The Relational Algebra

Database Management Systems

Handout 6

Chapter Outline

- Example Database Application (COMPANY)
- Relational Algebra
  - Unary Relational Operations
  - Relational Algebra Operations From Set Theory
  - Binary Relational Operations
  - Additional Relational Operations
  - Examples of Queries in Relational Algebra
- Relational Calculus
  - Tuple Relational Calculus

Database State for COMPANY

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

Database State for COMPANY

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS_ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

Database State for COMPANY

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

Relational Algebra

- The basic set of operations for the relational model is known as the relational algebra. These operations enable a user to specify basic retrieval requests.

- The result of a retrieval is a new relation, which may have been formed from one or more relations. The **algebra operations** thus produce new relations, which can be further manipulated using operations of the same algebra.

- A sequence of relational algebra operations forms a **relational algebra expression**, whose result will also be a relation that represents the result of a database query (or retrieval request).

Unary Relational Operations – Select (σ)

SELECT operation is used to select a *subset* of the tuples from a relation that satisfy a **selection condition**.

Example: To select the EMPLOYEE tuples whose department number is four or those whose salary is greater than $30,000 the following notation is used:

$$\boldsymbol{\sigma_{\text{DNO} = 4 \text{ OR SALARY} > 30{,}000} (\text{EMPLOYEE})}$$

- **S**elect operation is denoted by $\boldsymbol{\sigma}_{\text{<selection condition>}}(R)$

- <selection condition> is a Boolean expression specified on the attributes of relation R

Unary Relational Operations

SELECT Operation Properties

- The SELECT operation $\sigma_{\text{<selection condition>}}(R)$ produces a relation S that has the same schema as R

- The SELECT operation σ is **commutative;** i.e.,

  $\sigma_{\text{<condition1>}}(\sigma_{\text{< condition2>}}(R)) = \sigma_{\text{<condition2>}}(\sigma_{\text{< condition1>}}(R))$

  – A cascaded SELECT operation **may be applied in any order;** i.e.,

  $\sigma_{\text{<condition1>}}(\sigma_{\text{< condition2>}}(\sigma_{\text{<condition3>}}(R))$

  $= \sigma_{\text{<condition2>}}(\sigma_{\text{< condition3>}}(\sigma_{\text{< condition1>}}(R)))$

  – A cascaded SELECT operation may be replaced by a single selection with a conjunction of all the conditions; i.e.,

  $\sigma_{\text{<condition1>}}(\sigma_{\text{< condition2>}}(\sigma_{\text{<condition3>}}(R))$

  $= \sigma_{\text{<condition1> AND < condition2> AND < condition3>}}(R)))$

Unary Relational Operations (cont.)

Results of SELECT and PROJECT operations.

(a) $\sigma_{\text{(DNO=4 AND SALARY>25000) OR (DNO=5 AND SALARY>30000)}}(\text{EMPLOYEE})$.

(b) $\pi_{\text{LNAME, FNAME, SALARY}}(\text{EMPLOYEE})$. (c) $\pi_{\text{SEX, SALARY}}(\text{EMPLOYEE})$

(a)

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
Franklin	T	Wong	333445555	1955-12-08	638 Voss,Houston,TX	M	40000	888665555	5
Jennifer		Wallace	987654321	1941-06-20	291 Berry,Bellaire,TX	F	43000	888665555	4
Ramesh		Narayan	666884444	1962-09-15	975 FireOak,Humble,TX	M	38000	333445555	5

(b)

LNAME	FNAME	SALARY
Smith	John	30000
Wong	Franklin	40000
Zelaya	Alicia	25000
Wallace	Jennifer	43000
Narayan	Ramesh	38000
English	Joyce	25000
Jabbar	Ahmad	25000
Borg	James	55000

(c)

SEX	SALARY
M	30000
M	40000
F	25000
F	43000
M	38000
M	25000
M	55000

Unary Relational Operations – Project (π)

Project operation selects certain *columns* from the table and discards the other columns.

