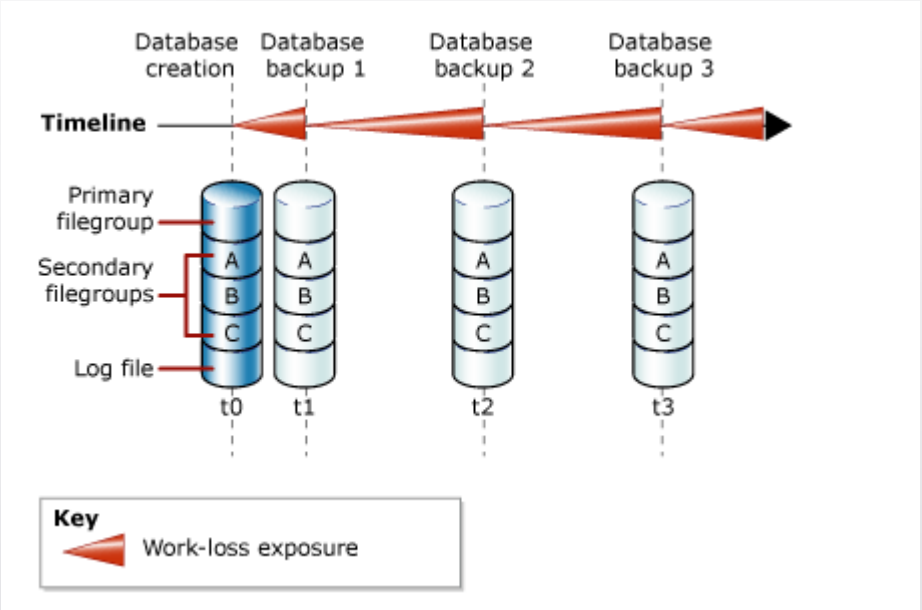
Full Backups

Full Database Backups

- A Full database backup backs up the whole database.
- This includes part of the transaction log so that the full database can be recovered after a full database backup is restored.
- Full database backups represent the database at the time the backup finished.
- It is the foundation of any kind of backup.
- This is a complete copy, which stores all the objects of the database: Tables, procedures, functions, views, indexes etc. Having a full backup.
- you will be able to easily restore a database in exactly the same form as it was at the time of the backup.
- A full backup must be done at east once before any of the other types of backups can be run—this is the foundation for every other kind of backup.

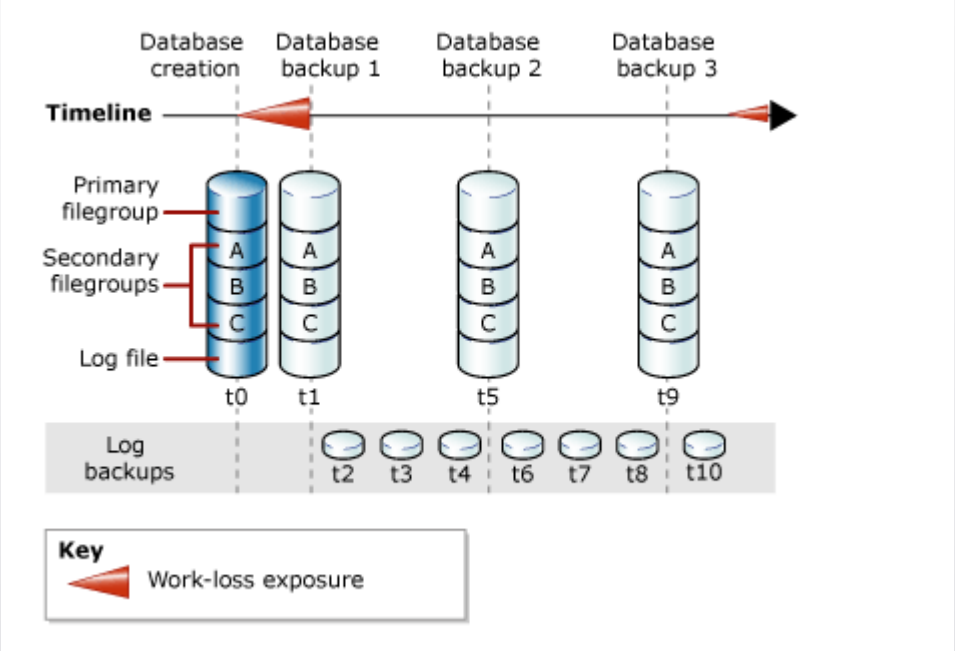
Full Database Backups in Simple Recovery

- Under the simple recovery model, after each backup, the database is exposed to potential work loss if a disaster were to occur.
- The work-loss exposure increases with each update until the next backup.
- when the work-loss exposure returns to zero and a new cycle of work-loss exposure starts.
- Work-loss exposure increases over time between backups.



