About the examination

- The examination is a closed book test, but allows taking one A4 page note (handwritten).

- Write relational algebra expressions and SQL statements (queries, assertions, triggers, authorizations, transactions etc.) to meet different requests.
- Create tables with different constraints. Inserting, deleting, and updating tuples are also requested.
- Understand the implications of candidate key, primary key, and foreign key.

- Draw E-R diagram in terms of different requests and convert E-R diagram to relations.

- Use Armstrong’s Axioms to prove a FD.
- Calculate F^+, α^+, F_C.
- List candidate keys. Test whether a relation is in BCNF or 3NF, and decompose it if necessary. Test whether a decomposition is a lossless-join decomposition and dependency preserving.

- Write XML files to store specific information.
- Write DTD/XMLSchema file that a specific XML file conforms. How to define the constraints, e.g., the min and max numbers of occurrences.
- Use Xpath/Xquery (FLWOR) to query XML files, and evaluate the results of Xpath/Xquery expressions.

- How to implement variable-length records.
- Estimate the number of blocks to store data and index (dense and sparse) files.
- Give the structure of a file after inserting and deleting a record.
- Determine the data in the buffer after a series of operations, and determine the pages to be removed under a specific policy (as well as the actions to take).

- The definition and requirements of B$^+$-trees.
- In a disk based implementation, calculate the maximum number of pointers in a node/block (4 Bytes are usually used to store a point).
- Calculate the height and the number of nodes of a B$^+$-tree (half-filled and full-filled).
- The updates on a B$^+$-tree after inserting or deleting data items.
- The query process of B$^+$-tree.

- The algorithms to implement select, external sorting, and join operations (including nested-loop join, block nested-loop join, indexed nested-loop join, merge join, and hash join).
- The methods to estimate the cost of these algorithms (under worst, best, and fixed buffer size cases).

- Equivalence rules.
- Estimate the size of intermediate results, as well as the number of distinct values.
- Statistical information based query optimization.
- Steps of heuristic optimization.
- Optimizing nested subqueries.
- How to maintain a materialized view.
- Materialized view and index selection.

- ACID features of transaction.
- Conflict serializability and the method to justify (precedence graph).
- Recoverable and cascadeless schedule.
- 2PL.
- Multiple granularity locking scheme.
- Implementation of locking.
- Deadlock detection (wait-for graph, lock table)

- The processes of log based crash recovery in serial and concurrent cases.
- Determine the values of data items after recovery.
- The sequence of logs generated during the recovery procedure (ARIES).
- If a system crashes during a recovery process, how to handle.
- Recovery with early lock release.