جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الامن السيبراني
Department of Cyber Security

Subject:

Database

Class:

Second

Lecturer:

M.Sc. Ali Haider Alazam

M.Sc. Mohammad Baqer Haleem

Lecture: (2)

Relational Model

Relational Model

The relational model is the dominant data model for commercial data-processing systems due to its simplicity and flexibility. Its straightforward structure makes database programming easier than earlier models such as hierarchical and network models. Over time, the relational model has evolved by incorporating advanced features, including object-relational capabilities, support for XML, and mechanisms for managing semi-structured data. Its independence from underlying storage structures has enabled it to remain relevant despite modern innovations such as column-oriented storage systems used for large-scale data analytics. This chapter introduces the fundamental concepts of the relational model, while later chapters explore its theoretical foundations in database design, query processing, and formal relational languages.

Structure of Relational Databases

A relational database is composed of multiple tables, each with a unique name. Each table consists of columns (attributes) and rows (tuples), where each row represents a single record. For example, an instructor table stores information such as ID, name, department, and salary for each instructor, while a course table records details including course ID, title, department, and credits. In each table, a specific column—such as ID for instructors or course id for courses—uniquely identifies each record.

ID	*name*	*dept_name*	*salary*
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

Figure 2.1 The *instructor* relation.

Figure 2.3 shows a third table, prereq, which stores the prerequisite courses for each course. The table has two columns, course id and prereq id. Each row consists of a pair of course identifiers such that the second course is a prerequisite for the first course. Thus, a row in the prereq table indicates that two courses are related in the sense that one course is a prerequisite for the other. As another example, when we consider the table instructor, a row in the table can be thought of as representing the relationship.

course_id	*title*	*dept_name*	*credits*
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

Figure 2.2 The *course* relation.

course_id	*prereq_id*
BIO-301	BIO-101
BIO-399	BIO-101
CS-190	CS-101
CS-315	CS-101
CS-319	CS-101
CS-347	CS-101
EE-181	PHY-101

Figure 2.3 The *prereq* relation.

In the relational model, a table represents a mathematical relation, where each row corresponds to a tuple and each column corresponds to an attribute. A tuple is an ordered sequence of values that represents a relationship among multiple attributes. Thus, a relation is a collection of tuples, and a specific set of tuples at a given time is called a relation instance. The order of tuples in a relation is irrelevant, since a relation is defined as a set. Each attribute in a relation has a defined domain, which specifies the set of allowable values for that attribute.

Database Schema

A database schema defines the logical structure of a database, while a database instance represents the actual data stored at a specific point in time. A relation schema specifies the attributes and their domains and is analogous to a type definition in programming, whereas a relation instance corresponds to a variable's value and may change over time. Although schemas remain relatively stable, instances are updated as data changes. Common attributes across relation schemas are used to relate data from different relations, enabling meaningful queries. In a university database, multiple relation schemas—such as instructor, department, course, section, and teaches—collectively represent different aspects of the

academic environment, with each schema modeling a specific real-world concept or relationship.

Keys

Keys are used in the relational model to uniquely identify tuples within a relation. A superkey is any set of one or more attributes whose values uniquely identify a tuple, while a candidate key is a minimal superkey with no extraneous attributes. From the candidate keys, one is selected as the primary key, which serves as the principal identifier and represents a real-world constraint. Primary keys may consist of single or multiple attributes and should be chosen so that their values are stable and rarely change.

Foreign-key constraints enforce relationships between relations by ensuring referential integrity. A foreign key in a referencing relation must match a primary-key value in a referenced relation, preventing invalid references. Foreign keys are a special case of referential integrity constraints, which more generally require that attribute values in one relation correspond to existing values in another. These constraints ensure consistency and correctness across related relations in a database.

Schema Diagrams

Schema diagrams visually represent a database schema along with its primary-key and foreign-key constraints. In such diagrams, each relation is shown as a box containing its attributes, with primary-key attributes underlined. Foreign-key constraints are depicted using arrows from the referencing attributes to the primary key of the referenced relation, while double-headed arrows indicate referential integrity constraints that are not foreign keys. Although many database systems provide graphical tools for creating schema diagrams, they should not be confused with entity–relationship diagrams, which use a different notation and are discussed separately.

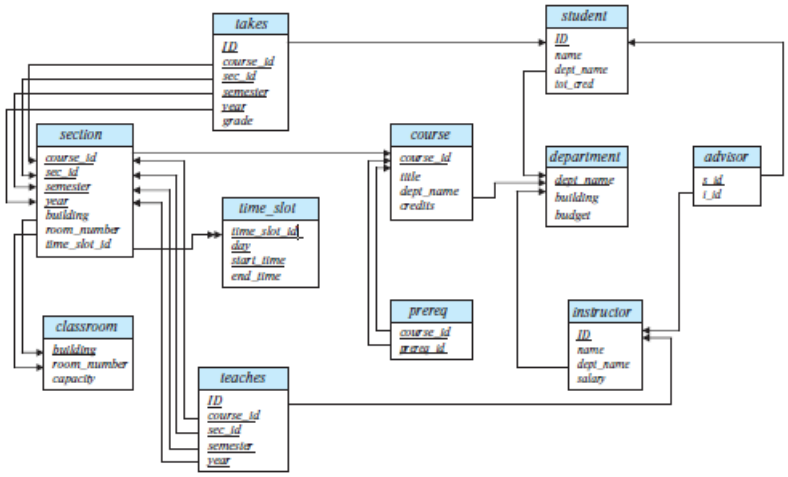

Figure 2.9 Schema diagram for the university database.

Relational Query Languages

A query language enables users to request information from a database at a level higher than traditional programming languages. Query languages are classified as imperative, where users specify the exact sequence of operations; functional, where queries are expressed as evaluations of side-effect-free functions; and declarative, where users describe what data is needed without specifying how to obtain it. Pure query languages, such as relational algebra and relational calculus, provide formal foundations for data retrieval, while practical languages like SQL combine elements of imperative, functional, and declarative approaches.

The Relational Algebra

Relational algebra is a formal procedural language consisting of operations that take one or two relations as input and produce a new relation.

- Fundamental Operations

    1- Select (σ): Extracts tuples that satisfy a specific condition.

    Example: $\sigma_{salary > 90000}(instructor)$

2- Project (Π): Returns specific columns and eliminates duplicates.

   Example: $\Pi_{ID, name, salary}(instructor)$

3- Union ($\cup$): Combines tuples from two compatible relations.
4- Set Difference ($- house$): Finds tuples in one relation but not in another.
5- Cartesian Product ($\times$): Combines every tuple from one relation with every tuple from another.
6- Rename (ρ): Gives a name to the result of an algebraic expression.

- Join Operations

  Natural Join ($\bowtie$): Joins two relations based on common attributes, keeping only one copy of the common columns.