DBMS CHAPTER 5

The Relational Data Model and Relational Database Constraints

Prepared for B.Sc(H)Comp.Sc - sem IV

By: Ms.Shweta Wadhera

CHAPTER OUTLINE

- Relational Model Concepts
- Relational Model Constraints and Relational Database Schemas
- Update Operations and Dealing with Constraint Violations

RELATIONAL MODEL CONCEPTS

- The relational Model of Data is based on the concept of a *Relation*
  - In this chapter we talk about the basic characteristics of the Relational Data Model and its constraints .
  - This Model uses the concept of a Mathematical relation , which looks like a table of values .
  - ***A Relation is a mathematical concept based on the idea of sets.***

RELATIONAL MODEL CONCEPTS

- The model was first proposed by Dr. E.F. Codd of IBM Research in 1970 in the following paper:
  - "A Relational Model for Large Shared Data Banks," Communications of the ACM, June 1970

- The above paper caused a *major revolution* in the field of database management and earned Dr.Codd the coveted ACM Turing Award .

- The 1st commercial implementation of the relational model became available in 1980s .

INFORMAL DEFINITIONS

- The relational model represents the database as a collection of relations .
- Each relation resembles a Table of Values.
- A relation typically contains a **set of rows**.
- The data elements in each **row** represent certain facts that correspond to a real-world **entity** or **relationship**

  In the formal model, rows are called tuples

- Each **column** has a column header that specify how to interpret the Data Values in each row .

In the formal model, the column header is called an attribute name (or just attribute)

EXAMPLE OF A RELATION

Relation Name → **STUDENT**

Attributes → (column headers)

Tuples → (rows)

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

Figure 5.1
The attributes and tuples of a relation STUDENT.

INFORMAL DEFINITIONS

- **Key of a Relation:**
  - Each row has a value of a data item (or set of items) that uniquely identifies that row in the table ***called the key***

In the STUDENT table, ***SSN is the key.***

FORMAL DEFINATIONS

- ***Domain*** :- 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 relational model is concerned .

- We generally give a name to the domain , ***because it helps in interpreting its values.***

- A datatype or format is also specified for each domain.

FORMAL DEFINITIONS - DOMAIN

- A **domain** has a logical definition:
  - Example: “phone_numbers_of India ” are the set of 10 digit mobile numbers valid in the India.
- A domain also has a data-type or a format defined for it.
  - The phone_numbers_India may have a format: (ddd)ddd-dddd where each d is a decimal digit.
  - Dates have various formats such as year, month, date formatted as yyyy-mm-dd, or as dd mm,yyyy etc.
- The attribute name designates the role played by a domain in a relation:
  - Used to interpret the meaning of the data elements corresponding to that attribute
  - Example: The domain Date may be used to define two attributes named “Invoice-date” and “Payment-date” with different meanings

FORMAL DEFINITIONS - SCHEMA

- **The Schema (or description) of a Relation:**
  - A relation schema is used to describe a relation.
  - Denoted by R (A1, A2, ...An)
  - R is the **name** of the relation
  - D is called the domain of Ai and is denoted by dom(Ai).
  - The **attributes** of the relation are A1, A2, ..., An
  - A relation schema ***R(A1, A2,….,An)*** , is made up of a relation named 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 .

CONTD ...

- **The degree of a relation** is the number of attributes n of its relation schema .

- Example:

  CUSTOMER (Cust-id, Cust-name, Address, Phone#)

  - CUSTOMER is the relation name
  - Defined over the four attributes: Cust-id, Cust-name, Address, Phone#

- Each attribute has a **domain** or a set of valid values.
  - For example, the domain of Cust-id is 6 digit numbers.

RELATIONAL DATABASE SCHEMA

- **Relational Database Schema:**
  - A set S of relation schemas that belong to the same database.
  - S is the name of the whole **database schema**
  - S = {R1, R2, ..., Rn}
  - R1, R2, …, Rn are the names of the individual **relation schemas** within the database S
- Following slide shows a COMPANY database schema with 6 relation schemas

COMPANY DATABASE SCHEMA

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 5.5
Schema diagram for the COMPANY relational database schema.

FORMAL DEFINITIONS - TUPLE

- **A tuple is an ordered set of values** (enclosed in angled brackets ‘< … >’)
- Each value is derived from its **corresponding *domain*.**

- A row in the CUSTOMER relation is a 4-tuple and would consist of four values, for example:
    - <632895, "John Smith", "101 Main St. Atlanta, GA 30332", "(404) 894-2000">
    - This is called a 4-tuple as it has 4 values

- A relation is a **set** of such tuples (rows)

FORMAL DEFINITIONS – *RELATION STATE*

- **A Relation State** :- The **relation state** is a **subset of** the Cartesian product of the domains of its attributes
  - each domain contains the set of all possible values the attribute can take.
- **Example:** attribute Cust-name is defined over the domain of character strings of maximum length 25
  - dom(Cust-name) is varchar(25)
- The role these strings play in the CUSTOMER relation is that of the *name of a customer*.

