Database Management System (DBMS)
B.Sc. CSIT
Fourth Semester

Unit 4.
The Relational Data Model and Relational Database Constraints

Date: **20th March, 2023**

Unit 4. The Relational Data Model and Relational Database Constraints

3 hours

Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations, Transactions, and Dealing with Constraint Violations

RELATIONAL MODEL CONCEPTS

- The relational Model of Data is based on the concept of a Relation.
- A Relation is a mathematical concept based on the ideas of sets.
- The strength of the relational approach to data management comes from the formal foundation provided by the theory of relations.
- We review the essentials of the relational approach in this unit.

Relational Model Concepts

- The model was first proposed by Dr. E.F. Codd of IBM 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 Ted Codd the coveted ACM Turing Award.

INFORMAL DEFINITIONS

- RELATION: A table of values

  - A relation may be thought of as a **set of rows**.
  - A relation may alternately be thought of as a **set of columns**.
  - Each row represents a fact that corresponds to a real-world **entity** or **relationship**.
  - Each row has a value of an item or set of items that uniquely identifies that row in the table.
  - Sometimes row-ids or sequential numbers are assigned to identify the rows in the table.
  - Each column typically is called by its column name or column header or attribute name.

FORMAL DEFINITIONS

- A **Relation** may be defined in multiple ways.
- The **Schema** of a Relation: *R* (A1, A2, ...An)

Relation schema *R* is defined over **attributes** A1, A2, ...An

For Example -

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

Here, CUSTOMER is a relation defined over the four attributes Cust-id, Cust-name, Address, Phone#, each of which has a **domain** or a set of valid values. For example, the domain of Cust-id is 6 digit numbers.

- A **tuple** is an ordered set of values
- Each value is derived from an appropriate domain.
- Each row in the CUSTOMER table may be referred to as a tuple in the table and would consist of four values.
- **<632895, "John Smith", "101 Main St. Atlanta, GA 30332", "(404) 894-2000">**
  is a tuple belonging to the CUSTOMER relation.
- A relation may be regarded as a ***set of tuples*** (rows).
- Columns in a table are also called attributes of the relation.

- A **domain** has a logical definition: e.g., "USA_phone_numbers" are the set of 10 digit phone numbers valid in the U.S.
- A domain may have a data-type or a format defined for it. The USA_phone_numbers may have a format: (ddd)-ddd-dddd where each d is a decimal digit. E.g., Dates have various formats such as monthname, date, year or yyyy-mm-dd, or dd mm,yyyy etc.
- An attribute designates the **role** played by the domain. E.g., the domain Date may be used to define attributes "Invoice-date" and "Payment-date".

- The relation is formed over the cartesian product of the sets; each set has values from a domain; that domain is used in a specific role which is conveyed by the attribute name.
- For example, attribute Cust-name is defined over the domain of strings of 25 characters. The role these strings play in the CUSTOMER relation is that of the name of customers.
- Formally,

  Given $R(A_1, A_2, \ldots\ldots\ldots., A_n)$

  $$r(R) \subset dom\ (A_1)\ X\ dom\ (A_2)\ X\ \ldots.X\ dom(A_n)$$
- R: schema of the relation
- r of R: a specific "value" or population of R.
- R is also called the **intension** of a relation
- r is also called the **extension** of a relation

- Let S1 = {0,1}
- Let S2 = {a,b,c}

- Let R $\subset$ S1 X S2

- Then for example: r(R) = {<0,a> , <0,b> , <1,c> }

  is one possible “state” or “population” or “extension” r of the relation R, defined over domains S1 and S2. It has three tuples.

DEFINITION SUMMARY

Informal Terms	Formal Terms
Table	Relation
Column	Attribute/Domain
Row	Tuple
Values in a column	Domain
Table Definition	Schema of a Relation
Populated Table	Extension

Example

Relation name

Attributes

Tuples

STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	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
	Charles Cooper	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

Characteristics of Relations

- **Ordering of tuples in a relation r(R)**: The tuples are *not* considered to be ordered, even though they appear to be in the tabular form.
- **Ordering of attributes in a relation schema R** (and of values within each tuple): We will consider the attributes in $R(A_1, A_2, ..., A_n)$ and the values in $t=<v_1, v_2, ..., v_n>$ to be *ordered* .