Example: To list each employee’s first and last name and salary, the following is used:

$$\pi_{\textbf{LNAME, FNAME,SALARY}}\textbf{(EMPLOYEE)}$$

General form: $\pi_{\text{<attribute list>}}(R)$,

- π(pi) is the symbol used to represent the project operation.

- <attribute list> is the desired list of attributes from the attributes of relation R.

- π *removes any duplicate tuples,* result is a set of tuples (hence a valid relation).

Unary Relational Operations (cont.)

PROJECT Operation Properties

- The number of tuples in the result of projection $\pi_{\text{<list>}}(R)$ is always less or equal to the number of tuples in R.
- If the list of attributes includes a key of R, then the number of tuples is equal to the number of tuples in R.
- $\pi_{\text{<list1>}}(\pi_{\text{<list2>}}(R)) = \pi_{\text{<list1>}}(R)$ as long as <list2> contains the attributes in <list1>

Unary Relational Operations (cont.)

- **Rename Operation**

  We may want to apply several relational algebra operations one after the other.

  - Write the operations as a single **relational algebra expression** by nesting the operations

  - We can apply one operation at a time and create **intermediate result relations**. (give names to the relations that hold the intermediate results.)

  Example: To retrieve the first name, last name, and salary of all employees who work in department number 5

  $\pi_{\text{FNAME, LNAME, SALARY}}(\sigma_{\text{DNO=5}}(\text{EMPLOYEE}))$

  $\text{DEP5\_EMPS} \leftarrow \sigma_{\text{DNO=5}}(\text{EMPLOYEE})$

  $\text{RESULT} \leftarrow \pi_{\text{FNAME, LNAME, SALARY}}(\text{DEP5\_EMPS})$

Unary Relational Operations (cont.)

Results of relational algebra expressions.
(a) $\pi_{\text{LNAME, FNAME, SALARY}}(\sigma_{\text{DNO}=5}(\text{EMPLOYEE}))$. (b) The same expression using intermediate relations and renaming of attributes.

(a)

FNAME	LNAME	SALARY
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

(b)

TEMP	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
	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

	FIRSTNAME	LASTNAME	SALARY
	John	Smith	30000
	Franklin	Wong	40000
	Ramesh	Narayan	38000
	Joyce	English	25000

Relational Algebra Operations From Set Theory

- **UNION Operation ($\cup$)**

  The result of this operation, denoted by $R \cup S$, is a relation that includes all tuples that are either in R or in S or in both R and S. Duplicate tuples are eliminated. The two operands must be "type compatible".

  Example: To retrieve the social security numbers of all employees who either work in department 5 or directly supervise an employee who works in department 5, we can use the union operation as follows:

  DEP5_EMPS $\leftarrow \sigma_{DNO=5}$ (EMPLOYEE)
  RESULT1 $\leftarrow \pi_{SSN}$(DEP5_EMPS)
  RESULT2(SSN) $\leftarrow \pi_{SUPERSSN}$(DEP5_EMPS)
  RESULT $\leftarrow$ RESULT1 $\cup$ RESULT2

Relational Algebra Operations From Set Theory

UNION Example

STUDENT	FN	LN
	Susan	Yao
	Ramesh	Shah
	Johnny	Kohler
	Barbara	Jones
	Amy	Ford
	Jimmy	Wang
	Ernest	Gilbert

INSTRUCTOR	FNAME	LNAME
	John	Smith
	Ricardo	Browne
	Susan	Yao
	Francis	Johnson
	Ramesh	Shah

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

STUDENT ∪ INSTRUCTOR

Relational Algebra Operations From Set Theory

- **Type Compatibility**
  - The operand relations $R_1(A_1, A_2, ..., A_n)$ and $R_2(B_1, B_2, ..., B_n)$ must have the same number of attributes, and the domains of corresponding attributes must be compatible; that is, $dom(A_i)=dom(B_i)$ for $i=1, 2, ..., n$.
  - The resulting relation for $R_1 \cup R_2$, $R_1 \cap R_2$, or $R_1 - R_2$ has the same attribute names as the *first* operand relation R1 (by convention).

