KOÇ UNIVERSITY

Database Management Systems

Relational Model

M. Emre Gürsoy

Assistant Professor
Department of Computer Engineering
www.memregursoy.com

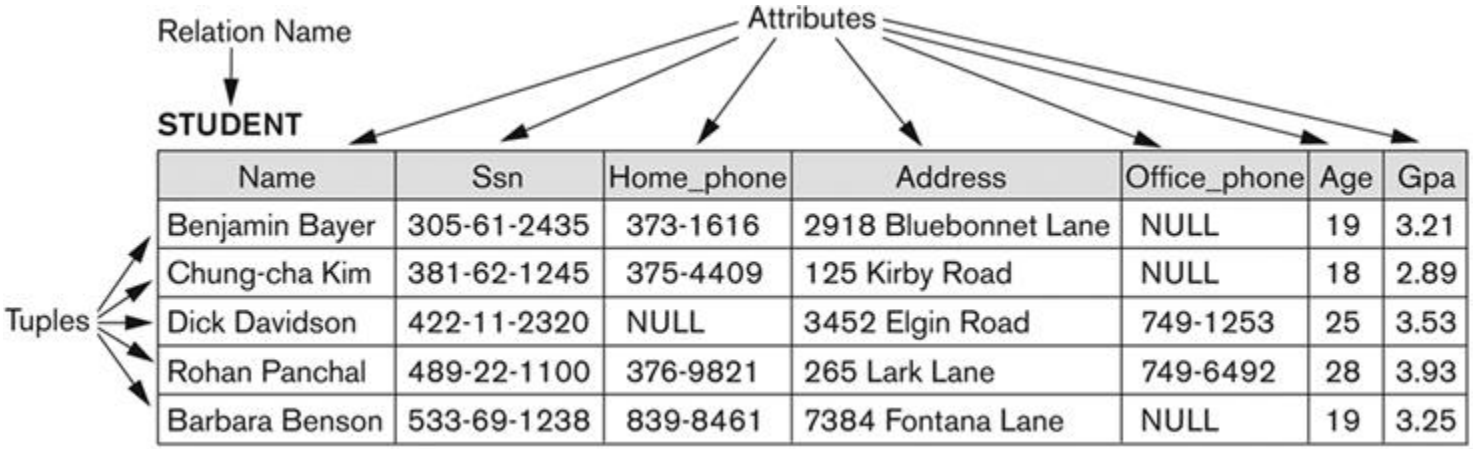
Introduction

- The **relational model** was first proposed by Dr. E. F. Codd from IBM Research in 1970:
  - "A relational model of data for large shared data banks", Communications of the ACM, June 1970
  - Revolutionary paper in the field of DBMS
  - Dr. Codd won the ACM Turing Award

- The **relational model** is based on the concept of a relation
  - Informally, a relation is a table with many rows
  - "Header" at the top (schema of the relation)
  - Rows are data entries (tuples in the relation)

A Relation

- The schema of a relation specifies how it is defined
  - Name of relation, name of attributes, attribute domains
  - Contents (tuples) are not included in the schema
- We typically write: R(A1, A2, ..., An)
  - Relation name R, attributes A1, A2, ..., An
  - Example: STUDENT(Name, Ssn, Home_phone, Address, Office_phone, Age, Gpa)



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

Formal Definitions

- Formally, given R(A1, A2, ..., An)
  - R is the **name** of the relation
  - A1, A2, …, An are the **attributes** of the relation
  - R(A1, A2, …, An) is the **schema** of the relation
  - Each attribute has a "**domain**" of allowed values: dom(Aj)

- r(R): a specific **state** of relation R – this is a *set of tuples* (rows) that currently exist in the relation
  - r(R) = {t1, t2, …} where each ti is a tuple
  - ti = <v1, v2, …, vn> where each vj ∈ dom(Aj)

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) $\subseteq$ dom(A1) X dom(A2)
- For example: r(R) could be {<0,a> , <0,b> , <1,c> }
  - This is one possible state (or "population") r of relation R
  - It has three tuples: <0,a> , <0,b> , <1,c>

Tuples

- We use notation t[Aj] to mean the value tuple t has for attribute Aj.
- A special NULL value is used to represent values that are unknown/unavailable in the database.
  - Below, for some tuples, t[Office_phone] = NULL.
- Tuples in a relation are not ordered (why?)
  - I can shuffle the table arbitrarily, r(R) stays the same
- All tuples in a relation are unique (why?)

