

## Database Management System – 21 (Additional Relational Operations)

Ajay James  
Asst. Prof in CSE  
Government Engineering College Thrissur

### Outline

- Aggregate Functions
- Recursive Closure operation
- Outer Join
- Outer Union
- Exercise 4

## Aggregate Functions and Grouping

- To specify mathematical **aggregate functions** on collections of values from the database
- Examples
  - Retrieve the average or total salary of all employees
  - Total number of employee tuples
- SUM, AVERAGE, MAXIMUM, and MINIMUM
- COUNT function is used for counting tuples or values

## Aggregate Functions and Grouping

<grouping attributes>  $\mathcal{F}$  <function list> (R)

- To retrieve each department number, the number of employees in the department, and their average salary

Dno  $\mathcal{F}$  COUNT Ssn, AVERAGE Salary (EMPLOYEE)

Dno	Count_ssn	Average_salary
5	4	33250
4	3	31000
1	1	55000

## Example

$\rho_R(\text{Dno}, \text{No\_of\_employees}, \text{Average\_sal}) (\text{Dno } \mathcal{F} \text{ COUNT Ssn, AVERAGE Salary (EMPLOYEE)})$

R

Dno	No_of_employees	Average_sal
5	4	33250
4	3	31000
1	1	55000

$\mathcal{F} \text{ COUNT Ssn, AVERAGE Salary (EMPLOYEE)}$

Count_ssn	Average_salary
8	35125

## Examples

- $\mathcal{F}_{\text{MAX Salary}} (\text{Employee})$ 
  - retrieves the maximum salary value from the Employee relation
- $\mathcal{F}_{\text{MIN Salary}} (\text{Employee})$ 
  - retrieves the minimum Salary value from the Employee relation
- $\mathcal{F}_{\text{SUM Salary}} (\text{Employee})$ 
  - retrieves the sum of the Salary from the Employee relation

## Recursive Closure Operations

- Operation that cannot be specified in the basic original relational algebra is **recursive closure**
- Example
  - To retrieve all SUPERVISEES of an EMPLOYEE  $e$  at all levels—that is, all EMPLOYEE  $e'$  directly supervised by  $e$ ; all employees  $e''$  directly supervised by each employee  $e'$ ; all employees  $e'''$  directly supervised by each employee  $e''$ ; and so on
- It is possible to retrieve employees at each level and then take their union
- We cannot, in general, specify a query such as “retrieve the supervisees of ‘James Borg’ at all levels” without utilizing a looping mechanism
- SQL3 standard includes syntax for recursive closure

## Example

**SUPERVISION**

(Borg's Ssn is 888665555)  
(Ssn) (Super\_ssn)

Ssn1	Ssn2
123456789	333445555
333445555	888665555
999887777	987654321
987654321	888665555
666884444	333445555
453453453	333445555
987987987	987654321
888665555	null

**RESULT1**

Ssn
333445555
987654321

(Supervised by Borg)

**RESULT2**

Ssn
123456789
999887777
666884444
453453453
987987987

(Supervised by  
Borg's subordinates)

**RESULT**

Ssn
123456789
999887777
666884444
453453453
987987987
333445555
987654321

(RESULT1  $\cup$  RESULT2)

## OUTER JOIN Operations

- In NATURAL JOIN tuples without a *matching (or related)* tuple are eliminated from the join result
  - Tuples with null in the join attributes are also eliminated
  - Loss of information
- Outer joins, can be used when we want to keep all the tuples in R, or all those in S, or all those in both relations in the result of the join, regardless of whether or not they have matching tuples in the other relation
- Left outer join** operation keeps every tuple in the *first* or *left* relation R in  $R \bowtie S$ 
  - if no matching tuple is found in S, then the attributes of S in the join result are filled or “padded” with null values
- Right outer join**, keeps every tuple in the *second* or right relation S in the result of  $R \bowtie S$
- Full outer join**, denoted by  $\bowtie$ : keeps all tuples in both the left and the right relations when no matching tuples are found, padding them with null values as needed

EMPLOYEE

Fname	Minit	Lname	Ssn	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

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT\_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

