Scenario

Our university's enrollment database was processing transactions when a sudden power outage occurred. Some student payments were saved, some were not. The system shut down unexpectedly.

list the Possible causes of a database crash(e.g., hardware failure, software bug, network outage, human error).

Hardware Failure

Storage problems

File System Corruption

software bug

Immediate steps you think the IT team should take to recover the data.

Isolate the System: Immediately cut off all access to the affected database to prevent further data corruption.

- Analyze the Crash: Review system and transaction logs to pinpoint the cause of the crash and the extent of the damage.
- Restore from Backup: Restore the database to its last known good state using the most recent full backup.
- Apply Transaction Logs: Use transaction logs to redo all committed transactions that occurred after the backup, bringing the database up to the point of failure.
- Rollback Incomplete Transactions: Undo any transactions that were in progress when the crash happened to ensure data consistency.
- Verify Integrity: Run checks and tests to confirm that all data is intact and the database is functioning correctly before allowing users to reconnect.

Which suggested recovery steps were closest to real DBMS techniques?

- Checking Transaction Logs: This is the core of DBMS recovery. The system uses redo and undo logs to determine which transactions were complete and which were not.
- Rolling Back Incomplete Transactions: This technique ensures Atomicity. The DBMS automatically reverses any partial transactions that didn't finish before the crash, as if they never happened.
- Redoing Committed Transactions: This ensures Durability. The DBMS re-applies completed transactions from the logs if their changes weren't saved to the main database files before the system failed.
- Restoring from Backups + Applying Logs: This combines a backup with transaction logs to restore the database to its most recent, consistent state, preventing data loss from transactions completed after the last backup.

Why are regular backups alone not always enough?

- Backups can be outdated – if the database crashes after the last backup, all new data since then is lost.,
- They don't capture in-progress transactions – only transaction logs can record and replay those.,
- Backups can be corrupted or incomplete – relying only on them is risky if they fail.,
- Recovery can take longer – restoring a full backup without logs may delay system availability.

How does the concept of ACID properties (especially Durability and Atomicity) relate to what we discussed?

- Atomicity – Ensures a transaction is all-or-nothing. In the crash scenario, payments that weren't finished must be completely rolled back, so no "half-saved" transactions remain.,
- Durability – Guarantees that once a transaction is committed, it stays saved even after a crash. The database uses transaction logs and recovery processes to redo those committed changes if needed.