Relational Algebra Operations From Set Theory

Two UNION compatible relations

STUDENT	FN	LN
	Susan	Yao
	Ramesh	Shah
	Johnny	Kohler
	Barbara	Jones
	Amy	Ford
	Jimmy	Wang
	Ernest	Gilbert

INSTRUCTOR	FNAME	LNAME
	John	Smith
	Ricardo	Browne
	Susan	Yao
	Francis	Johnson
	Ramesh	Shah

Relational Algebra Operations From Set Theory

- **INTERSECTION OPERATION**

The result of this operation, denoted by R ∩ S, is a relation that includes all tuples that are in both R and S. The two operands must be "type compatible"

Example: The result of the intersection operation (figure below) includes only those who are both students and instructors.

STUDENT ∩ INSTRUCTOR

FN	LN
Susan	Yao
Ramesh	Shah

Relational Algebra Operations From Set Theory

- **Set Difference (or MINUS) Operation**

The result of this operation, denoted by R - S, is a relation that includes all tuples that are in R but not in S. The two operands must be "type compatible".

Example: The figure shows the names of students who are not instructors, and the names of instructors who are not students.

STUDENT-INSTRUCTOR

FN	LN
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

Relational Algebra Operations From Set Theory

- Notice that both union and intersection are *commutative operations;* that is

  $$\mathbf{R \cup S = S \cup R, \text{ and } R \cap S = S \cap R}$$

- Both union and intersection can be treated as n-ary operations applicable to any number of relations as both are *associative operations;* that is

  $$\mathbf{R \cup (S \cup T) = (R \cup S) \cup T, \text{ and } (R \cap S) \cap T = R \cap (S \cap T)}$$

- The minus operation is *not commutative;* that is, in general

  $$\mathbf{R - S \neq S - R}$$

Relational Algebra Operations From Set Theory

- **CARTESIAN (or cross product) Operation**
  - Combine tuples from two relations in a combinatorial fashion. In general,
  - $R(A_1, A_2, \ldots, A_n) \times S(B_1, B_2, \ldots, B_m)$ is a relation Q with degree n + m attributes $Q(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$, in that order.
  - The resulting relation Q has one tuple for each combination of tuples—one from R and one from S.
  - Hence, if R has n_R tuples (denoted as $|R| = n_R$), and S has n_S tuples, then

    $| R \times S |$ will have $n_R * n_S$ tuples.
  - The two operands do NOT have to be "type compatible”

Relational Algebra Operations From Set Theory

Example:

FEMALE_EMPS $\leftarrow \sigma_{\text{SEX='F'}}$(EMPLOYEE)
EMPNAMES $\leftarrow \pi_{\text{FNAME, LNAME, SSN}}$ (FEMALE_EMPS)
EMP_DEPENDENTS $\leftarrow$ EMPNAMES x DEPENDENT

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

EMPNAMES	FNAME	LNAME	SSN
	Alicia	Zelaya	999887777
	Jennifer	Wallace	987654321
	Joyce	English	453453453

EMP_DEPENDENTS	FNAME	LNAME	SSN	ESSN	DEPENDENT_NAME	SEX	BDATE	• • •
	Alicia	Zelaya	999887777	333445555	Alice	F	1986-04-05	• • •
	Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	• • •
	Alicia	Zelaya	999887777	333445555	Joy	F	1958-05-03	• • •
	Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	• • •
	Alicia	Zelaya	999887777	123456789	Michael	M	1988-01-04	• • •
	Alicia	Zelaya	999887777	123456789	Alice	F	1988-12-30	• • •
	Alicia	Zelaya	999887777	123456789	Elizabeth	F	1967-05-05	• • •
	Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	• • •
	Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	• • •
	Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	• • •
	Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	• • •
	Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	• • •
	Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	• • •
	Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	• • •
	Joyce	English	453453453	333445555	Alice	F	1986-04-05	• • •
	Joyce	English	453453453	333445555	Theodore	M	1983-10-25	• • •
	Joyce	English	453453453	333445555	Joy	F	1958-05-03	• • •
	Joyce	English	453453453	987654321	Abner	M	1942-02-28	• • •
	Joyce	English	453453453	123456789	Michael	M	1988-01-04	• • •
	Joyce	English	453453453	123456789	Alice	F	1988-12-30	• • •
	Joyce	English	453453453	123456789	Elizabeth	F	1967-05-05	• • •