- EMPLOYEE \* DEPARTMENT

## Example

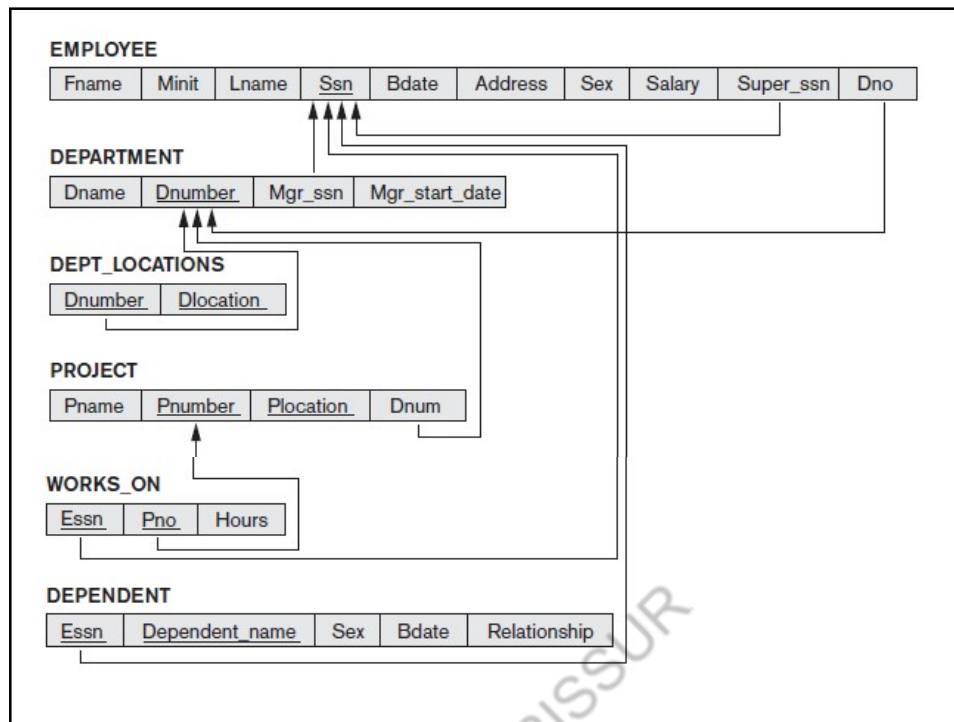
$TEMP \leftarrow (EMPLOYEE \bowtie_{Ssn=Mgr\_ssn} DEPARTMENT)$   
 $RESULT \leftarrow \pi_{Fname, Minit, Lname, Dname}(TEMP)$

### RESULT

Fname	Minit	Lname	Dname
John	B	Smith	NULL
Franklin	T	Wong	Research
Alicia	J	Zelaya	NULL
Jennifer	S	Wallace	Administration
Ramesh	K	Narayan	NULL
Joyce	A	English	NULL
Ahmad	V	Jabbar	NULL
James	E	Borg	Headquarters

## OUTER UNION Operation

- To take the union of tuples from two relations if the relations are *not union compatible*
- Union of tuples in two relations  $R(X, Y)$  and  $S(X, Z)$  that are **partially compatible**
  - Only some of their attributes, say  $X$ , are union compatible
- Attributes that are union compatible are represented only once in the result
- Those attributes that are not union compatible from either relation are also kept in the result relation  $T(X, Y, Z)$
- Example:**
  - STUDENT(Name, SSN, Department, Advisor)
  - INSTRUCTOR(Name, SSN, Department, Rank)
  - STUDENT\_OR\_INSTRUCTOR will have the following attributes:  
**STUDENT\_OR\_INSTRUCTOR (Name, SSN, Department, Advisor, Rank)**



### Exercise – 4 (Relational algebra queries)

1. Retrieve the name and address of all employees who work for the 'Research' department and whose salary is greater than 25000
2. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name and salary
3. Find the names of employees who work on all the projects controlled by department number 5 and whose salary is greater than 25000.

## Exercise – 4 (Relational algebra queries)

4. 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
5. List the names of all employees with two or more dependents
6. List the names of managers who have at least one dependent.

## Airline Reservation System

### AIRPORT

<u>Airport_code</u>	Name	City	State
---------------------	------	------	-------

### FLIGHT

<u>Flight_number</u>	Airline	Weekdays
----------------------	---------	----------

### FLIGHT\_LEG

<u>Flight_number</u>	<u>Leg_number</u>	Departure_airport_code	Scheduled_departure_time
		Arrival_airport_code	Scheduled_arrival_time

### LEG\_INSTANCE

<u>Flight_number</u>	<u>Leg_number</u>	<u>Date</u>	Number_of_available_seats	Airplane_id	
		Departure airport code	Departure time	Arrival airport code	Arrival time



# Airline Reservation System

## FARE

<u>Flight_number</u>	<u>Fare_code</u>	Amount	Restrictions
----------------------	------------------	--------	--------------

## AIRPLANE\_TYPE

<u>Airplane_type_name</u>	Max_seats	Company
---------------------------	-----------	---------

## CAN LAND

<u>Airplane_type_name</u>	<u>Airport_code</u>
---------------------------	---------------------

## AIRPLANE

<u>Airplane_id</u>	Total_number_of_seats	Airplane_type
--------------------	-----------------------	---------------

## SEAT RESERVATION

<u>Flight_number</u>	<u>Leg_number</u>	<u>Date</u>	<u>Seat_number</u>	<u>Customer_name</u>	<u>Customer_phone</u>
----------------------	-------------------	-------------	--------------------	----------------------	-----------------------

## Exercise - 4

7. For each flight, list the flight number, the departure airport for the first leg of the flight, and the arrival airport for the last leg of the flight.
8. List the flight numbers and weekdays of all flights or flight legs that depart from Houston Intercontinental Airport (airport code 'iah') and arrive in Los Angeles International Airport (airport code 'lax').
9. List the flight number, departure airport code, scheduled departure time, arrival airport code, scheduled arrival time, and weekdays of all flights or flight legs that depart from some airport in the city of Houston and arrive at some airport in the city of Los Angeles.
10. List all fare information for flight number 'co197'.
11. Retrieve the number of available seats for flight number 'co197' on '2009-10-09'.

## Reference

- Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education 6<sup>th</sup> edition and 7<sup>th</sup> edition

Thank you