Course Code: IS201
Course Title : Database System

Dr. Mahmoud Zaher

Egyptian Russian University (ERU)

Faculty of Artificial Intelligence

Text Book

The concepts and presentation of this course are drawn from:

- R. ElMasri & S. Navathe, “Fundamentals of Database Systems”, Addison Wesley, Fifth Edition, 20011.
- Carlos M. Coronel - Database Systems_ Design, Implementation, & Management-Cengage Learning (2018).
- Learn SQL Database Programming_ Query and manipulate databases from popular relational database servers using SQL-Packt Publishing (2020)

The Relational Algebra and Relational Calculus

The Relational Algebra and Relational Calculus

- **Relational algebra**
  - Basic set of operations for the relational model
  - $\{\sigma, \pi, \cup, \rho, -, \times\}$
- **Relational algebra expression**
  - Sequence of relational algebra operations
- **Relational calculus**
  - Higher-level declarative language for specifying relational queries

Unary Relational Operations: SELECT and PROJECT

- **Unary**
  - Applied to a single relation
- The SELECT Operation (select rows)

$$\sigma_{<\text{selection condition}>}(R)$$

  - Subset of the tuples from a relation that satisfies a selection condition:
    - Boolean expression contains clauses of the form <attribute name> <comparison op> <constant value>

      or
    - <attribute name> <comparison op> <attribute name>
    - Tuple is (A row or record in a database table.)

Unary Relational Operations: SELECT and PROJECT (cont'd.)

- Example:

$$\sigma_{(\text{Dno}=4 \textbf{ AND } \text{Salary}>25000) \textbf{ OR } (\text{Dno}=5 \textbf{ AND } \text{Salary}>30000)}(\text{EMPLOYEE})$$

- <selection condition> applied independently to each individual tuple *t* in *R*
  - If condition evaluates to TRUE, tuple selected
- Boolean conditions **AND**, **OR**, and **NOT**

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

Unary Relational Operations: SELECT and PROJECT (cont'd.)

- **Selectivity**
  - Fraction of tuples selected by a selection condition
- SELECT operation commutative
- **Cascade** SELECT operations into a single operation with **AND** condition

Select

Original table

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

SELECT ALL yields

New table

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

SELECT only PRICE less than $2.00 yields

P_CODE	P_DESCRIPT	PRICE
213345	9v battery	1.92
254467	100W bulb	1.47

SELECT only P_CODE = 311452 yields

P_CODE	P_DESCRIPT	PRICE
311452	Powerdrill	34.99

The PROJECT Operation

- Selects columns from table and discards the other columns:

$$\pi_{\langle \text{attribute list} \rangle}(R) \qquad \pi_{\text{Fname, Lname, Salary}}(\text{EMPLOYEE})$$

- **Degree**
  - Number of attributes in <attribute list>

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

Sequences of Operations and the RENAME Operation

- **In-line** expression:

$$\pi_{\text{Fname, Lname, Salary}}(\sigma_{\text{Dno}=5}(\text{EMPLOYEE}))$$

- Sequence of operations:

$$\text{DEP5\_EMPS} \leftarrow \sigma_{\text{Dno}=5}(\text{EMPLOYEE})$$
$$\text{RESULT} \leftarrow \pi_{\text{Fname, Lname, Salary}}(\text{DEP5\_EMPS})$$

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

Sequences of Operations and the RENAME Operation

- **Rename** ρ attributes in intermediate results
  - RENAME operation

$$\rho_{S(B1,\, B2,\, \ldots,\, Bn)}(R) \quad \text{or} \quad \rho_{S}(R) \quad \text{or} \quad \rho_{(B1,\, B2,\, \ldots,\, Bn)}(R)$$

Operating on Relations (Cont.)

- **Projection**:
  - Projection ***selects certain columns*** from a table.
  - For example, you might want to see just the Name, Customer Number (Cust#), and City fields from the Cust-address table.

Operating on Relations (Cont.)

- The *projection operator* takes a table and a set of field names as input and returns another table as output.
- The resulting table contains the same number of rows but fewer columns.
- For example, Project Name, Cust#, City produces the following table:

Project

Original table

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

New table

PROJECT PRICE yields

PRICE
5.26
25.15
10.99
1.92
1.47
34.99

PROJECT P_DESCRIPT and PRICE yields

P_DESCRIPT	PRICE
Flashlight	5.26
Lamp	25.15
Box Fan	10.99
9v battery	1.92
100W bulb	1.47
Powerdrill	34.99

PROJECT P_CODE and PRICE yields

P_CODE	PRICE
123456	5.26
123457	25.15
123458	10.99
213345	1.92
254467	1.47
311452	34.99

Relational Algebra Operations from Set Theory

- **UNION**, **INTERSECTION**, and **MINUS**
  - Merge the elements of two sets in various ways
  - Binary operations
  - Relations must have the same type of tuples
- UNION
  - $R \cup S$
  - Includes all tuples that are either in *R* or in *S* or in both *R* and *S*
  - Duplicate tuples eliminated

Operating on Relations (Cont.)

