**Question 1**

**Explain why you would not recommend creating indexes on every attribute in a table?**

Don't use indexes in all the tables because it will slow down the searches . Cause index will only get bigger as the search goes on.

**Explain why a Query processor/optimiser may not opt to use an index on a column even though it is available? Mention the role of Statistics**

SQL for creating an index

CREATE INDEX index\_name

ON table\_name (column\_name)

Create index of non key column and see which different queries involving that column actually use the index if any.

When testing to see if the index was being used by certain queries we chose a course code that had a very high repeition factor and one with a very low repeition factor

**Question 2**

**Explain the term multi-threading.**

a technique by which a single set of code can be used by several processors at different stages of execution.

**Explain how multi-threading is of significance to DBMS architecture e.g. load balancing, availability.**

**Question 3**

**Write the Undo/Redo recovery procedure that would be able to process this type of**

**transaction schedule for recovery.**

**Recovery Procedure (Undo/Redo)** (general recovery procedure; buffers independent, ACC)

Using the above example: at restart time the following recovery procedure is used.

1. retrieve the last checkpoint record

2. create two lists called UNDO and REDO. Set UNDO equal to the list of all transactions given in the checkpoint record.

3. search through the log from the checkpoint time forward.

4. If a BEGIN TRANSACTION log entry is found for transaction T, add T to the UNDO list.

5. If a COMMIT entry is found for transaction T, move T from UNDO to REDO.

Note : ROLLBACK and implications.

6. When the end of the log is reached both list identify the transactions that require partial changes to be undone and unpropagated modifications for committed transactions(i.e. not logged) to be redone.

In the above example T3 and T5 must be undone. T2 and T4 redone.

O

UNDO: rectifies undesirable changes made by a transaction. Requires Before\_value in log

REDO: ensures that changes made by a transaction are correctly recorded by redoing the action(s). Requires the After\_value in the log.