4 : Relational Algebra

IT2306 – Database Systems I

Level I - Semester 2

Detailed Syllabus

4.1. Relational algebra (RA) Operations

- Traditional Set Operations
  - Union
  - Intersection
  - Difference
  - Cartesian Product

Detailed Syllabus

4.2. Special Relational Operations

- Select or Restrict
- Project
- Join
- Different types of join
  - theta join
  - equi-join
  - natural join
  - outer joins
- Division
- Minimal set of operations
- Simple and Complex queries using RA.

4.1. Relational algebra (RA) Operations

Relational Algebra

- Relational algebra is a formal language associated with the relational model.
- The basic set of operations of the formal relational model is the relational algebra.
- Informally, relational algebra is a (high-level) procedural language.

Relational Algebra

- Relational algebra operations work on one or more relations to define another relation without changing the original relations.
- Since the result of a relational algebra is a relation, it can become operand to another relational algebra operation.
- This allows operations to be nested, just as in arithmetic. [This property is called <u>**closure**</u>.]

Sequence of Operations

- For most queries we may want to apply several relational algebra operations one after the other.

- We can write the operation in two ways:

    1. A single relational algebra expression by nesting the operations (in-line expression)
    2. Apply one operation at a time and create intermediate result relations

Sequence of Operations

- Example:
- Retrieve the first name, last name and salary of all employees who work in department no.5

1. A single relational algebra expression by nesting the operations
   - $\pi_{\text{Fname,Lname,Salary}} (\sigma_{\text{Dno=5}} (\text{Employee}))$

2. As a sequence of expressions giving names to the intermediate results
   - DEP5_EMPS $\leftarrow \sigma_{\text{Dno=5}}$ (Employee)
   - RESULT $\leftarrow \pi_{\text{Fname,Lname,Salary}}$ (DEP5_EMPS)

Relational Algebra

- Relational algebra operations enable a user to specify basic retrieval requests as *relational algebra expressions*.
- 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).

Relational Algebra Operations

- Relational algebra operations can be divided into two groups:

1. Traditional set operations (from mathematical set theory)
   - Union, Intersection, Set Difference and Cartesian Product
2. Operations specific to relational databases
   - Select, Project, Join, etc.

Traditional Set Operations – Union

- Union of relations R and S is denoted by R ∪ S.

- The result of this operation 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.

- R and S must be union compatible. (i.e. the two relations must have the same type of tuples)

- If the number of tuples in the two relations are I and J, the result of union will contain a maximum of (I+J) tuples.

Traditional Set Operations

- Union, intersection, set difference and Cartesian product are binary operations (i.e. applied to two sets of tuples).

- To apply union, intersection or set difference on two sets of tuples, they must be union compatible (or type compatible).
- The two relations $R(A_1, A_2, \ldots, A_n)$ and $S(B_1, B_2, \ldots, B_n)$ are said to be union compatible if R and S contain the tuples of same type.
  - Both R and S have the same degree n and
  - $dom(A_i) = dom(B_i)$ for $1 \le i \le n$.

Traditional Set Operations – Union

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

UNION

STUDENT ∪ INSTRUCTOR

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

Traditional Set Operations – Union

Example:

- Retrieve the EmpNo of all employees who either work in department 5 or directly supervise an employee who works in department 5.

- DEP5_EMPS ← $\sigma_{\text{Dno=5}}$ (Employee)
- RESULT1 ← π_{EmpNo} (DEP5_EMPS)
- RESULT2(EmpNo) ← π_{SuperNo}(DEP5_EMPS)
- RESULT ← RESULT1 ∪ RESULT2

Traditional Set Operations – Set Difference

- Also called as MINUS operation or EXCEPT operation.

- Set difference between the two relations R and S us denoted by R – S

- The result of this operation is a relation that includes all tuples that are in R but not in S.

- R and S must be union compatible

Traditional Set Operations – Set Difference

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

STUDENT - INSTRUCTOR

Fn	Ln
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR - STUDENT

Fname	Lname
John	Smith
Ricardo	Browne
Francis	Johnson

Traditional Set Operations – Set Difference

Example:

- Retrieve the EmpNo of all employees who directly supervise an employee who works in department 5 but does not work in department 5.

