1. Batch vs Script vs Transaction vs Backup

Batch

A collection of SQL statements or jobs executed together as a single unit, typically scheduled to run automatically at specific times (e.g., nightly data processing, bulk updates). Focuses on *grouping operations* for efficiency.

Script

A file containing SQL commands or programming code that can be executed to perform database tasks. Scripts are *reusable and portable*, used for database setup, migrations, or repetitive tasks.

Transaction

A logical unit of work containing one or more database operations that must succeed or fail as a whole (ACID properties). Uses `BEGIN`, `COMMIT`, and `ROLLBACK` to ensure *data consistency and integrity*.

Backup

A copy of database data and structure saved to external storage for disaster recovery and data protection. Types include full, incremental, and differential backups. Focuses on *data preservation*.

2. Transaction Logging

Transaction logging is the process of recording all changes made to a database in a log file (transaction log or write-ahead log).

Purpose:

- **Recovery:** Restore database to consistent state after crashes by replaying or undoing logged operations
- **Durability:** Ensures committed transactions persist even if system fails (ACID's Durability)
- **Rollback:** Allows reverting uncommitted transactions
- **Replication:** Enables data synchronization across servers
- **Point-in-time recovery:** Restore database to specific moment

How it works: Before modifying data, changes are written to the log. If a failure occurs, the database uses the log to redo committed transactions and undo incomplete ones.

3. Soft Delete vs Hard Delete

Hard Delete

Permanently removes records from the database using `DELETE` or `TRUNCATE` commands. Data is irretrievable unless restored from backup.

Advantages: Saves storage space, complies with data retention policies, maintains cleaner database

Disadvantages: Irreversible, potential for accidental data loss, breaks referential integrity

Soft Delete

Marks records as deleted (e.g., setting `is_deleted = TRUE` or `deleted_at = timestamp`) without physically removing them. Records remain in database but are excluded from queries.

Advantages: Recoverable, maintains audit trail, preserves referential integrity, supports data analysis

Disadvantages: Consumes storage, requires modified queries (`WHERE is_deleted = FALSE`), potential privacy concerns

Common Implementation:

```
ALTER TABLE users ADD COLUMN deleted_at TIMESTAMP NULL;
UPDATE users SET deleted_at = NOW() WHERE id = 123;
SELECT * FROM users WHERE deleted_at IS NULL;
```