RAMAIAH
Institute of Technology

UNIT-II

RAMAIAH Institute of Technology

Relational Model Concepts:

- The relational model represents the **database as a collection of relations**. When a relation is thought of as a **table of values**, each row in the table represents a collection of related data values.

- In the formal relational model terminology, **a row is called a tuple**, a **column header**
- **is called an attribute**, and the **table is called a relation.**

Domains, Attributes, Tuples, and Relations

A **domain D** is a set of **atomic values**. By atomic we mean that each value in the domain is indivisible as far as the formal relational model is concerned.

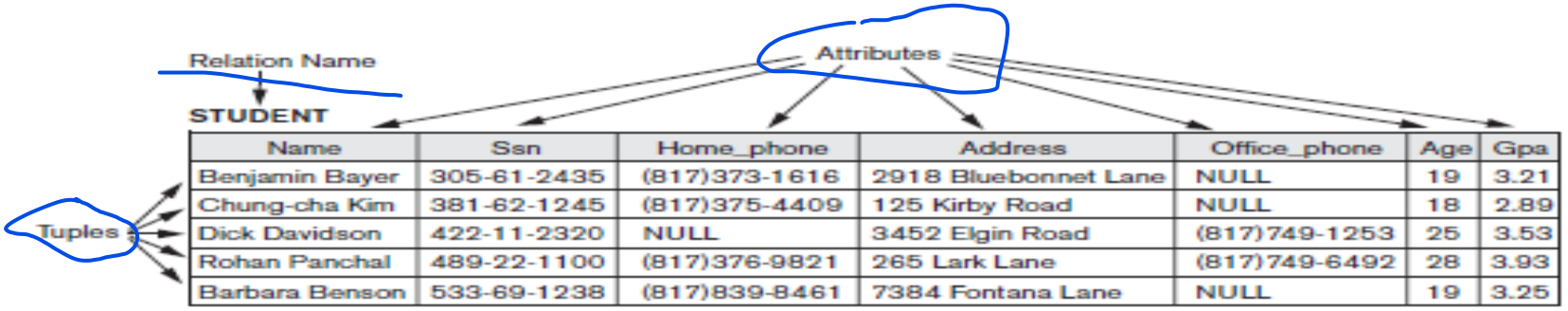
A relation (or relation state) r of the relation schema R(A1, A2, ..., An), also denoted **by r(R)**, is a set of n-tuples r = {t1, t2, ..., tm}. Each n-tuple t is an ordered list of n values t =<v1, v2, ..., vn>,

where each value **vi, 1 ≤ i ≤ n, is an element of dom (Ai)** or is a special NULL value.

$$r(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times \ldots \times \text{dom}(A_n))$$

RAMAIAH
Institute of Technology

Entity-Relationship Model



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 3.1
The attributes and tuples of a relation STUDENT.

- Usa_phone_numbers. The set of ten-digit phone numbers valid in the United States.
- Local_phone_numbers. The set of seven-digit phone numbers valid within a particular area code in the United States. The use of local phone numbers is quickly becoming obsolete, being replaced by standard ten-digit numbers.
- Social_security_numbers. The set of valid nine-digit Social Security numbers. (This is a unique identifier assigned to each person in the United States for employment, tax, and benefits purposes.)
- Names: The set of character strings that represent names of persons.
- Grade_point_averages. Possible values of computed grade point averages; each must be a real (floating-point) number between 0 and 4.
- Employee_ages. Possible ages of employees in a company; each must be an integer value between 15 and 80.
- Academic_department_names. The set of academic department names in a university, such as Computer Science, Economics, and Physics.
- Academic_department_codes. The set of academic department codes, such as 'CS', 'ECON', and 'PHYS'.

RAMAIAH
Institute of Technology

A **relation schema R**, denoted by **R(A1, A2, ...,An)**, is made up of a relation name R and a list of attributes, A1, A2, ..., An.

- Each attribute Ai is the name of a role played by some domain D in the relation schema R.
- **D is called the domain of Ai** and is denoted by dom(Ai).
- A relation schema is used to describe a relation; R is called the name of this relation.
- The **degree (or arity) of a relation** is the number of attributes n of its relation schema.

Domains for some of the attributes of the STUDENT relation:

- dom(Name) = Names;
- dom(Ssn) = Social_security_numbers;
- dom(HomePhone) =USA_phone_numbers3,
- dom(Office_phone) = USA_phone_numbers, and
- dom(Gpa) = Grade_point_averages.

RAMAIAH Institute of Technology

Characteristics of Relations

Ordering of Tuples in a Relation: A relation is defined as a set of tuples. Mathematically, elements of a set have **no order among them**; hence, tuples in a relation do not have any particular order.

Ordering of Values within a Tuple and an Alternative Definition of a Relation:
According to the preceding definition of a relation, an n-tuple is an ordered list of n values, so the **ordering of values in a tuple**—and hence of attributes in a relation schema—is important.

Values and NULLs in the Tuples. Each value in a tuple is an atomic value; An important concept is that of NULL values, which are used to represent the values of attributes that may be unknown or may not apply to a tuple.

- In general, we can have several meanings for NULL values, such as
  - **value unknown,**
  - **value exists but is not available, or attribute does not apply to this tuple (also known as value undefined).**

RAMAIAH Institute of Technology

Relational Model Constraints and Relational Database Schemas

- Model-based constraints or implicit constraints
- **Schema-based constraints or explicit constraints.**
- Application-based or semantic constraints or business rules.

