

Database Administration Security

Cybersecurity Department

Course Code: CBS 214



Practical Lecture 10: Database Backup in MS SQL Server

Practical Lab Exercise: Database Backup in MS SQL Server

Block 1: Create a test database and sample data

```
USE master;
GO
IF DB_ID('BackupLab') IS NOT NULL DROP DATABASE BackupLab;
GO
CREATE DATABASE BackupLab;
GO
USE BackupLab;
GO
CREATE TABLE Students (
    ID INT IDENTITY PRIMARY KEY,
    Name VARCHAR(50),
    Grade INT
);
INSERT INTO Students VALUES ('Berivan', 85), ('Dilshad', 90), ('Hewa', 78);
SELECT * FROM Students;
GO
```

Block 2: Full database backup

```
BACKUP DATABASE BackupLab
TO DISK = 'C:\BackupLab\BackupLab_Full.bak'
WITH INIT, COMPRESSION;
PRINT 'Full backup created.';
GO
```

- `BACKUP DATABASE` - Creates a complete copy of the whole database.
- `TO DISK = '...'` - Saves the backup to `C:\BackupLab\` (the folder you created).
- `WITH INIT` - Overwrites any existing file with the same name.
- `COMPRESSION` - Reduces file size (about 70-80% smaller).

A full backup is the foundation - you must have one before using differential or log backups.

Block 3: Make changes and create a differential backup

```
USE BackupLab;
GO
INSERT INTO Students VALUES ('Shilan', 95);
UPDATE Students SET Grade = 88 WHERE Name = 'Dilshad';
DELETE FROM Students WHERE Name = 'Hewa';
SELECT * FROM Students;
GO

BACKUP DATABASE BackupLab
TO DISK = 'C:\BackupLab\BackupLab_Diff.bak'
WITH DIFFERENTIAL, INIT;
PRINT 'Differential backup created.';
GO
```

- **Changes made:** Insert Shilan (95), update Dilshad to 88, delete Hewa. Now we have 3 rows: Berivan(85), Dilshad(88), Shilan(95). `WITH DIFFERENTIAL`, this backup stores **only the changes** that happened since the last full backup.
- **Benefit:** Much smaller and faster than a full backup. (Typical strategy: full weekly, differential daily.)

Block 4: Make more changes and create a transaction log backup

```
USE BackupLab;
GO
INSERT INTO Students VALUES ('Rozhin', 100);
UPDATE Students SET Grade = 92 WHERE Name = 'Berivan';
SELECT * FROM Students;
GO

-- Log backups require FULL recovery model
ALTER DATABASE BackupLab SET RECOVERY FULL;
GO

BACKUP LOG BackupLab
TO DISK = 'C:\BackupLab\BackupLab_Log.trn'
WITH INIT;
PRINT 'Transaction log backup created.';
GO
```

- **More changes:** Insert Rozhin (100), update Berivan to 92. Now we have 4 rows.
- `ALTER DATABASE ... SET RECOVERY FULL` – Needed for log backups (default is SIMPLE, which does not allow log backups).
- `BACKUP LOG` - Captures every transaction (INSERT, UPDATE, DELETE) that happened after the last log backup.
- **Why log backups?** They allow point-in-time recovery (e.g., “restore to 10 minutes before an accidental delete”).

Block 5: Simulate a disaster – drop the database

```
USE master;  
GO  
DROP DATABASE BackupLab;  
PRINT 'Database BackupLab has been deleted. Now we restore...';  
GO
```

- `DROP DATABASE` - Completely removes the database from SQL Server. This simulates a disaster (ransomware, human error, hardware failure). Our backups will bring it back.

Block 6: Restore from backups (Full → Differential → Log)

```
-- Step 1: Restore FULL backup with NORECOVERY
RESTORE DATABASE BackupLab
FROM DISK = 'C:\BackupLab\BackupLab_Full.bak'
WITH NORECOVERY, REPLACE;
PRINT 'Full backup restored (database in restoring state).';
GO

-- Step 2: Restore DIFFERENTIAL backup with NORECOVERY
RESTORE DATABASE BackupLab
FROM DISK = 'C:\BackupLab\BackupLab_Diff.bak'
WITH NORECOVERY;
PRINT 'Differential backup restored.';
GO

-- Step 3: Restore LOG backup with NORECOVERY
RESTORE LOG BackupLab
FROM DISK = 'C:\BackupLab\BackupLab_Log.trn'
WITH NORECOVERY;
PRINT 'Log backup restored.';
GO

-- Step 4: Bring database online
RESTORE DATABASE BackupLab WITH RECOVERY;
PRINT 'Database is now online.';
GO
```

- **NORECOVERY** - Leaves the database in a “restoring” state so you can apply more backups.
- **Order matters:** Full → Differential → Log (in the same sequence they were created).
- **REPLACE** - Overwrites any existing database with the same name (useful when restoring over an old copy).
- Final **WITH RECOVERY** - Makes the database available for normal use.

Block 7: Verify the restored data

```
USE BackupLab;  
GO  
SELECT * FROM Students;  
GO
```

The **SELECT** should show **4 rows**: Berivan(92), Dilshad(88), Shilan(95), Rozhin(100). All changes (even those made after the differential backup) are recovered because the log backup brought them back. This proves the backup chain works.

Block 8: Clean up (optional)

```
USE master;  
GO  
DROP DATABASE BackupLab;  
PRINT 'Lab database removed. Backup files remain at C:\BackupLab\';  
GO
```