Binary Relational Operations

- **JOIN Operation**
  - The sequence of Cartesian product followed by select is used quite commonly to identify and select related tuples from two relations, a special operation, called **JOIN**. It is denoted by a $\bowtie$
  - This operation is very important for any relational database with more than a single relation, because it allows us to process relationships among relations.
  - The general form of a join operation on two relations $R(A_1, A_2, \ldots, A_n)$ and $S(B_1, B_2, \ldots, B_m)$ is:

$\bowtie$

$R \bowtie_{\text{<join condition>}} S$

where R and S can be any relations that result from general *relational algebra expressions*.

Binary Relational Operations (cont.)

Example: Suppose that we want to retrieve the name of the manager of each department. To get the manager's name, we need to combine each DEPARTMENT tuple with the EMPLOYEE tuple whose SSN value matches the MGRSSN value in the department tuple. We do this by using the join $\bowtie$ operation.

DEPT_MGR ← DEPARTMENT $\bowtie_{\text{MGRSSN=SSN}}$ **EMPLOYEE**

JOIN Operation

DEPT_MGR ← DEPARTMENT $\bowtie_{MGRSSN=SSN}$ EMPLOYEE

DEPT_MGR	DNAME	DNUMBER	MGRSSN	• • •	FNAME	MINIT	LNAME
	Research	5	333445555	• • •	Franklin	T	Wong
	Administration	4	987654321	• • •	Jennifer	S	Wallace
	Headquarters	1	888665555	• • •	James	E	Borg

Result of the JOIN operation DEPT_MGR ← DEPARTMENT $\bowtie_{MGRSSN=SSN}$ EMPLOYEE

Binary Relational Operations (cont.)

- **EQUIJOIN Operation**

  The most common use of join involves join conditions with equality comparisons only. Such a join, where the only comparison operator used is =, is called an EQUIJOIN. In the result of an EQUIJOIN we always have one or more pairs of attributes (whose names need not be identical) that have *identical values* in every tuple.

  The JOIN seen in the previous example was EQUIJOIN.

- **NATURAL JOIN Operation**

  Because one of each pair of attributes with identical values is superfluous(extra), a new operation called natural join—denoted by *—was created to get rid of the second (extra) attribute in an EQUIJOIN condition.

  The standard definition of natural join requires that the two join attributes, or each pair of corresponding join attributes, have the **same name** in both relations. If this is not the case, a renaming operation is applied first.

Equijoin: Example

Salior

Sid	Sname	Rating	Age
22	Justin	7	23
31	Petter	8	21
58	James	10	34

Reserves

Sid	Bid	Day
22	101	2001/01/23
58	103	2002/01/01

Results of Sailor $\bowtie_{\text{Salior.Sid = Reserves.Sid}}$ Reserves

Sid	Sname	Rating	Age	Sid	Bid	Day
22	Justin	7	23	22	101	2001/01/23
58	James	10	34	58	103	2002/01/01

Natural Join: Example

Salior

Sid	Sname	Rating	Age
22	Justin	7	23
31	Petter	8	21
58	James	10	34

Reserves

Sid	Bid	Day
22	101	2001/01/23
58	103	2002/01/01

Results of Sailor * Reserves

Sid	Sname	Rating	Age	Bid	Day
22	Justin	7	23	101	2001/01/23
58	James	10	34	103	2002/01/01

Complete Set of Relational Operations

- The set of operations including **select σ, project π , union ∪, set difference - , and cartesian product X** is called a complete set because any other relational algebra expression can be expressed by a combination of these five operations.