The schema-based constraints include :

- **Domain constraints,**
- **Key constraints,**
- **Entity integrity constraints, and**
- **Referential integrity constraints.**

RAMAIAH
Institute of Technology

Domain Constraints:

- Domain constraints specify that **within each tuple, the value of each attribute A must be an atomic value** from the **domain dom(A)**
    - The data types associated with domains typically include standard **numeric data types** for integers (such as short integer, integer, and long integer) and **real numbers** (float and double precision float). **Characters, Booleans, fixed-length strings, and variable-length strings** are also available, as are **date, time, timestamp**, and money, or other special data types.

Key Constraints and Constraints on NULL Values:

- a relation is defined as a set of tuples. By definition, all elements of a set are distinct; hence, **all tuples in a relation must also be distinct**.
    - This means that **no two tuples can have the same combination of values for all their attributes.**

RAMAIAH Institute of Technology

Key Constraints and Constraints on NULL Values:

- Suppose that we denote one such subset of attributes by SK; then for any two distinct tuples t1 and t2 in a relation state r of R, we have the constraint that:

  t1[SK] != t2[SK] Any such set of attributes SK is called a **super key of the relation schema R**.

- A super key SK specifies a uniqueness constraint that **no two distinct tuples in any state r of R can have the same value for SK.**

- A key K of a relation schema R is a super key of R with the additional property that removing any attribute A from K leaves a set of attributes K' that is not a super key of R any more. Hence, a key satisfies two properties:

  1. **Two distinct tuples in any state of the relation cannot have identical values** for (all) the attributes in the key.

  2. **It is a minimal super key**—that is, a super key from which we cannot remove any attributes and still have the uniqueness constraint in condition 1 hold.

RAMAIAH Institute of Technology

Key Constraints and Constraints on NULL Values:

- The attribute set **{Ssn} is a key of STUDENT** because no two student tuples can have the same value for Ssn.
  - Any set of attributes that includes Ssn—for example, **{Ssn, Name, Age}—is a super key.**
  - However, the super key {Ssn, Name, Age} is not a key of STUDENT because removing Name or Age or both from the set still leaves us with a super key.

- In general, **a relation schema may have more than one key.** In this case, each of the keys is called a **candidate key**.
  - For example, the CAR relation in Figure 3.4 has two candidate keys: **License_number and Engine_serial_number**. It is common to designate one of the candidate keys as the primary key of the relation.

CAR

License_number	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

Figure 3.4
The CAR relation, with two candidate keys: License_number and Engine_serial_number.

RAMAIAH Institute of Technology

- Figure 3.5 shows a relational database schema that we call COMPANY = {EMPLOYEE, DEPARTMENT, DEPT_LOCATIONS, PROJECT, WORKS_ON, DEPENDENT}.