- DEP5_EMPS ← $\sigma_{Dno=5}$ (Employee)
- ALL_SUP ← $\pi_{SuperNo}$ (Employee)
- RESULT ← ALL_SUP – DEP5_EMPS

Traditional Set Operations – Intersection

- Intersection between the two relations R and S us denoted by $R \cap S$

- The result of this operation is a relation that includes all tuples that are in both R and S.

- R and S must be union compatible

- This can be expressed using basic operations:
    - $R \cap S = R - (R - S)$

Traditional Set Operations – Intersection

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

Intersection

STUDENT ∩ INSTRUCTOR

Fn	Ln
Susan	Yao
Ramesh	Shah

Traditional Set Operations – Intersection

Example:

- Retrieve the EmpNo of all employees who directly supervise an employee who works in department 5 and also work in department 5.

- DEP5_EMPS ← $\sigma_{\text{Dno=5}}$ (Employee)
- ALL_SUP ← π_{SuperNo} (Employee)
- RESULT ← ALL_SUP ∩ DEP5_EMPS

Traditional Set Operations

- Both union and intersection are commutative
  - $R \cup S = S \cup R$ and $R \cap S = S \cap R$

- Both union and intersection are associative. Therefore they can be applied to any number of relations.
  - $R \cup (S \cup T) = (R \cup S) \cup T$ and
  - $R \cap (S \cap T) = (R \cap S) \cap T$

- Difference operation is not commutative
  - $R – S \neq S - R$

Traditional Set Operations – Cartesian Product

- Also called as Cross Product or Cross Join
- Cross product between the relations R and S is denoted as R x S
- The result combine every tuple from one relation with every tuple from the other relation.
- R and S do not have to be union compatible.

Traditional Set Operations – Cartesian Product

- The result of $R(A_1, A_2, \ldots, A_n) \times S(B_1, B_2, \ldots, B_m)$ is a relation Q with n + m attributes in the order:
  - $Q (A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$

  - Q has one tuple for each combination of tuples from tuples from R and S.
  - If R and S have n_R and n_S tuples respectively then Q will have $n_R * n_S$ tuples.

Traditional Set Operations – Cartesian Product

EMPNAMES

Fname	Lname	Emp_no
Alicia	Zelaya	999887777
Jennifer	Wallace	987654321

DEPS

D_Emp_no	Dependent_name
333445555	Alice
333445555	Theodore
333445555	Joy
987654321	Abner

EMPNAMES x DEPS

Fname	Lname	Emp_no	D_Emp_no	Dependent_name
Alicia	Zelaya	999887777	333445555	Alice
Alicia	Zelaya	999887777	333445555	Theodore
Alicia	Zelaya	999887777	333445555	Joy
Alicia	Zelaya	999887777	987654321	Abner
Jennifer	Wallace	987654321	333445555	Alice
Jennifer	Wallace	987654321	333445555	Theodore
Jennifer	Wallace	987654321	333445555	Joy
Jennifer	Wallace	987654321	987654321	Abner

Traditional Set Operations – Cartesian Product

Example

- Retrieve a list of names of each female employee's dependents.

- EMPNAMES← $\pi_{\text{Fname, Lname, Emp\_no}}(\sigma_{\text{Gender='F'}}$ (Employee))
- DEPS (D_Emp_no, D_name)← $\pi_{\text{Emp\_no, Dependent\_name}}$ (Dependent)
- EMP_DEPS← EMPNAMES x DEPS
- DEP_INFO← $\sigma_{\text{Emp\_no=D\_Emp\_no}}$ (EMP_DEPS)

Traditional Set Operations – Cartesian Product

- n-ary CARTESIAN PRODUCT operation produces new tuples by concatenating all possible combinations of tuples from n underlying relations.

- The Cartesian product is usually followed by a selection that matches values of attributes coming from the component relations.

4.2. Special Relational Operations

RA Operation – Selection

- Also called as restriction
- SELECT operation on the relation R based on a selection condition is denoted by:
  - $\sigma_{\text{<selection condition>}}(R)$
- Result of the operation a subset of the tuples from relation R that satisfies the **selection condition.**
- Selection condition is a Boolean expression specified on the attributes of the relation.

