

**In the name of
ALLAH**



Introduction to Database Systems



Teacher: Dr. Nayyer Masood

DATA MANIPULATION LANGUAGES





Relational Algebra

- Relational algebra operations work on one or more relations to define another relation leaving the original intact.



Relational Algebra

- Both operands and results are relations, so output from one operation can become input to another operation.



Relational Algebra

- Allows expressions to be nested, just as in arithmetic. This property is called ***closure***.



Relational Algebra

- 5 basic operations in relational algebra: Selection, Projection, Cartesian product, Union, and Set Difference.



Relational Algebra

- These perform most of the data retrieval operations needed.
- Also have Join, Intersection, and Division operations, which can be expressed in terms of 5 basic operations.



Five Basic Operators

- Unary Operators
 1. Select
 2. Project
- Binary Operators
 3. Union
 4. Set Difference
 5. Cartesian Product



Select

- Selects tuples that satisfy a given predicate
- Notation is Greek symbol sigma:

$$\sigma_{PREDICATE}(RELATION)$$



Select

- To process a selection,
 - Look at each tuple
 - See if we have a match (based on the condition)



Select

- The Degree of the resulting relation is the same as the degree of the relation

$$| \sigma | = | r(R) |$$

- σ is Commutative

$$\sigma_{c1} (\sigma_{c2}(R)) = \sigma_{c2} (\sigma_{c1}(R))$$



Selection Example

stId	stName	stAdr	prName	curSem
S1020	Sohail Dar	H#14, F/8-4, Islamabad.	MCS	4
S1038	Shoaib Ali	H#23, G/9-1, Islamabad	BCS	3
S1015	Tahira Ejaz	H#99, Lala Rukh Wah.	MCS	5
S1018	Arif Zia	H#10, E-8, Islamabad.	BIT	5



Selection Example

1. $\sigma_{\text{Curr_Sem} > 3} (\text{STUDENT})$
2. $\sigma_{\text{Studid} = \text{'S1038'}} (\text{STUDENT})$



Selection Example Cont'd)

FacID	Fname	Address	Salary	Rank
F2345	Usman	H#Car409	21000	lecturer
F3456	Tahir	H#Bac100	23000	Asso Prof
F4567	Ayesha	H#Car301	27000	Asso Prof
F5678	Samad	H#Irv342	32000	Professor



Selection Example Cont'd)

1. $\sigma_{\text{Salary} > 27000}$ (FAC)
2. $\sigma_{(\text{Salary} > 26000) \text{ and } (\text{Rank} = \text{'Asso Prof'})}$ (FAC)



Projection

- Unary operation that returns its argument relation with certain attributes left out.
- Any duplicate rows are eliminated.



Projection

- Notation

$$\Pi_{a_1, a_2, \dots, a_k}(R)$$

where $a_1 \dots a_k$ are attribute names and R is a relation name



Projection

- Works on a single relation R and defines a relation that contains a vertical subset of R , extracting the values of specified attributes and eliminating duplicates.



Projection

- Reduces duplicate columns created by cross product.
- Creates a New relation



Projection Example

FacID	Fname	Dept	Salary	Rank
F2345	Usman	CSE	21000	lecturer
F3456	Tahir	CSE	23000	Asso Prof
F4567	Ayesha	ENG	27000	Asso Prof
F5678	Samad	MNG	32000	Professor



Projection Example

- π FacID, Salary (Faculty)
- π Fname, Rank (Faculty)
- π Facid, Salary, Rank (Faculty)



Union

- Assuming R & S are union compatible...
- Union: $R \cup S$ is the set of tuples in either R or S or both.



Union

- Since it is a set, there are no duplicate tuples
- Union is Commutative

$$R \cup S = S \cup R$$



Union Example

CID	ProgID	Cred_Hrs	CourseTitle
C2345	P1245	3	Operating Sytems
C3456	P1245	4	Database Systems
C4567	P9873	4	Financial Management
C5678	P9873	3	Money & Capital Market

CID	ProgID	Cred_Hrs	CourseTitle
C4567	P9873	4	Financial Management
C8944	P4567	4	Electronics



Union Example

Course 1 U Course 2

CID	ProgID	Cred_Hrs	CourseTitle
C2345	P1245	3	Operating Sytems
C3456	P1245	4	Database Systems
C4567	P9873	4	Financial Management
C5678	P9873	3	Money & Capital Market
C8944	P4567	4	Electronics



Intersection (\cap)

- Assuming that R & S are union compatible
- Intersection: $R \cap S$ is the set of tuples in both R and S



Intersection (\cap)

- Intersection is Commutative

$$R \cap S = S \cap R$$



Intersection (\cap) Example

Course 1 \cap Course 2

CID	ProgID	Cred_Hrs	CourseTitle
C4567	P9873	4	Financial Management



Difference (-)

- Difference: $R - S$ is the set of tuples that appear in R but do not appear in S .



