

Database Management Systems

Part II: The Relational Model

Lecture 4

Relational Data Objects: Domains and Relations

Contents

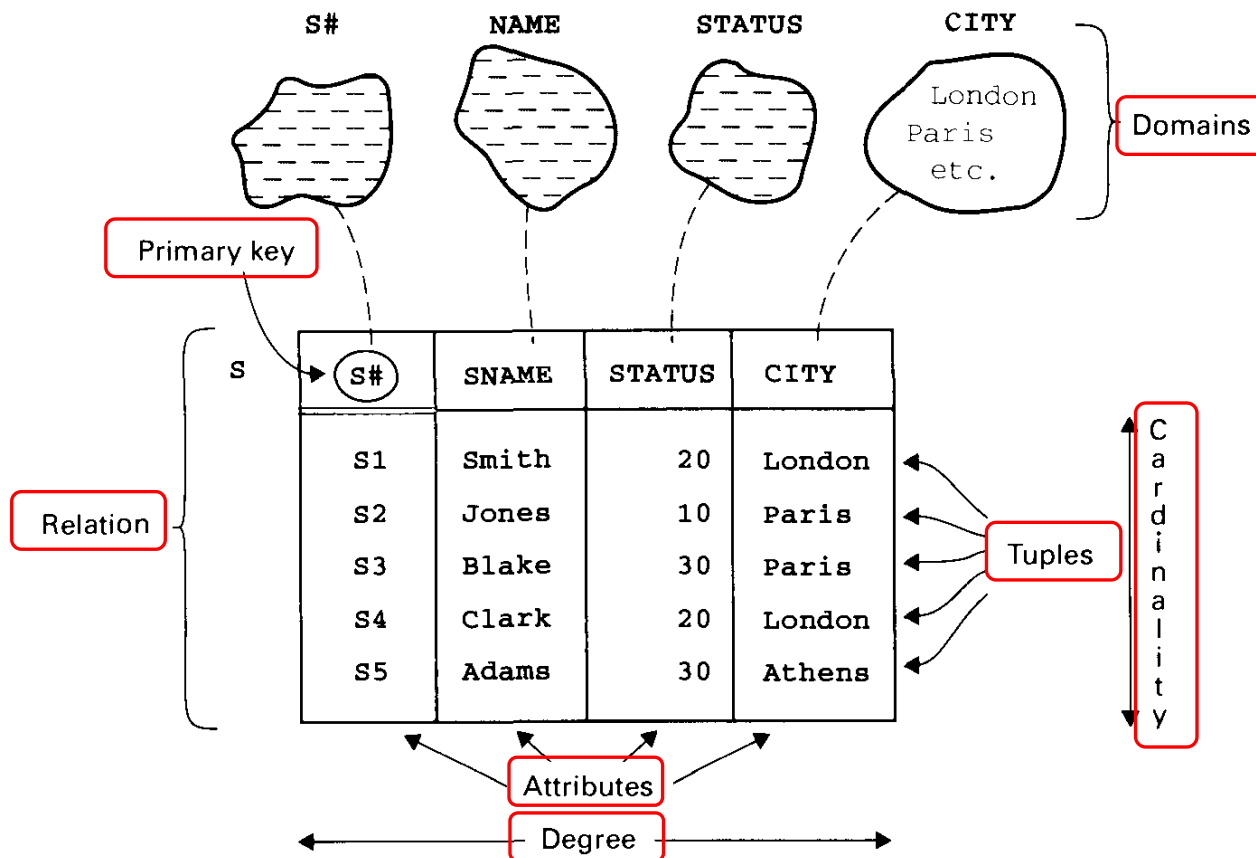
- Introduction
- Relational Data Objects
- Domains
- Relations
- Properties of Relations

Introduction

- The **relational model** is divided into **three parts**:
 - ✓ **data structure (or objects),**
 - ✓ **data integrity, and**
 - ✓ **data manipulation (or operators).**
- All three parts have their own special terms.

Relational Data Objects

- The most important terms used in connection with the **relation data object** part are shown in figure.