  (However, a more general *alternative definition* of relation does not require this ordering).
- **Values in a tuple**: All values are considered *atomic* (indivisible). A special **null** value is used to represent values that are unknown or inapplicable to certain tuples.

- Notation:

- We refer to **component values** of a tuple t by $t[A_i] = v_i$ (the value of attribute A_i for tuple t).

  Similarly, $t[A_u, A_v, ..., A_w]$ refers to the subtuple of t containing the values of attributes $A_u, A_v, ..., A_w$, respectively.

	A1	A2	A3	A4	A5	A6	A7
STUDENT	Name	SSN	HomePhone	Address	OfficePhone	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
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Katherine Ashly	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

$$t[A1, A2, A3, A4, A5, A6, A7]$$

$$= [v1, v2, v3, v4, v5, v6, v7]$$

$$= [\text{'Charles Cooper'}, \text{'489-22-1100'}, \text{'376-9821'},$$

$$\text{'265 Lark Lane'}, \text{'749-6492'}, 28, 3.93]$$

RELATIONAL INTEGRITY CONSTRAINTS

- Constraints are *conditions* that must hold on *all* valid relation instances. There are three main types of constraints:
  1. **Key** constraints
  2. **Entity integrity** constraints
  3. **Referential integrity** constraints

KEY CONSTRAINTS

A	B	C
1	a	x
2	b	y
3	c	y
4	a	z

Keys: A, BC

A [A$\rightarrow$BC]

$A^+ \rightarrow$ [ABC]

BC [BC$\rightarrow$A]

$(BC)^+ \rightarrow$ [ABC]

KEY CONSTRAINTS

Suppose
R(A,B,C,D) is a schema or relation R with attributes A,B,C and D respectively.
and if

$$ABC \rightarrow D$$
$$AB \rightarrow CD$$
$$A \rightarrow BCD$$

then,
Super Keys = {ABC, AB, A}, Candidate Key = {A}

Note:

ABC→D *means we can use attributes A,B and C to find the value of attribute D. This means functional dependency of attribute D on attributes A, B and D. Thus, closure of ABC gives ABCD.*
Similar for other keys.

KEY CONSTRAINTS

Suppose
R(A,B,C,D) is a schema or relation R with attributes A,B,C and D respectively.
and if

$B \rightarrow ACD$
$ACD \rightarrow B$

then,
Super Keys = {B, ACD}, Candidate Key = {B, ACD}

KEY CONSTRAINTS

- **<u>Superkey</u>** <u>of R:</u> A set of attributes SK of R such that no two tuples *in any valid relation instance 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].
- **<u>Candidate Key</u>** <u>of R:</u> A "minimal" superkey; that is, a superkey K such that removal of any attribute from K results in a set of attributes that is not a superkey.

<u>Example</u><u>:</u> The CAR relation schema:

CAR(<u>State</u>, <u>Reg#</u>, SerialNo, Make, Model, Year)

has two keys Key1 = {State, Reg#}, Key2 = {SerialNo}, which are also superkeys. {SerialNo, Make} is a superkey but *not* a candidate key.

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

KEY CONSTRAINTS

Figure 7.4 The CAR relation with two candidate keys: LicenseNumber and EngineSerialNumber.

