Unit II: Data Modeling and Database Design

Chapter 04: Relational Data Model Concepts And Relational Database Constraints

1. Relational Model Concepts

- ✓ The relational Model represents the database as a collection of Relations.
- ✓ A Relation is corresponds to concept of entity type or relationship type in ER-Model
- ✓ A table is called **relation**, a row is called a **tuple** and column reader/name is called an **attribute**
- ✓ Each tuple in the relation represents a collection of related data values that corresponds to the real-world entity or relationship.
- ✓ A Relation is a mathematical concept based on the ideas of set theory.
- ✓ The model was first proposed by Dr. E.F. Codd of IBM in 1970 and revoluted in the field of Database Management

2. Formal Definitions

- ✓ A **Relation** may be defined in multiple ways.

- ✓ The **Schema** of a Relation denoted by *R* (A1, A2, ...An) is defined over the list of **attributes** A1, A2, ...An. e.g -

  STUDENT (RollNo, Name, Address, Phone#)

Here, STUDENT is a relation defined over the four attributes RollNo, Name, Address, Phone#, each of which has a **domain** or a set of valid values.

For example, the domain of RollNo is 7 digit numbers.

- ✓ A **tuple** is an ordered set of values
- ✓ Each value is derived from an appropriate domain.
- ✓ Each row in the STUDENT table is referred to as a tuple in the table and would consist of four values.

- ✓ **<20207789 , “Tashi", “GCIT, Mongar", “17898874" >**

is a tuple belonging to the STUDENT relation.

- ✓ A relation can be defined as a ***set of tuples*** (rows).
- ✓ Columns in a table are also called attributes of the relation.
- ✓ A **domain** is a set of atomic values. Atomic means that each value in the domain is indivisible. examples:

"MobilePhone_numbers" : the set of 8 digits that represents phone numbers in Bmobile "CId_numbers" : the set of 11 digits that represent identity card numbers in Bhutan

"Names": the set of character string that represents a name of a person.

- ✓ A domain may have a data-type or a format defined for it.

  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.
- ✓ e.g: attribute Name is defined over the domain of strings of 25 characters.
- ✓ Formally,

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

$r(R) \quad \subset \quad \text{dom}(A_1) \text{ X dom}(A_2) \text{X} \ldots \text{X dom}(A_n)$

Where R: schema of the relation
r of R: a specific "value" or population of R.

- Let S1 = {0,1}
- Let S2 = {a,b,c}
- Let R $\subset$ S1 X S2

- Then $r(R) = \{<0,a>, <0,b>, <1,c>\}$ is one possible “state” or “population” 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

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

Tuples

Example contd…..

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

3. Relational Database/Model Constraints

- ✓ Constraints are *conditions* that must hold on *all* valid relation instances.
- ✓ Constraints are specified to ensure data integrity i.e consistency of data while manupulating the database.

Types of constraints:

1. **Domain** constraints
2. **Key** constraints
3. Not Null constraints
4. **Entity integrity** constraints
5. **Referential integrity** constraints

1. Domain & Not Null Constraints

- Domain Constraint:

- ➢ Specifies that within each tuple, the value of each attribute A must be an atomic value from the domain of A
- ➢ Data type specified for domain must be one of:

-integer, float, char, varchar, enum, boolean, date, time, timestamp

- ➢ Cannot specify more than one datatype for a domain

- Not Null Constraint:
  - ➢ Specifies whether null values are permitted or not on an attribute.
  - ➢ By default, null values are permitted

2. Key Constraints

- **Superkey** of R: A set of attributes SK of R such that no two tuples *in any valid relation state* will have the same value for SK. That is, for any distinct tuples t1 and t2, $t1[SK] \neq t2[SK]$
  - ➢ Every relation has atleast one default super key i.e the set of all its attributes

- **Key** of R: A "minimal" superkey; that is, a superkey K from which we can not remove any attribute and still have the uniqueness constraint on any two tuples of the relation.

Example: The CAR relation schema:

CAR(State, Reg#, SerialNo, Make, Model, Year)

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

- If a relation has *more than one key, each of the key is called* **candidate key.** One is chosen arbitrarily to be the **primary key** and *underlined it*.

Key Constraints

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

The Car Relation with two Candidate keys: LicenseNumber & EngineSerialNumber

3. Entity Integrity Constraint

- **Relational Database Schema**: A set S of relation schemas that belong to the same database. If 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. $t[PK] \neq$ null for any tuple t 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.

