WEEK 10

TRANSACTION – MOTIVATION FOR THE NEED FOR TRANSACTIONS

STUDENT OBJECTIVES

- Upon completion of this video, you should be able to:
  - Give 2 scenarios where transactions are required
  - Define the term *transaction*
  - Give the meaning of each of the 4 properties in acronym ACID
  - Give an example of at least 3 types of things that could happen that would cause problems
  - List the 3 SQL commands used to build transactions
  - Explain why logs are necessary in database management systems

IMAGINE:

- **Situation 1**: Homer moving 50 dollars from his savings account to his checking account where:
  - *Get the account balance of the savings account*
  - *If the Homers&Marge's savings account has more than 50 dollars in the savings account, subtract the 50 bucks from the savings.*
  - *Add the 50 bucks to the checking balance.*
- **Situation 2**: Marge is going to deduct 75 dollars from the SAME savings account as Homer
  - *Get the account balance of the savings account*
  - *If the Homers&Marge's savings account has more than 75 dollars in the savings account subtract the 75 bucks from the savings.*

2 POSSIBLE PROBLEMS:

QUESTION: Just consider problem 1 on it's own. What could go wrong with just problem 1?

UPDATE jointaccount SET balance = balance - 50 WHERE cusID = "homer123" AND type="saving";

UPDATE jointaccount SET balance = balance + 50 WHERE cusID="homer123" AND type="chequing";

QUESTION: Now consider problem 2 with 2 different things happening at the same time. What could go wrong?
Assume account has 100 dollars.

Homer does:
SELECT balance FROM jointaccount WHERE cusID = "homer123";
UPDATE jointaccount SET balance = balance - 50 WHERE cusID="homer123";

Marge does:
SELECT balance FROM jointaccount WHERE cusID = "marg123";
UPDATE jointaccount SET balance = balance - 75 WHERE cusID = "marg123";

WHAT IS A TRANSACTION?

- A transaction is a sequence of database operations, where the execution of the operations preserves the consistency of the database (a Logical Unit of Work)

SOLUTION: TRANSACTIONS:

- **TRANSACTION 1**: A Transaction for Homer moving 50 dollars from his savings account to his checking account where:
  - **Begin Transaction**
    - Get the account balance of the savings account
    - If the Homers&Marge's savings account has more than 50 dollars in the savings account, subtract the 50 bucks from the savings.
    - Add the 50 bucks to the checking balance.
  - **Commit** ← take both action works.
- **TRANSACTION 2**: Another transaction for Marge, she is going to deduct 75 dollars from the SAME savings account as Homer
  - **Begin Transaction**
    - Get the account balance of the savings account
    - If the Homers&Marge's savings account has more than 75 dollars in the savings account subtract the 75 bucks from the savings.
  - **Commit**

PROBLEMS THAT CAN OCCUR DURING THE MIDDLE OF A TRANSACTION:

1. **System Crash:** Example --> Main Memory Failure
2. **Transaction or System Error:** Example → Division by zero
3. **Local/Exception Errors:** Example → Not being able to access or find data
4. **Concurrency control enforcement:** Example → Concurrency may abort a transaction and restart it later

two action happen together

5. **Disk Failure:** Example → Read/Write Head Crash
6. **Catastrophes:** Example → air conditioning failure, fire, theft, overwriting disks by mistake

THE ACID TEST

- The transaction must take the database from a **consistent state** to another **consistent state,** thus the initial state before the *Begin Transaction* of transaction 1 is a **consistent state,** and after the *Commit* it is a consistent state but in-between that the database is in an **inconsistent state** because it violates integrity and semantic rules.
- The DBMS must be able to recover the database to a previous consistent state if for some reason some happens in the middle of the transaction.

- **EVERY TRANSACTION MUST PASS THE ACID TEST:**
  - **A**tomic: All or nothing: All parts of the transaction MUST be executed, thus in the above example, all steps are thought of as 1 atomic transaction where either all are completed or none are completed
  - **C**onsistency: A transaction must transform the database from one consistent state to another consistent state.
  - **I**solation: Data used during the execution of the a transaction cannot be used by a second transaction until the first one is completed (runs as if in isolation)
  - **D**urability: When a transaction is completed, the database has reached a consistent state and that state cannot be lost, even if there is a system failure. The changes made by the transaction are durable, i.e. will survive certain types of systems crashes.

- Transactions are supported in SQL with the use of 3 commands:
  - **Begin Transaction** (start). the commit state is also the starting point for the next action.
  - **Commit**
  - **Rollback (also called Abort)**

```
BEGIN
    DECLARE '_rollback' BOOL DEFAULT 0;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET '_rollback' = 1;
    START TRANSACTION;
    INSERT INTO SAIDAS(data, hora) VALUES(CURDATE("yyyy-MM-dd"),CURTIME("hh:mm:ss"));
    UPDATE INTO ENTRADAS(sai) VALUES(@sai);

    IF '_rollback' THEN
        SET retcode = 0;   ← roll back if error
        ROLLBACK;
    ELSE
        SET retcode = 1;   ← commit if no error.
        COMMIT;
    END IF;
END$$
```

TRANSACTION LOGS

- DBMS use transaction logs to keep track of all updates on the system. The log file is used for recovery of a consistent state if there is an abnormal termination or crash
- *Example Log File*:

TID	Table	TupleID	Attribute	Before Value	After Value
101	*** Begin Transaction ***				
101	Inventory	645	On_Hand	243	109
101	AccRec	4324542	Balance	1200	4500
101	*** End Transaction: Committed				
102	*** Begin Transaction ***				
102	Inventory	645	On_Hand	109	43
102	AccRec	323453	Balance	4000	5600

we cannot roll back before this point

- QUESTION: Why would you want to have your log file on a different disk than your data file?

  ANSWER: Need the log to recreate the data, if the disk is gone and both the log and data are on the same disk, then you lose both!

- QUESTION: Why would you want to have your log file repeated on more than 1 disk?

  ANSWER: Need to have a backup of your log file.