FORMAL DEFINITIONS - SUMMARY

- Formally,
  - Given R(A1, A2, ..., An)
  - r(R) $\subset$ dom (A1) X dom (A2) X ....X dom(An)
- R(A1, A2, …, An) is the **schema** of the relation
- R is the **name** of the relation
- A1, A2, …, An are the **attributes** of the relation
- r(R): a specific **state** of relation R
  - r(R) = {t1, t2, …, tn} where each ti is an n-tuple
  - ti = <v1, v2, …, vn> where each vj *element-of* dom(Aj)

Defination of Relation State :

A relation r , of the relation schema R(A1,A2,….,An) also denoted by r(R), is a set of n-tuples . **Each tuple will be an n-tuple ,bcoz there are "n attributes"**

r (R)= { t1, t2, ….., tn }

CONTD ….

- **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 .**
- Null value …. Values unknown or which do not exist or may not apply to a particular entity . Ex : Off_phone_no.

FORMAL DEFINITIONS - EXAMPLE

- Let R(A1, A2) be a relation schema:
  - Let dom(A1) = {0,1}
  - Let dom(A2) = {a,b,c}
- Then: dom(A1) X dom(A2) is all possible combinations:

  {<0,a> , <0,b> , <0,c>, <1,a>, <1,b>, <1,c> }
- The relation state r(R) $\subset$ dom(A1) X dom(A2)
- For example: r(R) could be {<0,a> , <0,b> , <1,c> }
  - this is one possible state (or "population" or "extension") r of the relation R, defined over A1 and A2.
  - It has three 2-tuples: <0,a> , <0,b> , <1,c>

DEFINITION SUMMARY

Informal Terms		Formal Terms
Table		Relation
Column Header		Attribute
All possible Column Values		Domain
Row		Tuple
Table Definition		Schema of a Relation
Populated Table		State of the Relation

CHARACTERISTICS OF RELATIONS

- **Ordering of tuples in a relation r(R):**
  - 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.
  - Also , defination of a relation does not specify any order.
  - So there is no preference for one logical ordering over another.
  - Hence relation states with different ordering of tuples are considered as identical to each other.

EXAMPLE – A RELATION STUDENT

Relation Name

Attributes

Tuples

STUDENT

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

Figure 5.1
The attributes and tuples of a relation STUDENT.

SAME STATE AS PREVIOUS FIGURE (BUT WITH DIFFERENT ORDER OF TUPLES)

Figure 5.2
The relation STUDENT from Figure 5.1 with a different order of tuples.

STUDENT

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

- ***Ordering of attributes in a relation schema R (and of values within each tuple):***

  - We will consider the attributes in R(A1, A2, ..., An) and the values in t = <v1, v2, ..., vn> to be ordered .

  - According to the preceding defination of a relation , ***an n-tuple is an ordered list of n-values*** , so the ordering of values in a tuple is there…..
  - and hence ordering of attributes in a relation schema is important .

CHARACTERISTICS OF RELATIONS

- **Values in a tuple:**
  - All values are considered *atomic (indivisible).*
  - So composite and Multivalued attributes are NOT allowed .
  - Each value in a tuple must be from the domain of the attribute for that column
    - If tuple t = <v1, v2, …, vn> is a tuple (row) in the relation state r of R(A1, A2, …, An)
    - Then each *vi* must be a value from *dom(Ai)*
  - A *special* ***null*** *value* is used to represent values that are unknown or inapplicable to certain tuples.

CHARACTERISTICS OF RELATIONS

- **Notation:**
  - We refer to **component values** of a tuple t by:
    - t [Ai]   or   t . Ai
    - This is the value vi of attribute Ai for tuple t

RELATIONAL MODEL CONSTRAINTS

- *Constraints are conditions that must hold on all valid relation states.*

- There are generally many **restrictions or constraints** on the actual values in a DB state .

CATEGORIES OF CONSTRAINTS

- Model Based Constraints :- Constraints that are inherent in the data model .

- Schema Based Constraints :- Constraints that can be directly expressed in the schemas of the data model.(by specifying them in DDL data defination Lang).

- Application based Constraints :- Constraints that cannot be directly expressed in the schemas of the data model, and hence must be expressed and enforced by the application programs .

SCHEMA BASED CONSTRAINTS

These include ……

- Domain Constraints
- Key Constraints
- Constraints on Nulls
- Entity Integrity constraints
- Referential Integrity constraints

- **Domain Constraints**

Domain Constraints specify that within each tuple, the value of each attribute Ai must be an atomic value from the domain dom(Ai).

