# LECTURE 3

Unit 2

# INTRODUCTION

- Relational algebra is the basic set of operations for the relational model.
- These operations enable a user to specify **basic retrieval requests** (or **queries**).
- The result of an operation is a *new relation*, which may have been formed from one or more *input* relations.

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

# OPERATIONS – 1/2

- **Unary Relational Operations**
  - SELECT (symbol: σ (sigma))
  - PROJECT (symbol: π (pi))
  - RENAME (symbol: ρ (rho))

- **Relational Algebra Operations From Set Theory**
  - UNION ( U ), INTERSECTION ( ∩ ), DIFFERENCE (or MINUS, – )
  - CARTESIAN PRODUCT ( **x** )

# OPERATIONS – 1/2

- **Binary Relational Operations**
  - JOIN
  - EQUIJOIN
  - NATURAL JOIN
- **Additional Relational Operations**
  - AGGREGATE FUNCTIONS (These compute summary of information: for example, SUM, COUNT, AVG, MIN, MAX)
  - LEFT JOIN, RIGHT JOIN, OUTER JOIN, OUTER UNION

# UNARY RELATIONAL OPERATIONS

- The **relational algebra** is a set of operators that take and return relations

- Unary operations take one relation, and return one relation:
  - SELECT operation
  - PROJECT operation
  - RENAME operation

# SELECT OPERATION – 1/4

- The SELECT operation (denoted by σ (sigma)) is used to select a *subset* of the tuples from a relation based on a **selection condition**.
- The selection condition acts as a **filter.**
- Keeps only those tuples that satisfy the qualifying condition.
- Tuples satisfying the condition are *selected* whereas the other tuples are discarded (*filtered out*).

# SELECT OPERATION – 2/4

- Examples:
- Select the EMPLOYEE tuples whose department number is 4:

  $\sigma_{DNO = 4}(EMPLOYEE)$

- Select the employee tuples whose salary is greater than $30,000:

  $\sigma_{SALARY > 30,000}(EMPLOYEE)$

# SELECT OPERATION – 3/4

- In general, the *select* operation is denoted by

  **σ <selection condition>(R)**

where

- The symbol σ (sigma) is used to denote the *select* operator
- The selection condition is a Boolean (conditional) expression specified on the attributes of relation R

# SELECT OPERATION – 4/4

- Tuples that make the condition **true** are selected and appear in the result of the operation

- Tuples that make the condition **false** are filtered out discarded from the result of the operation

# PROPERTIES OF SELECT – 1/2

- The SELECT operation

  **σ <selection condition>(R)**

  produces a relation S that has the same schema (same attributes) as R

- SELECT σ is **commutative**:

  **σ <condition1>(σ < condition2> (R))**

  **≡**

  **σ <condition2> (σ < condition1> (R))**

# PROPERTIES OF SELECT – 2/2

- A **cascade** of SELECT operations may be replaced by **a single selection** with a **conjunction** of all the conditions:

**σ<cond1>(σ< cond2> (σ<cond3>(R))**

≡

**σ <cond1> AND < cond2> AND < cond3>(R)))**

- The number of tuples in the result of a SELECT is less than (or equal to) the number of tuples in the input relation R
- For any selection-condition c, we have

$$| \sigma_{<c>} (R) | \leq | R |$$

# THE PROJECT OPERATION – 1/3

- The PROJECT operation , selects certain columns from a relation and discards the columns that are not in the PROJECT list

- The PROJECT operation is denoted by:

$$\pi_{\text{<attribute list>}}(R)$$

Where **attribute-list** is a list of attributes from the relation R

# THE PROJECT OPERATION – 2/3

- The **degree** of the relation resulting from a PROJECT operation is equal to the number of attributes in **attribute-list**.

- If only non-key attributes appear in an attribute-list, duplicate tuples are likely to occur. The PROJECT operation, however, removes any duplicate tuples –this is called **duplicate elimination**

- The PROJECT operation **is not commutative**

# THE PROJECT OPERATION – 3/3

Results of SELECT and PROJECT operations.

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

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

(c) $\pi_{SEX,\ SALARY}(EMPLOYEE)$.