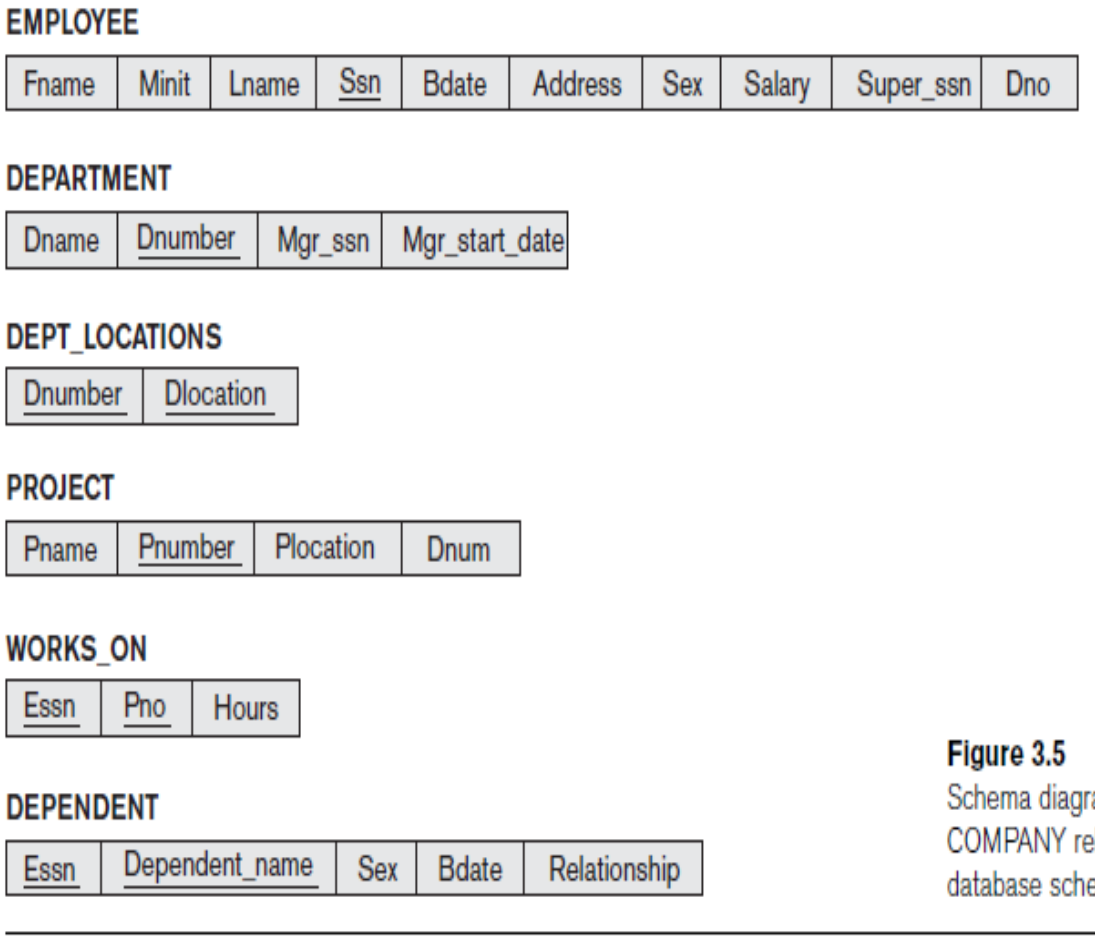


Figure 3.5
Schema diagram for the COMPANY relational database schema.

- A relational database schema S is a set of relation schemas **S = {R1, R2, ..., Rm}** and a set of integrity constraints IC.
- A relational database state DB of S is a set of relation states **DB = {r1, r2, ..., rm}** such that each ri is a state of Ri and such that the ri relation states satisfy the integrity constraints specified in IC.

RAMAIAH
Institute of Technology

Integrity, Referential Integrity, and Foreign Keys:

- The **entity integrity constraint** states that no primary key value can be NULL.
  - This is because the primary key value is used to identify individual tuples in a relation. Having NULL values for the primary key implies that we cannot identify some tuples.
- Key constraints and entity integrity constraints are specified on individual relations.
- The **referential integrity constraint** is specified between two relations and is used to maintain the consistency among tuples in the two relations.
- For example, in Figure 3.6, the attribute **Dno of EMPLOYEE** gives the department number for which each employee works; hence, its value in every EMPLOYEE tuple must match the **Dnumber** value of some tuple in the **DEPARTMENT** relation.

RAMAIAH Institute of Technology

Referential Integrity, and Foreign Keys

A **referential integrity constraint** between the two **relation schemas R1 and R2**. A set of attributes FK in relation schema R1 is a foreign key of R1 that references relation R2 if it satisfies the following rules:

1. The attributes in **FK have the same domain(s) as the primary key attributes PK** of R2; the attributes FK are said to reference or refer to the relation R2.

2. A value of FK in a tuple t1 of the current state r1(R1) either occurs as a value of PK for some tuple t2 in the current state r2(R2) or is NULL. In the former case, we have **t1[FK] = t2[PK], and we say that the tuple t1 references or refers to the tuple t2.**

In this definition, R1 is called the referencing relation and R2 is the referenced relation. If these two conditions hold, a referential integrity constraint from R1 to R2 is said to hold.

Figure 3.6
One possible database state for the COMPANY relational database schema.