KEY CONSTRAINT

- A relation is a set of Tuples .
- All elements of a set *are distinct* .

→ All tuples in a relation *must be distinct .*

→ No two tuples can have the same combination of values.

- Suppose we denote one subset of values by sk .
- Then for any two distinct tuples t1 and t2 in a relation state r of R , we have the constraint

  t1[sk] != t2[sk]

→ Any such set of attributes sk is called a superkey of R .

➔ In a way we can say that …..

A superkey sk specifies the uniqueness constraint ….

i.e No two distinct tuples in any state r of R can have the same value for sk.

➔ Infact every relation has atleast one default superkey ---- the set of all its attributes .

KEY CONSTRAINTS

- **Superkey** of R:
  - Is a set of attributes SK of R with the following condition:
    - No two tuples in any valid relation state r(R) will have the same value for SK
    - That is, for any distinct tuples t1 and t2 in r(R), t1[SK] $\neq$ t2[SK]
    - This condition must hold in *any valid state* r(R)
- **Key** of R:
  - A "minimal" superkey
  - That is, a key is a superkey K such that removal of any attribute from K results in a set of attributes that is not a superkey (does not possess the superkey uniqueness property)

- Example
-

Sno	Pno	Qty_sold
s1	p1	500
s1	p2	200
s1	p3	500
s2	p1	500
s2	p2	200
s2	p3	500
s3	p2	200

(Sno,Pno,Qty_sold) ….. SuperKey

(Sno.Pno) ………. Minimal Superkey

KEY CONSTRAINTS (CONTINUED)

- Example: Consider the CAR relation schema:
  - CAR(State, Reg#, SerialNo, Make, Model, Year)
  - CAR has two keys:
    - Key1 = {State, Reg#}
    - Key2 = {SerialNo}
  - Both are also superkeys of CAR
  - {SerialNo, Make} is a superkey but *not* a key.
- In general:
  - Any *key* is a *superkey* (but not vice versa)
  - Any set of attributes that *includes a key* is a *superkey*
  - A *minimal* superkey is also a key

KEY CONSTRAINTS (CONTINUED)

- If a relation has several **candidate keys**, one is chosen arbitrarily to be the **primary key**.
  - The primary key attributes are <u>underlined</u>.

- Example: Consider the CAR relation schema:
  - CAR(State, Reg#, <u>SerialNo</u>, Make, Model, Year)
  - We chose SerialNo as the primary key

- The primary key value is used to *uniquely identify* each tuple in a relation
  - Provides the tuple identity
- Also used to *reference* the tuple from another tuple

  - General rule: Choose as primary key the smallest of the candidate keys (in terms of size)
  - Not always applicable – choice is sometimes subjective

CAR TABLE WITH TWO CANDIDATE KEYS – LICENSENUMBER CHOSEN AS PRIMARY KEY

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 5.4
The CAR relation, with two candidate keys: License_number and Engine_serial_number.

ENTITY INTEGRITY

- **Entity Integrity:**
  - The *primary key attributes* PK of each relation schema R in S cannot have null values in any tuple of r(R).
    - This is because primary key values are used to *identify* the individual tuples.
    - t[PK] $\neq$ null for any tuple t in r(R)
    - If PK has several attributes, null is not allowed in any of these attributes
  - Note: Other attributes of R may be constrained to disallow null values, even though they are not members of the primary key.

REFERENTIAL INTEGRITY

- A constraint involving **two** relations
  - The previous constraints involve a single relation.
- Used to specify a **relationship** among tuples in two relations:
  - The **referencing relation** and the **referenced relation**.

REFERENTIAL INTEGRITY

- Tuples in the **referencing relation** R1 have attributes FK (called **foreign key** attributes) that reference the primary key attributes PK of the **referenced relation** R2.
  - A tuple t1 in R1 is said to **reference** a tuple t2 in R2 if t1[FK] = t2[PK].
- A referential integrity constraint can be displayed in a relational database schema as a directed arc from R1.FK to R2.