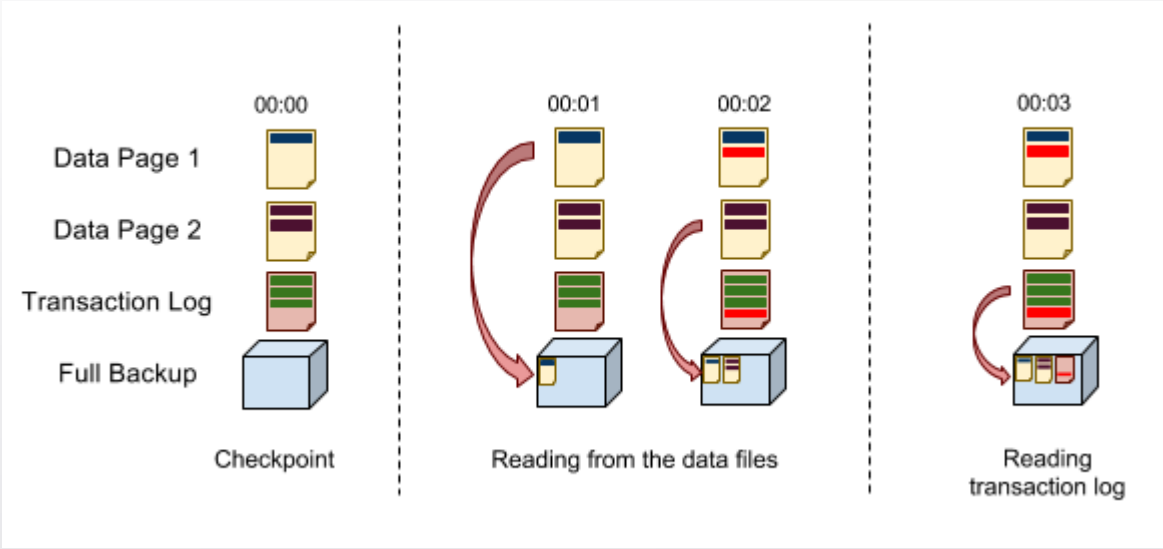
Full Database Backups in Full Recovery

- For databases that use full and bulk-logged recovery, database backups are necessary but not sufficient.
- Transaction log backups are also required.



Full Database Backups in Full Recovery

- A full database backup backs up all data files and active part of the transaction log.
- The active part of the transaction log is necessary to restore database to a transactionally consistent point.
- **00:00** Full backup starts. It forces a database checkpoint and this flushes all dirty pages in the disk. Please note, if you use a simple recovery model the log is truncated, meaning that all committed transactions are removed from the log automatically. In full and bulk-logged recovery models it does not occur.
- **00:01** The backup process reads Data Page 1 and adds it to the backup file.
- **00:02** The backup process reads Data Page 2 and simultaneously some changes in Data Page 1 take place. These changes are added to the Transaction Log as well. Now the backup contains Data Page 2 and the old version of Data Page 1 (as it was at 00:01).
- **00:03** Data reading operation is complete and followed by Transaction Log reading. Transaction Log contains changes made in Data Page 1 (at 00:01) and they are added into the backup to be applied to Data Page 1 later during the restore process recovery stage.



Backup database command and options

- **Full backup :** BACKUP DATABASE AdventureWorks TO DISK = 'C:\AdventureWorks.BAK'
- **File level backup :** BACKUP DATABASE TestBackup FILE = 'TestBackup' TO DISK = 'C:\TestBackup_TestBackup.FIL'
- **File group backup :** BACKUP DATABASE TestBackup FILEGROUP = 'ReadOnly' TO DISK = 'C:\TestBackup_ReadOnly.FLG'
- **Full backup multiple files** : BACKUP DATABASE AdventureWorks TO DISK = 'C:\AdventureWorks_1.BAK',DISK = 'D:\AdventureWorks_2.BAK',DISK = 'E:\AdventureWorks_3.BAK'
- **Full backup with password :** BACKUP DATABASE AdventureWorks TO DISK = 'C:\AdventureWorks.BAK' WITH PASSWORD = 'Q!W@E#R$'
- **Full backup with stats :** BACKUP DATABASE AdventureWorks TO DISK = 'C:\AdventureWorks.BAK' WITH STATS
- **Full backup with description :** BACKUP DATABASE AdventureWorks TO DISK = 'C:\AdventureWorks.BAK' WITH DESCRIPTION = 'Full backup for AdventureWorks'
- **Full backup with mirror :** BACKUP DATABASE AdventureWorks TO DISK = 'C:\AdventureWorks.BAK‘ MIRROR TO DISK = 'D:\AdventureWorks_mirror.BAK' WITH FORMAT