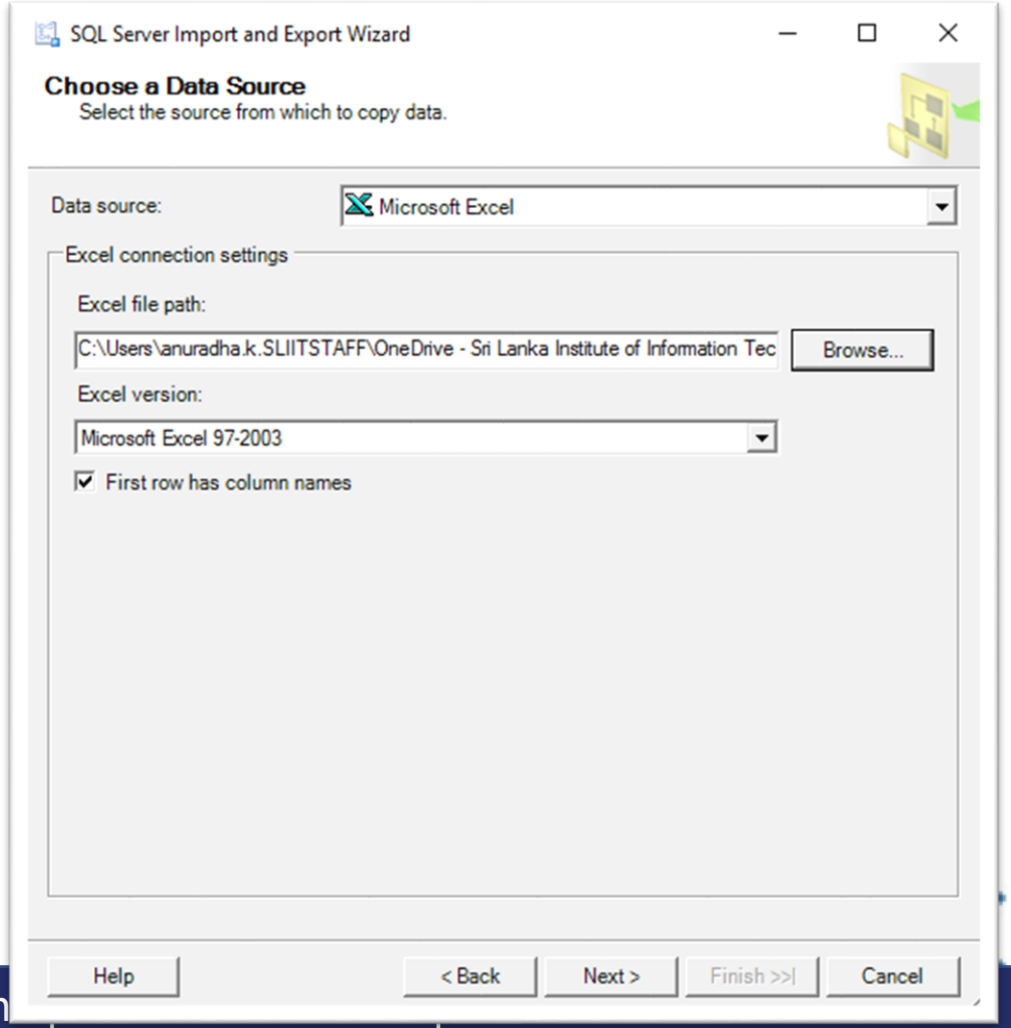
Step 1 – Exporting data from an excel file

- Log in to the SQL sever
- Right click on a database name and select export data sub menu from the task menu.
- Click Next in the welcome screen which appears.