4. Referential Integrity Constraint

- 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

- Functional Dependency Constraints: used as tools

- to analyze the quality of relational designs
- to normalize relations to improve their quality

Figure 7.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS_ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

One Possible Relational Database State Corresponding to the COMPANY schema

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John		Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin		Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia		Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer		Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh		Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce		English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad		Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James		Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

DEPT_LOCATIONS	DNUMBER	DLOCATION
		Houston
		Stafford
		Bellaire
		Sugarland

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

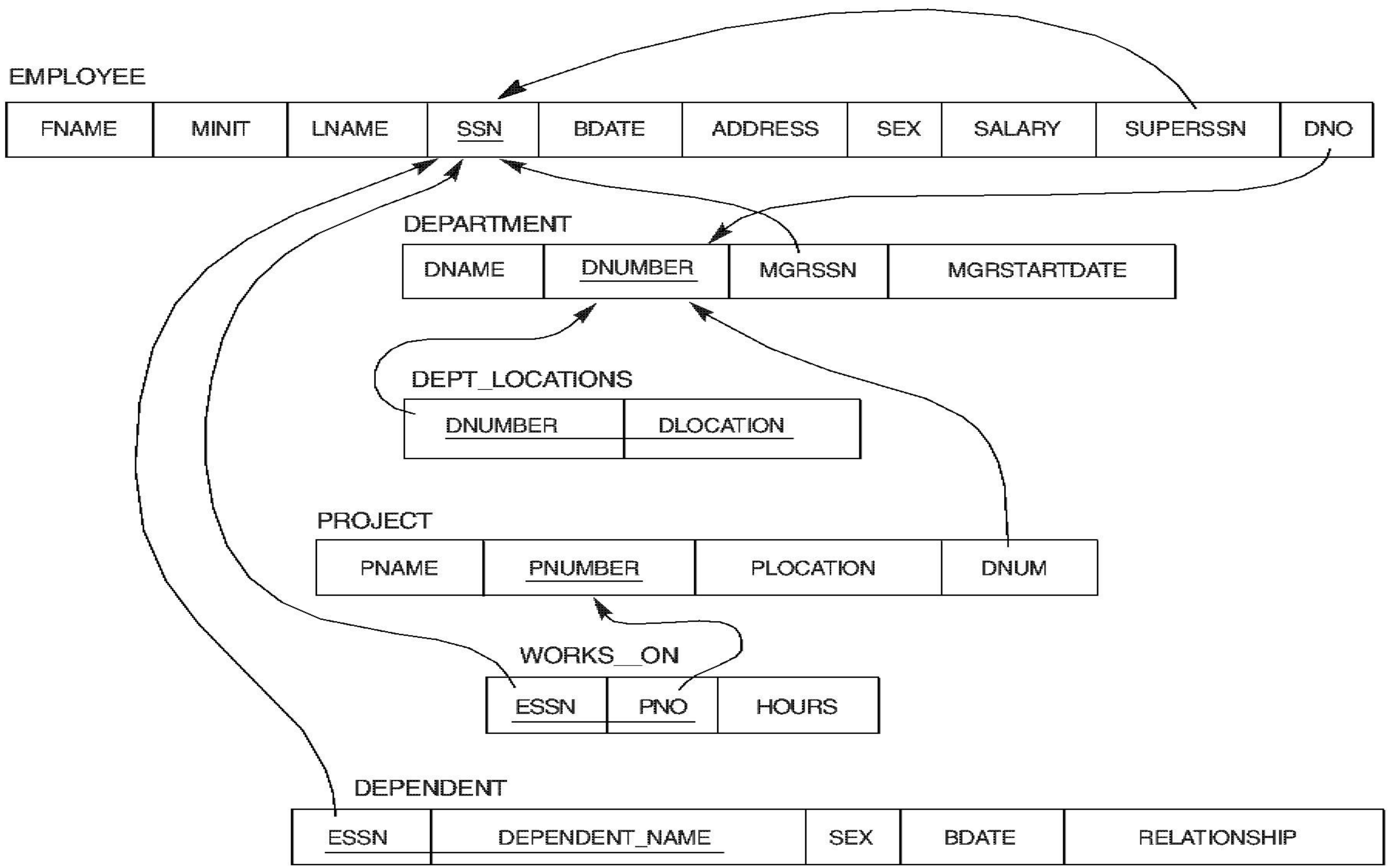
One Possible Relational Database State Condt………..

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 7.7 Referential integrity constraints displayed on the COMPANY relational database schema diagram.



Update Operations on Relations & Dealing with Constraint violations

- **Update Operations:**
  - INSERT a tuple; DELETE a tuple; MODIFY a tuple.
- **Constraint Violations:**
  - 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.
- 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(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.