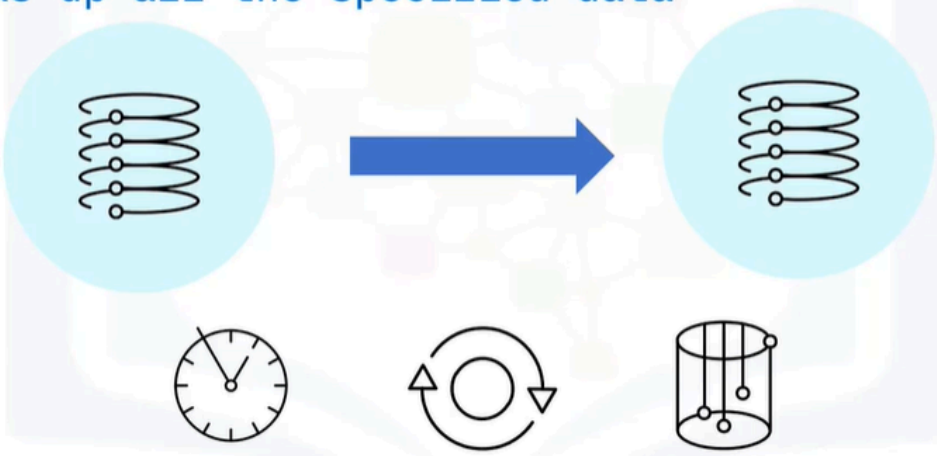
Types of Backup

When we talk about backups, think of it like taking a snapshot of your favorite moments in life. Just like you might take photos to remember special events, backups help you save your important data so you can recover it if something goes wrong. There are several types of backups, but the main ones are full backups, differential backups, and incremental backups.

- There are many different types of backup that you can use to meet your backup and restore needs. The choice of which are available to you depends on the database system that you are using. The commonly used types include: Full backup Point in time backup Differential backup Incremental backup A full backup is a complete copy of all of the data in the object or objects that you are backing up.