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

Formal vs Informal

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

Relational Database

- Collection of several relations (schema+state)
- The schema of Company DB:

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.

Key + Superkey

- **Superkey** SK of relation R is a set of attributes such that:
  - Two tuples cannot have the same value for all SK attributes, i.e., for distinct t1, t2: $t1[SK] \neq t2[SK]$
  - This condition must hold for all valid states of R
- **Key** K of relation R:
  - A "minimal" superkey, i.e., no redundant attributes
  - If you remove any attribute from K, uniqueness no longer holds (every attribute is necessary)

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

Key + Superkey

- True/False?
  - A key is always a superkey.
  - A superkey is always a key.
  - Any set of attributes that includes a key is a superkey.

- Can there be multiple keys?
  - CAR(State, Reg#, SerialNo, Make, Model, Year)
  - CAR has two keys:
    - Key1 = {State, Reg#}
    - Key2 = {SerialNo}
  - Both are also superkeys
  - {SerialNo, Make} is a superkey but not a key

Primary Key

- If a relation has several candidate keys (eg: CAR), one of them is chosen (by the administrator) as the **primary key**.
- We underline the primary key in the schema.
  - CAR(State, Reg#, SerialNo, Make, Model, Year)
  - STUDENT(Name, SSN, Home_phone, Address, Office_phone, Age, GPA)
- The primary key is used to uniquely identify each tuple in a relation (it becomes the "identity" of the tuple).
- Recommendation (but not a rule) for choosing the primary key: Choose the smallest of the candidate keys.
  - Key1 = {State, Reg#} has 2 attributes
  - Key2 = {SerialNo} has 1 attribute

Foreign Key

- In a DB with many relations, attributes in some relations may reference attributes in other relations.
- Tuples in the referencing relation (eg: ENROLLMENT) have attributes FK called **foreign keys** that reference the primary key attributes PK of the referenced relation (eg: STUDENTS).
  - sid is a primary key in STUDENTS; it is a foreign key in ENROLLMENT

