CSIE4105 Database Systems

Homework # 4

Due on 01/03/2024

1. (15%) Consider the universal relation **R = {A, B, C, D, E, F, G, H, I, J}** and the set of functional dependencies F = { {A, B}→{C}, {A}→{D, E}, {B}→{F}, {F}→{G, H}, {D}→{I, J} }.
   (a) (5%) What is the key for R? **Explain your answer.**
   (b) (5%) Decompose R into 2NF.
   (c) (5%) Based on your result in (b), decompose R into 3NF.

2. (20%) Consider the relation R, which has attributes that hold schedules of courses and sections at a university; **R = {Course_no, Sec_no, Offering_dept, Credit_hours, Course_level, Instructor_ssn, Semester, Year, Days_hours, Room_no, No_of_students}**. Suppose that the following functional dependencies hold on R:

   {Course_no} → {Offering_dept, Credit_hours, Course_level}

   {Course_no, Sec_no, Semester, Year} → {Days_hours, Room_no, No_of_students, Instructor_ssn}

   {Room_no, Days_hours, Semester, Year} → {Instructor_ssn, Course_no, Sec_no}

   (a) (5%) What normal form is the relation in? **Explain your answer.**
   (b) (3%) Try to determine which sets of attributes form **keys** of R.
   (c) (12%) Apply normalization until you cannot decompose the relations further. **State the reasons behind each decomposition.**

3. (10%) Describe at least *three* methods to **make data access more efficient on disk**.

4. (40%) Consider a disk with block size B = 512 bytes. A block pointer is P = 6 bytes long, and a record pointer is P_R = 7 bytes long. A file has r = 30,000 EMPLOYEE records of *fixed length*. Each record has the following fields: **Name** (30 bytes), **Ssn** (9 bytes), **Department_code** (9 bytes), **Address** (40 bytes), **Phone** (10 bytes), **Birth_date** (8 bytes), **Sex** (1 byte), **Job_code** (4 bytes), and **Salary** (4 bytes, real number). An additional byte is used as a *deletion marker*.

   (a) (4%) Calculate the record size R in bytes (including the *deletion marker*).
   (b) (6%) Calculate the blocking factor *bfr* and the number of disk blocks *b*, assuming an **unspanned** organization.
   (c) (15%, each 3%) Suppose that the file is ***ordered*** by the key field Ssn and we want to construct a ***primary index*** on **Ssn**. Calculate (i) the index blocking factor bfr_i (which is also the index fan-out *fo*); (ii) the number of first-level index entries and the number of first-level index blocks; (iii) the number of levels needed if we make it into a multilevel index; (iv) the total number of blocks required by the multilevel index; and (v) the number of block accesses needed to search for and retrieve a record from the file—given its Ssn value—using **the primary index**.

(d) (15%, each 3%) Suppose that the file is ***not ordered*** by the key field Ssn and we want to construct a ***secondary index*** on **Ssn**. Repeat the previous exercise **(part c)** for **the secondary index**.

5. (15%) What are the differences among the **primary**, **secondary**, and **clustering** indexes?