Difference (-) Example

Course1 – Course2

CID	ProgID	Cred_Hrs	CourseTitle
C2345	P1245	3	Operating Sytems
C3456	P1245	4	Database Systems
C5678	P9873	3	Money & Capital Market



Cartesian Product (X)

- Sets do not have to be union compatible.

$$R(A_1, A_2, \dots, A_N)$$
$$X$$
$$S(B_1, B_2, \dots, B_M) \text{ is}$$



Cartesian Product (X)

$Q (A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$

- If R has C tuples and S has D tuples, the result is $C \times D$ tuples.



Cartesian Product (X)

- Also called “cross product”
- Note that union, intersection and cross product are commutative and associative



Cartesian Product (X) Example

Course X Registration

Course

CID	CourseTitle
C3456	Database Systems
C4567	Financial Management
C5678	Money & Capital Market

Registration

SID	StudName
S101	Ali Tahir
S103	Farah Hasan



Cartesian Product (X) Example

CID	CourseTitle	SID	StudName
C3456	Database Systems	S101	Ali Tahir
C4567	Financial Management	S101	AliTahr
C5678	Money & Capital Market	S101	Ali Tahir
C3456	Database Systems	S103	Farah Hasan
C4567	Financial Management	S103	Farah Hasan
C5678	Money & Capital Market	S103	Farah Hasan



Types of Join

- A special form of cross product of two tables
- Different types
- Theta, Equi, Natural, Semi, Outer Joins are different types



Theta Join

We apply the condition through select on one relation and then only those rows are used in the cross product with the second relation

$$R \bowtie_{\theta} S$$



Theta Join Example

COURSE

facId	facName	dept	salary	rank
F2345	Usman	CSE	21000	lecturer
F3456	Tahir	CSE	23000	Asso Prof
F4567	Ayesha	ENG	27000	Asso Prof
F5678	Samad	MNG	32000	Professor

crCode	crTitle	fId
C3456	Database Systems	F2345
C4567	Financial Management	
C5678	Money & Capital Market	F4567
C3425	Introduction to Accounting	F2345

$(\sigma_{\text{rank} = \text{'Asso Prof'}}(\text{FACULTY})) \times \text{COURSE}$



σ rank = 'Asso Prof' (FACULTY) X COURSE

facId	facName	dept	salary	Rank	crCode	crTitle	fId
F3456	Tahir	CSE	23000	Asso Prof	C3456	Database Systems	F2345
F3456	Tahir	CSE	23000	Asso Prof	C4567	Financial Management	
F3456	Tahir	CSE	23000	Asso Prof	C5678	Money & Capital Market	F4567
F3456	Tahir	CSE	23000	Asso Prof	C3425	Introduction to Accounting	F2345
F4567	Ayesha	ENG	27000	Asso Prof	C3456	Database Systems	F2345
F4567	Ayesha	ENG	27000	Asso Prof	C4567	Financial Management	
F4567	Ayesha	ENG	27000	Asso Prof	C5678	Money & Capital Market	F4567
F4567	Ayesha	ENG	27000	Asso Prof	C3425	Introduction to Accounting	F2345



Equijoin

- Rows are joined on the basis of values of a common attribute between the two relations
- Rows having the same value in the common attribute are joined



Equijoin

- Common attributes appears twice in the output
- Common attribute with the same name is qualified with the relation name in the output



FACULTY

Equijoin Example

COURSE

facId	facName	dept	salary	rank
F2345	Usman	CSE	21000	lecturer
F3456	Tahir	CSE	23000	Asso Prof
F4567	Ayesha	ENG	27000	Asso Prof
F5678	Samad	MNG	32000	Professor

crCode	crTitle	fId
C3456	Database Systems	F2345
C4567	Financial Management	
C5678	Money & Capital Market	F4567
C3425	Introduction to Accounting	F2345

FACULTY



FACULTY.facId = COURSE.fId

COURSE



Equijoin Example

FACULTY \bowtie FACULTY.facId = COURSE.fld COURSE

facId	facName	dept	salary	Rank	crCode	crTitle	fld
F2345	Usman	CSE	21000	lecturer	C3456	Database Systems	F2345
F4567	Ayesha	ENG	27000	Asso Prof	C5678	Money & Capital Market	F4567
F2345	Usman	CSE	21000	lecturer	C3425	Introduction to Accounting	F2345



Equijoin Example

FACULTY \bowtie FACULTY.facId = COURSE.facId COURSE

FACULTY. facId	acName	dept	salary	Rank	crCode	crTitle	COURSE.fa cId
F2345	Usman	CSE	21000	lecturer	C3456	Database Systems	F2345
F4567	Ayesha	ENG	27000	Asso Prof	C5678	Money & Capital Market	F4567
F2345	Usman	CSE	21000	lecturer	C3425	Introduction to Accounting	F2345



Natural Join

- Also called simply the join, most general form of join
- Same as equijoin with common column appearing once