RA Operation – Selection

- Select creates a horizontal partition of the relation, filtering out tuples that does not satisfy the condition.
- The resulting relation has the same attributes as R (has the same degree as the R)
- The resulting relation contains the same or less number of tuples than R.
- The fraction of tuples selected by a selection condition is referred to as the **selectivity** of the condition.
  - Selectivity = $|\sigma_C(R)| / |R|$

RA Operation – Selection

EMPLOYEE

Fname	Minit	Lname	Emp_no	Bdate	Address	Sex	Salary	Super_ssn	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	1968-01-19	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

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

Fname	Minit	Lname	Emp_no	Bdate	Address	Sex	Salary	Super_ssn	Dno
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

RA Operation – Selection

Example:

- Retrieve all employees who either work in department 4 and make over $25,000 per year, or work in department 5 and make over $30,000

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

RA Operation – Selection

- SELECT is a unary operation
- SELECT operation is commutative.
  - I.e. the results of applies same set of SELECTs in any order gives the same result.
  - $\sigma_{<cond1>}(\sigma_{<cond2>}(R)) = \sigma_{<cond2>}(\sigma_{<cond1>}(R))$
- A sequence of SELECTs can be combined into a single SELECT operation with a conjunctive (AND) condition
  - Ex:
  - $\sigma_{<c1>}(\sigma_{<c2>}(\ldots(\sigma_{<cn>}(R))\ldots)) = \sigma_{<c1>AND<c2>AND\ldots<cn>}(R))$

RA Operation – Projection

- PROJECT operation on the relation R based on a list of attributes is denoted by:
  - $\pi_{\text{<attribute list>}}(R)$
- This operation project the relation over only the attributes specified in the <attribute list>.
- The result contains a set of tuples corresponding to the tuples in R extracting only the values of specified attributes and eliminating duplicates.

RA Operation – Projection

- Project creates a vertical partition of the relation, filtering out attributes for the tuples that are not specified in the projection list.
- The degree of the resulting relation is equal to the number of attributes specified in <attribute list>.
- The result of the PROJECT operation is a set of *distinct* tuples.
    - The PROJECT operation removes any duplicate tuples.
    - The resulting relation contains the same or less number of tuples than R.

RA Operation – Projection

- PROJECT is a unary operation.
- PROJECT operation is NOT commutative.
- A sequence of PROJECTs can be re-written as follows:
  - $\pi_{<list1>}(\pi_{<list2>}(\ldots(\pi_{<listn>}(R))\ldots)) = \pi_{<list1>}(R)$

RA Operation – Projection

EMPLOYEE

Fname	Minit	Lname	Emp_no	Bdate	Address	Sex	Salary	Super_ssn	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	1968-01-19	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

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

$\pi_{\text{Lname, Fname, Salary}}$ (EMPLOYEE)

Sex	Salary
M	30000
M	40000
F	25000
F	43000
M	38000
M	25000
M	55000

$\pi_{\text{Sex, Salary}}$ (EMPLOYEE)

RA Operation - Join

- Join operation between relations R and S is denoted by $R \bowtie_{\text{<join condition>}} S$.

- The operation is generate a set of tuples by combining tuples from two related relations R and S into a single tuples where the two tuples satisfies the join condition.

- Join operation allows to process relationships among relations.

RA Operation - Join

- The join operation $R(A_1, A_2, \dots, A_n) \bowtie_C S(B_1, B_2, \dots, B_m)$ produce a new relation $Q (A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$.
- Degree of Q is equal to the sum of the degree values of R and S.
- If the number of tuples in R and S are n_R and n_S respectively; the number of tuples in Q n_Q is less than or equal to $(n_R * n_S)$.

RA Operation - Join

Example:

- Retrieve the name of the manager of each department.