- For example:

$R \cap S = (R \cup S) - ((R - S) \cup (S - R))$

$R \bowtie_{\text{<join condition>}} S = \sigma_{\text{<join condition>}} (R \times S)$

Binary Relational Operations (cont.)

- **DIVISION Operation**

  Let R and S are two relations

  X is a set of attributes of S, Z is a set of attributes of R

  R(Z) ÷ S(X), where X subset Z.

  Let Y = Z - X (and hence Z = X ∪ Y); that is, let Y be the set of attributes of R that are not attributes of S.

- The result of DIVISION is a relation T(Y) that includes a tuple t if tuples t_R appear in R with t_R [Y] = t, and with

  t_R [X] = t_s *for every tuple* t_s in S.

- For a tuple t to appear in the result T of the DIVISION, the values in t must appear in R in combination with *every* tuple in S.

Division Operation: Example 1

R

Lecturer	Module
Brown	Compilers
Brown	Databases
Green	Prolog
Green	Databases
Lewis	Prolog
Smith	Databases

S

Subject
Prolog

R ÷ S

Lecturer
Green
Lewis

Division Operation: Example 2

R

Lecturer	Module
Brown	Compilers
Brown	Databases
Green	Prolog
Green	Databases
Lewis	Prolog
Smith	Databases

S

Subject
Databases
Prolog

R ÷ S

Lecturer
Green

Division Operation: Example 3

R

Lecturer	Module
Brown	Compilers
Brown	Databases
Green	Prolog
Green	Databases
Lewis	Prolog
Smith	Databases

S

Subject
Compilers
Prolog

R ÷ S

Lecturer

Division Operation: Exercise

A

sno	*pno*
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

B1

pno
p2

B2

pno
p2
p4

B3

pno
p1
p2
p4

Find A ÷ B1, A ÷ B2, A ÷ B3

Division Operation: Example

Retrieve the names of employees who work on all the projects that ‘John Smith’ works on

$SMITH \leftarrow \sigma_{FNAME='John' \text{ AND } LNAME='Smith'}(EMPLOYEE)$

$SMITH\_PNOS \leftarrow \pi_{PNO}(WORKS\_ON \bowtie_{ESSN=SSN} SMITH)$

$SSN\_PNOS \leftarrow \pi_{ESSN,PNO}(WORKS\_ON)$

$SSNS(SSN) \leftarrow SSN\_PNOS \div SMITH\_PNOS$

$RESULT \leftarrow \pi_{FNAME, LNAME}(SSNS * EMPLOYEE)$

Additional Relational Operations

- **Aggregate Functions and Grouping**
  - Specify mathematical **aggregate functions** on collections of values from the database.
  - Examples: Retrieve the average salary of all employees, Total number of employees.
  - Common functions applied to collections of numeric values include SUM, AVERAGE, MAXIMUM, and MINIMUM. The COUNT function is used for counting tuples or values.

Additional Relational Operations (cont.)

Use of the Functional operator $\mathcal{F}$

- $\mathcal{F}_{\text{MAX } (\textit{Salary})}$ **(Employee)** retrieves the maximum salary value from the Employee relation
- $\mathcal{F}_{\text{MIN } (\textit{Salary})}$ **(Employee)** retrieves the minimum Salary value from the Employee relation
- $\mathcal{F}_{\text{SUM } (\textit{Salary})}$ **(Employee)** retrieves the sum of the Salary from the Employee relation
- $_{\text{DNO}}\,\mathcal{F}_{\textit{COUNT (SSN), AVERAGE (Salary)}}$ **(Employee)** groups employees by DNO (department number) and computes the count of employees and average salary per department.

  [Note: count just counts the number of rows, without removing duplicates]

Aggregate Functions and Grouping-Example

Retrieve each department number, the number of employees in the department, and their average salary.

Result(DNO, NO_OF_EMPLOYEES, AVERAGE_SAL) $\leftarrow$ ($_{\text{DNO}}\,\mathcal{F}_{\text{COUNT(SSN), AVERAGE(SALARY)}}$ (EMPLOYEE))

Exercise

Q0: Retrieve the name and address of all employees who work for the department no=2.

