G53RDB: Theory of Relational Databases

Lecture 16

Natasha Alechina
School of Computer Science & IT
nza@cs.nott.ac.uk

Plan of the lecture

- Summary of the course
- Format of the exam
- Advice on revision
- Particular topics
  - Query optimisation
  - Multivalued dependencies

Summary of the course

- Main topic: relational algebra and relational model
- Also:
  - extensions of relational algebra (nulls, bags,…)
  - relational algebra and first order logic
  - relational algebra and Datalog
  - relational algebra and SQL
- Normal forms, dependencies, reasoning about dependencies

Format of the exam

- 4 questions out of 6.
- Questions are similar to informal coursework:
  - Define English queries in relational algebra, Datalog, SQL
  - Define relational algebra queries in Datalog, SQL queries in relational algebra etc.
  - Which queries are less expensive to evaluate… optimise a given query…
  - Normalise a relation…
  - Given that a relation satisfies these dependencies, which other dependencies does it satisfy (Armstrong closure)

Revision

- Relational algebra: chapter 5 of Ullman and Widom
- Extensions - ch.5; lecture slides; Abiteboul, Hull, Vianu.
- SQL and relational algebra: chapter 6 of Ullman and Widom
- Dependencies, normal forms: chapter 3.6, 3.7 of Ullman and Widom. Additional material: see lecture slides.
- First order logic (relational calculus): lecture slides; any textbook on mathematical logic
- Datalog: chapter 10 of Ullman and Widom

Optimisation

- Exam 2004-5 question 5.

(a) Give a query tree for the following relational algebra expression:

$\pi_{Student.StName}(\sigma_{Lecturer.LecName= Jones}(\sigma_{Student.School= 'CS'}($
$(Student \bowtie Module) \bowtie Lecturer)))$

Explain in English what does the query compute, given that relation Student(ID, StName, School) stores IDs, names and schools of all students in the university, Module(ModCode, Title, ID) stores codes and titles of modules and IDs of students who take the module, and Lecturer(LecName, ModCode) stores names of lecturers who teach the module. (5)

Optimisation

- Answer for (a):
- The query asks for names of CS students who take a module taught by Jones.

Optimisation

(b) Optimise this query (give a new relational algebra expression and a query tree). (10)

(c) Explain why your answer is an improvement on the original query. Assume that there are 50,000 students and 3,000 lecturers at the university, 500 students at the School of CS, 1,000 modules taught at the university, and only five modules are taught by someone called Jones. (10)

Optimisation

(b) Optimise this query (give a new relational algebra expression and a query tree).

$\pi_{\text{Student.Name}}$ (

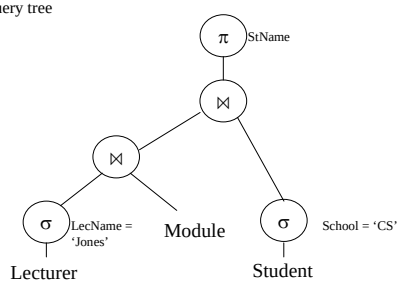
((σ$_{\text{Lecturer.Name='Jones'}}$ Lecturer) ⋈ Module) ⋈ σ$_{\text{Student.School = 'CS'}}$.Student

)

(query tree on the next slide)

Optimisation

(b) Query tree



Optimisation

- (c) for both versions (a) and (b) of the query,
  - Either just state cardinality of each intermediate relation
  - Or, in addition, how many tuples have to be matched to compute it.

Optimisation

- Student

ID	StName	School
		CS
		CS
		CS

500

50,000

Lecturer

LecName	ModCode
3000 x modules per lecturer	

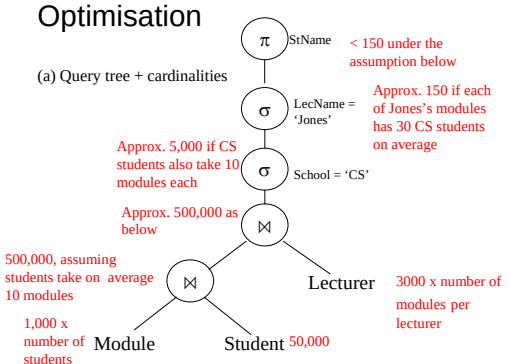
Module

ModCode	Title	ID

3000 x student-per-module

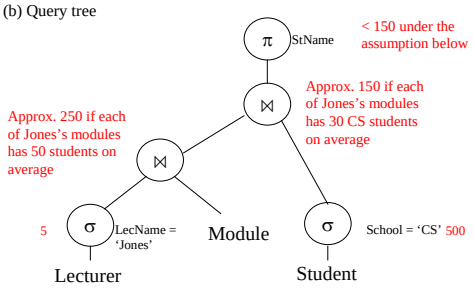
Optimisation

(a) Query tree + cardinalities



Optimisation

(b) Query tree



Multivalued dependencies

- Exam 2004-5 question 3.

(a) What is a multivalued dependency? (5)

(b) When is a relation in 4NF? Explain what kind of anomalies are avoided by normalising to 4NF as opposed to BCNF. (3)

Multivalued dependencies

(a) Given a relation R over schema U and X,Y $\subseteq$ U, a multivalued dependency X $\rightarrow\!\!\rightarrow$ Y holds if for any two tuples t and s in R, if t(X) = s(X), then there is a tuple v in R such that v(X)=t(X), v(Y) = t(Y) and v(Z)=s(Z) where Z = U – (X $\cup$ Y).

(b) A relation R is in 4NF is when for any non-trivial multivalued dependency X $\rightarrow\!\!\rightarrow$ Y in R, X is a superkey. Normalisation to 4NF eliminates update anomaly caused by redundancy, when every possible value of Y given X has to be repeated with every possible value of the rest of the attributes given X.

Multivalued dependencies

(c) Normalise the relation Tutee below to BCNF and 4NF and show that the resulting set of relations represents the same information. StudentID is a unique indentifier for every student; each student has exactly one tutor; each student takes several modules and may have several interests. (12)

StudentID	Tutor	Module	Interests
abc00u	xyz	G51PRG	football
abc00u	xyz	G51CSA	football
abc00u	xyz	G51MCS	football
abc00u	xyz	G51CUA	football
abc00u	xyz	G51SCI	football
abc00u	xyz	G5AIAI	football
abc00u	xyz	G51PRG	photography
abc00u	xyz	G51CUA	photography
....			

Multivalued dependencies

- Decomposition to BCNF. First of all, find candidate keys. The only one is (StudentID, Module, Interests). We have a partial dependency

StudentID $\rightarrow$ Tutor,

so the relation is not even in 2NF. Decompose into R1(StudentID, Tutor) and R2(StudentID, Module, Interests).

- In R1, the only candidate key is StudentID. The only non-trivial fd is StudentID $\rightarrow$ Tutor, so it is in BCNF (the only determinant is a (super)key. In R2, the key is still (StudentID, Module, Interests). No fds where determinant is not a (super)key. So they are both in BCNF. But we have mvds

StudentID $\rightarrow\!\!\rightarrow$ Module and StudentID $\rightarrow\!\!\rightarrow$ Interests.

Decompose into R2a(StudentID, Module) and R2b(StudentID,Interests).

- The join of R1 and R2 is lossless by the Heath's theorem for fds, so they represent the same information as the original table. The join of R2a and R2b is lossless by the Heath theorem for mvds.