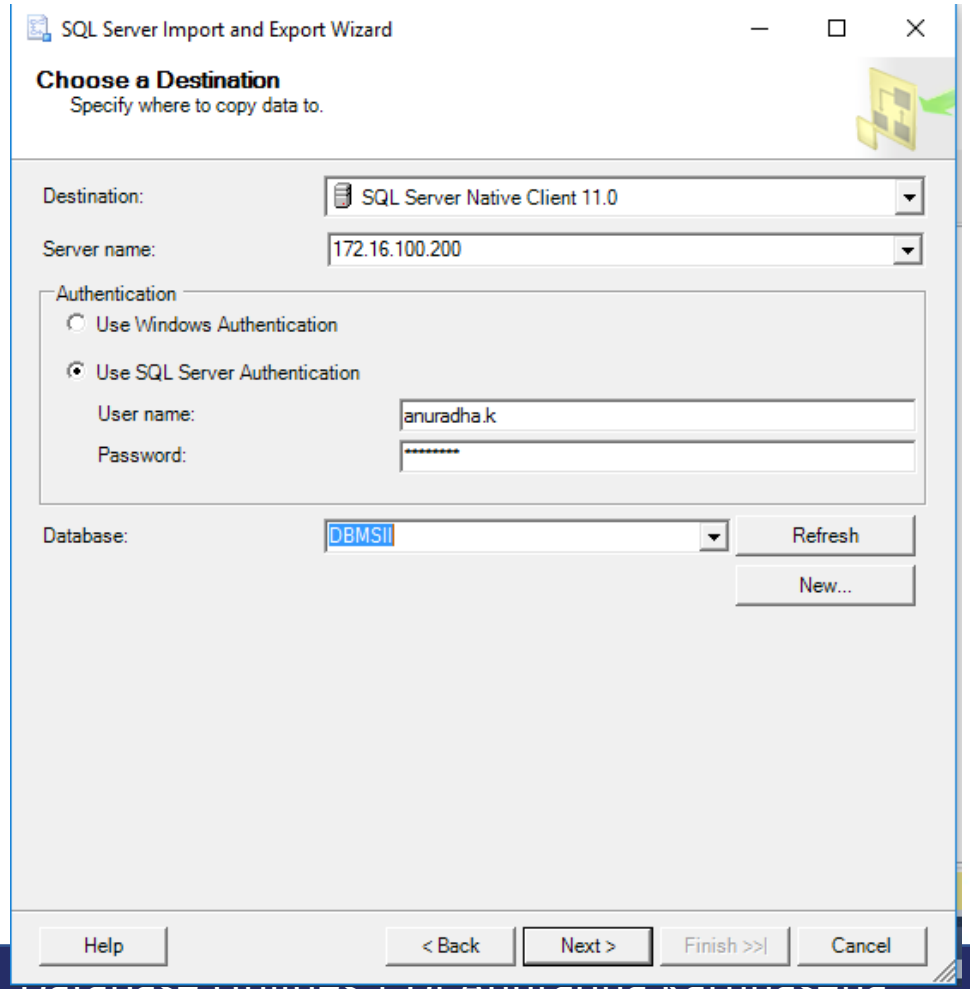
Step 2 – Exporting data from an excel file

- On the next screen select the data source.
- In this scenario the data source is an excel file.
- Browse and select the location of the file.
- Tick the check box on the bottom of the screen if the column names are in the first row of the file.
- Click Next.



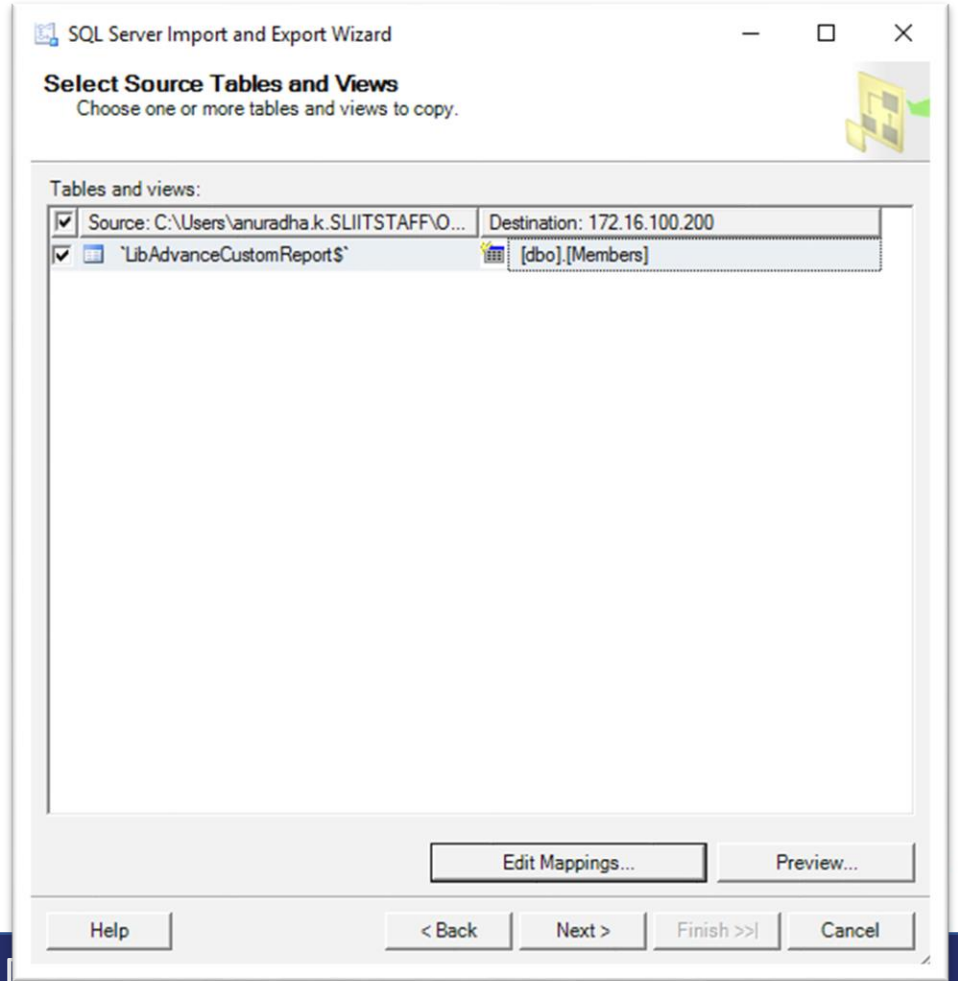
Step 3 – Exporting data from an excel file

