HW2 Structured Unstructured Data

1

Employer = (Employer_name, Project_name, salary, department_name, department_manager)

- An employer can participate in multiple projects
- An employer can get paid for each project they participate in
- Each project is managed by one department
- Each department has only one manager

A

Give the functional dependencies that express the above data constraints. You may put down a schema diagram as illustrated in the lecture slides.

(Employer_name, Project_name) → salary

> Note: it is unclear if the 2nd bullet point indicates employers can have various salaries depending on project, thus there is the above dependency. If an employer's salary is independent of the project then the dependency would be: Employer_name → salary.

Project_name → department_name

department_name → department_manager

B

Give a primary key of the relation Employer

Employer_name, Project_name

C

Is Employer in the second normal form (2NF)? If not, decompose it into 2NF

It is not in 2NF.

We need to decompose it into 2NF by removing partial dependencies.

R1(Employer_name, Project_name, salary)

R2(Project_name, department_name, department_manager)

Note: department_manager is a transitive dependency.

D

Is Employer in the third normal form (3NF)? If not, decompose it into 3NF

It is not in 3NF.

We need to decompose it into 3NF by removing transitive dependencies.

R1(Employer_name, Project_name, salary)

R2(Project_name, department_name)

R3(department_name, department_manager)

2

Consider the relation $R(A, B, C, D, E, F)$. Given this set of Functional Dependencies $C \rightarrow B, E \rightarrow D, B \rightarrow A, CE \rightarrow B$

A

There is only one candidate key for R. Determine what it is

The only canidate key is $\{C, E, F\}$

B

Which highest normal form is R in? 1NF, 2NF or 3NF? Explain your choice

Partial dependency $C \rightarrow B$ violates 2NF, therefore R is in 1NF.

3

Given two tables: r with schema R, and s with schema S, where r contains nr tuples and s contains ns tuples, and $ns > nr > 0$. Give the maximum and minimum possible tuples for the result of the following relational algebra expressions

A $r \cup s$

Max: $ns + nr$ Min: ns

B $s - r$

Max: ns Min: $ns - nr$

C $\sigma_{A>3}(r \times s)$

Max: $ns \times nr$ Min: 0

D $\sigma_{A=2}(\pi_{AB}(s))$

Max: ns Min: 0

4

Given the following schemas: **Student**(Sid,Sname,Age,Major) **Course**(Cno, Cname, Prerequisite, Ccredit) **Enroll**(Cno,Sno,Grade)

A

Express the following queries in relational algebra select σ, project $\prod$, Cartesian product $\times$, join (theta-join), with logic expressions ($e.g., \wedge, \vee$) comparisons ($e.g., >, <, =, \neq$) if needed

Q1

Find the students' name of all students majoring in computer science

$\pi_{Sname}(\sigma_{Major="ComputerScience"}(Student))$

Q2

List all the course numbers taken by students whose major is 'CS'

$\pi_{Cno}(\sigma_{Major="CS"}(Student) \bowtie_{Sid=Sno} Enroll)$

Q3

List the names and students who have enrolled in the course "Introduction to Data Science"

$\pi_{Sname,Sid}(\sigma_{Cname="IntroductionntoDataScience"}(Course) \quad \bowtie_{Cno} \quad Enroll \quad \bowtie_{Sno=Sid} Student)$

Q4

List the student IDs (Sid) of the students who didn't choose the course named 'CSDS 133'

$\pi_{Sno}(\sigma_{Cname\neq'CSDS133'}(Course) \bowtie_{Cno} Enroll)$

Alternatively

$\pi_{Sid}(Student) - \pi_{Sno}(\sigma_{Cname='CSDS133'}(Course) \bowtie_{Cid} Enroll)$

Q5

List the student IDs (Sid) of the students who enrolled in both 'CSDS 101' and 'CSDS 234'

$\pi_{Sno}(\sigma_{Cname="CSDS101"}(Course) \bowtie_{Cno} Enroll) \cap \pi_{Sno}(\sigma_{Cname="CSDS101"}(Course) \bowtie_{Cno} Enroll)$

B

Give a natural language description for the following RA expressions

Q6

$\sigma_{Major='EE'}(Students)$

Select all students whose major is EE

Q7

$\pi_{Sname}(\sigma_{Age>22}(Students))$–$\pi_{Sname}(\sigma_{Major='Bio'}(Students))$

List the names of students whose age is greater than 22 and are not majoring in Bio

Q8

$\pi_{sname}(Students) - \pi_{S1.sname}(\sigma_{S1.age>S2.age}(\rho(S1, Students) \times \rho(S2, Students)))$

List the names of students who are not older than other students. (Select the youngest student)

5

Consider a data table **Student**(Sid, Sname, Age, Major), where Sid has values that are 4-byte integers, Sname are 8-byte string, Age are 4-byte Strings, and Major are 8-byte strings. Answer the following questions

A

Assume a data block has size 512 bytes. We use data blocks to store B+ tree nodes as fixed-length records. A pointer takes 8 bytes. How many B+ tree nodes can a data block holds, if we define an B+ tree index with degree 6, on Sid, Sname, Age, or Major, respectively?

One tree node can hold 6 pointers and 5 keys. A thus one tree node is $6(8) + 5(A)$ bytes, where A is the size of a key.

Sid is 4 bytes, Sname is 8 bytes, Age is 4 bytes, and Major is 8 bytes. Therefore, Sid and Age will hold the same number of nodes whereas Sname and Major will hold different number of nodes.

For Sid and Age, one data block holds $512 = X(6(8) + 5(4))$ bytes. For Sname and Major, one data block holds $X = \frac{512}{68} = 7.5$ tree nodes.

For Sname and Major, one data block holds $X = \frac{512}{48+40} = \frac{512}{88} = 5.17$ tree nodes.

B

If we define a B+ tree index on Sname, and are allowed to use one data block (that can hold 512 bytes) to store a single B+ tree node with no bound on its degree. For such a B+ tree index, how many records can it index if its height is 3?

> Sname keys are 8 bytes, pointers are 8 bytes. A data block holds n pointers and $n-1$ keys. So a 512 byte data block holds $512 = 8(n-1) + 8n$, or $n = \frac{504}{16} = 31.5$. Meaning that since we have a height of 3, we can index at most $31^3 - 31^2$ records (according to slides).