1. **What is the goal/use of query processing? (5 points)**

Activities involved in retrieving data from the

database.

**Aims of QP:**

– transform query written in high-level language

(e.g. SQL), into correct and efficient execution

strategy expressed in low-level language

(implementing RA);

– execute strategy to retrieve required data.

1. **Explain what a transaction is? Why transaction is important in DBMS? What problems do they solve? (10 points)**

**Transaction**: Action, or series of actions, carried out by user or

application, which reads or updates contents of

database.  
**Benefits:**

The primary benefit of using transactions is data integrity.

Many database uses require storing data to multiple tables, or multiple rows to the same table in order to maintain a consistent data set.

A secondary benefit of using transactions is speed.

There is often an overhead associated with actually committing the data to the database. If you've got 1000 rows to insert, committing after every row can cause quite a performance hit compared to committing once after all the inserts.

1. **Transaction ACID properties describe each one (16 points total)**

**Atomicity:** ‘All or nothing’ property.

**Consistency:** Must transform database from one consistent

state to another.

**Isolation:** Partial effects of incomplete transactions

should not be visible to other transactions.

**Durability:** Effects of a committed transaction are

permanent and must not be lost because of later failure.

1. **Issues that can occur when concurrent access to the database is allowed** in a multiuser environment (16 points total)

Three examples of potential problems caused by

concurrency:

– Lost update problem.

Successfully completed update is overridden by

another user.

􀂋 T1 withdrawing £10 from an account with balx,

initially £100.

􀂋 T2 depositing £100 into same account.

􀂋 Serially, final balance would be £190.

􀂋 Loss of T2’s update avoided by preventing T1

from reading balx until after update.

– Uncommitted dependency problem.

Occurs when one transaction can see

intermediate results of another transaction

before it has committed.

􀂋 T4 updates balx to £200 but it aborts, so balx

should be back at original value of £100.

􀂋 T3 has read new value of balx (£200) and uses

value as basis of £10 reduction, giving a new

balance of £190, instead of £90.

􀂋 Problem avoided by preventing T3 from

reading balx until after T4 commits or aborts.

– Inconsistent analysis problem.

Occurs when transaction reads several values

but second transaction updates some of them

during execution of first.

􀂋 Sometimes referred to as dirty read or

unrepeatable read.

􀂋 T6 is totaling balances of account x (£100),

account y (£50), and account z (£25).

􀂋 Meantime, T5 has transferred £10 from balx to

balz, so T6 now has wrong result (£10 too high).

1. **When transaction x requests data currently hold by transaction y, x is allowed to wait only if it has a timestamp smaller than the timestamp of y (x is older than y). Otherwise x is rolled back (dies) so this is: (5 points total)**
2. **Normalization:**

**Explain the concept of normalization?**

Normalization is a technique for producing a

set of suitable relations that support the data

requirements of an enterprise.

**What does first normal form do?**

A relation in which the intersection of each row

and column contains one and only one value.

**What does second normal form do?**

A relation that is in 1NF and every nonprimary-

key attribute is fully functionally

dependent on the primary key.

**What does third normal form do?**

A relation that is in 1NF and 2NF and in which

no non-primary-key attribute is transitively

dependent on the primary key.

**Materialization vs pipelining Explain with examples (14 points)**

**Materialization**

* Materialized evaluation walks the parse or expression tree of the relational algebra operation, and performs the innermost or leaf-level operations first
* The intermediate result of each operation is materialized — an actual, but temporary, relation —and becomes input for subsequent operations.
* The cost of materialization is the sum of the individual operations plus the cost of writing the intermediate results to disk — a function of the blocking factor (number of records per block) of the temporaries.
* The problem with materialization is that — lots of temporary files, lots of I/O.

**Pipelining**

* With pipelined evaluation, operations form a queue, and results are passed from one operation to another as they are calculated, hence the technique’s name.
* With pipelined evaluation, operations form a queue, and results are passed from one operation to another as they are calculated, hence the technique’s name.
* General approach: restructure the individual operation algorithms so that they take streams of tuples as both input and output.
* Limitation: General approach: restructure the individual operation algorithms so that they take streams of tuples as both input and output.
* So, for instance, algorithms that require sorting can only use pipelining if the input is already sorted beforehand, since sorting by nature cannot be performed until all tuples to be sorted are known.

**Assignments Solutions**

**Assignment 8 – Week 12**

------------------------------------------------------------------------------------------------------------------------------------------

1. What is meant by granularity? Give examples.  
   ANS:

granularity is the size of data items chosen as unit of protection by concurrency control protocol.

1. Discuss the types of failure that may occur in a database environment. Explain why it is important for a multi-user DBMS to provide a recovery mechanism.

ANS:

**System crashes:** they can be due to hardware or software errors, resulting

in loss of main memory.

**Media failures:** such as head crashes or unreadable media resulting in loss

of parts of secondary storage.

**Application software errors:** logical errors in the program that is accessing the database

causing one or more transactions to fail.

**Natural physical disasters** such as floods, fires or power failures

**Carelessness** or unintentional destruction of data, hardware or software facilities by operators

or users.

**Sabotage**, intentional corruption or destruction of data, hardware, or software facilities.

in a multi-user DBMS to provide a recovery mechanism since there is high risk of

one of the users to be careless or intentionally destruct data and every single user would

therefore, loose data.

1. Discuss how the log file (or journal) is a fundamental feature in any recovery mechanism. Explain what is meant by forward and backward recovery and describe how the log file is used in forward and backward recovery.

ANS:

A log file or journal is fundamental feature in any recovery mechanism since it keeps

information about all updates to the database. Each DBMS maintains one. The log file contains

data on transaction records and checkpoint records. It has the before and after images of the

database.

Forward recovery can be explained as a process to redo changes using after-images of the

database.

Backward recovery can be explained as a process to undo changes using before-images of the

database.

1. What is the significance of the write-ahead log protocol? How do checkpoints affect the recovery protocol?

ANS:

The write-ahead protocol states that it is essential that log records are written before the

corresponding write to the database.

A checkpoint is a point of synchronization between the database and the transaction log file. It

limits the amount of searching and subsequent processing that we need to carry out on the log

file, we can checkpoint.

1. Compare and contrast the deferred update and immediate update recovery protocols.

ANS:

**Deferred Update**

Since transactions do not affect the database

until they are committed, there is no need for

undo of any transaction.

Since all locks are released only after commit, no

transaction can read data elements that are

being modified by another uncommitted

transaction, hence, no possibility of cascading

rollbacks.

Heavy disk operation during commits the undo operations are performed in the reverse

order to which they were written to the log.

**Immediate Update**

Since updated are applied to the database as

they occur without waiting to reach the commit

point, there is a need for a redo of committed

updates following transaction failure.

For any transaction for which the log contains a

transaction start record but not a transaction

commit record, we need to undo that transaction.

The undo operations are performed in the reverse

order to which they were written to the log.