Relational Data Objects (Cont.)

Brief Definitions

- **Relation:** A relation corresponds to a **table**.
- **Tuple:** A tuple corresponds to a **row** of such a table.
- **Attribute:** An attribute corresponds to a **column**.

Relational Data Objects (Cont.)

Brief Definitions

- **Cardinality:** The **number of tuples** is called the cardinality.
- **Degree:** The **number of attributes** is called the degree.
- **Primary key:** The primary key is a **unique identifier** for the table.
- **Domain:** Finally, a domain is a **pool of values** from which specific attributes of specific relations draw their actual values.

Relational Data Objects (Cont.)

Terminology

- Figure shows the terminology in relational data objects.

| Formal relational term | Informal equivalents |
|------------------------|----------------------|
| relation | table |
| tuple | row or record |
| cardinality | number of rows |
| attribute | column or field |
| degree | number of columns |
| primary key | unique identifier |
| domain | pool of legal values |

Domains

- For each attribute of a relation, there is a set of permitted values called a domain of that attribute.
- For example, consider the following instructor table which stores information about instructors.

| <i>ID</i> | <i>name</i> | <i>dept_name</i> | <i>salary</i> |
|-----------|-------------|------------------|---------------|
| 10101 | Srinivasan | Comp. Sci. | 65000 |
| 12121 | Wu | Finance | 90000 |
| 15151 | Mozart | Music | 40000 |
| 22222 | Einstein | Physics | 95000 |
| 32343 | El Said | History | 60000 |
| 33456 | Gold | Physics | 87000 |
| 45565 | Katz | Comp. Sci. | 75000 |
| 58583 | Califieri | History | 62000 |
| 76543 | Singh | Finance | 80000 |
| 76766 | Crick | Biology | 72000 |
| 83821 | Brandt | Comp. Sci. | 92000 |
| 98345 | Kim | Elec. Eng. | 80000 |

Domains (Cont.)

- According to the instructor table,
 - ✓ the domain of the ID attribute is the set of all possible instructors' ID,
 - ✓ the domain of the name attribute is the set of all possible instructor names,
 - ✓ the domain of the dept_name attribute is the set of all possible department names, and
 - ✓ the domain of the salary attribute is the set of all possible salary values.

Domains (Cont.)

- A common method of specifying a domain is **to specify a name** for the domain, to help in interpreting its values.
- It is also useful **to specify a data type** from which the data values forming the domain are drawn.

Domains (Cont.)

- Some examples of **domain names** are as follows.
 - ✓ **Student_ID**: To represent the identification number of students in a university
 - ✓ **Student_Names**: To represent the names of students
 - ✓ **Department_Names**: The set of academic department names, such as Computer Science, Economics, and Physics
 - ✓ **GPA**: Possible values of computed grade point averages

Domains (Cont.)

- Some examples of **data type** for domains are as follows.
 - ✓ **Student_ID**: Each must be **character string** (such as CS-1)
 - ✓ **Student_Names**: The set of **character strings**
 - ✓ **Department_Names**: The set of **character strings**
 - ✓ **GPA**: Each must be **a real number**

Domains (Cont.)

- A relation of degree four, which stores information about university students, would contain four attributes describing each student, as follows;

STUDENT (ID, Name, Major, GPA)

- Using the data type of each attribute, the definition is sometimes written as:

STUDENT (ID: string, Name: string, Major: string, GPA: real)

Domains (Cont.)

- **Additional information** for interpreting the values of a domain can also be given.
- For example, a numeric domain such as **Weights** should have the units of measurement, such as **Weights_In_Pounds** or **Weights_In_Kilograms**.

Relations

- A **relation R** on a collection of domains D_1, D_2, \dots, D_n consists of two parts:
 - ✓ **Heading**
 - ✓ **Body**

Relations (Cont.)

Heading

- The heading consists of a **fixed set of attributes**, or more precisely `<attribute-name:domain-name>` pairs,

$\{ \langle A1:D1 \rangle, \langle A2:D2 \rangle, \dots, \langle An:Dn \rangle \}$

Relations (Cont.)

