Databases

Lesson 06

Transaction Models, Query Processing and Data Recovery

Transaction

- Means execution of interrelated instructions in a sequence for a specific operation on a database
- Database transaction models must maintain data integrity and must enforce a set of rules called ACID rules

ACID Rules

1. Atomicity
2. Consistency
3. Isolation
4. Durability

1. Atomicity

- All operations of a transaction must be complete
- In case, a transaction cannot be completed; it must be undone (rolled back)
- Operations in a transaction are assumed to be one indivisible unit (atomic unit)

2. Consistency

- A transaction must be such that it preserves the integrity constraints and follows the declared consistency rules for a given database
- Consistency means the data is not in a contradictory state after the transaction

Consistency

- The amount transferred must be subtracted from account *A* and added into account *B*
- Consistency means that the sum total of the balances in accounts *A* and *B* is the same as it was before the transaction

3. Isolation

- If two transactions are carried out simultaneously, there should not be any interference between the two
- Further, any intermediate results in a transaction should be invisible to any other transaction

4. Durability

- After a transaction is completed, it must persist and cannot be aborted or discarded
- For example, in a transaction entailing transfer of a balance from account *A* to account *B*, once the transfer is completed and finished there should be no roll back

ADO.NET (ActiveX Data Objects in .NET)

- BeginTransaction: It is used to begin a transaction.
- Any operation after BeginTransaction is assumed to be a part of the transaction till the CommitTransaction command or the RollbackTransaction command

Auto-commit mode

- Means that the transaction is finished automatically even if an error occurs in between
- set autocommit = 1

Query Processing

- During a transaction with a database, queries sent to read and get the records from the database
- Contacts and SavedNumbers
- Contacts stores the rows of records consisting of first character (firstChar) of name, contact-name (cName), and contact telephone number (cTelNum)

Query Processing

- Querying of Record in Contacts by firstChar, cName, or cTelNum
- DialledNumbers stores the rows of records consisting of dialling sequence number (seqNum), time of call (cTime), and dialled telephone number (dTelNum). A record in DialledNumbers can be searched by seqNum, cTime, or dTelNum

SQL Query

- SELECT cName, cTelNum FROM Contacts, DialledNumbers WHERE Contacts.firstChar = “R” AND Contacts.cTelNum = DialledNumbers.dTelNum

