

Introduction to Backup and Restore

Objectives

After watching this video, you will be able to:

- Describe common backup and restore scenarios
- Explain the difference between physical and logical backup
- List the items that you can back up and restore
- Explain what you need to consider when backing up databases and their objects

Backup and restore scenarios

- Saving a copy of data for protection
- Recovering from data loss
 - After unplanned shutdown
 - Accidental deletion
 - Data corruption
- Move to a different database system
- Share data with business partners
- Use a copy of the data, e.g. dev or test



Physical vs. logical backups

- Logical backup like create table insert
- Contains DDL and DML commands to recreate database
 - Can reclaim wasted space
 - Slow and may impact performance
 - Granular
 - Backup/restore, import/export, dump & load utilities

- Physical backup
- Copy of physical files, including logs, and configuration
 - Smaller and quicker
 - Less granular if it contains data for more than one, it won't be easily recoverable
 - Can only restore to similar RDBMS
 - Common for specialized storage and Cloud

What to back up



Database



Schema



Tables



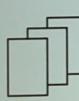
Subset of data



Other objects

Key considerations

- Check that your backup is valid
- Check that your restore plan works
- Ensure that your back up files are secure



Backup options

- Compression:
 - Reduces size for storage and transmission
 - Increases time for backup and restore processes

- Encryption:
 - Reduces the risk of data being compromised
 - Increases time for backup and restore processes

Types of Backup

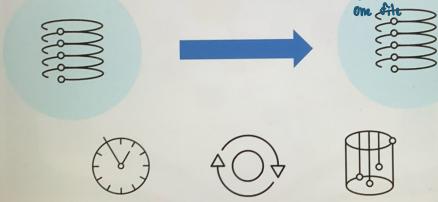
Objectives

After watching this video, you will be able to:

- Identify common types of database backup
- Describe the advantages and disadvantages of different types of database backup

Full backups = complete copy of all of data.

- Backs up all the specified data



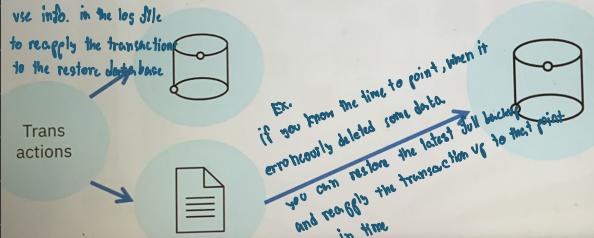
When restore full backup it, it will be back to that state. It means data base may have

Full backups

- Multiple copies of the backup means storing many instances of a large file
- Only storing one copy risks data loss if file is corrupt, then can not restore the data.
- Could be needlessly backing up unchanging data
- Must secure backup files

Point-in-time recovery

- Uses logged transactions to restore to an earlier point in time



processed
many transactions
so to implement

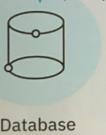
Differential backups

- A copy of any data that has changed since the last full backup was taken → smaller than a full backup file.

Ex. you could perform a full backup once a week

If you need to restore the database on Tuesday

can skip because that file are also included in backup from Tuesday,

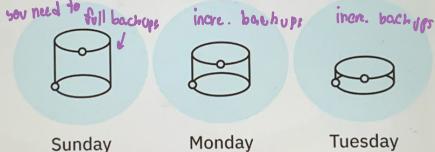


Incremental backups

- A copy of any data that has changed since the last backup of any type was taken

Ex. you could perform a full backup once a week

If you need to restore the database on Tuesday



Database

Sunday

Monday

Tuesday

↳ Restoring from full and incre. backup will take longer than

simply restoring a full backup and diff.

↳ but incre. back likely to be less than needed for a diff back up.

↳ restore date less frequently than you back it up
you are likely to save time overall (diff. take long time
to full backups, but ↓)

Backup Policies

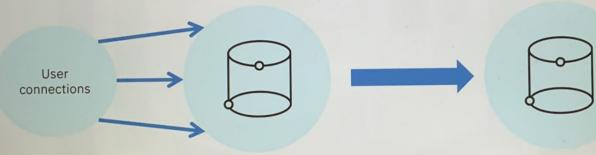
Objectives

After watching this video, you will be able to:

- Explain the difference between hot and cold backups
- Determine an appropriate backup policy

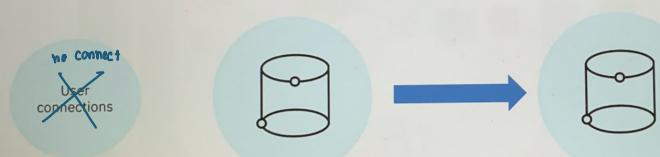
Hot vs. cold backups

- Hot backup - taken while data is in use
 - ↳ or Online backup
 - ↳ advantage: It has no impact on availability and users can continue their activities throughout the backup period

