# Database Management Systems

# Part II: The Relational Model

## Lecture 7

## Relational Operators II: Relational Calculus

# Contents

- Relational Calculus
- Relational Calculus vs. Relational Algebra
- Tuple-Oriented Relational Calculus
- Domain-Oriented Relational Calculus

# Relational Calculus

- Beside the relational algebra, there exists another formal language called the relational calculus in the relational model.
- A relational calculus is based on the branch of mathematical logic called predicate calculus.
- The relational calculus expression creates a new relation.
- A calculus expression specifies what is to be retrieved rather than how to retrieve it.
- Therefore, the relational calculus is considered to be a nonprocedural language.

# Relational Calculus (Cont.)

- There are two types of relational calculus:
  - ✓ tuple relational calculus, and
  - ✓ domain relational calculus.
- In tuple relational calculus, variables range over tuples.
- Whereas, in domain relational calculus, variables range over the domains (values) of attributes.

# Relational Calculus vs. Relational Algebra

- The calculus is closer to natural language; the algebra is perhaps more like a programming language.
- The calculus simply describes what the problem is, the algebra prescribes a procedure for solving that problem.
- The calculus is nonprocedural; the algebra is procedural.
- Whereas the algebra defines a set of operations for the relational model, the relational calculus provides a higher-level declarative language for specifying relational queries.

# Tuple-Oriented Relational Calculus

- A fundamental feature of the calculus is the concept of the tuple variable, also known as a range variable.
- A tuple variable is a variable that ranges over some relation, that is a variable whose only permitted values are tuples of that relation.
- The tuple relational calculus is based on specifying a number of tuple variables.
- It describes the desired information without giving a specific procedure for obtaining that information.

# Tuple-Oriented Relational Calculus (Cont.)

- A simple tuple relational calculus query is of the form:

  {t | COND(t)}

  where t is a tuple variable and COND(t) is a formula or a conditional (Boolean) expression involving t that evaluates to either TRUE or FALSE for different assignments of tuples to the variable t.

- The result of such a query is the set of all tuples t that evaluate COND(t) to TRUE.

# Tuple-Oriented Relational Calculus (Cont.)

- One possible database state for the COMPANY relational database schema is as shown in figure.
- These relations will be used in the following example queries.

**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 |

**WORKS_ON**

| Essn | Pno | Hours |
|---|---|---|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | NULL |

**PROJECT**

| Pname | Pnumber | Plocation | Dnum |
|---|---|---|---|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

**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 |

# Tuple-Oriented Relational Calculus (Cont.)

## Example Query 1

- To find all employees whose salary is above $50,000, we can write the following tuple calculus expression:

  {t | EMPLOYEE(t) AND t.Salary>50000}

- The condition EMPLOYEE(t) specifies that the range relation of tuple variable t is EMPLOYEE.
- As a result, each EMPLOYEE tuple t that satisfies the condition t.Salary>50000 will be retrieved.

# Tuple-Oriented Relational Calculus (Cont.)

## Example Query 2

- In the previous query, all attribute values for each selected EMPLOYEE tuple t will be retrieved.
- To retrieve only some of the attributes—say, the first and last names—we write:

{t.Fname, t.Lname | EMPLOYEE(t) AND t.Salary>50000}

# Tuple-Oriented Relational Calculus (Cont.)

## Example Query 3

- Retrieve the birth date and address of the employee (or employees) whose name is John B. Smith.

{t.Bdate, t.Address | EMPLOYEE(t) AND
t.Fname='John' AND
t.Minit='B' AND
t.Lname='Smith'}

# Tuple-Oriented Relational Calculus (Cont.)

## Example Query 3

- In tuple relational calculus, we first specify the requested attributes t.Bdate and t.Address for each selected tuple t.
- Then, we specify the condition for selecting a tuple following the bar (|)—namely, that t be a tuple of the EMPLOYEE relation whose Fname, Minit, and Lname attribute values are ‘John’, ‘B’, and ‘Smith’, respectively.

# Tuple-Oriented Relational Calculus (Cont.)

## Quantifiers

- In the expression of tuple relational calculus, COND is a condition or formula.
- In COND, the two special symbols called quantifiers can appear.
- These quantifiers are:
  - ✓ the existential quantifier (∃ or EXISTS) and
  - ✓ the universal quantifier (∀ or FORALL).

# Tuple-Oriented Relational Calculus (Cont.)

## Quantifiers: Existential Quantifier

- The (∃) or EXISTS quantifier is called an existential quantifier because a formula (∃t)(F) is TRUE if there exists some tuple that makes F TRUE.
- The expression is as follow.

EXISTS t (F) or (∃t)(F)

# Tuple-Oriented Relational Calculus (Cont.)

## Quantifiers: Universal Quantifier

- The (∀) or FORALL quantifier is called a universal quantifier because a formula (∀t)(F) is TRUE if all tuples t make F TRUE.
- The expression is as follow.

FORALL t (F) or (∀t)(F)

# Tuple-Oriented Relational Calculus (Cont.)

## Free and Bound Variables

- Each occurrence of a tuple variable within a well-formed formula (WFF) is either free or bound.
- Informally, a tuple variable t is bound if it is quantified, meaning that it appears in an (∃t) or (∀t) clause; otherwise, it is free.

