Database Backup and Recovery

- A **backup** is a copy of data from your database that can be used to reconstruct that data. Backups can be divided into physical backups and logical backups.
- **Physical backups** are backups of the physical files used in storing and recovering your database, such as data files, control files, and archived redo logs. Every physical backup is a copy of files storing database information to some other location, whether on disk or some offline storage such as tape.
- **Logical backups** contain logical data (for example, tables or stored procedures) exported from a database management system (DBMS) and stored in a binary file for later re-importing into a database using the corresponding DBMS restore/import utility.
- The Database Administrator (DBA) must always back up data regularly as a protective action. All system elements should be backed up at intervals that depend on how frequently they are updated. For instance, if the database is updated daily, the backup should also be performed daily.
- A DBMS uses a **transaction log** to keep track of all transactions that update the database. The information stored in this log is used by the DBMS for a recovery requirement triggered by a ROLLBACK statement. Some DBMS uses the transaction log to recover a database forward to a currently consistent state.
- While the DBMS executes transactions that modify the database, it also automatically updates the transaction log.
- Keeping and maintaining logs in real time and in a real environment may fill out all the memory space available in the system. As time passes, the log file may grow too big to be handled at all.
- A **checkpoint** is a mechanism where all the previous logs are removed from the system and stored permanently in a storage disk. Checkpoint declares a point before which the DBMS was in a consistent state, and all the transactions were committed.
- Backing up data in a database should include at least the following measures:
  - Some DBMSs include tools to ensure **automatic backup and recovery** of either the whole database or portions of it.
  - **Proper backup identification**. Backups must be clearly identified through detailed descriptions and date information, thus enabling the DBA to ensure that the correct backups are used to recover the database. Nowadays, the most common backup mediums are disk-based backup devices and cloud-based data storage.
  - **Convenient and safe backup storage**. Multiple backups of the same data are required, and each backup copy must be stored in a different location. The storage locations must include sites inside and outside the organization. (Keeping different backups in the same place defeats the purpose of having multiple backups.)
  - **Physical protection of both hardware and software**. Protection might include the use of closed installations with restricted access, as well as preparation of the computer sites to provide air conditioning, backup power, and fire protection.
  - Physical protection also includes a backup computer and DBMS to be used in case of emergency. For example, when Hurricane Sandy hit the east coast of North America in 2012, the U.S. Northeast suffered widespread destruction of its communications infrastructure.

- **Database recovery** restores a database from a given state (usually inconsistent) to a previous consistent state.
  Critical events can cause a database to stop working and compromise the integrity of the data. Examples of critical events are:
- *Hardware/software failures.* A failure of this type could be a hard disk media failure, a bad capacitor on a motherboard, or a failing memory bank. Other causes of errors under this category include application program or operating system errors that cause data

to be overwritten, deleted, or lost. This can be considered as one of the most common sources of database problems.

- Human-caused incidents. This type of event can be categorized as *unintentional* or *intentional*.
  - An *unintentional failure* is caused by a careless end-user. Such errors include deleting the wrong rows from a table, pressing the wrong key on the keyboard, or shutting down the main database server by accident.
  - *Intentional events* are of a more severe nature and normally indicate that the company data is at serious risk. Under this category are security threats caused by hackers trying to gain unauthorized access to data resources and virus attacks caused by disgruntled employees trying to compromise the database operation and damage the company.
- The recovery technique that must be selected is dependent on the extent of the damage that has occurred to the database. There are two cases that can be considered and are described below:

1. If the database has been extensively damaged: For example, a disk-based database crash has occurred and destroyed the entire database.
   - In this event, it is necessary to restore the last backup copy of the database and reapply the update operations of the committed transactions using the log file. This assumes that the log file has not been damaged as well.
2. If the database has not been physically damaged but has become inconsistent: For example, the system crashed while transactions were executing.
   - In this situation, it is necessary to undo the changes that caused the inconsistency and to redo some transactions to ensure that the updates performed successfully.

- There are two (2) common techniques that can be used if the database enters an inconsistent state. These are the *deferred update* and *immediate update*.
- When the recovery procedure uses a *deferred-write technique* (also called a deferred update), the transaction operations do not immediately update the physical database. Instead, only the transaction log is updated. The database is physically updated only with data from committed transactions, using information from the transaction log. If the transaction aborts before it reaches its commit point, no changes (no ROLLBACK or undo) need to be made to the database because it was never updated.
- When the recovery procedure uses a *write-through technique* (also called an immediate update), the database is immediately updated by transaction operations during the transaction’s execution, even before the transaction reaches its commit point. If the transaction aborts before it reaches its commit point, a ROLLBACK or undo operation needs to be done to restore the database to a consistent state. In that case, the ROLLBACK operation will use the transaction log “before” values.

REFERENCES

Coronel, C. and Morris, S. (2018). *Database systems design, implementation, & management (13th ed.)*. Cengage Learning.

Elmasri, R. & Navathe, S. (2016). *Fundamentals of Database systems (7th ed.)*. Pearson Higher Education.

Kroenke, D. & Auer, D. *Database processing: Fundamentals, design, and implementation (12th ed.)*. Pearson Higher Education.

Silberschatz A., Korth H.F., & Sudarshan, S. (2019). *Database system concepts (7th ed.).* McGraw-Hill Education.