Q1: Retrieve the name and address of all employees who work for the 'Research' department.

Q2:For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate.

Q3:Find the names of employees who work on *all the projects controlled by department number 5.*

Q4:Make a list of project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project.

Q5:List the names of all employees with two or more dependents.

Q6: Retrieve the names of employees who have no dependents.

Exercise

Q1:

$DEPT2 \leftarrow \sigma_{(Dname=\text{'Research'})}(DEPARTMENT)$

$EMPS \leftarrow (DEPT2 \bowtie_{DNUMBER=DNO} (EMPLOYEE)$

$RESULT \leftarrow \pi_{(FNAME,\ LNAME,\ ADDRESS)}(EMPS)$

Q2:

$STAFFORD\_PROJS \leftarrow \sigma_{(PLOCATION=\text{'Stafford'})}(PROJECT)$

$CONTR\_DEPT \leftarrow (STAFFORD\_PROJS \bowtie_{DNUM=DNUMBER}$
$DEPARTMENT)$

$PROJ\_DEPT\_MGR \leftarrow (CONTR\_DEPT \bowtie_{MGRSSN=SSN} EMPLOYEE)$

$RESULT \leftarrow \pi_{(PNUMBER,\ DNUM,\ LNAME,\ ADDRESS,\ BDATE)}(PROJ\_DEPT\_MGR)$

Exercise

Q3:

DEPT5_PROJS(PNO) $\leftarrow \pi_{\text{PNUMBER}}(\sigma_{\text{DNUM}=5}(\text{PROJECT}))$

EMP_PRJO(SSN, PNO) $\leftarrow \pi_{\text{ESSN, PNO}}(\text{WORKS\_ON})$

RESULT_EMP_SSNS $\leftarrow$ EMP_PRJO $\div$ DEPT5_PROJS

RESULT $\leftarrow \pi_{\text{LNAME, FNAME}}(\text{RESULT\_EMP\_SSNS} * \text{EMPLOYEE})$

Q4:

SMITHS(ESSN) $\leftarrow \pi_{\text{SSN}}(\sigma_{\text{LNAME='Smith'}}(\text{EMPLOYEE}))$

SMITH_WORKER_PROJ $\leftarrow \pi_{\text{PNO}}(\text{WORKS\_ON} * \text{SMITHS})$

MGRS $\leftarrow \pi_{\text{LNAME, DNUMBER}}(\text{EMPLOYEE} \bowtie_{\text{SSN=MGRSSN}} \text{DEPARTMENT})$

SMITH_MANAGED_DEPTS (DNUM) $\leftarrow \pi_{\text{DNUMBER}}(\sigma_{\text{LNAME= 'Smith'}}(\text{MGRS}))$

SMITH_MGR_PROJS(PNO) $\leftarrow \pi_{\text{PNUMBER}}(\text{SMITH\_MANAGED\_DEPTS} * \text{PROJECT})$

RESULT $\leftarrow$(SMITH_WORKER_PROJS $\cup$ SMITH_MGR_PROJS)

Exercise

Q5:

T1(SSN, NO_OF_DEPTS) $\leftarrow$ $_{\text{ESSN}}\,\mathcal{F}_{\text{COUNT (DEPENDENT\_NAME)}}$ (DEPENDENT)

T2 $\leftarrow \sigma_{\text{NO\_OF\_DEPTS=2}}$ (T1)

RESULT $\leftarrow \pi_{\text{LNAME, FNAME}}$ (T2 * EMPLOYEE)

Q6:

ALL_EMPS $\leftarrow \pi_{\text{SSN}}$(EMPLOYEE)

EMPS_WITH_DEPS(SSN) $\leftarrow \pi_{\text{ESSN}}$(DEPENDENT)

EMPS_WITHOUT_DEPS $\leftarrow$ (ALL_EMPS - EMPS_WITH_DEPS)

RESULT $\leftarrow \pi_{\text{LNAME, FNAME}}$ (EMPS_WITHOUT_DEPS * EMPLOYEE)