- Select the destination for the exported data.
- In this case the destination is the SQL sever database.
- Provide the server name, authentication information and the name of the target database in this screen.
- On the next, select whether all data from a file should be copied or specific data should be drawn through querying.
- In this case all data from the file should be copied.



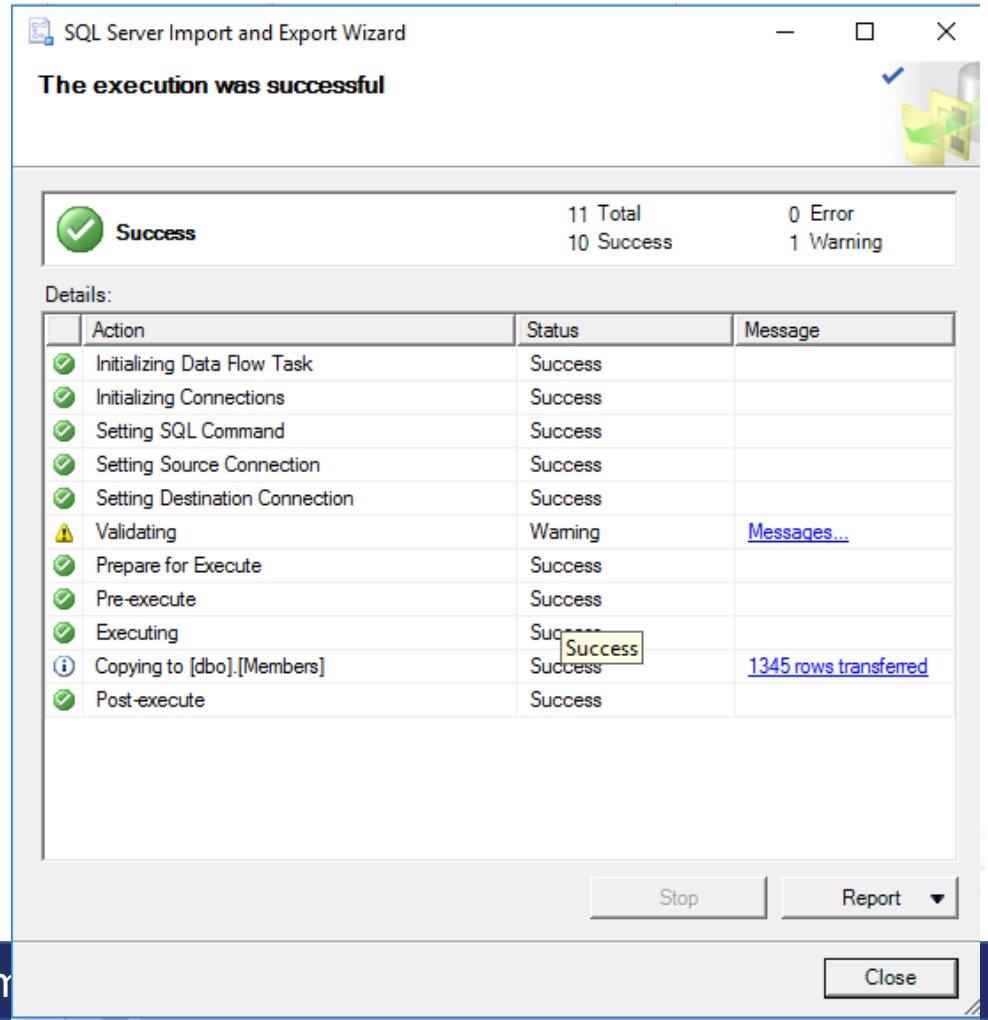
Step 4 – Exporting data from an excel file