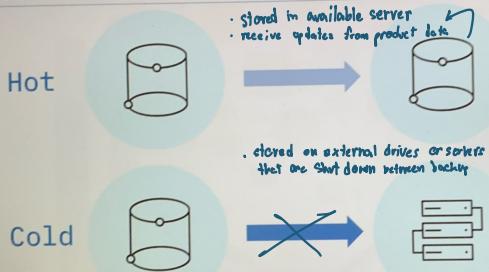


Hot vs. cold backups

- Cold backup - data is offline
 - ↳ offline backup
 - ↳ eliminates the data integrity risks associated with hot backups



Hot vs. cold backups



Backup policies

- Physical or logical
- Full, differential, or incremental
- Hot or cold
- Compression
- Encryption

Backup policies

- Frequency:
 - Is data regularly changing or being added?
 - Is the existing table large?
- Schedule:
 - Is the data accessed equally across the 24-hour day?
 - Is it accessed at weekends?
- Automated

Managed cloud backups

Options dependant upon RDBMS and cloud service provider include:

- Preconfigured automated backup
- Configurable automated backup
- Manual backups
- Third party tools

Using Database Transaction Logs for Recovery

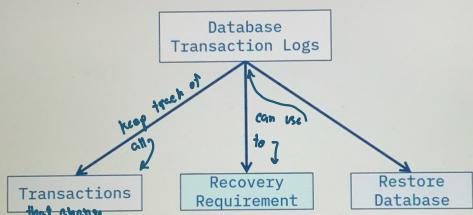
Objectives

After watching this video, you will be able to:

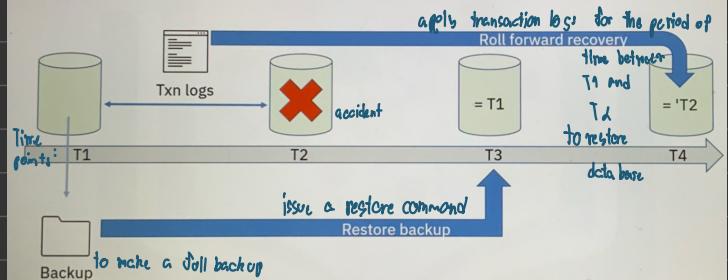
- Describe database transaction logs
- Explain the primary uses of transaction logs
- Indicate where transactional logs are stored and how to access them
- Identify the typical structure of a transactional log

This process is called
Roll forward

Database transaction logs



Transaction log usage example



Storing transaction log files

- Configure log file location
- Db2:
 - sqlogdir
- PostgreSQL:
 - pg_xlog
- Write-ahead log (WAL)

```

db2 get db cfg for sample

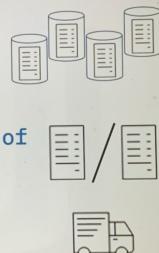
...
Log file size (4KB)
Number of primary log files
Number of secondary log files
Number of redo log streams
Path to log files
Overflow log path
Mirror log path
First active log file
Block log on disk full
(BLK_LOG_DISK_FULL) = NO

OVERFLOWLOGPATH = C:\DB2\NODE0000\SQL00002\SQLLOGDIR
(MIRRORLOGPATH) = 
First active log file = S0000002.LOG
(BLK_LOG_DISK_FULL) = NO
  
```

C:\DB2\NODE0000\SQL00002\SQLLOGDIR

Storing transaction log files

- Isolate logs on different volumes from data
 - Performance
 - Recoverability
- Log mirroring - store second copy of log files in an alternative location
- Log shipping - copy and send logs to replica or standby servers



Accessing transaction log files

- Logging may or may not be enabled by default
 - Log settings are usually configurable
- MySQL - To view transaction log files:
 - SHOW BINARY LOGS;
- Transaction logs:
 - Typically in binary format
 - May be encrypted
 - View using special tools



Accessing transaction log files

- MySQL: mysqlbinlog -v DB_Bin_Logs.000001


```

# at 218 #080828 15:03:08 server id 1 end_log_pos 258 Write_rows: table id 17 flags:
STMT_END_F BINLOG 'FA535BMAAAALAAAANOAAAAABEAAAAAAABHR1c3QAAQXQAAWMPGCIUAAQ=
FA535BcRAAAAKAAAAATRAAAOABFAAAAAAEEAA//RADAQAAVhHR5Q== /**/;
## INSERT INTO test.t ## SET ## @1=1 ## @2='apple' ## @3=NULL
  
```
- Db2: db2fmtlog S0000002.LOG -replayonlywindow


```

LREC | 4812 0003F9E7 000000000197
LREC | Record LSO = 53456942
      Record TID = 000000000197
| DDL Statement = create table t5ba.t1 (a int, b long varchar) in t1
  
```

Sources: MySQL - <https://dev.mysql.com/doc/refman/8.0/en/mysqlbinlog-row-events.html>
DB2 - <https://www.ibm.com/support/pages/db2fmtlog-%E2%80%93-format-and-display-log-file-information-tool>

Basic anatomy of a database log

