UNIT 2

Lecture 1

RELATIONAL MODEL CONCEPTS

INTRODUCTION

- Relational model represents database as a **collection of data values**
- A **relation looks like a table of values.** 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 gives an indication** of the meaning of the data items in that column

INTRODUCTION

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

- **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
  - In the STUDENT relation, SSN is the key

EXAMPLE OF RELATION

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

DOMAIN

- A **domain is the set of all possible values that an attribute can take.**
- These values are **atomic** i.e. each value in the domain is indivisible
- A domain also has a **data-type or a format defined** for it.
- The attribute name designates the role played by a domain in a relation.

TUPLE

- A **tuple is an ordered set of values enclosed in angled** brackets '< … >'
- **Each value is derived from an appropriate *domain*.**
- A row in the CUSTOMER relation is a 4-tuple and would consist of four values

  <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)**

SCHEMA

- The Schema (or description) of a Relation

  R(A1, A2, ...An)
  R is the name of the relation
  The attributes of the relation are A1,..., An

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

- **Each attribute has a domain or a set of valid values.**

STATE – 1/2

- The **relation state is a subset of the Cartesian** product of the domains of its attributes

Attribute Cust-name is defined over the domain of character strings of maximum length 25

Hence, **dom(Cust-name)** is varchar(25)

STATE – 2/2

R(A1, A2, ..., An) → **Schema,** R is **relation name** and A1, A2, …, An are the **attributes** of the relation

r(R) ⊂ dom (A1) X dom (A2) X ....X dom(An)

r(R) → a specific **state of relation** R (*set of tuples*)

r(R) = {t1, t2, …, tn} where each t is an n-tuple

t = <v1, v2, …, vn> where each v *element-of dom(A)*

COMPARISON OF TERMS

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 - 1/2

- **Ordering of tuples in a relation**
  - 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)**
  - Consider the attributes in R(A1, A2, ...,An) and the values in t=<v1, v2, ..., vn> to be ordered

  - Tuple can be considered as a set of **(<attribute, value>) pairs**, where each pair gives the value of the mapping from attribute Ai to value vi from dom(Ai)

CHARACTERISTICS OF RELATIONS - 2/2

- **Values in a tuple:**
  - All values are considered **atomic (indivisible)**
  - 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.

RELATIONAL CONSTRAINTS

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

- There are four different types of constraints in the relational model
  - Domain constraints
  - Key constraints
  - Entity integrity constraints
  - Referential integrity constraints

DOMAIN CONSTRAINTS

- Specify value of **each value A must be atomic** value from **domain dom(A)**

- Data types associated with domains
  - Short integer, integer, long integer, float, double precision float
  - Characters, fixed length strings, variable length strings
  - Date, time, currency data types
  - Subrange of values from data type or as enumerated data type

KEY CONSTRAINTS – 1/7

- All elements of a set are distinct hence all **tuples in relation must also be distinct**

- **No two tuples have the same combination of values for all their attributes**

- There are other subset of attributes of relation schema R with the property that no two tuples in any relation state r of R should have the same combination of values for these attributes

KEY CONSTRAINTS – 2/7

- Denote one such subset as SK, then for any two **distinct tuples t1 and t2** in relation state r of R, we have

  t1 [SK] ≠ t2 [SK]

- Any **such set of attributes is called superkey** of R

- **SK specifies a uniqueness constraint that no two distinct tuples in state r of R can have same values for SK**

KEY CONSTRAINTS – 3/7

- **Every relation has one default superkey – set of all its attributes**
- Key K of R is superkey of R with additional property that removing any attribute A from K leaves a set of attributes K' that is not superkey of R
- **Key is a minimal superkey – superkey from which we cannot remove any attributes and still have uniqueness constraint hold**

KEY CONSTRAINTS – 4/7

- For STUDENT relation

  {SSN} is a key of student

  {SSN, Name} is a superkey

  However, {SSN, Name} is not key of student, since removing Name from set still leaves us with superkey

- **Relation schema may have more than one key**

- **Each key here is called candidate key**

KEY CONSTRAINTS – 5/7

- 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 – 6/7

- Eg. CAR relation

  LicenseNumber and EngineSerialNumber are two candidate keys

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

- **The primary key attributes are underlined.**

- **The primary key value is used to *uniquely identify* each tuple in a relation**

KEY CONSTRAINTS – 7/7

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 has two candidate keys :

LicenseNumber and EngineSerialNumber

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

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] ≠ null for any tuple t in r(R)**
  - If PK has several attributes, null is not allowed in any of these attributes.

REFERENTIAL INTEGRITY – 1/3

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

REFERENTIAL INTEGRITY – 2/3

- 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.