CAR	LicenseNumber	EngineSerialNumber	Make	Model	Year
	Texas ABC-739	A69352	Ford	Mustang	96
	Florida TVP-347	B43696	Oldsmobile	Cutlass	99
	New York MPO-22	X83554	Oldsmobile	Delta	95
	California 432-TFY	C43742	Mercedes	190-D	93
	California RSK-629	Y82935	Toyota	Camry	98
	Texas RSK-629	U028365	Jaguar	XJS	98

ENTITY INTEGRITY

- **Relational Database Schema**: A set S of relation schemas that belong to the same database. S is the *name* of the **database**.

$$S = \{R_1, R_2, ..., R_n\}$$

- **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.

$$t[PK] \neq \text{null for any tuple } t \text{ in } r(R)$$

- Note: Other attributes of R may be similarly 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**.
- Tuples in the *referencing relation* R_1 have attributes FK (called **foreign key** attributes) that reference the primary key attributes PK of the *referenced relation* R_2. A tuple t_1 in R_1 is said to **reference** a tuple t_2 in R_2 if $t_1[FK] = t_2[PK]$.
- A referential integrity constraint can be displayed in a relational database schema as a directed arc from $R_1.FK$ to R_2.

REFERENTIAL INTEGRITY CONSTRAINT

Statement of the constraint

- The value in the foreign key column (or columns) FK of the the **referencing relation** R_1 can be either:

  (1) a value of an existing primary key value of the corresponding primary key PK in the **referenced relation** $R_{2,}$, or

  (2) a null.

- In case (2), the FK in R_1 should not be a part of its own primary key.

Other Types of Constraints

Semantic Integrity Constraints:

- based on application semantics and cannot be expressed by the model
- E.g., “the max. no. of hours per employee for all projects he or she works on is 56 hrs per week”
- A *constraint specification language* may have to be used to express these
- SQL-99 allows triggers and ASSERTIONS to allow for some of these

UPDATE OPERATIONS ON RELATIONS

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: *Schema diagram for COMPANY relational database*