(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 | S | Wallace | 987654321 | 1941-06-20 | 291 Berry,Bellaire,TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | 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 |

# SET OPERATIONS

- Two relations $R_1$ and $R_2$ are said to be **union compatible** if they have the **same degree** and all their **attributes** (correspondingly) have the **same domain.**

- **The UNION, INTERSECTION, and SET DIFFERENCE operations are applicable on union compatible relations**

- The **resulting relation has the same attribute names as the first relation**

# THE UNION OPERATION – 1/2

- The result of UNION operation on two relations, $R_1$ and $R_2$, is a relation, $R_3$, that **includes all tuples that are either in $R_1$, or in $R_2$, or in both $R_1$ and $R_2$.**

- The UNION operation is denoted by:

$$R_3 = R_1 \cup R_2$$

- The UNION operation **eliminates duplicate tuples**

# THE UNION OPERATION – 2/2

**RESULT = RESULT1 ∪ RESULT2**

| RESULT1 | SSN |
|---|---|
| | 123456789 |
| | 333445555 |
| | 666884444 |
| | 453453453 |

| RESULT2 | SSN |
|---|---|
| | 333445555 |
| | 888665555 |

| RESULT | SSN |
|---|---|
| | 123456789 |
| | 333445555 |
| | 666884444 |
| | 453453453 |
| | 888665555 |

# THE INTERSECTION OPERATION

- The result of INTERSECTION operation on two relations, $R_1$ and $R_2$, is a relation, $R_3$, that **includes all tuples that are in both $R_1$ and $R_2$.**

- The INTERSECTION operation is denoted by:

$$R_3 = R_1 \cap R_2$$

- The both UNION and INTERSECTION operations are **commutative** and **associative** operations

# THE SET DIFFERENCE OPERATION – 1/2

- The result of SET DIFFERENCE operation on two relations, $R_1$ and $R_2$, is a relation, $R_3$, that **includes all tuples that are in $R_1$ but not in $R_2$**.

- The SET DIFFERENCE operation is denoted by:

$$R_3 = R_1 - R_2$$

- The SET DIFFERENCE (or MINUS) operation **is not commutative**

# THE SET DIFFERENCE OPERATION – 2/2

(a) Two union-compatible relations (b) STUDENT ∪ INSTRUCTOR.

(c) STUDENT ∩ INSTRUCTOR (d) STUDENT – INSTRUCTOR.

(e) INSTRUCTOR – STUDENT

(a)

| 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 |

(b)

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

(c)

| FN | LN |
|---|---|
| Susan | Yao |
| Ramesh | Shah |

(d)

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

(e)

| FNAME | LNAME |
|---|---|
| John | Smith |
| Ricardo | Browne |
| Francis | Johnson |

# RENAME OPERATION – 1/3

- We can write a single relational algebra expression as follows:

$$\Pi_{FName,LName,Salary}(\sigma_{DNo=5}(EMPLOYEE))$$

- We can explicitly show the sequence of operations, giving a name to each intermediate relation:

$$DEPS\_EMPS \leftarrow \sigma_{DNo=5}(EMPLOYEE)$$

$$RESULT \leftarrow \Pi_{FName,LName,Salary}(DEPS\_EMPS)$$

# RENAME OPERATION – 2/3

- Rename the attributes in the intermediate and result relations

DEPS_EMPS← $\sigma_{DNo=5}$(EMPLOYEE)

**R(FirstName, LastName, Salary) ← $\Pi_{FName,LName,Salary}$(DEPS_EMPS)**

- Define a formal RENAME operation-which can rename either the relation name or the attribute names, or both

# RENAME OPERATION – 3/3

- Has three forms:
  - **Rename both relation and attributes**

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

  - **Rename only relation**

    $\rho_{S}(R)$

  - **Rename only attributes**

    $\rho_{(B1,B2,\ldots Bn)}(R)$

# CARTESIAN PRODUCT OPERATION – 1/3

- This operation (also known as CROSS PRODUCT or CROSS JOIN) denoted by:

$$R_3 = R_1 \times R_2$$

  The resulting relation, $R_3$, includes all combined tuples from two relations $R_1$ and $R_2$

- Degree ($R_3$) = Degree ($R_1$) + Degree ($R_2$)

- $|R_3| = |R_1| \times |R_2|$

# CARTESIAN PRODUCT OPERATION – 2/3

The CARTESIAN PRODUCT (CROSS PRODUCT)

| 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 | • • • |

# CARTESIAN PRODUCT OPERATION – 3/3

The CARTESIAN PRODUCT (CROSS PRODUCT)

| ACTUAL_DEPENDENTS | FNAME | LNAME | SSN | ESSN | DEPENDENT_NAME | SEX | BDATE | • • • |
|---|---|---|---|---|---|---|---|---|
| | Jennifer | Wallace | 987654321 | 987654321 | Abner | M | 1942-02-28 | • • • |

| RESULT | FNAME | LNAME | DEPENDENT_NAME |
|---|---|---|---|
| | Jennifer | Wallace | Abner |