- DEPT_MGR $\leftarrow$ DEPARTMENT $_{\text{Mgr\_emp\_no=Emp\_no}} \bowtie$ EMPLOYEE
- RESULT $\leftarrow \pi_{\text{Dname, Lname, Fname}}$(DEPT_MGR)

RA Operation - Join

- JOIN operation can be specified by the basic relational algebra operations:

  - $R \bowtie_C S = \sigma_C (R \times S)$

- Example:
- $EMP\_DEPS \leftarrow EMPNAMES \times DEPS$
- $DEP\_INFO \leftarrow \sigma_{Emp\_no=D\_Emp\_no} (EMP\_DEPS)$

- $DEP\_INFO \leftarrow EMPNAMES \bowtie_{Emp\_no=D\_Emp\_no} DEPS$

RA Operation - Join

Theta Join

- A theta join is a join operation with a join condition of the form:

  <cond> AND <cond> AND … AND <cond>

  and

  each <cond> is of the form $A_i \Theta B_i$ where Θ is one of the comparison operators $\{=, <, \leq, >, \geq, \neq\}$

- Tuples whose join attributes are NULL or for which the join condition is FALSE do not appear in the result.

RA Operation - Join

Equijoin

- A equijoin is a join operation involves a join condition with equality comparisons only.

- For the general form of the join condition <cond> AND <cond> AND … AND <cond> ; each <cond> is of the form $A_i = B_i$

- The result of an EQUIJOIN always contains one or more pairs of attributes (for attribute pairs used in the join condition) that have identical values in every tuple.

RA Operation - Join

Equijoin

- Example
- Combine each PROJECT tuple with the DEPARTMENT tuple that controls the project.
- Assuming the attribute name of department number in PROJECT and DEPARTMENT are Dnum and Dnumber respectively.

- RESULT $\leftarrow$ PROJECT $\bowtie_{\text{Dnum=Dnumber}}$ DEPARTMENT

RA Operation - Join

Natural Join

- Denoted by *
- Natural join is performed only on a pair of relations with the join attributes have the same name in the two relations.
- This operation is used to get rid of the second attribute in an equijoin condition.
- All the attribute pairs with the same name will be considered as join attributes.

RA Operation - Join

Natural Join

- Example
- Combine each PROJECT tuple with the DEPARTMENT tuple that controls the project.
  - Since the attribute name for department number in PROJECT and DEPARTMENT are different, we can rename the attributes in DEPARTMENT and before applying Natural join operation.

- RESULT $\leftarrow$ PROJECT * $\rho_{(\text{Dname, Dnum, } \ldots)}$ DEPARTMENT

or

- DEPT $\leftarrow$ $\rho_{(\text{Dname, Dnum, } \ldots)}$ DEPARTMENT
- RESULT $\leftarrow$ PROJECT * DEPT

RA Operation - Join

Outer Join

- Outer join operation includes tuples in the results that do not have matching values in the join attribute.
- R ⟕ S
  - (Left) outer join is join in which tuples from R that do not have matching values for the common attributes of S are also included in result relation.

RA Operation - Join

Outer Join

- Example:
- Retrieve names of managers of the departments names of departments that they manage

- Temp $\leftarrow$ Employee $⟕_{emp\_no=mgr\_emp\_no}$ Department
- Result $\leftarrow$ $\pi_{Fname,Lname,Dname}$ (Temp)

RA Operation - Division

- Division between two relations R and S is denoted by R ÷ S

- Division operation results in a relation over the attributes C that consists of set of tuples from R that match combination of every tuple in S.

RA Operation - Division

Example:

- Retrieve the names of employees who work on <u>all</u> projects that ‘Sunil Silva’ works on.

- Sunil ← $\sigma_{\text{Fname = ‘Sunil’ AND Lname=‘Silva’}}$ (Employee)
- Sunil_Pnos← π_{Pno} (Works_on $\bowtie_{\text{e\_id= emp\_id}}$ Sunil)
- Empid_Pnos ← $\pi_{\text{E\_id, Pno}}$ (Works_on)
- Empids ← Empid_Pnos ÷ Sunil_Pnos
- RESULT ← $\pi_{\text{Fname, Lname}}$ (Empids * Employee)

Minimal Set of RA Operations

- There are five basic operations in relational algebra which are required to perform most of the data retrieval operations.
  - Selection
  - Projection
  - Cartesian product
  - Union
  - Set difference
- Any other operations such as Join, Intersection and Division operation can be expressed in terms of these basic operations.
- Therefore these set of RA operations are called as a complete set.