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

RAMAIAH
Institute of Technology

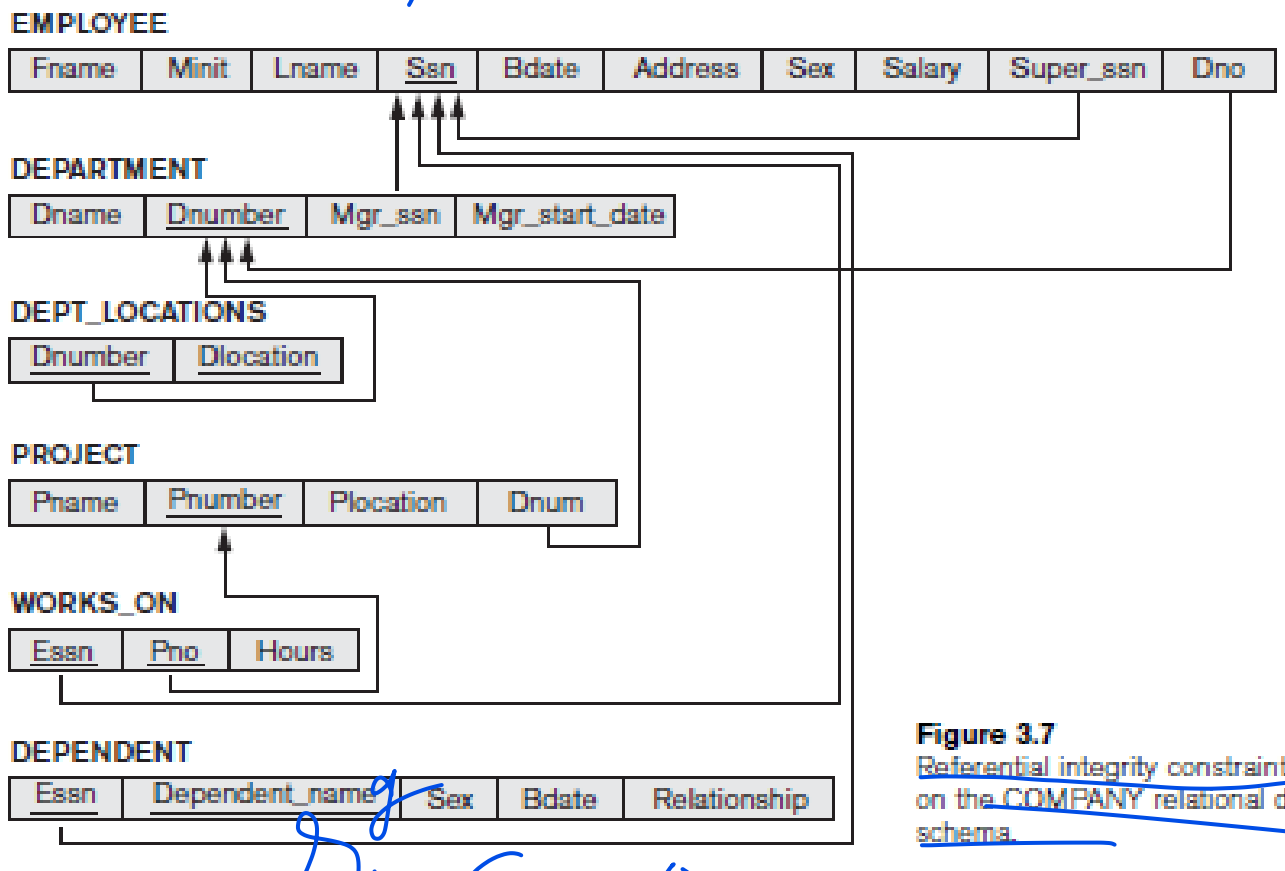


Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

RAMAIAH Institute of Technology

Other Types of Constraints

The Insert Operation:

- **Domain constraints** can be violated if an attribute value is given that does not appear in the corresponding domain or is not of the appropriate data type.
- **Key constraints** can be violated if a key value in the new tuple t already exists in another tuple in the relation r(R).
- **Entity integrity** can be violated if any part of the primary key of the new tuple t is NULL.
- **Referential integrity** can be violated if the value of any foreign key in t refers to a tuple that does not exist in the referenced relation

- *Operation*:
  Insert <'Cecilia', 'F', 'Kolonsky', NULL, '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, NULL, 4> into EMPLOYEE.
  Result: This insertion violates the entity integrity constraint (NULL for the primary key Ssn), so it is rejected.
- *Operation*:
  Insert <'Alicia', 'J', 'Zelaya', '999887777', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, '987654321', 4> into EMPLOYEE.
  Result: This insertion violates the key constraint because another tuple with the same Ssn value already exists in the EMPLOYEE relation, and so it is rejected.
- *Operation*:
  Insert <'Cecilia', 'F', 'Kolonsky', '677678989', '1960-04-05', '6357 Windswept, Katy, TX', F, 28000, '987654321', 7> into EMPLOYEE.
  Result: This insertion violates the referential integrity constraint specified on Dno in EMPLOYEE because no corresponding referenced tuple exists in DEPARTMENT with Dnumber = 7.

RAMAIAH
Institute of Technology

The Delete Operation:

- The Delete operation can **violate only referential integrity.** This occurs if the tuple being deleted is referenced by foreign keys from other tuples in the database.

- *Operation*:
  Delete the WORKS_ON tuple with Essn = '999887777' and Pno = 10.
  Result: This deletion is acceptable and deletes exactly one tuple.
- *Operation*:
  Delete the EMPLOYEE tuple with Ssn = '999887777'.
  Result: This deletion is not acceptable, because there are tuples in WORKS_ON that refer to this tuple. Hence, if the tuple in EMPLOYEE is deleted, referential integrity violations will result.
- *Operation*:
  Delete the EMPLOYEE tuple with Ssn = '333445555'.
  Result: This deletion will result in even worse referential integrity violations, because the tuple involved is referenced by tuples from the EMPLOYEE, DEPARTMENT, WORKS_ON, and DEPENDENT relations.

RAMAIAH
Institute of Technology

The Update Operation:

- The Update (or Modify) operation is used to change the values of one or more attributes in a tuple (or tuples) of some relation R. It is necessary to specify a condition on the attributes of the relation to select the tuple (or tuples) to be modified.

  - *Operation*:
    Update the salary of the EMPLOYEE tuple with Ssn = '999887777' to 28000.
    Result: Acceptable.
  - *Operation*:
    Update the Dno of the EMPLOYEE tuple with Ssn = '999887777' to 1.
    Result: Acceptable.
  - *Operation*:
    Update the Dno of the EMPLOYEE tuple with Ssn = '999887777' to 7.
    Result: Unacceptable, because it violates referential integrity.
  - *Operation*:
    Update the Ssn of the EMPLOYEE tuple with Ssn = '999887777' to '987654321'.
    Result: Unacceptable, because it violates primary key constraint by repeating a value that already exists as a primary key in another tuple; it violates referential integrity constraints because there are other relations that refer to the existing value of Ssn.

RAMAIAH
Institute of Technology

Text Books/Reference:

Elmasri and Navathe: Fundamentals of Database Systems, 5th Edition, Addison-Wesley, 2011.

RAMAIAH
Institute of Technology

Database Systems

Subject Code: CS52 (Credits: 3:1:0)

Ganeshayya Shidaganti
Dept.CSE
Ramaiah Institute of Technology ,Bangalore

RAMAIAH
Institute of Technology

Unit-II

Relational Model and Relational Algebra:

- Relational Model Concepts,
- Relational Model Concepts,
- Relational Model Constraints and
- Relational Database Schema Update Operations,
- Transactions and Dealing with Constraint violations,

Relational Algebra and Relational Calculus

- Unary Relational operations,
- Relational Algebra Operations from Set Theory,
- Binary Relational Operations, JOIN and DIVISION,
- Additional Relational Operations,
- Examples of Queries in Relational Algebra Relational
- Database Design Using ER- to-Relational Mapping.