# Tuple-Oriented Relational Calculus (Cont.)

## Free and Bound Variables

- For example, consider the following formulas:

  F1 : d.Dname='Research'

  F2 : (∃t)(d.Dnumber=t.Dno)

  F3 : (∀d)(d.Mgr_ssn='333445555')

- The tuple variable d is free in both F1 and F2, whereas it is bound to the (∀) quantifier in F3.
- Variable t is bound to the (∃) quantifier in F2.

# Tuple-Oriented Relational Calculus (Cont.)

## Example Query 1

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

  Q1: {t.Fname, t.Lname, t.Address | EMPLOYEE(t) AND

  (∃d) (DEPARTMENT(d) AND d.Dname='Research' AND d.Dnumber=t.Dno)}

- In Q1, t is the only free variable; it is then bound successively to each tuple.

# Tuple-Oriented Relational Calculus (Cont.)

## Example Query 2

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

  Q2: {p.Pnumber, p.Dnum, m.Lname, m.Bdate, m.Address | PROJECT(p) AND EMPLOYEE(m) AND p.Plocation=‘Stafford’ AND ((∃d)(DEPARTMENT(d) AND p.Dnum=d.Dnumber AND d.Mgr_ssn=m.Ssn))}

- In Q2 there are two free tuple variables, p and m.
- Tuple variable d is bound to the existential quantifier.

# Tuple-Oriented Relational Calculus (Cont.)

## Example Query 3

- List the name of each employee who works on some project controlled by department number 5.

Q3:{e.Fname, e.Lname | EMPLOYEE(e) AND ((∃x)(∃w)(PROJECT(x) AND WORKS_ON(w) AND x.Dnum=5 AND w.Essn=e.Ssn AND x.Pnumber=w.Pno))}

# Domain-Oriented Relational Calculus

- There is another type of relational calculus called the domain relational calculus, or simply, domain calculus.
- Domain calculus differs from tuple calculus in the type of variables used in formulas.
- In domain calculus, tuple variables are replaced by domain variables, that is variables that range over a domain instead of a relation.

# Domain-Oriented Relational Calculus (Cont.)

- To form a relation of degree n for a query result, we must have n of these domain variables—one for each attribute.
- An expression of the domain calculus is of the form:

$$\{x1, x2, ..., xn \mid COND(x1, x2, ..., xn, xn+1, xn+2, ..., xn+m)\}$$

where x1, x2, ..., xn, xn+1, xn+2, ..., xn+m are domain variables that range over domains (of attributes), and COND is a condition or formula of the domain relational calculus.

# Domain-Oriented Relational Calculus (Cont.)

## Example Query 1

- List the birth date and address of the employee whose name is 'John B. Smith'.
- According to the EMPLOYEE relation (described in slide 8), we need ten domain variables, one to range over each of the domains of attributes of EMPLOYEE in order.
- We will use ten lowercase letters q, r, s, ..., x, y, z for domain variables.

# Domain-Oriented Relational Calculus (Cont.)

## Example Query 1

- We first specify the requested attributes, Bdate and Address, by the free domain variables u for BDATE and v for ADDRESS.
- Then we specify the condition for selecting a tuple following the bar (|)—namely, that the sequence of values assigned to the variables qrstuvwxyz be a tuple of the EMPLOYEE relation and that the values for q (Fname), r (Minit), and s (Lname) be equal to 'John', 'B', and 'Smith', respectively.

# Domain-Oriented Relational Calculus (Cont.)

## Example Query 1

- As a result, we can write the query 1 as follow.

Q1: {u, v | (∃q) (∃r) (∃s) (EMPLOYEE(qrstuvwxyz) AND q='John' AND r='B' AND s='Smith')}

- Only u and v are free variables because they appear to the left of the bar.

# Domain-Oriented Relational Calculus (Cont.)

## Example Query 2

- Retrieve the name and address of all employees who work for the ‘Research’ department.

Q2: {q, s, v | (∃z) (∃l) (∃m) (EMPLOYEE(qrstuvwxyz) AND

DEPARTMENT(lmno) AND l=‘Research’ AND m=z)}

# Domain-Oriented Relational Calculus (Cont.)

## Example Query 3

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

Q3: {i, k, s, u, v | (∃j)(∃m)(∃n)(∃t)(PROJECT(hijk) AND

EMPLOYEE(qrstuvwxyz) AND DEPARTMENT(lmno) AND k=m AND n=t AND j=‘Stafford’)}

# Summary

- The relational calculus is descriptive where the algebra is prescriptive.
- The calculus exists in two forms, tuple calculus and domain calculus.
- The variables of the tuple calculus are tuple variables while the variables of the domain calculus are domain variables.
- Relational calculus have two quantifiers called universal quantifier and existential quantifier.

# Next Lecture

The SQL Language

- Data Definition
- Data Manipulation

# Textbook and References

**Textbook**

- C. J. Date, “An Introduction to Database Systems”, 6th Edition, 1994.

**Additional References**

- Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database System Concepts”, 6th Edition, 2011.
- Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, 6th Edition, 2010.