Update Operations on Relations

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

Figure continued on next slide…

UPDATE OPERATIONS ON RELATIONS

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

Figure: *One possible relational database state corresponding to COMPANY Schema*

UPDATE OPERATIONS ON RELATIONS

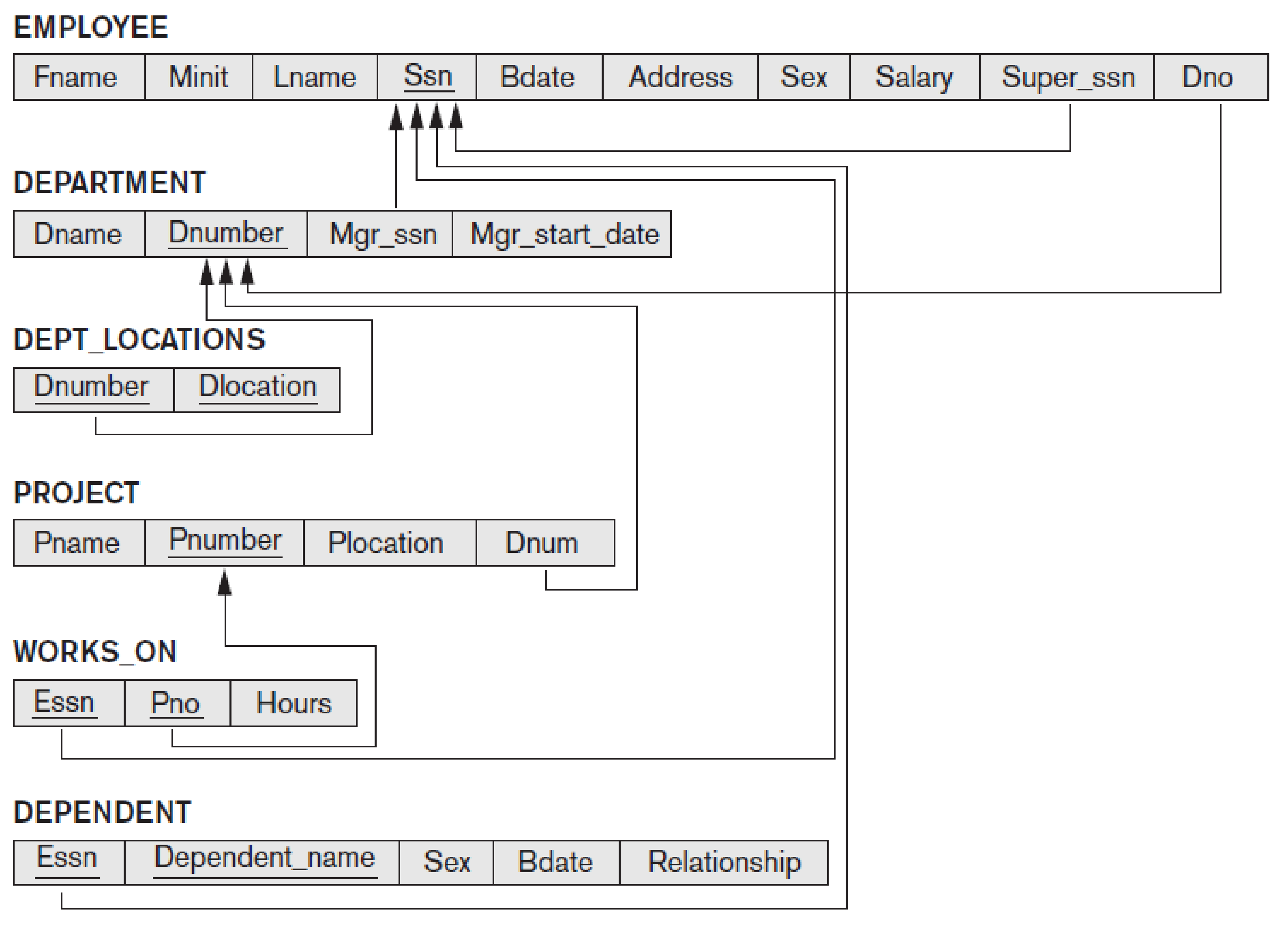

Figure: *Referential integrity constraints displayed on the COMPANY relational database schema diagram.*

UPDATE OPERATIONS ON RELATIONS

- INSERT a tuple.
- DELETE a tuple.
- MODIFY a tuple.

- Integrity constraints should not be violated by the update operations.
- Several update operations may have to be grouped together.
- Updates may *propagate* to cause other updates automatically. This may be necessary to maintain integrity constraints.

DEALING WITH CONSTRAINT VIOLATIONS

- In case of integrity violation, several actions can be taken:
  - Cancel the operation that causes the violation (REJECT option)
  - Perform the operation but inform the user of the violation
  - Trigger additional updates so the violation is corrected (CASCADE option, SET NULL option)
  - Execute a user-specified error-correction routine

In Class Exercise

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

STUDENT(<u>SSN</u>, Name, Major, Bdate)

COURSE(<u>Course#</u>, Cname, Dept)

ENROLL(<u>SSN</u>, <u>Course#</u>, <u>Quarter</u>, Grade)

BOOK_ADOPTION(<u>Course#</u>, <u>Quarter</u>, Book_ISBN)

TEXT(<u>Book_ISBN</u>, Book_Title, Publisher, Author)

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

SOLUTION

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

- the attribute SSN of relation ENROLL that references relation STUDENT
- the attribute Course# in relation ENROLL that references relation COURSE
- the attribute Course# in relation BOOK_ADOPTION that references relation COURSE
- the attribute Book_ISBN of relation BOOK_ADOPTION that references relation TEXT

TRANSACTION

→ In DBMS, a transaction is a sequence of one or more database operations that are treated as a single, indivisible unit of work. A transaction is an all-or-nothing proposition, meaning that either all the operations in the sequence are completed successfully, or none of them are, leaving the database in its original state.

→ A transaction typically involves reading or modifying data stored in a database, such as inserting, updating, or deleting records. Examples of transactions might include transferring funds between bank accounts, updating a customer's order status, or modifying an employee's record.

→ Transactions are important in DBMS because they help ensure data consistency, reliability, and integrity. By grouping multiple database operations into a single transaction, the system can guarantee that the data is always in a valid and consistent state, even in the face of concurrent updates or hardware failures. If an error occurs during a transaction, the system can "rollback" or undo all the operations in the transaction, restoring the database to its original state before the transaction began.

Thank You!!!