RAMAIAH Institute of Technology

Relational Algebra

a set of operations that is used to manipulate entire set of relations

- The basic set of operations for the relational model is the relational algebra.
  - The relational algebra is very important for several reasons. First, it provides a **formal foundation for relational model operations**.
  - Second, it is used as a basis for **implementing and optimizing queries in the query processing and optimization modules** that are integral parts of relational database management systems (RDBMSs)
  - Third, some of its **concepts are incorporated into the SQL** standard query language for RDBMSs.
  - The relational algebra is often considered to be an integral part of the relational data model. Its operations can be divided into two groups. One group includes **set operations from mathematical set theory**;
  - Set operations include UNION, INTERSECTION, SET DIFFERENCE, and CARTESIAN PRODUCT (also known as CROSS PRODUCT).
  - The other group consists of operations developed specifically for relational databases—these include SELECT, PROJECT, and JOIN, among others.

RAMAIAH
Institute of Technology

First, we describe the **SELECT and PROJECT operations** in Section 6.1 because they are **unary operations** that operate on single relations.

Then we discuss set operations in Section 6.2. In Section 6.3, we discuss **JOIN and other complex binary operations**, **which operate on two tables** by combining related tuples (records) based on join conditions

Figure 6.1
Results of SELECT and PROJECT operations. (a) $\sigma_{(Dno=4\ \textbf{AND}\ Salary>25000)\ OR\ (Dno=5\ \textbf{AND}\ Salary>30000)}(EMPLOYEE)$. (b) $\pi_{Lname,\ Fname,\ Salary}(EMPLOYEE)$. (c) $\pi_{Sex,\ Salary}(EMPLOYEE)$.

(a)

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	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 Fire Oak, 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

RAMAIAH Institute of Technology

Unary Relational Operations: SELECT and PROJECT

The SELECT Operation

- The SELECT operation is used to **choose a subset of the tuples** from a relation that satisfies a **selection condition.**
- The SELECT operation can also be **visualized as a horizontal partition** of the relation into two sets of tuples—those tuples that satisfy the condition and are selected, and those tuples that do not satisfy the condition and are discarded.
- For example, to select the EMPLOYEE tuples whose department is 4, or those whose salary is greater than $30,000
- where the **symbol σ (sigma) is used to denote the SELECT operator** and the selection condition is a Boolean expression (condition) specified on the attributes of relation R.

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

$$\sigma_{Salary>30000}(EMPLOYEE)$$

In general, the SELECT operation is denoted by

$$\sigma_{<selection\ condition>}(R)$$

RAMAIAH Institute of Technology

The **SELECT operation is commutative**; that is,

σ<cond1>(σ<cond2>(R)) = σ<cond2>(σ<cond1>(R))

The SELECT operator is unary; that is, it is applied to a single relation; **The degree of the relation** resulting from a SELECT operation—its number of attributes—is the same as the degree of R. The number of tuples in the resulting relation is always less than or equal to the number of tuples in R.

<attribute name> <comparison op> <constant value>
or
<attribute name> <comparison op> <attribute name>

- where <attribute name> is the name of an attribute of R, <comparison op> is normally
- one of the **operators {=, <, ≤, >, ≥, ≠}, and <constant value> is a constant value** from the attribute domain.

- For example, to select the tuples for 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, we can specify the following SELECT operation:

σ(Dno=4 AND Salary>25000) OR (Dno=5 AND Salary>30000)(EMPLOYEE)

RAMAIAH
Institute of Technology

The PROJECT Operation: The SELECT operation chooses some of the rows from the table while discarding other rows. The PROJECT operation, on the other hand, **selects certain columns from the table** and discards the other columns

- If we are **interested in only certain attributes of a relation**, we use the PROJECT operation to project the relation over these attributes only. Therefore, the result of the PROJECT operation can be visualized as a **vertical partition of the relation** into two relations:

- For example, to list each employee's first and last name and salary, we can use the PROJECT operation as follows:

  π Lname, Fname, Salary(EMPLOYEE)

- The general form of the PROJECT operation is
  π<attribute list>(R)

- where **π (pi) is the symbol used to represent the PROJECT operation**, and <attribute list> is the desired sublist of attributes from the attributes of relation R.

The result of the PROJECT operation has **only the attributes specified in <attribute list> in the same order as they appear in the list**. Hence, its degree is equal to the number of attributes in <attribute list>.

The **PROJECT operation removes any duplicate tuples**, so the result of the PROJECT operation is a **set of distinct tuples**, and hence a valid relation. This is known as duplicate elimination. For example, consider the following PROJECT operation:

πSex, Salary(EMPLOYEE)

In SQL, the PROJECT attribute list is specified in the SELECT clause of a query. For example, the following operation:

πSex, Salary(EMPLOYEE)

would correspond to the following SQL query:

```
SELECT DISTINCT Sex, Salary
FROM EMPLOYEE
```

(a)

Fname	Lname	Salary
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

Figure 6.2
Results of a sequence of operations. (a) $\pi_{Fname, Lname, Salary}(\sigma_{Dno=5}(EMPLOYEE))$. (b) Using intermediate relations and renaming of attributes.