Simple and Complex Queries using RA

Company DB Schema

EMPLOYEE

Fname	Minit	Lname	Eid	Bdate	Address	Sex	Salary	Super_eid	Dno

DEPARTMENT

Dname	Dnumber	Mgr_eid	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Eid	Pno	Hours

DEPENDENT

Eid	Dependent_name	Sex	Bdate	Relationship

Example 1

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

With named intermediate results:

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

RESEARCH_EMPS

$\leftarrow (\text{RESEARCH\_DEPT} \bowtie_{\text{Dnumber=Dno}} \text{EMPLOYEE})$

$\text{RESULT} \leftarrow \pi_{\text{Fname, Lname, Address}}(\text{RESEARCH\_EMPS})$

- As a single in-line expression:

$\pi_{\text{Fname, Lname, Address}} (\sigma_{\text{Dname='Research'}}(\text{DEPARTMENT} \bowtie_{\text{Dnumber=Dno}} (\text{EMPLOYEE}))$

Example 2

For every project located in ‘Kandy’, list the project number, the controlling department number, and the department manager’s last name, address, and birth date

$\text{KANDY\_PROJS} \leftarrow \sigma_{\text{Plocation='Kandy'}}(\text{PROJECT})$

$\text{CONTR\_DEPTS} \leftarrow (\text{KANDY\_PROJS} \bowtie_{\text{Dnum=Dnumber}} \text{DEPARTMENT})$

$\text{PROJ\_DEPT\_MGRS} \leftarrow (\text{CONTR\_DEPTS} \bowtie_{\text{Mgr\_eid=eid}} \text{EMPLOYEE})$

$\text{RESULT} \leftarrow \pi_{\text{Pnumber, Dnum, Lname, Address, Bdate}}(\text{PROJ\_DEPT\_MGRS})$

Example 3

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

DEP5_PROJS $\leftarrow \pi_{\text{Pnumber}}(\sigma_{\text{Dnum=5}}$ (PROJECT))

EMP_PROJ $\leftarrow \rho_{\text{(Eid, Pnumber)}}$ ($\pi_{\text{Eid, Pno}}$(WORKS_ON))

RESULT_EMP_EIDS $\leftarrow$ EMP_PROJ $\div$ DEP5_PROJS

RESULT $\leftarrow \pi_{\text{Lname, F\_name}}$(RESULT_EMP_EIDS * EMPLOYEE)

Example 4

Make a list of project numbers for projects that involve an employee whose last name is ‘Silva’, either as a worker or as a manager of the department that controls the project.

$SIL\_EIDS \leftarrow \pi_{Eid}(\sigma_{Lname='Silva'}(EMPLOYEE))$

$SIL\_WORKER\_PROJS \leftarrow \pi_{Pno}(WORKS\_ON * SIL\_EIDS)$

$SIL\_MANAGED\_DEPS(Dnum)$

$\leftarrow \pi_{Dnumber}(SIL\_EIDS \bowtie_{Eid=Mgr\_eid} (DEPARTMENT))$

$SIL\_MGR\_PROJS(Pno)$

$\leftarrow \pi_{Pnumber}(SIL\_MANAGED\_DEPS * PROJECT)$

$RESULT \leftarrow (SIL\_WORKER\_PROJS \cup SIL\_MGR\_PROJS)$

Example 5

List the names of all employees who have no dependents.

ALL_EMPS $\leftarrow \pi_{Eid}$(EMPLOYEE)

EMPS_WITH_DEPS$\leftarrow \pi_{Eid}$(DEPENDENT)

EMPS_WITHOUT_DEPS $\leftarrow$ (ALL_EMPS – EMPS_WITH_DEPS)

RESULT $\leftarrow \pi_{Lname, Fname}$(EMPS_WITHOUT_DEPS * EMPLOYEE)

Example 6

List the names of all managers who have at least one dependents.

$MGRS\ (Eid) \leftarrow \pi_{Mgr\_eid}\ (DEPARTMENT)$

$EMPS\_WITH\_DEPS \leftarrow \pi_{Eid}\ (DEPENDENT)$

$MGRS\_WITH\_DEPS \leftarrow (MGRS \cap EMPS\_WITH\_DEPS)$

$RESULT \leftarrow \pi_{Lname,\ Fname}(MGRS\_WITH\_DEPS\ *\ EMPLOYEE)$