Full backups

- Backs up all the specified data

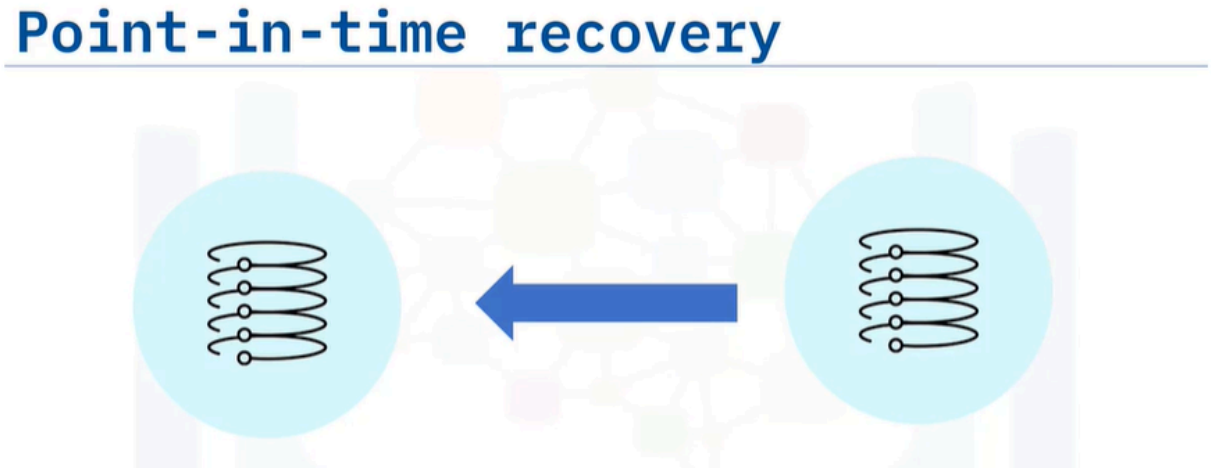


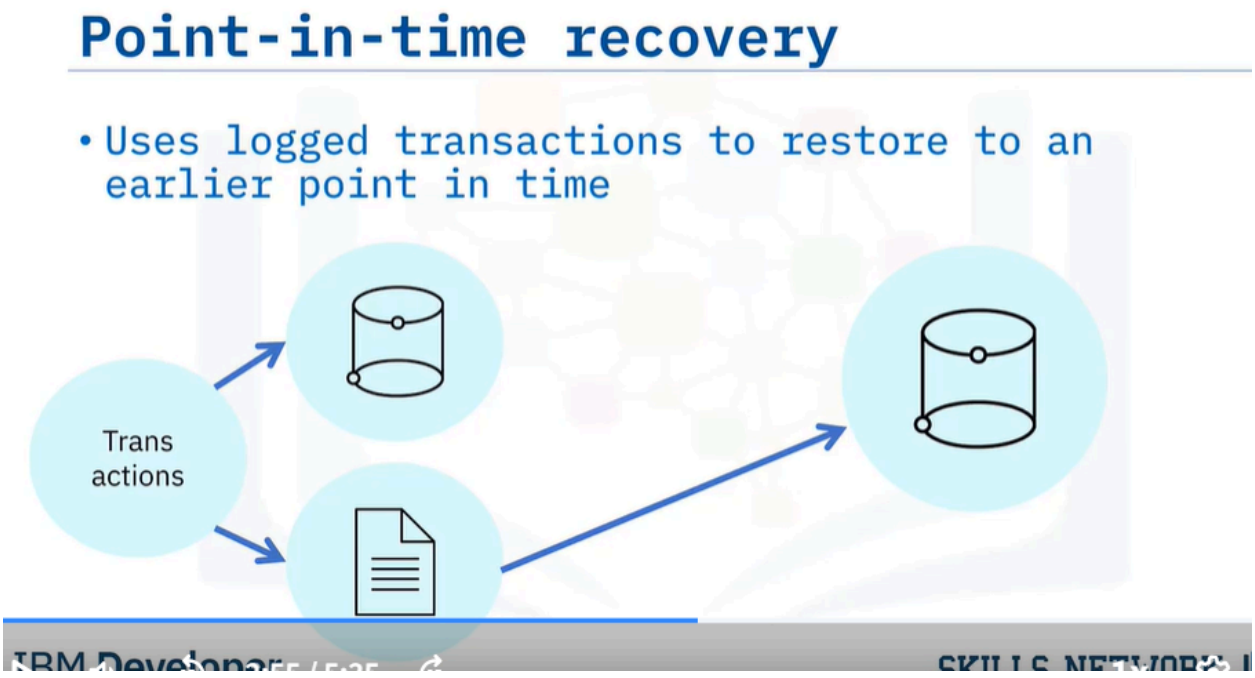
Full backups

- Multiple copies of the backup means storing many instances of a large file
- Only storing one copy risks data loss if file is corrupt
- Could be needlessly backing up unchanging data
- Must secure backup files

- Performing full backups simplifies the restore and recovery processes, because you just need to locate the latest full backup and restore that one file. However, as the size of your database increases, the time, bandwidth, and storage for the backup file also increases.
- And if you decide to keep previous copies of the backup for safety, you will be storing many instances of a large file. However, if you only keep one copy, you must accept the risk that if the backup file is corrupt, you cannot restore your data. Also consider that if only a subset of your data is regularly changing, you

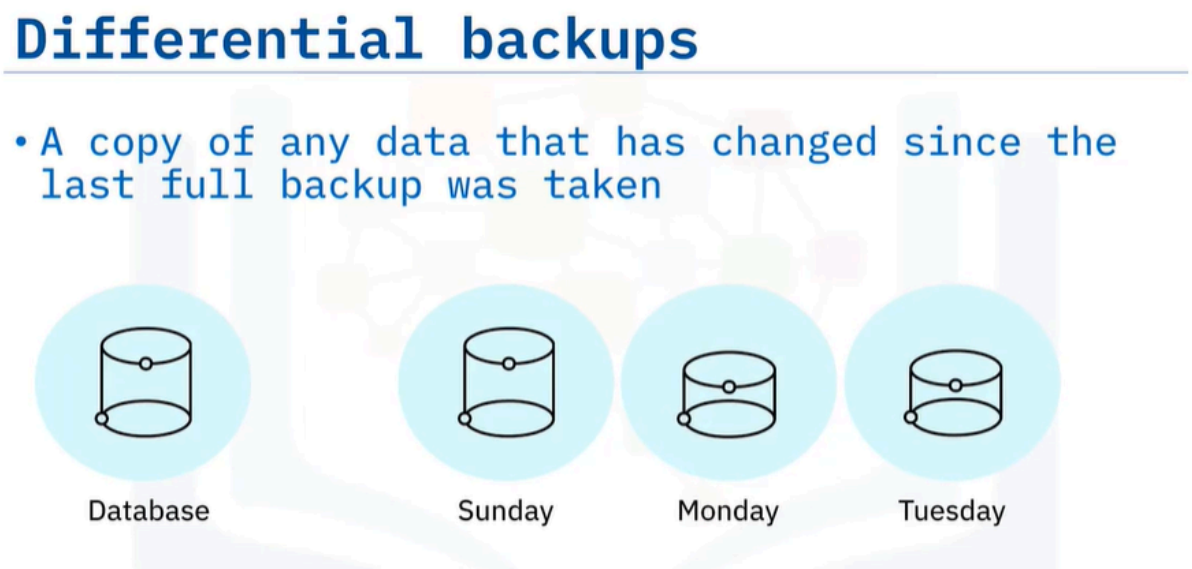
could be needlessly backing up the same data many times. Storing a complete copy of your data outside of the RDBMS means that you must ensure it is adequately secured and that it cannot be accessed by unauthorized users