(b)

TEMP

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
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

R

First_name	Last_name	Salary
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

RAMAIAH Institute of Technology

Relational Algebra Operations from Set Theory

The UNION, INTERSECTION, and MINUS Operations:

- For 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 ← σDno=5(EMPLOYEE)
RESULT1 ← πSsn(DEP5_EMPS)
RESULT2(Ssn) ← πSuper_ssn(DEP5_EMPS)
RESULT ← RESULT1 ∪ RESULT2

RESULT1

Ssn
123456789
333445555
666884444
453453453

RESULT2

Ssn
333445555
888665555

RESULT

Ssn
123456789
333445555
666884444
453453453
888665555

Figure 6.3
Result of the UNION operation RESULT ← RESULT1 ∪ RESULT2.

RAMAIAH
Institute of Technology

Several set theoretic operations are used to merge the elements of two sets in various ways, including UNION, INTERSECTION, and SET DIFFERENCE (also called MINUS or EXCEPT).

We can define the three operations UNION, INTERSECTION, and SET DIFFERENCE on two union-compatible relations R and S as follows:

- UNION: The result of this operation, denoted by R ∪ 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.

- INTERSECTION: The result of this operation, denoted by R ∩ S, is a relation that includes all tuples that are in both R and S.

- **SET DIFFERENCE (or MINUS):** The result of this operation, denoted by R – S, is a relation that includes all tuples that are in R but not in S.

RAMAIAH
Institute of Technology

- UNION and INTERSECTION are commutative operations; that is,

  $R \cup S = S \cup R$ and $R \cap S = S \cap R$

- Both UNION and INTERSECTION can be treated as n-ary operations applicable to any number of relations because both are also associative operations; that is,

  $R \cup (S \cup T) = (R \cup S) \cup T$ and $(R \cap S) \cap T = R \cap (S \cap T)$

- The MINUS operation is not commutative; that is, in general,

  $R - S \neq S - R$

Figure 6.4
The set operations UNION, INTERSECTION, and MINUS. (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

RAMAIAH Institute of Technology

The CARTESIAN PRODUCT (CROSS PRODUCT) Operation

- The **CARTESIAN PRODUCT** operation—also known as **CROSS PRODUCT** or **CROSS JOIN—which is denoted by ×.**
- In general, the result of R(A1, A2, ..., An) × S(B1, B2, ..., Bm) is a relation Q with degree n + m attributes Q(A1, A2, ..., An, B1, B2, ..., Bm), 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 nR tuples (denoted as |R| = nR), and S has nS tuples, then R × S will have nR * nS tuples.
- For example, suppose that we want to retrieve a list of names of each female employee's dependents. We can do this as follows:

FEMALE_EMPS ← σSex='F'(EMPLOYEE)
EMPNAMES ← πFname, Lname, Ssn(FEMALE_EMPS)
EMP_DEPENDENTS ← EMPNAMES × DEPENDENT
ACTUAL_DEPENDENTS ← σSsn=Essn(EMP_DEPENDENTS)
RESULT ← πFname, Lname, Dependent_name(ACTUAL_DEPENDENTS)

Figure 6.5
The Cartesian Product (Cross Product) operation.

FEMALE_EMPS

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Alicia	J	Zelaya	999887777	1968-07-19	3321Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291Berry, Bellaire, TX	F	43000	888665555	4
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

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	...

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

RAMAIAH Institute of Technology

Binary Relational Operations: JOIN and DIVISION

URO RA BRO

The JOIN Operation

- The JOIN operation, denoted by ⋈, is used to **combine related tuples** from two relations into single "longer" tuples.

- To illustrate JOIN, 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 Mgr_ssn value** in the department tuple

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

- The JOIN operation can be specified as a CARTESIAN PRODUCT operation followed by a SELECT operation.

$$\mathbf{EMP\_DEPENDENTS \leftarrow EMPNAMES \times DEPENDENT}$$
$$\mathbf{ACTUAL\_DEPENDENTS \leftarrow \sigma_{Ssn=Essn}(EMP\_DEPENDENTS)}$$

These two operations can be replaced with a single JOIN operation as follows:

$$ACTUAL\_DEPENDENTS \leftarrow EMPNAMES \bowtie_{Ssn=Essn} DEPENDENT$$

RAMAIAH
Institute of Technology

DEPT_MGR

Dname	Dnumber	Mgr_ssn	...	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	...

Figure 6.6
Result of the JOIN operation DEPT_MGR ← DEPARTMENT $\bowtie_{Mgr\_ssn=Ssn}$ EMPLOYEE.

A general join condition is of the form:

<condition> AND <condition> AND...AND <condition>

- where each <condition> is of the form Ai θ Bj, Ai is an attribute of R, Bj is an attribute of S, Ai and Bj have the same domain, and θ (theta) is one of the comparison operators {=, <, ≤, >, ≥, ≠}. A JOIN operation with such a general join condition is called a **THETA JOIN.**

RAMAIAH Institute of Technology

Variations of JOIN: The EQUIJOIN and NATURAL JOIN

- 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.**

- For example, in Figure 6.6, the values of the attributes **Mgr_ssn and Ssn are identical in every tuple of DEPT_MGR (the EQUIJOIN result)** because the equality join condition specified on these two attributes requires the values to be identical in every tuple in the result.

- a new operation called **NATURAL JOIN—denoted by *—**was created to get rid of the second (superfluous) attribute in an EQUIJOIN condition.

  - The standard definition of **NATURAL JOIN requires that the two join attributes (or each pair of join attributes) have the same name in both relations**

RAMAIAH
Institute of Technology

Suppose we want to **combine each PROJECT tuple with the DEPARTMENT** tuple that controls the project.