REFERENTIAL INTEGRITY (OR FOREIGN KEY) CONSTRAINT

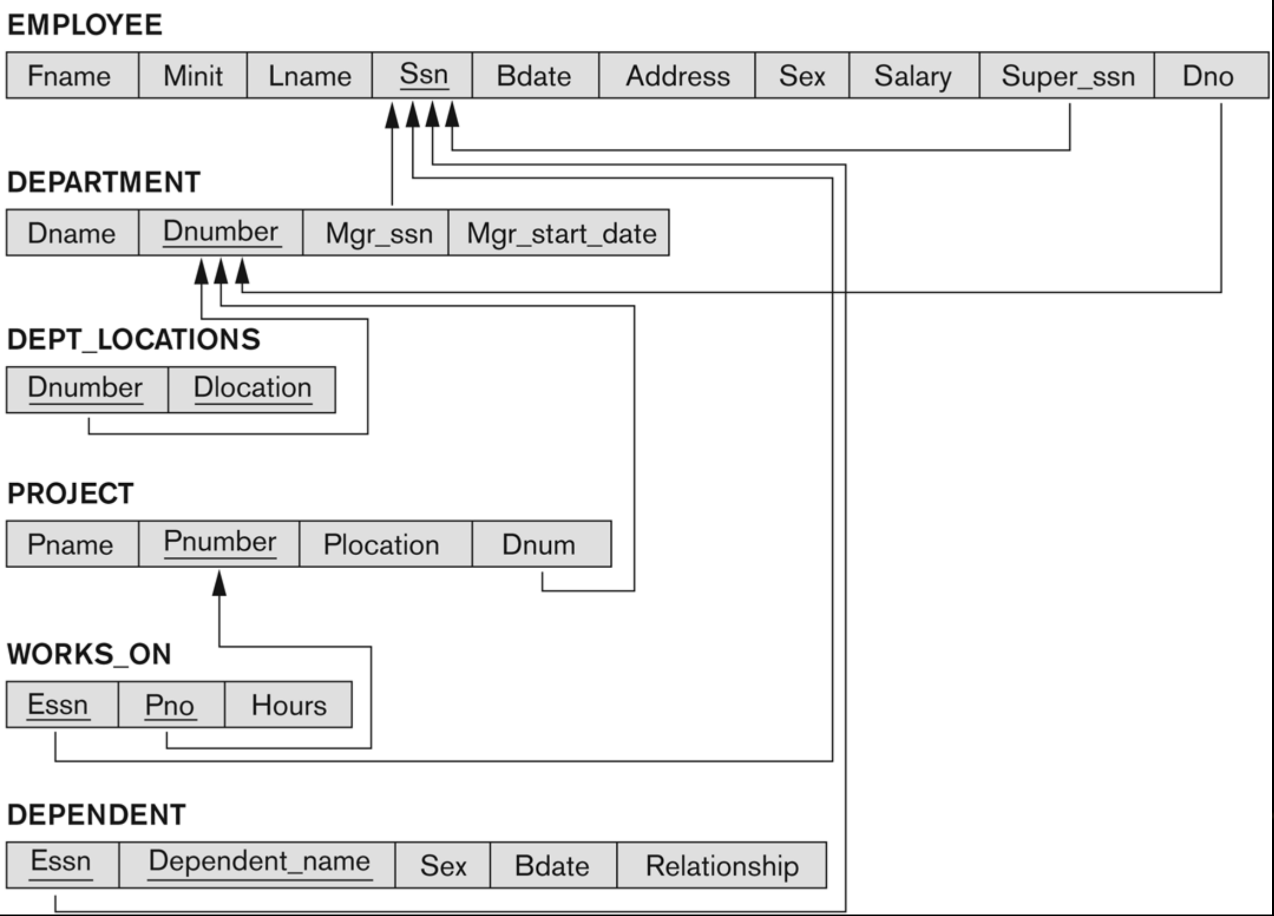
- Statement of the constraint
  - The value in the foreign key column (or columns) FK of the the **referencing relation** R1 can be **either**:
    - (1) a value of an existing primary key value of a corresponding primary key PK in the **referenced relation** R2, <u>or</u>
    - (2) a **null**.
- In case (2), the FK in R1 should **not** be a part of its own primary key.

DISPLAYING A RELATIONAL DATABASE SCHEMA AND ITS CONSTRAINTS

- Each relation schema can be displayed as a row of attribute names
- The name of the relation is written above the attribute names
- The primary key attribute (or attributes) will be underlined
- A foreign key (referential integrity) constraints is displayed as a directed arc (arrow) from the foreign key attributes to the referenced table
  - Can also point the the primary key of the referenced relation for clarity
- Next slide shows the COMPANY **relational schema diagram**

Referential Integrity Constraints for COMPANY database

Figure 5.7
Referential integrity constraints displayed on the COMPANY relational database schema.



SUMMARY

- Presented Relational Model Concepts
  - Definitions
  - Characteristics of relations
- Discussed Relational Model Constraints and Relational Database Schemas
  - Domain constraints'
  - Key constraints
  - Entity integrity
  - Referential integrity

IN-CLASS EXERCISE

(Taken from Exercise 5.15)

Consider the following relations for a database that keeps track of student enrollment in courses and the books adopted for each course:

STUDENT(SSN, Name, Major, Bdate)

COURSE(Course#, Cname, Dept)

ENROLL(SSN, Course#, Quarter, Grade)

BOOK_ADOPTION(Course#, Quarter, Book_ISBN)

TEXT(Book_ISBN, Book_Title, Publisher, Author)

Draw a relational schema diagram specifying the foreign keys for this schema.