Practical-2

Aim : Write relational algebra queries on the tables created in Practical-1 of ER diagram

Relational Algebra

• Procedural Algebra

Queries in relational algebra are applied to relation instances, result of a query is again a relation instance

Six basic operators in relational algebra :

select	σ	selects a subsets of tuples from reln
project	⊓	deletes unwanted columns from reln
Cartesian Product	x	allows to combine two relations
Set difference	-	tuples in reln.1, but not in reln. 2
Union	U	tuples in reln1 plus tuples in reln2
Rename	ρ	Renames attribute(s) and relation

• The operators take one or two relations as input and give a new relation as a result (relational algebra is "closed").

Assume the following relations :

BOOKS(DocId, Title, Publisher, Year)

STUDENTS(StId, StName, Major, Age)

AUTHORS(AName, Address)

Borrows(DocId, StId, Date)

has-written(DocId, Aname)

describes(DocId, Keyword)

• List the year and title of each book.

$\Pi_{year, Title}(BOOKS)$

• List all information about students whose major is CS.

$\sigma_{Major = 'CS'}(STUDENTS)$

• List all students with the books they can borrow.

STUDENTS $\times$ BOOKS

• List all books published by MCGraw- Hill before 1990

$\sigma_{Publisher = 'McGraw- Hill' \wedge Year < 1990}(BOOKS)$

• List the name of those authors who are living in Davis.

$\pi_{Aname}(\sigma_{Address\ like '\%Davis\%'}(AUTHORS))$

• List the name of students who are older than 30 and who are not studying CS.

$\pi_{StName}(\sigma_{Age>30}(STUDENTS)) -$

$\pi_{StName}(\sigma_{Major = 'CS'}(STUDENTS))$

• Rename Aname in the relation AUTHORS to Name

$\rho_{AUTHORS(Name, Address)}(AUTHORS)$

Example of Compound Queries

1. List the names of all students who have borrowed a book and who are CS majors.

$\pi_{StName}(\sigma_{STUDENTS.StId = borrows.StId}(\sigma_{Major='CS'}(SUDENTS) \times borrows))$

2. List the title of books written by the author 'Silberschatz'.

$\pi_{Title}(\sigma_{AName = 'Silberschatz'} \sigma_{has\text{-}written.DocId = BOOK.DocId}(has\text{-}written \times BOOKS)))$

Or

$\pi_{Title}(\sigma_{has\text{-}written.DocId = BOOK.DocId}(\sigma_{Aname = 'Silberschatz'}(has\text{-}written \times BOOKS))$

'database'.

...as for 2...

$\pi_{Title}(\sigma_{describes.DocId=BOOKS.DocId}(\sigma_{Keyword = 'database'}(describes) \times BOOKS))$

4. Find the name of the youngest student.

$\pi_{StName}(\sigma_{STUDENTS})-$

$\pi_{S1.StName}(\sigma_{S1.Age>S2.Age}(\rho_{S1}(STUDENTS) \times \rho_{S2}(STUDENTS)))$

5. Find the title of the oldest book.

$\pi_{Title}(BOOKS)-$

$\pi_{B1.Title}(\sigma_{B1.Year>B2.Year}(\rho_{B1}(BOOKS) \times \rho_{S2}(BOOKS)))$