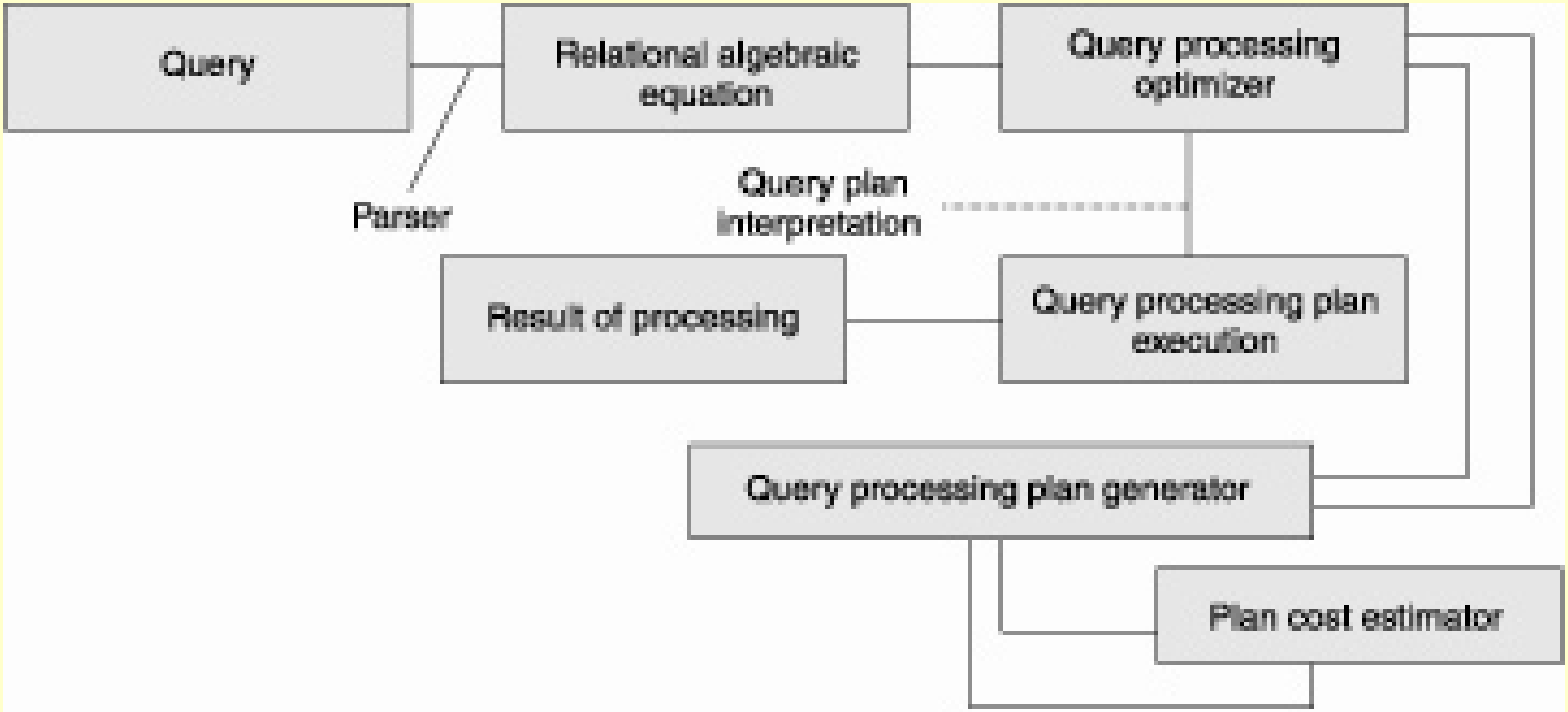
Query processing

- Efficient processing of queries needs optimization of steps for query processing
- Query processing means making a correct as well as efficient execution strategy by *query decomposition* and *query-optimization*. A relational-algebraic equation defines a set of operations needed during query processing

Queries optimization

- Based on cost (number of micro-operations in processing) by evaluating the costs of sets of equivalent expressions
- Based on a heuristic approach consisting of the following steps: perform the selection steps and projection steps as early as possible and eliminate duplicate operations

Query processing architecture



Number of reasons warranting database recovery

- Media failure
- System failure
- Transaction abortion
- Data destruction due to intentional external attack or due to unintentional (due to careless handling) user carelessness

Number of reasons warranting database recovery

- Data may also be destroyed due to destruction of the physical media hoarding the data
- Logical program errors and a transaction may not materialize
- Finally, there may be loss of main memory due to system errors (hardware or software)

Non-recoverable Data

- In case of media failure, intentional attack on the database and transactions logging data, or physical media destruction
- However, data recovery possible in other cases