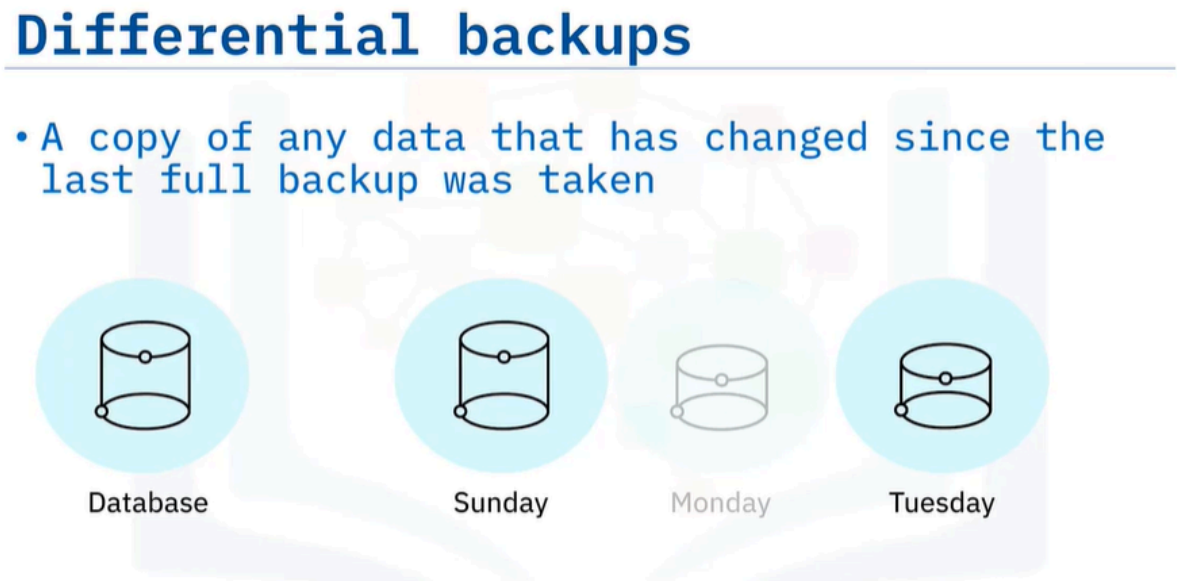




- When you restore a full backup, you restore the data to the state that it was in when the backup was taken. But the database may have processed many transactions since that time, which should ideally be restored too.

- One way around this issue is to enable logging of each transaction on your database and then you can use the information in the log file to reapply the transactions to the restore database. The process of reapplying transactions after restoring a database backup is known as recovery and it enables you to recover the data to a state that it was in at a particular point in time, hence the name, point-in-time recovery. For example, if you know that a DML statement run at 11:05am erroneously deleted some data, you can restore the latest full backup and then reapply the transactions up to that point in time, minimizing the loss of data changes that occurred between the last full backup and the moment that the wrong data was deleted. Different database systems use different terminology for the log containing the transaction information, for example, MySQL calls it the binary log, Postgres calls it the write-ahead log, and Db2 on Cloud calls it the transaction log.

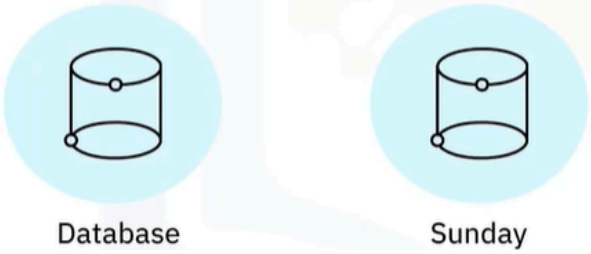




- Because full backups can require a lot of time, bandwidth, and storage, one alternative is to use them in conjunction with differential backups.
- A differential backup consists of a copy of any data that has changed since the last full backup was taken. This makes the differential backup file much smaller than a full backup file, reducing the time, bandwidth, and storage needs for the backup while still enabling you to restore a recent copy of the data. For example, you could perform a full backup once a week on a Sunday and then run a differential backup every day of the week. Each differential backup contains all the changes since the full backup was taken on the Sunday,
- so if you need to restore the database on a Tuesday, you first restore the full backup from Sunday and then restore the differential backup from Tuesday. You do not need to restore Monday's differential backup, because all the changes in that file are also included in the differential backup from Tuesday. Restoring data from full and differential backups will take longer than simply restoring a full backup, but because you restore data less frequently than you back it up, you are likely to save time overall.