- first we rename the **Dnumber attribute of DEPARTMENT** to Dnum—so that it has the same name as the **Dnum attribute in PROJECT**—and then we apply NATURAL JOIN:

$$PROJ\_DEPT \leftarrow PROJECT * \rho_{(Dname,\ Dnum,\ Mgr\_ssn,\ Mgr\_start\_date)}(DEPARTMENT)$$

The same query can be done in two steps by creating an intermediate table DEPT as follows:

$$DEPT \leftarrow \rho_{(Dname,\ Dnum,\ Mgr\_ssn,\ Mgr\_start\_date)}(DEPARTMENT)$$
$$PROJ\_DEPT \leftarrow PROJECT * DEPT$$

(a)

PROJ_DEPT

Pname	Pnumber	Plocation	Dnum	Dname	Mgr_ssn	Mgr_start_date
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	Mgr_ssn	Mgr_start_date	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

Figure 6.7
Results of two NATURAL JOIN operations. (a) PROJ_DEPT ← PROJECT * DEPT.
(b) DEPT_LOCS ← DEPARTMENT * DEPT_LOCATIONS.

RAMAIAH Institute of Technology

The DIVISION Operation:

The DIVISION operation, **denoted by ÷,** is useful for a special kind of query that sometimes occurs in database applications. An example is Retrieve the names of employees who work on all the projects that 'John Smith' works on.

First, retrieve the list of project numbers that 'John Smith' works on in the intermediate relation SMITH_PNOS:

SMITH ← σFname='John' AND Lname='Smith'(EMPLOYEE)
SMITH_PNOS ← πPno(WORKS_ON Essn=SsnSMITH)

Next, create a relation that includes a tuple <Pno, Essn> whenever the employee whose Ssn is Essn works on the project whose number is Pno in the intermediate relation SSN_PNOS:

SSN_PNOS ← πEssn, Pno(WORKS_ON)

Finally, apply the DIVISION operation to the two relations, which gives the desired employees' Social Security numbers:

SSNS(Ssn) ← SSN_PNOS ÷ SMITH_PNOS
RESULT ← πFname, Lname(SSNS * EMPLOYEE)

Figure 6.8
The DIVISION operation. (a) Dividing SSN_PNOS by SMITH_PNOS. (b) $T \leftarrow R \div S$.

(a)

SSN_PNOS

Essn	Pno
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987654321	30
987654321	20
888665555	20

SMITH_PNOS

Pno
1
2

SSNS

Ssn
123456789
453453453

(b)

R

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

S

A
a1
a2
a3

T

B
b1
b4

RAMAIAH
Institute of Technolo

Table 6.1 Operations of Relational Algebra

OPERATION	PURPOSE	NOTATION
SELECT	Selects all tuples that satisfy the selection condition from a relation R.	$\sigma_{\text{<selection condition>}}(R)$
PROJECT	Produces a new relation with only some of the attributes of R, and removes duplicate tuples.	$\pi_{\text{<attribute list>}}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{\text{<join condition>}} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{\text{<join condition>}} R_2$, OR $R_1 \bowtie_{(\text{<join attributes 1>}), (\text{<join attributes 2>})} R_2$
NATURAL JOIN	Same as EQUIJOIN except that the join attributes of R_2 are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R_1 *_{\text{<join condition>}} R_2$, OR $R_1 *_{(\text{<join attributes 1>}), (\text{<join attributes 2>})} R_2$ OR $R_1 * R_2$
UNION	Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cup R_2$
INTERSECTION	Produces a relation that includes all the tuples in both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cap R_2$
DIFFERENCE	Produces a relation that includes all the tuples in R_1 that are not in R_2; R_1 and R_2 must be union compatible.	$R_1 - R_2$
CARTESIAN PRODUCT	Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2.	$R_1 \times R_2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$

RAMAIAH
Institute of Technology

Aggregate Functions and Grouping:

- **Aggregate Functions :** Examples of such functions include **retrieving the average or total salary** of all employees or the **total number of employee** tuples

- Common functions applied to collections of numeric values include **SUM, AVERAGE, MAXIMUM, and MINIMUM. The COUNT** function is used for counting tuples or values.

- Another common type of request involves **grouping the tuples in a relation** by the value of some of their attributes and then applying an aggregate function independently to each group.
  - An example would be to group EMPLOYEE tuples by Dno, so that each group includes the tuples for employees working in the same department.

- an AGGREGATE FUNCTION operation, using the **symbol ℑ (pronounced script F)**, to specify these types of requests as follows:

<grouping attributes> ℑ <function list> (R)

- For example, to retrieve each department number, the number of employees in the department, and their average salary, while renaming the resulting attributes as indicated below, we write:

ρ R(Dno, No_of_employees, Average_sal) (Dno ℑ COUNT Ssn, AVERAGE Salary (EMPLOYEE))

- **If no renaming is applied**, then the attributes of the resulting relation that correspond to the function list will each be the concatenation of the function name with the attribute name in the form <function>_<attribute>

Dno ℑ COUNT Ssn, AVERAGE Salary(EMPLOYEE)

- **If no grouping attributes are specified**, the functions are applied to all the tuples in the relation, so the resulting relation has a single tuple only.

ℑ COUNT Ssn, AVERAGE Salary(EMPLOYEE)

RAMAIAH
Institute of Technology

Figure 6.10
The aggregate function operation.

a. $\rho_{R(\text{Dno, No\_of\_employees, Average\_sal})}(_{\text{Dno}}\Im_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE}))$.

b. $_{\text{Dno}}\Im_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$.

c. $\Im_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$.