- **Union:**
  - The *union* of two tables with N and M rows, respectively, is obtained by *concatenating* them into one table with a total of (N+M) rows.
  - Union makes sense only if the schemes of the two tables match, i.e., if they have the same number of fields with matching attributes in each field.

Union

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

UNION

P_CODE	P_DESCRIPT	PRICE
345678	Microwave	160.00
345679	Dishwasher	500.00
123458	Box Fan	10.99

yields

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99
345678	Microwave	160
345679	Dishwasher	500

Relational Algebra Operations from Set Theory (cont'd.)

- INTERSECTION
  - $R \cap S$
  - Includes all tuples that are in both R and S
- SET DIFFERENCE (or MINUS)
  - $R - S$
  - Includes all tuples that are in R but not in S

Intersect

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Franklin	Johnson
Martin	Lopez

INTERSECT

EMP_FNAME	EMP_LNAME
Franklin	Lopez
William	Turner
Franklin	Johnson
Susan	Rogers

yields

STU_FNAME	STU_LNAME
Franklin	Johnson

Operating on Relations (Cont.)

- **Difference:**
  - The *difference* operator can be used to find records that are in one relation (table), but not in another.

Difference

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Franklin	Johnson
Martin	Lopez

DIFFERENCE

EMP_FNAME	EMP_LNAME
Franklin	Lopez
William	Turner
Franklin	Johnson
Susan	Rogers

yields

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Martin	Lopez

The CARTESIAN PRODUCT (CROSS PRODUCT) Operation

- **CARTESIAN PRODUCT**
  - **CROSS PRODUCT** or **CROSS JOIN**
  - Denoted by $\times$
  - Binary set operation
  - Relations do not have to be union compatible
  - Useful when followed by a selection that matches values of attributes

Operating on Relations (Cont.)

- **Product:**
  - The product *multiplies two tables* so that if one table has N rows and the other has M rows, the product table will contain (N*M) rows.
  - Thus if we take the product of the following two tables:

Product

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

PRODUCT

STORE	AISLE	SHELF
23	W	5
24	K	9
25	Z	6

yields

P_CODE	P_DESCRIPT	PRICE	STORE	AISLE	SHELF
123456	Flashlight	5.26	23	W	5
123456	Flashlight	5.26	24	K	9
123456	Flashlight	5.26	25	Z	6
123457	Lamp	25.15	23	W	5
123457	Lamp	25.15	24	K	9
123457	Lamp	25.15	25	Z	6
123458	Box Fan	10.99	23	W	5
123458	Box Fan	10.99	24	K	9
123458	Box Fan	10.99	25	Z	6
213345	9v battery	1.92	23	W	5
213345	9v battery	1.92	24	K	9
213345	9v battery	1.92	25	Z	6
311452	Powerdrill	34.99	23	W	5
311452	Powerdrill	34.99	24	K	9
311452	Powerdrill	34.99	25	Z	6
254467	100W bulb	1.47	23	W	5
254467	100W bulb	1.47	24	K	9
254467	100W bulb	1.47	25	Z	6

Operating on Relations (Cont.)

Cust-address

Name	Cust#	City
Gucci	3577	BevHls
SAG	2667	BevHls

Cust-credit

Name	Credit
Gucci	Good
SAG	Average

Operating on Relations (Cont.)

- We obtain the following table with 2 X 2 = 4 rows:

Name	Cust#	Ci	Name	Credit
Gucci	3577	BevHls	Gucci	Good
SAG	2667	BevHls	Gucci	Good
Gucci	3577	BevHls	SAG	Average
SAG	2667	BevHls	SAG	Average

Operating on Relations (Cont.)

- Where the first three columns are from the first table and the last two columns, from the second table.
- The resulting table does not look right as it stands.
- Why should we have a record that refers both to Gucci's location and SG's credit record.

Operating on Relations (Cont.)

- We can clean up the table by selecting those rows where the names in columns 1 and 4 match to obtain the following table:

Name	Cust#	City	Name	Credit
Gucci	3577	BevHls	Gucci	Good
SAG	2667	BevHls	SAG	Average

Operating on Relations (Cont.)

- We can then project out one of the <u>*Name*</u> fields to get:

Name	Cust#	City	Credit
Gucci	3577	BevHls	Good
SAG	2667	BevHls	Average

Binary Relational Operations: JOIN and DIVISION

- The **JOIN** Operation
  - Denoted by $\bowtie$
  - Combine related tuples from two relations into single “longer” tuples
  - General join condition of the form <condition> **AND** <condition> **AND**...**AND** <condition>
  - Example:

$$\text{DEPT\_MGR} \leftarrow \text{DEPARTMENT} \bowtie_{\text{Mgr\_ssn=Ssn}} \text{EMPLOYEE}$$
$$\text{RESULT} \leftarrow \pi_{\text{Dname, Lname, Fname}}(\text{DEPT\_MGR})$$

Operating on Relations (Cont.)

- **Joining Tables**:
  - The *join* operator allows us to join several tables together.
  - Given two tables, A and B, the *join* operator allows one to apply a condition to all rows that are formed by concatenating rows from A and rows from B.