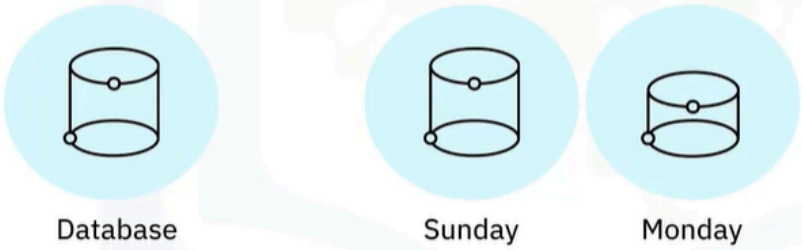
Incremental backups

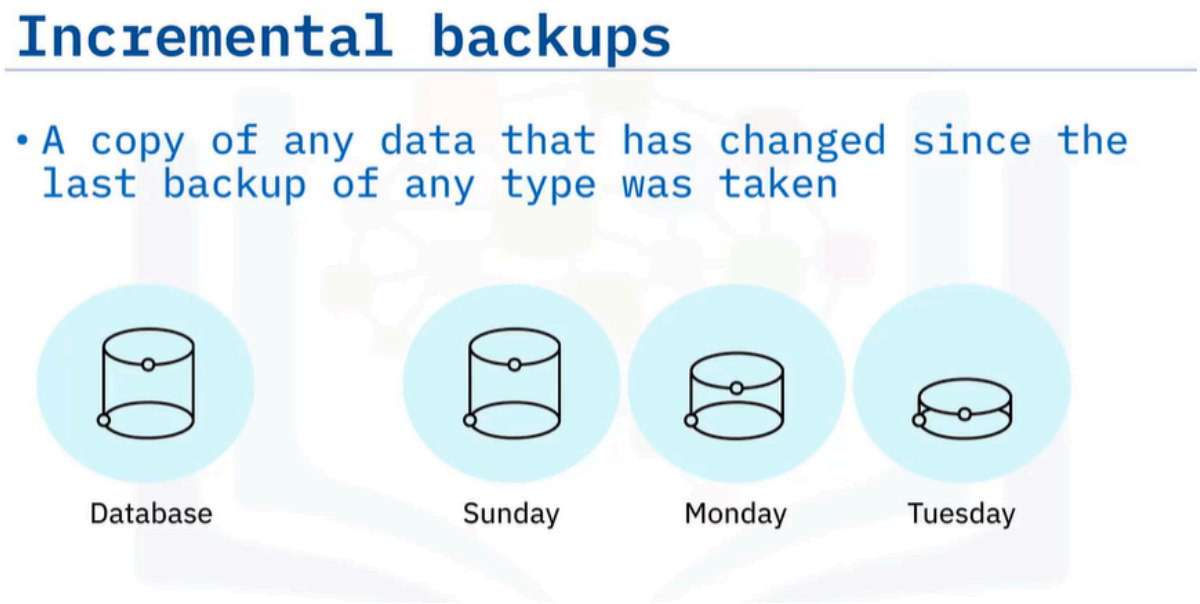
- A copy of any data that has changed since the last backup of any type was taken



Incremental backups

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





Incremental backups are similar to differential backups, but they only contain data that has changed since the last backup of any type was taken. For example, you could perform a full backup once a week on a Sunday and then run an incremental backup every day of the week. Each incremental backup only contains changes since the backup was taken on the previous day, so if you need to restore the database on a Tuesday, you first restore the full backup from Sunday, then the incremental backup from Monday, and then the incremental backup from Tuesday. Restoring data from full and incremental backups will take longer than simply restoring a full backup or restoring differential backups, but the time taken to perform the incremental backups is likely to be less than needed for a differential backup.

Summary

In this video, you learned that:

- Full backups are simple to create and restore, but can be slow to run and result in large files
- Point-in-time recovery provides a more granular recovery model than just using full database backups
- Differential backups are quicker to run than full backups, but the restore process can take longer
- Incremental backups are even quicker to run, but the restore process can take even longer