R

(a)

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

RAMAIAH Institute of Technology

Examples of Queries in Relational Algebra

Query 1. Retrieve the name and address of all employees who work for the 'Research' department.

$$RESEARCH\_DEPT \leftarrow \sigma_{Dname='Research'}(DEPARTMENT)$$
$$RESEARCH\_EMPS \leftarrow (RESEARCH\_DEPT \bowtie_{Dnumber=Dno} EMPLOYEE)$$
$$RESULT \leftarrow \pi_{Fname,\ Lname,\ Address}(RESEARCH\_EMPS)$$

As a single in-line expression, this query becomes:

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

RAMAIAH
Institute of Technology

Query 2. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date.

$$\text{STAFFORD\_PROJS} \leftarrow \sigma_{\text{Plocation='Stafford'}}(\text{PROJECT})$$
$$\text{CONTR\_DEPTS} \leftarrow (\text{STAFFORD\_PROJS} \bowtie_{\text{Dnum=Dnumber}} \text{DEPARTMENT})$$
$$\text{PROJ\_DEPT\_MGRS} \leftarrow (\text{CONTR\_DEPTS} \bowtie_{\text{Mgr\_ssn=Ssn}} \text{EMPLOYEE})$$
$$\text{RESULT} \leftarrow \pi_{\text{Pnumber, Dnum, Lname, Address, Bdate}}(\text{PROJ\_DEPT\_MGRS})$$

RAMAIAH
Institute of Technology

Query 3. Find the names of employees who work on all the projects controlled by department number 5.

$$DEPT5\_PROJS \leftarrow \rho_{(Pno)}(\pi_{Pnumber}(\sigma_{Dnum=5}(PROJECT)))$$
$$EMP\_PROJ \leftarrow \rho_{(Ssn,\ Pno)}(\pi_{Essn,\ Pno}(WORKS\_ON))$$
$$RESULT\_EMP\_SSNS \leftarrow EMP\_PROJ \div DEPT5\_PROJS$$
$$RESULT \leftarrow \pi_{Lname,\ Fname}(RESULT\_EMP\_SSNS * EMPLOYEE)$$

RAMAIAH
Institute of Technology

Query 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.

$$SMITHS(Essn) \leftarrow \pi_{Ssn}(\sigma_{Lname='Smith'}(EMPLOYEE))$$
$$SMITH\_WORKER\_PROJS \leftarrow \pi_{Pno}(WORKS\_ON * SMITHS)$$
$$MGRS \leftarrow \pi_{Lname,\ Dnumber}(EMPLOYEE \bowtie_{Ssn=Mgr\_ssn} DEPARTMENT)$$
$$SMITH\_MANAGED\_DEPTS(Dnum) \leftarrow \pi_{Dnumber}(\sigma_{Lname='Smith'}(MGRS))$$
$$SMITH\_MGR\_PROJS(Pno) \leftarrow \pi_{Pnumber}(SMITH\_MANAGED\_DEPTS * PROJECT)$$
$$RESULT \leftarrow (SMITH\_WORKER\_PROJS \cup SMITH\_MGR\_PROJS)$$

RAMAIAH
Institute of Technology

Query 5. List the names of all employees with two or more dependents.

- Strictly speaking, this query cannot be done in the basic (original) relational algebra. We have to use the AGGREGATE FUNCTION operation with the COUNT aggregate function. We assume that dependents of the same employee have distinct Dependent_name values.

$$T1(\text{Ssn}, \text{No\_of\_dependents}) \leftarrow {}_{\text{Essn}}\Im_{\text{COUNT Dependent\_name}}(\text{DEPENDENT})$$
$$T2 \leftarrow \sigma_{\text{No\_of\_dependents}>2}(T1)$$
$$\text{RESULT} \leftarrow \pi_{\text{Lname, Fname}}(T2 * \text{EMPLOYEE})$$

RAMAIAH
Institute of Technology

Query 6. Retrieve the names of employees who have no dependents.

- This is an example of the type of query that uses the MINUS (SET DIFFERENCE) operation.

$$\text{ALL\_EMPS} \leftarrow \pi_{\text{Ssn}}(\text{EMPLOYEE})$$
$$\text{EMPS\_WITH\_DEPS(Ssn)} \leftarrow \pi_{\text{Essn}}(\text{DEPENDENT})$$
$$\text{EMPS\_WITHOUT\_DEPS} \leftarrow (\text{ALL\_EMPS} - \text{EMPS\_WITH\_DEPS})$$
$$\text{RESULT} \leftarrow \pi_{\text{Lname, Fname}}(\text{EMPS\_WITHOUT\_DEPS} * \text{EMPLOYEE})$$

RAMAIAH
Institute of Technology

Query 7. List the names of managers who have at least one dependent

$$MGRS(Ssn) \leftarrow \pi_{Mgr\_ssn}(DEPARTMENT)$$
$$EMPS\_WITH\_DEPS(Ssn) \leftarrow \pi_{Essn}(DEPENDENT)$$
$$MGRS\_WITH\_DEPS \leftarrow (MGRS \cap EMPS\_WITH\_DEPS)$$
$$RESULT \leftarrow \pi_{Lname,\ Fname}(MGRS\_WITH\_DEPS * EMPLOYEE)$$


RAMAIAH
Institute of Technology


RAMAIAH
Institute of Technology