Natural Join Example

COURSE

facId	facName	dept	salary	rank
F2345	Usman	CSE	21000	lecturer
F3456	Tahir	CSE	23000	Asso Prof
F4567	Ayesha	ENG	27000	Asso Prof
F5678	Samad	MNG	32000	Professor

crCode	crTitle	facId
C3456	Database Systems	F2345
C4567	Financial Management	
C5678	Money & Capital Market	F4567
C3425	Introduction to Accounting	F2345

$\Pi_{\text{facName, crTitle}}(\text{FACULTY} \bowtie \text{COURSE})$

FACULTY.facId, COURSE.facId

COURSE)



Natural Join Example

FACULTY \bowtie FACULTY.facId, COURSE.facId COURSE

facId	facName	dept	salary	Rank	crCode	crTitle
F2345	Usman	CSE	21000	lecturer	C3456	Database Systems
F4567	Ayesha	ENG	27000	Asso Prof	C5678	Money & Capital Market
F2345	Usman	CSE	21000	lecturer	C3425	Introduction to Accounting



Types Of Joins

- Left Outer Join
- Right Outer Join
- Outer Join
- Semijoin



Left Outer Join

- Keep all of the tuples from the “left” relation
- Join with the right relation
- Pad the non-matching tuples with nulls



Left Outer Join Example

FACULTY

facId	facName	dept	salary	rank
F2345	Usman	CSE	21000	lecturer
F3456	Tahir	CSE	23000	Asso Prof
F4567	Ayesha	ENG	27000	Asso Prof
F5678	Samad	MNG	32000	Professor

COURSE

crCode	crTitle	facId
C3456	Database Systems	F2345
C4567	Financial Management	
C5678	Money & Capital Market	F4567
C3425	Introduction to Accounting	F2345

FACULTY \bowtie COURSE



facId	facName	dept	salary	Rank	crCode	crTitle
F2345	Usman	CSE	21000	lecturer	C3456	Database Systems
F4567	Ayesha	ENG	27000	Asso Prof	C5678	Money & Capital Market
F2345	Usman	CSE	21000	lecturer	C3425	Introduction to Accounting
F5678	Samad	MNG	32000	Professor		
F3456	Tahir	CSE	23000	Asso Prof		



Right Outer Join

Same as the left, but keep tuples from the “right” relation



Right Outer Join Example

FACULTY

COURSE

facId	facName	dept	salary	rank
F2345	Usman	CSE	21000	lecturer
F3456	Tahir	CSE	23000	Asso Prof
F4567	Ayesha	ENG	27000	Asso Prof
F5678	Samad	MNG	32000	Professor

crCode	crTitle	facId
C3456	Database Systems	F2345
C4567	Financial Management	
C5678	Money & Capital Market	F4567
C3425	Introduction to Accounting	F2345

FACULTY \bowtie COURSE



Outer Join

Same as left, but keep all tuples from both relations



Outer Join Example

FACULTY

facId	facName	dept	salary	rank
F2345	Usman	CSE	21000	lecturer
F3456	Tahir	CSE	23000	Asso Prof
F4567	Ayesha	ENG	27000	Asso Prof
F5678	Samad	MNG	32000	Professor

COURSE

crCode	crTitle	facId
C3456	Database Systems	F2345
C4567	Financial Management	
C5678	Money & Capital Market	F4567
C3425	Introduction to Accounting	F2345



Semijoin

- First take the natural join of two tables
- Then take the projection on the attributes of first table

FACULTY \bowtie COURSE



Semijoin Example

FACULTY

facId	facName	dept	salary	rank
F2345	Usman	CSE	21000	lecturer
F3456	Tahir	CSE	23000	Asso Prof
F4567	Ayesha	ENG	27000	Asso Prof
F5678	Samad	MNG	32000	Professor

COURSE

crCode	crTitle	facId
C3456	Database Systems	F2345
C4567	Financial Management	
C5678	Money & Capital Market	F4567
C3425	Introduction to Accounting	F2345

COURSE ► FACULTY





Example Schema

- PROGRAM (prName, totSem, prCredits)
- COURSE (crCode, crName, crCredits, crLevel)
- SEMESTER (semName, stDate, endDate)
- CROFRD (crCode, semName, facId)
- FACULTY (facId, fName, fQual, fSal, rank)
- STUDENT (stId, stName, stFName, stAdres, stPhone, curSem, cgpa)
- ENROLL (stId, crCode, semName, mTerm, sMrks, fMrks, totMrks, grade, gp)
- SEM_RES (stId, semName, totCrs, totCrdts, totGP, gpa)



Practice Queries

- Show all the programs offered by university
- Show the total semesters and credits in each program
- Show the faculty name, salary and rank



Practice Queries

- Show the records of students who do not have telephone
- Show the mid-term marks, sessional and final marks of each student
- Show the semester result of all students for the current semester



Practice Queries

- Show the name, current semester result and the cgpa of all students
- Show the course names of all courses where course credits are 4



Practice Queries

- List the names of the courses, names of faculty who are teaching those courses in current semester
- Show the name of the student and names of the courses in which he has registered so far



- Show the name of the student and names of the courses in which he has registered in the current semester