Body

- The body consists of a **set of tuples**.
- Each tuple consists of a set of <attribute-name:attribute-value> pairs.

$\{ \langle A1:vi1 \rangle, \langle A2:vi2 \rangle, \dots, \langle An:vin \rangle \}$, ($i = 1, 2, \dots, m$), where m is number of tuples in the set).

- The value **m** and **n** are called the **cardinality** and the **degree** respectively of relation R .

Relations (Cont.)

- By way of example, **Table S** will be examined.
- First, that table does have **four underlying domains**, namely the domain of supplier numbers (S#), the domain of names (NAME), the domain of status values (STATUS), and the domain of city names (CITY).
- Next, the table certainly does have **two parts** – it has a row of column headings, and it has a set of data rows.

Relations (Cont.)

- Consider the row of column headings first:

(S#, SNAME, STATUS, CITY)

- This row really represents the following set of ordered pairs:

$$\{ \langle S\# \quad : S\# \quad \rangle, \\ \langle SNAME \quad : NAME \rangle, \\ \langle STATUS \quad : STATUS \rangle, \\ \langle CITY \quad : CITY \quad \rangle \}$$

- The first component in each pair is the **attribute name** and the second is the corresponding **domain name**.

Relations (Cont.)

- As for the rest of the table, it certainly does consist of a set, namely a set of rows.

(S1, Smith, 20, London)

- This row really represents the following set of ordered pairs:

{ < S# : 'S1' >,
< SNAME : 'Smith' >,
< STATUS : '20' >,
< CITY : 'London' > }

- The first component in each pair is the **attribute name** and the second is the corresponding **attribute value**.

Relations (Cont.)

- In the following figure, Table S represents **suppliers relation**.
- The **attribute names** are a supplier number (S#), a supplier name (SNAME), a rating or status value (STATUS), and a location (CITY).
- Here, the **domain** of supplier numbers is the set of all possible supplier numbers and its **data type is string**.

Relation S

| S# | SNAME | STATUS | CITY |
|----|-------|--------|--------|
| S1 | Smith | 20 | London |
| S2 | Jones | 10 | Paris |
| S3 | Blake | 30 | Paris |
| S4 | Clark | 20 | London |
| S5 | Adams | 30 | Athens |

Heading (Attribute names)

Body (Attribute values)

Relations (Cont.)

- In the following example, Table P represents **parts relation**.
- It includes **five domains**: a part number (P#), a part name (PNAME), a color (COLOR), a weight in pounds (WEIGHT), and a location where parts are stored (CITY).
- Here, **the data type** of WEIGHT should be **integer greater than 0**.

Relation P

| P# | PNAME | COLOR | WEIGHT | CITY |
|----|-------|-------|--------|--------|
| P1 | Nut | Red | 12 | London |
| P2 | Bolt | Green | 17 | Paris |
| P3 | Screw | Blue | 17 | Rome |
| P4 | Screw | Red | 14 | London |
| P5 | Cam | Blue | 12 | Paris |
| P6 | Cog | Red | 19 | London |

Heading (Attribute names)

Body (Attribute values)

Properties of Relations

- Relations have certain properties.
- The **four properties** are as follows.
 1. There are **no duplicate tuples**.
 2. Tuples are unordered, **top to bottom**.
 3. Attributes are unordered, **left to right**
 4. All attribute values are **atomic**.

Summary

- A **domain** is essentially a data type (possibly system-defined, more generally user-defined).
- Domain provides a **set of scalar values**, from which various attributes in various relations draw their actual values.
- A relation can be thought of as **a table** with the columns and the rows.
- A relation has **two parts**, a heading and a body.
- The heading is a **set of attributes** and the body is a **set of tuples**.

Next Lecture

Part II: The Relational Model

Relational Data Integrity

- Candidate Keys
- Primary Keys
- Foreign Keys
- Nulls

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.
- <https://www.guru99.com>
- [https:// www.includehelp.com](https://www.includehelp.com)
- [https:// www.tutorialspoint.com](https://www.tutorialspoint.com)