# BINARY RELATIONAL OPERATIONS

- An important operation for any relational database is the JOIN operation, because it enables us to combine **related tuples** from two relations into single tuple

- The JOIN operation is denoted by:

$$R_3 = R_1 \bowtie_{\text{<join condition>}} R_2$$

- The degree of resulting relation is

$$\text{degree}(R_1) + \text{degree}(R_2)$$

# THE JOIN OPERATION – 1/2

- The difference between CARTESIAN PRODUCT and JOIN is that the resulting relation from JOIN consists only those tuples that satisfy the **join condition**

- The JOIN operation is equivalent to CARTESIAN PRODUCT and then SELECT operation on the result of CARTESIAN PRODUCT operation, if the **select-condition** is the same as the **join condition**

# THE JOIN OPERATION – 2/2

Result of the JOIN operation

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

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

# THE EQUIJOIN OPERATION

- If the JOIN operation has equality comparison only (that is, = operation), then it is called an EQUIJOIN operation

- In the resulting relation on an EQUIJOIN operation, we always have one or more pairs of attributes that have **identical values** in every tuples

# THE NATURAL JOIN OPERATION – 1/2

- In EQUIJOIN operation, if the two attributes in the join condition have the same name, then in the resulting relation we will have two identical columns.
- In order to avoid this problem, we define the NATURAL JOIN operation
- The NATURAL JOIN operation is denoted by:

$$R_3 = R_1 * R_2$$

- In $R_3$ only one of the duplicate attributes from the list are kept

# THE NATURAL JOIN OPERATION – 2/2

(a) PROJ_DEPT ← PROJECT * DEPT.

(b) DEPT_LOCS ← DEPARTMENT * DEPT_LOCATIONS.

(a)

| PROJ_DEPT | PNAME | PNUMBER | PLOCATION | DNUM | DNAME | MGRSSN | MGRSTARTDATE |
|---|---|---|---|---|---|---|---|
| | ProductX | 1 | Bellaire | 5 | Research | 333445555 | 1988-05-22 |
| | ProductY | 2 | Sugarland | 5 | Research | 333445555 | 1988-05-22 |
| | ProductZ | 3 | Houston | 5 | Research | 333445555 | 1988-05-22 |
| | Computerization | 10 | Stafford | 4 | Administration | 987654321 | 1995-01-01 |
| | Reorganization | 20 | Houston | 1 | Headquarters | 888665555 | 1981-06-19 |
| | Newbenefits | 30 | Stafford | 4 | Administration | 987654321 | 1995-01-01 |

(b)

| DEPT_LOCS | DNAME | DNUMBER | MGRSSN | MGRSTARTDATE | LOCATION |
|---|---|---|---|---|---|
| | Headquarters | 1 | 888665555 | 1981-06-19 | Houston |
| | Administration | 4 | 987654321 | 1995-01-01 | Stafford |
| | Research | 5 | 333445555 | 1988-05-22 | Bellaire |
| | Research | 5 | 333445555 | 1988-05-22 | Sugarland |
| | Research | 5 | 333445555 | 1988-05-22 | Houston |

# ADDITIONAL RELATIONAL OPERATIONS

- Statistical queries cannot be specified in the basic relational algebra operation
- We define **aggregate functions** that can be applied over numeric values such as, SUM, AVERAGE, MAXIMUM, MINIMUM, and COUNT
- The aggregate function is denoted by

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

(a)

| R | DNO | NO_OF_EMPLOYEES | AVERAGE_SAL |
|---|---|---|---|
| | 5 | 4 | 33250 |
| | 4 | 3 | 31000 |
| | 1 | 1 | 55000 |

(b)

| DNO | COUNT_SSN | AVERAGE_SALARY |
|---|---|---|
| 5 | 4 | 33250 |
| 4 | 3 | 31000 |
| 1 | 1 | 55000 |

(c)

| COUNT_SSN | AVERAGE_SALARY |
|---|---|
| 8 | 35125 |

**FIGURE 6.9** The AGGREGATE FUNCTION operation. (a) $\rho_{R(\text{DNO, NO\_OF\_EMPLOYEES, AVERAGE\_SAL})}$ (${}_{\text{DNO}}\Im_{\text{COUNT SSN, AVERAGE SALARY}}$ (EMPLOYEE)). (b) ${}_{\text{DNO}}\Im_{\text{COUNT SSN, AVERAGE SALARY}}$ (EMPLOYEE). (c) $\Im_{\text{COUNT SSN, AVERAGE SALARY}}$ (EMPLOYEE).