- In this screen mapping between the source and destination could be done.
- Change the destination name (i.e. target table name) if it is required.
- Select edit mapping to provide more information on the mapping between the source and the destination
  - Ex: data types of the source table
- Click Next



Step 5 – Exporting data from an excel file

- Click finish to complete the migration on the next screen.
- Outcome of the data migration process would be displayed as shown on the screen.



BCP command

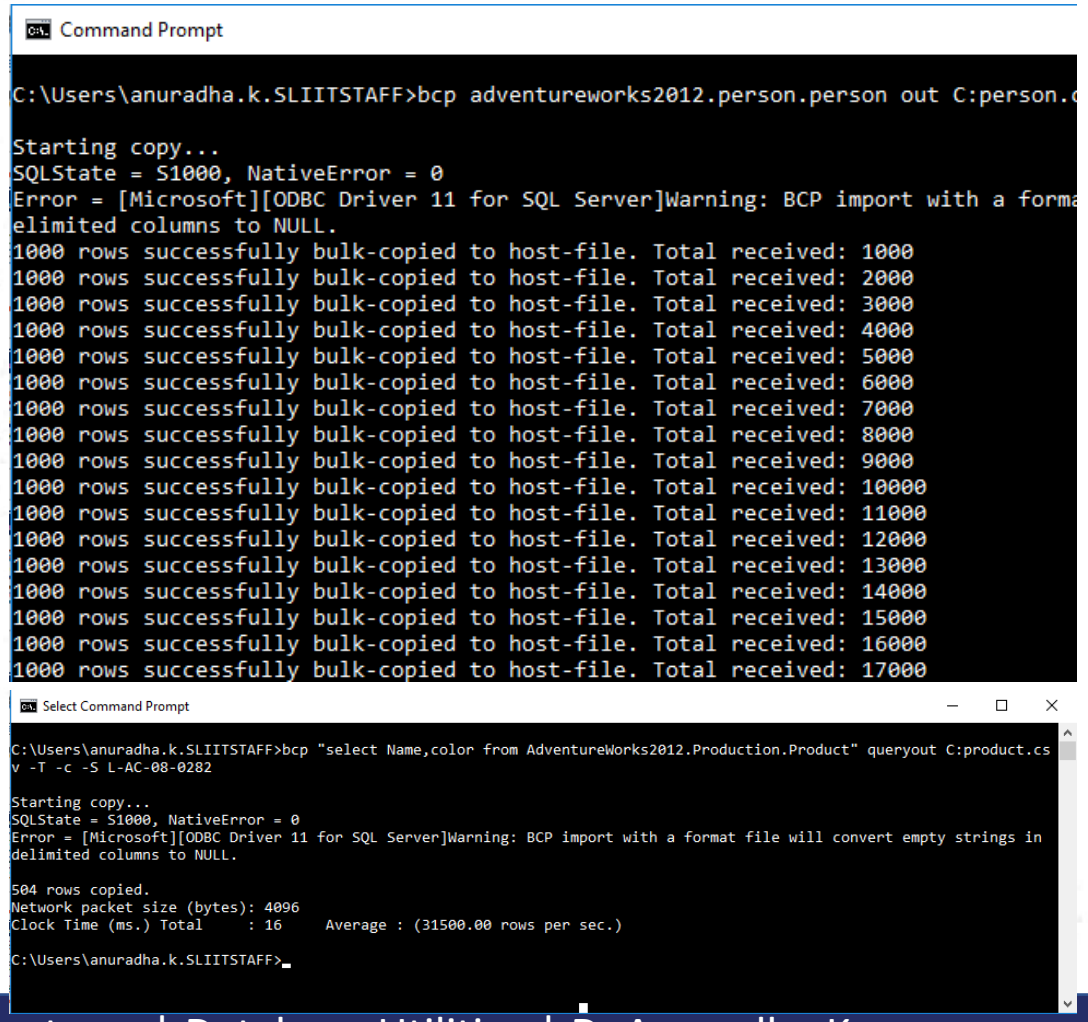
- BCP is a powerful command line utility that enables us to transfer large number of records between a SQL instance and a data file using a special file format.
- This tool is installed by default with SQL Server.
- With BCP the data migrated can be a table or a SQL query result.
- Type BCP command on the command line to obtain the options that could be used with the command.