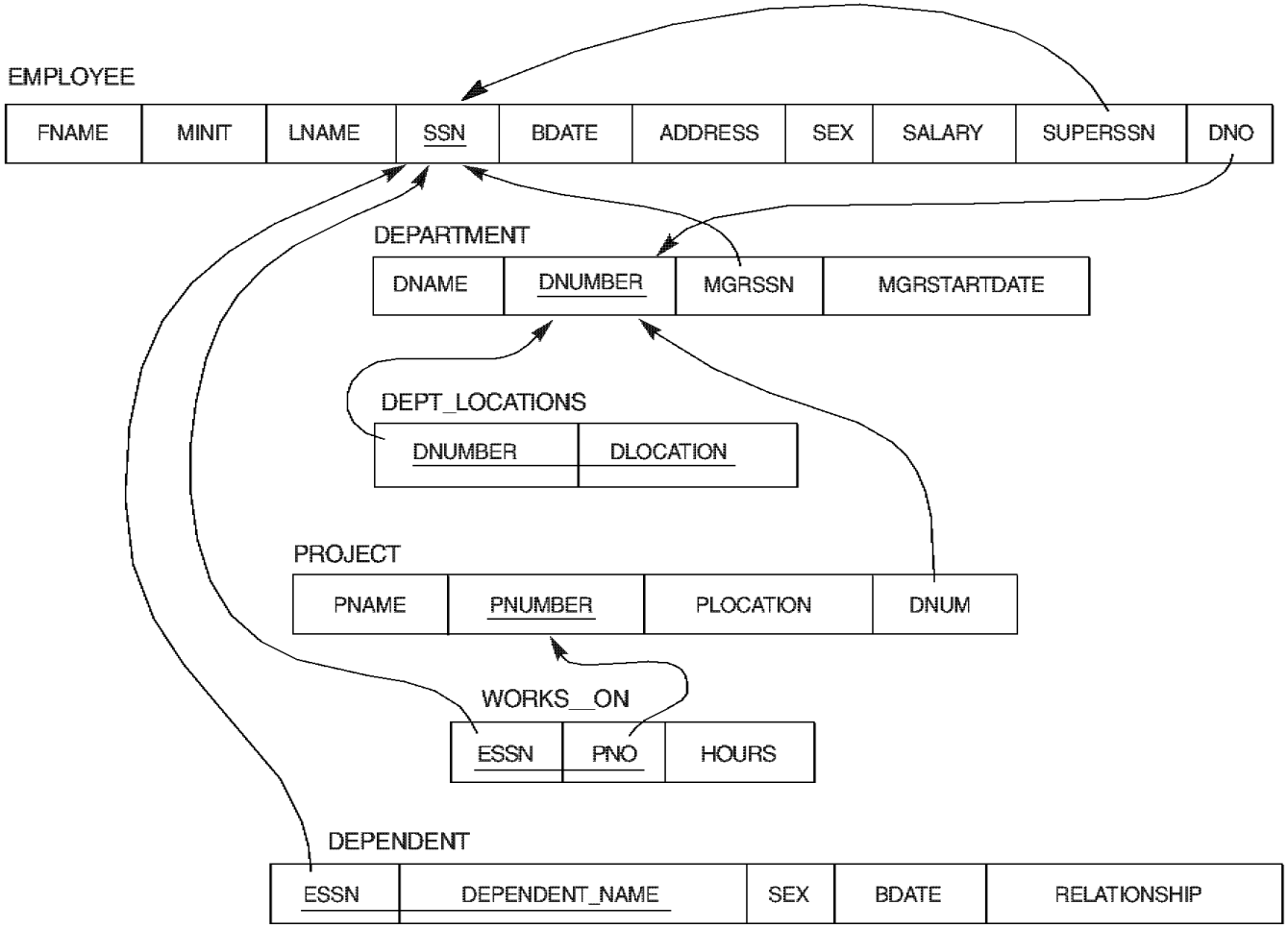
REFERENTIAL INTEGRITY – 3/3

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

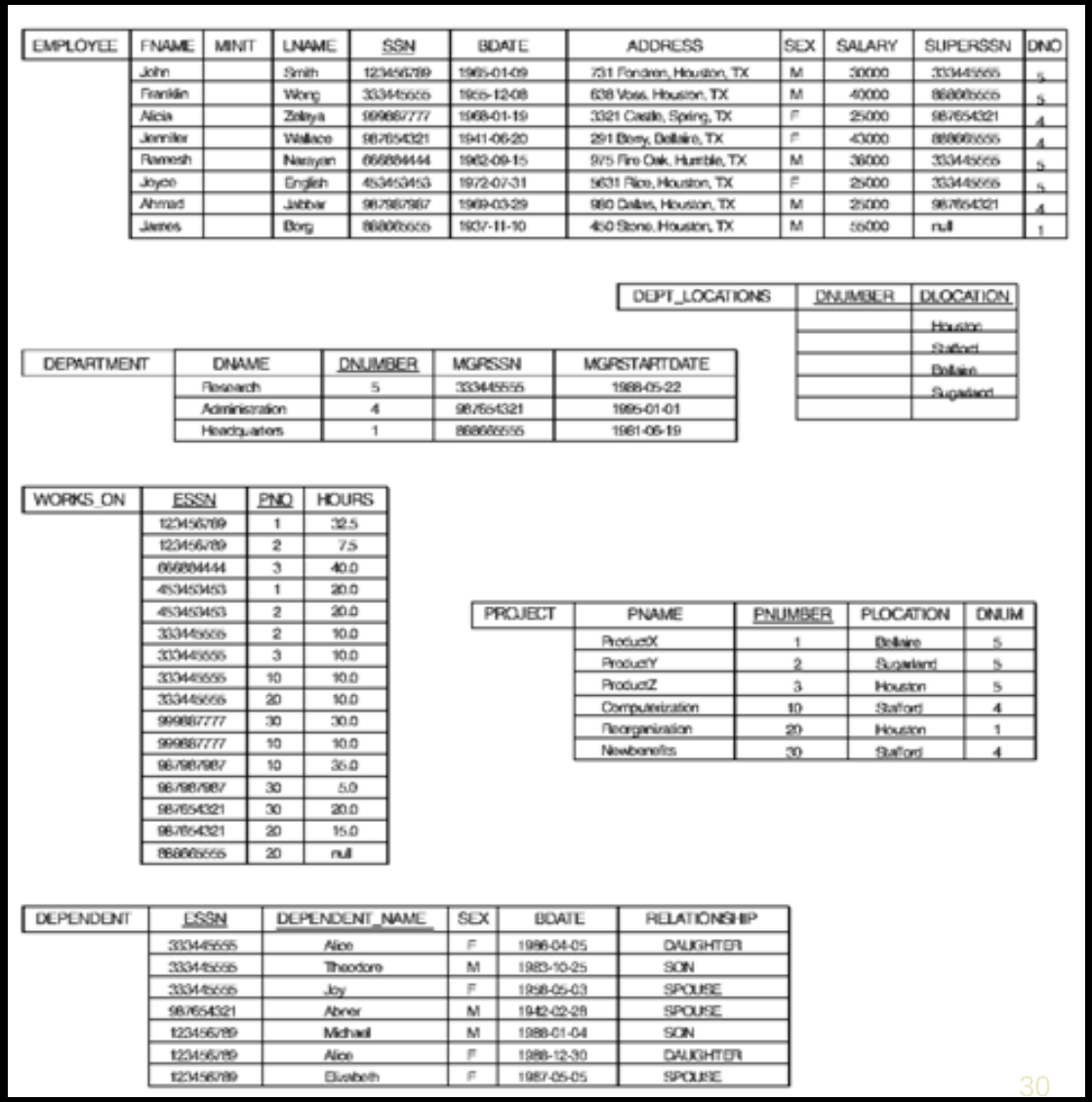
RELATIONAL SCHEMA REPRESENTATION

- 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 (or Can also point the primary key of the referenced relation for clarity)**

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



POSSIBLE DATABASE STATE FOR 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

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

UPDATE OPERATIONS ON A RELATION

- **INSERT** a new tuple in a relation
- **DELETE** an existing tuple from a relation
- **MODIFY** an attribute of an existing tuple

VIOLATIONS FOR INSERT OPERATION – 1/2

- INSERT may violate any of the constraints:
  - Domain constraint:
    - If one of the attribute values provided for the new tuple is not of the specified attribute domain
  - Key constraint:
    - if the value of a key attribute in the new tuple already exists in another tuple in the relation

VIOLATIONS FOR INSERT OPERATION – 2/2

- Referential integrity:
  - if a foreign key value in the new tuple references a primary key value that does not exist in the referenced relation
- Entity integrity:
  - if the primary key value is null in the new tuple

VIOLATIONS FOR DELETE OPERATION – 1/2

- DELETE may violate only referential integrity:
  - If the primary key value of the tuple being deleted is referenced from other tuples in the database
  - Can be remedied by several actions: RESTRICT, CASCADE, SET NULL
  - RESTRICT option: reject the deletion

VIOLATIONS FOR DELETE OPERATION – 2/2

- CASCADE option: propagate the new primary key value into the foreign keys of the referencing tuples
- SET NULL option: set the foreign keys of the referencing tuples to NULL
- One of the above options must be specified during database design for each foreign key constraint

VIOLATIONS FOR UPDATE OPERATION – 1/2

- UPDATE may violate domain constraint and NOT NULL constraint on an attribute being modified
- Any of the other constraints may also be violated, depending on the attribute being updated:
  - Updating the primary key (PK):
    - Similar to a DELETE followed by an INSERT
    - Need to specify similar options to DELETE

VIOLATIONS FOR UPDATE OPERATION – 2/2

- Updating a foreign key (FK):
  - May violate referential integrity
- Updating an ordinary attribute (neither PK nor FK):
  - Can only violate domain constraints