University of Glasgow

Friday 11 May 2018
9.30 am – 11.30 am
Duration: 2 Hours

DEGREES of MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

Database Systems H

(Answer All Questions)

This examination paper is worth a total of 120 marks

The use of a calculator is not permitted in this examination

INSTRUCTIONS TO INVIGILATORS

Please collect all exam question papers and exam answer scripts and retain for school to collect. Candidates must not remove exam question papers.

Note: The Exam Duration is Two (2) HOURS

Part A: Relational Modelling and SQL [Total: 40 Marks]

1. **(a)** Provide a definition of the foreign key attribute in a relation.

[5]

(b) Can the foreign key be `NULL`? Explain your answer with an example.

[5]

2. Consider the relation **Student**(<u>SSN</u>, SchoolID, GPA), where SSN is the Social Security Number (SSN) of the student, which is the Primary Key in the relation, GPA is the Grade Point Average, and SchoolID is the ID of the school.

(a) Write a SQL query that lists the school IDs of the schools having more than 50 students using the GROUP BY operator and the HAVING condition.

[5]

(b) Based on the SQL query in Question **2.(a)**, count the total number of students whose GPA is greater than 10, in schools having more than 50 students.

[5]

3. **(a)** Provide a definition and an example of the Functional Dependency (FD): $X \rightarrow Y$ between the attributes X and Y.

[5]

(b) Assume the relation **R**(A, B, C, D) with four attributes {A, B, C, D}. Consider the set of Functional Dependencies (FDs), which are the only dependencies that hold true in the relation **R:**

FDs: $C \rightarrow D$, $C \rightarrow A$, $B \rightarrow C$

(i) Identify the candidate key(s) for the relation **R**; Explain your answer.

[5]

(ii) Identify the **highest** Normal Form (NF) that relation **R** satisfies, i.e., one of: 1NF, 2NF, 3NF, or BCNF; Explain your answer.

[5]

(iii) If relation **R** is not in BCNF, decompose **R** into a set of BCNF relations; Explain your answer.

[5]

Part B: File Organization and Indexing Methods [40 Marks]

4. Assume the relation **Employee**(ID, Name, Age), which is stored in a file on a disk. We need 40 bytes for the integer Primary Key attribute ID, 59 bytes for the Name attribute and 1 byte for the Age attribute. Consider that the relation has r = 1000 tuples and the size of the file block is fixed 512 bytes.

(a) Given that the Database system adopts fixed-length records, such that each file record corresponds to each tuple of the relation, answer the following:

(i) Which will be the **blocking factor** (bfr) of the file block to accommodate the relation Employee?

[2]

(ii) How many blocks b does this file consist of?

[2]

(b) Estimate the expected cost in terms of **block accesses** (I/O blocks read/write) for the following SQL query *for each* type of file organization:

```
SQL: SELECT Name
     FROM   Employee
     WHERE  ID >= 1 AND ID <= 10
```

File Type Organization:

(i) Heap File;

[2]

(ii) Sequential File ordered by ID;

[2]

(iii) Hash File implemented through the External Hashing method over the hash key-field ID having M = 200 buckets and zero overflow buckets. Each bucket contains only one block.

[2]

Note: For convenience of calculations: $\log_2(200) = 8$, i.e., logarithm of 200 with base 2 is 8.

5. Given a file block size of B = 128 bytes, a key field of size V = 2 bytes, a data-pointer of size PR = 8 bytes and a tree-pointer of size P = 8 bytes, answer the following questions:

(a) Calculate the order p of a B Tree and a B+ Tree as a non-ordering/key Secondary Index such that each of the B Tree nodes, B+ Tree internal nodes, and B+ Tree leaf nodes occupies **only one file block**. Which of these indexes has the larger fun-out?

[10]

(b) How many blocks are required to store all the leaf nodes of the B+ Tree Index given a file of r = 100,000 records? **Note:** The B+ Tree index is built on a non-ordering/key attribute.

[10]

(c) Assume that every B+ Tree internal node and B+ Tree leaf node is 100% full. Which should be the level of the B+ Tree Index including the Root level and the leaf level to accommodate r = 100,000 key values? **Note:** For convenience of calculations: $\log_{13}(8334) \approx 3.5$, i.e., logarithm of 8334 with base 13 is 3.5.

[10]

Part C: Query Processing and Optimization [40 Marks]

6. Consider the following SQL query over the relations: **Employee**(<u>SSN</u>, SUPER_SSN, LName) and **Department**(<u>DNumber</u>, MGR_SSN)

```
SELECT E1.LNAME
FROM   EMPLOYEE E1, EMPLOYEE E2, DEPARTMENT D
WHERE  E1.SUPER_SSN = E2.SSN
AND    E2.SSN = D.MGR_SSN
AND    D.DNUMBER = 5
```

(a) Write this query in a Relation Algebra Expression.

[10]

(b) Draw the corresponding Relation Algebra Tree of your answer in Question **6.(a)**, which corresponds to the **optimal** query after applying heuristic optimization rules.

[10]

7. Consider two relations **Employee** E and **Department** D such that:

- relation E has n_E = 100 blocks and r_E = 1000 records
- relation D has n_D = 50 blocks and r_D = 10 records.

The SSN (Social Security Number) is the Primary Key of the relation Employee (E). The unique attribute Mgr_SSN (Manager's SSN) is the Foreign Key in relation Department (D) referencing to relation E on the SSN attribute. The memory of the Database system can accommodate n_B = 12 blocks for processing and the blocking factor for the join-results block is bfrRS = 10 records per block. Assume that we join the two relations E and D with respect to **equijoin** on the primary-foreign keys Mgr_SSN and SSN as shown in the following SQL-JOIN.

SQL-JOIN:
```
SELECT * FROM Employee E, Department D
WHERE  D.Mgr_SSN = E.SSN
```

(a) Based on the **nested-loop join strategy**, estimate the *total* expected cost in I/O block accesses. Report on the **best** nested-loop strategy and explain your answer.

[8]

(b) Estimate the *total* expected cost in I/O block accesses based on the naïve join strategy, i.e., producing the Cartesian product.

[2]

(c) Assume that there is a **Level 2 Secondary Index** over the Department relation for the unique attribute Mgr_SSN, i.e., level $x_D = 2$, and a **Level 2 Primary Index** over the Employee relation for the primary key SSN, i.e., level $x_E = 2$. Propose **two index-based nested-loop join strategies** and explain which one is the best using these indexes.

[8]

(d) Assume that both Employee and Department relations are already sorted by SSN and Mgr_SSN, respectively. What is the join cost for the **sort-merge join strategy**?

[2]