Operating on Relations (Cont.)

- Those rows satisfying the condition are returned, i.e. product of the two tables, *followed by* a selection that ensures that all the rows in the new table are meaningful, i.e. *the common fields* of *the tow tables have equal values.*

Operating on Relations (Cont.)

- The following tables show how the *joins* would affect the Suppliers and Product-composition tables:

Operating on Relations (Cont.)

<u>**Product-composition**</u>

Product	**Part-name**	**Part #**	**Qty**
fastener	**Lever**	**2021**	**1**
fastener	**Sprocket**	**2197**	**3**
fastener	**Cog**	**2876**	**4**
fastener	**Spring**	**2346**	**6**
adapter	**Spring**	**2346**	**5**
adapter	**Pulley**	**2477**	**5**
adapter	**Rivet**	**2498**	**21**
transformer	**Lever**	**2477**	**3**
transformer	**Cam**	**2021**	**1**
transformer	**Cam**	**2655**	**3**
transformer	**rivet**	**2498**	**12**
processor	**Cpul**	**9876**	**1**
processor	**8k-chip**	**9801**	**4**
processor	**led**	**9701**	**4**

Operating on Relations (Cont.)

Suppliers

Sup-name	Part#
Ace	20211
Ace	2346
Ace	2477
Jackson	2197
Campbell	2876
Trueman	2498

Operating on Relations (Cont.)

Join *(Product-composition table, Suppliers table, at part#* (common field)):

Product	Part-Name	Part#	Qty	Sup-Name
fastener	Lever	2021	1	Ace
fastener	sprocket	21973	Jackson	fastener
2876	4	Cam	bell	fastener
2346	6	Ace	dapter	2346
5	Ace	adapter	2477	5
Ace	adapter	2498	21	Trueman
transformer	Lever	2477	3	Ace
transformer	Cam	2021	1	Ace
transformer	Cam	2655	3	Trueman
transformer	rivet	2498	12	Trueman
rocessor	c	9876	1	Bitstream
rocessor	8	9801	4	Electra
rocessor	led	9701	4	Electra
.	.	.	.	.

Binary Relational Operations: JOIN and DIVISION (cont'd.)

- **THETA** JOIN
  - Each <condition> of the form $A_i \ \theta \ B_j$
  - A_i is an attribute of R
  - B_j is an attribute of S
  - A_i and B_j have the same domain
  - θ (theta) is one of the comparison operators:

$$\{=, <, \leq, >, \geq, \neq\}$$

Variations of JOIN: The EQUIJOIN and NATURAL JOIN

- **EQUIJOIN** =
  - Only = comparison operator used
  - Always have one or more pairs of attributes that have identical values in every tuple
- **NATURAL JOIN** *
  - Denoted by *
  - Removes second (superfluous) attribute in an EQUIJOIN condition

Variations of JOIN: The EQUIJOIN and NATURAL JOIN (cont'd.)

- **Join selectivity**
  - Expected size of join result divided by the maximum size $n_R * n_S$
- **Inner joins**
  - Type of match and combine operation
  - Defined formally as a combination of CARTESIAN PRODUCT and SELECTION

A Complete Set of Relational Algebra Operations

- Set of relational algebra operations

$$\{\sigma, \pi, \cup, \rho, -, \times\}$$

is a **complete set**

  - Any relational algebra operation can be expressed as a sequence of operations from this set

The DIVISION Operation

- Denoted by ÷
- Example: retrieve the names of employees who work on all the projects that ‘John Smith’ works on
- Apply to relations $R(Z) \div S(X)$
  - Attributes of *R* are a subset of the attributes of *S*

Divide

CODE	LOC
A	5
A	9
A	4
B	5
B	3
C	6
D	7
D	8
E	8

DIVIDE

CODE
A
B

yields

LOC
5

Operations of Relational Algebra

Table 6.1 Operations of Relational Algebra

OPERATION	PURPOSE	NOTATION
SELECT	Selects all tuples that satisfy the selection condition from a relation R.	$\sigma_{\text{<selection condition>}}(R)$
PROJECT	Produces a new relation with only some of the attributes of R, and removes duplicate tuples.	$\pi_{\text{<attribute list>}}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{\text{<join condition>}} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{\text{<join condition>}} R_2$, OR $R_1 \bowtie_{(\text{<join attributes 1>}),\ (\text{<join attributes 2>})} R_2$
NATURAL JOIN	Same as EQUIJOIN except that the join attributes of R_2 are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R_1 *_{\text{<join condition>}} R_2$, OR $R_1 *_{(\text{<join attributes 1>}),\ (\text{<join attributes 2>})} R_2$ OR $R_1 * R_2$

Operations of Relational Algebra (cont'd.)

Table 6.1 Operations of Relational Algebra

UNION	Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cup R_2$
INTERSECTION	Produces a relation that includes all the tuples in both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cap R_2$
DIFFERENCE	Produces a relation that includes all the tuples in R_1 that are not in R_2; R_1 and R_2 must be union compatible.	$R_1 - R_2$
CARTESIAN PRODUCT	Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2.	$R_1 \times R_2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$