BCP command (Contd.)

- Format of the BCP command is as follows
  - bcp {table|view|"query"} {out|queryout|in|format} {data_file|nul} {[optional_argument]...}
  - Table|view|query represents the source of the data
  - out|queryout|in|format determines the command's mode (direction)
    - Ex: **out option** exports data from a table or view into a data file.
    - Ex: queriyout option exports data retrieved through a query into a data file.
  - *data_file*|nul is the full path of the data file or, when a data file should not be specified, the **nul** value.

BCP Command – Examples

- Copying data from a table to a file
  - bcp adventureworks2012.person.person out C:person.csv -T -c -S L-AC-08-0282
  - -c argument is used to perform operations using a character type.
  - -T is used to connect using a Trusted connection (Windows Authentication).
  - -S is used to specify the SQL Server name.
- Exporting data from a SQL Server Query to a file.
  - bcp "select Name,color from AdventureWorks2012.Production.Product" queryout C:product.csv -T -c -S L-AC-08-0282



Bulk insert command

- BULK INSERT loads data from a data file into a table.
- This functionality is similar to that provided by the **in** option of the **bcp** command; however, the data file is read by the SQL Server process.
- Example :

```
BULK INSERT emp2 from "D:\emps.csv"
WITH (
FIELDTERMINATOR = ',',
ROWTERMINATOR = '\n'
)
SELECT * FROM emp2
```

SELECT INTO COMMAND

- SELECT INTO statement copies data from one table to another
- Tables can be on the same SQL Server or linked SQL servers or on different types of servers using distributed queries
- SELECT INTO operates just like Bulk Insert except it can’t read from an external file
- SELECT INTO can also create the destination table automatically before copying the data
- Example :
  - select * into emps from emp2

Securing data with backups

- Backing up your SQL Server databases, protects you from potentially catastrophic data loss.
- With valid backups of a database, you can recover your data from many failures, such as:
  - User errors, for example, dropping a table by mistake.
  - Hardware failures, for example, a damaged disk drive or permanent loss of a server.
  - Natural disasters.

Backup and Restore of databases

- The SQL Server backup and restore component provides an essential safeguard for protecting critical data stored in your SQL Server databases.
- To minimize the risk of catastrophic data loss, you need to back up your databases to preserve modifications to your data on a regular basis.
- A well-planned backup and restore strategy helps protect databases against data loss caused by a variety of failures.

Types of backups

- Full Backups : Full database backup takes a copy of the entire database including the part of the transaction log file.
- Differential Backups : Differential database backup includes only extents which were changed since the last full database backup.
- Transaction Log Backups :Transaction log backup captures all the transaction log records that have been written after the last full database backup or last transaction log backup.

Demo

JOBs in SQL server

- A job is a specified series of operations performed sequentially
- Use jobs to define an administrative task that can be run one or more times and monitored for success or failure. A job can run on one local server or on multiple remote servers.
- A job can perform a wide range of activities and can run repetitive or schedulable tasks
- Automatically notify users of job status

Running jobs in sql server

- Jobs can run in several ways:
  - According to one or more schedules.
  - In response to one or more alerts.
  - By executing the sp_start_job stored procedure.
- Jobs can be created by users in several roles including sysadmin user role
- A created job can be edited by only its owners or members of sysadmin role.

Schedules

- A *schedule* specifies when a job runs. More than one job can run on the same schedule, and more than one schedule can apply to the same job. A schedule can define the following conditions for the time when a job runs:
  - Whenever SQL Server Agent starts. (SQL server agent is the component of SQL sever responsible for automation)
  - Whenever CPU utilization of the computer is at a level you have defined as idle.
  - One time, at a specific date and time.
  - On a recurring schedule.

Demo

Maintenance Plans

- A database maintenance plan is a set of specific, proactive tasks that need to be performed regularly on **databases** to ensure their adequate performance and availability.
- Maintenance Plan Wizard and Designer is to cover those critical database maintenance tasks that, as a bare minimum, should be applied to all databases, to ensure *adequate* performance and availability.

Core maintenance plan tasks

- Backup Databases
- Verify the Integrity of Database
- Maintain a Database's Indexes
- Remove Older Data from msdb
- Remove Old Backups

Demo

Thank you!