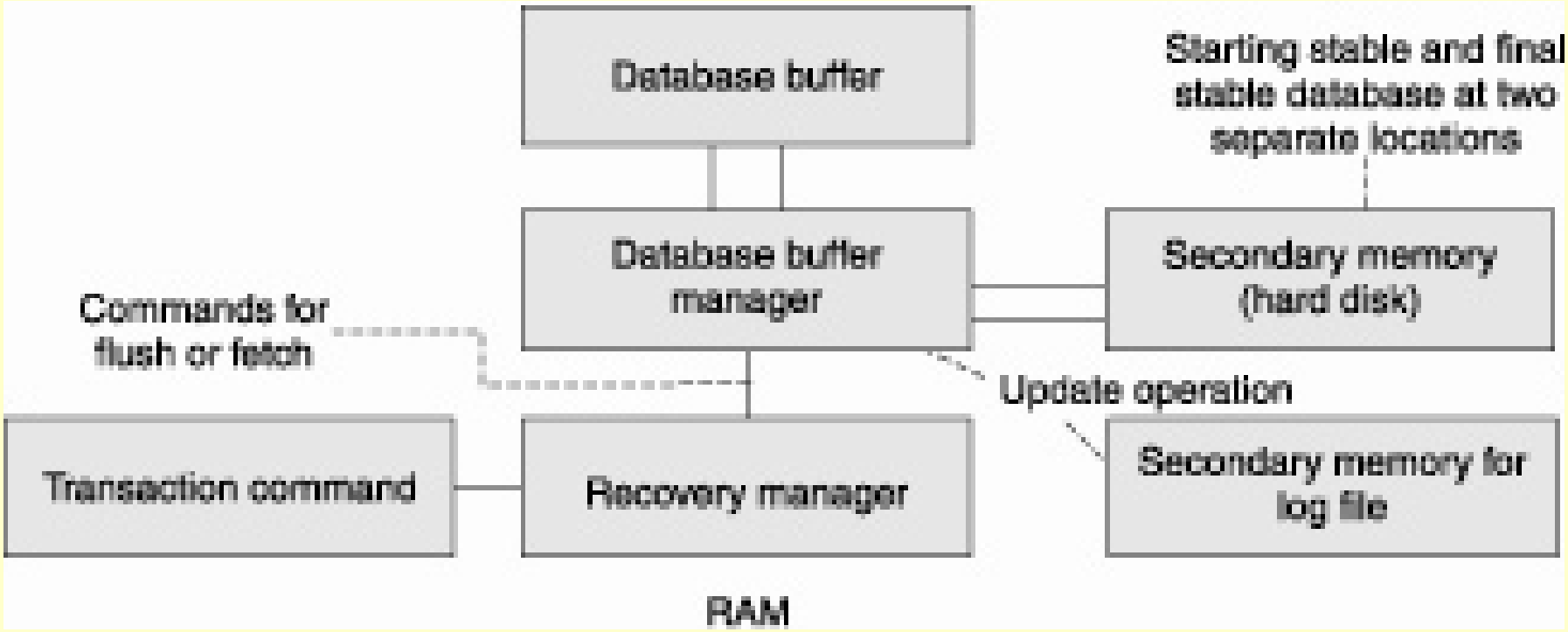
Example

- Assume that transactions started at time t0 and system crash or failure occurs at *t*0 + *T*.
- Assuming that transactions *T*0 to *Tn*–1 are required to be completed in sequence *T*0, *T*1, *T*2 , ..., *Tn*–1, the following cases are possible

Example

- Case 1: Last transactions incomplete
- Case 2: Initial and Last transactions incomplete

Recovery Management architecture



Recovery Manager

- Recovers or aborts a transaction using the logged entries

Recovery manager log file

- Each instruction for a transaction for update (insertion, deletion, replacement, and addition) logged.
- Database read instructions are not logged
- Log files stored at a different storage medium
- Log entries flushed out after the final stable state database is stored

Logged entry Fields

- Transaction type (begin, commit, or rollback transaction)
- Transaction ID
- Operation-type
- Object on which the operation performed
- Pre-operation and post-operation values of the object

Check Point based Recovery

- Uses the checkpoints for operations on the data during a set of transactions
- Recovery always made by back-scanning the logged records
- A checkpoint-based data recovery procedure defines the stage, up to which the back-scanning of logged operations in the secondary storage is to be done

Recovery Models

- Full recovery model
- Bulk logged recovery model
- Simple recovery model

Summary

- Atomicity in transactions
- Consistency in transactions
- Isolation in transactions
- Durability in Transactions
- Query
- Query processing
- Query Optimization

...

… Summary

- Data recovery Model
- Recovery manager
- Check Points
- Logged Fields help in recovery

End of Lesson 06

Transaction Models, Query Processing and Data Recovery