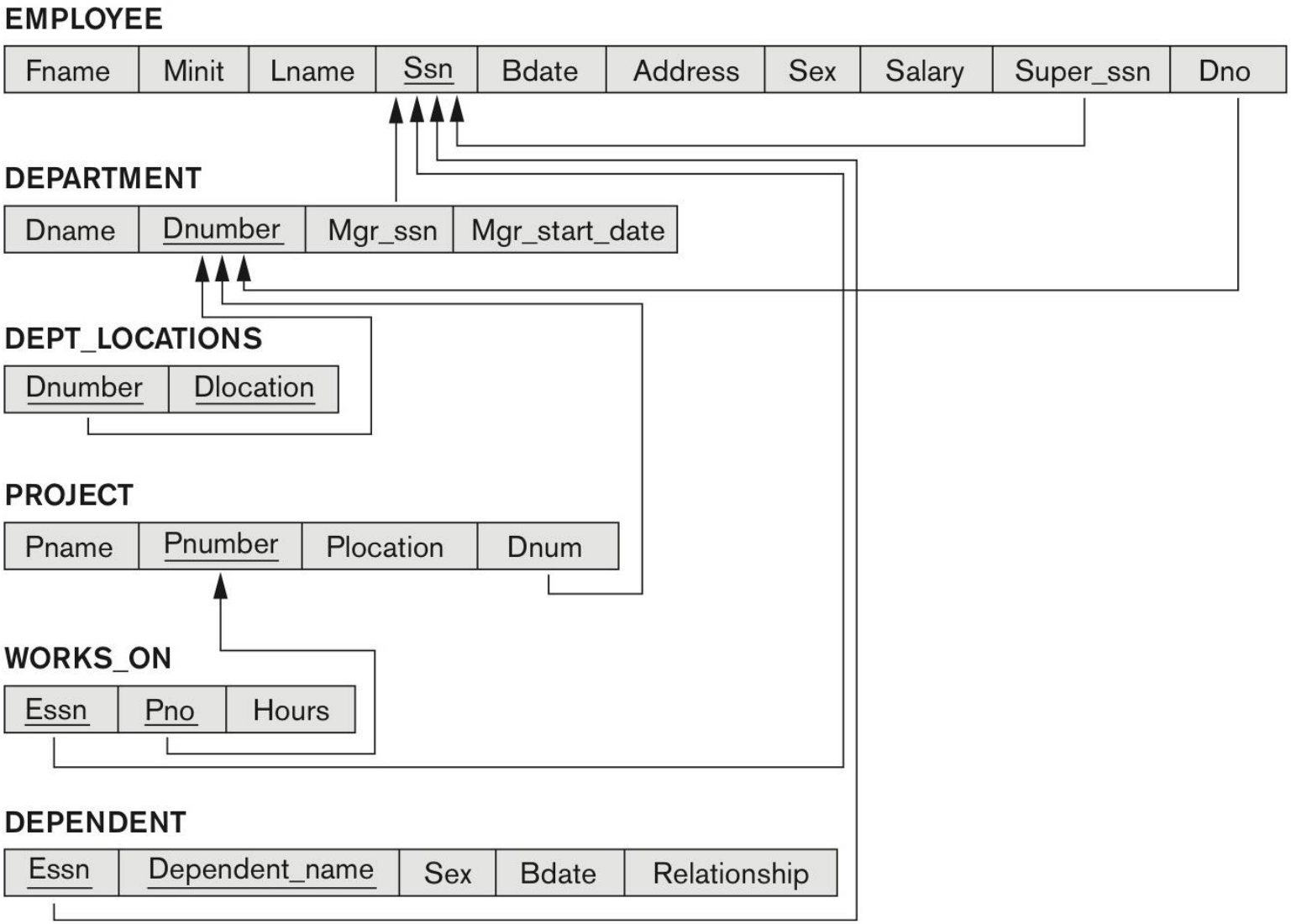
ENROLLMENT

sid	cid	grade
53688	Topology112	A
53666	History105	B

STUDENTS

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2

Examples of PK + FK



Company DB Example

Figure 5.6

One possible database state for the COMPANY relational database schema.

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

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

Creating Relations

```
CREATE TABLE Students
        (sid: CHAR(20),
         name: CHAR(20),
         login: CHAR(10),
         age: INTEGER,
         gpa: REAL,
         PRIMARY KEY (sid))
```

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Shero	shero@cs	18	3.2

```
CREATE TABLE Enrollment
        (sid: CHAR(20),
        cid: CHAR(30),
        grade: CHAR(2),
        PRIMARY KEY (sid, cid),
        FOREIGN KEY (sid)
            REFERENCES Students)
```

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Integrity Constraints

- 4 types of integrity constraints in the relational model:
  - **Domain constraint:** Every value of a tuple must be from the domain of its corresponding attribute (or NULL, if allowed).
  - **Key constraint:** Every tuple must be unique in terms of its key attributes.
  - **Entity integrity constraint:** Tuples cannot have NULL values for their primary key attributes.
  - **Referential integrity constraint:** There should be no dangling references between relations.
    - More detail on the next slides!

Referential Integrity

- What should be done if someone wants to...
  - Insert a tuple into Enrollment with a non-existent sid?
    - Reject it!
  - Delete a tuple from Students?
    - Also delete all Enrollment tuples that refer to that student tuple?
    - Disallow deletion of this tuple?
  - Update sid of a student in Students?
    - Disallow?
    - Modify that sid in Enrollment?

```
CREATE TABLE Enrollment
        (sid: CHAR(20),
        cid: CHAR(30),
        grade: CHAR(2),
        PRIMARY KEY (sid, cid),
        FOREIGN KEY (sid)
            REFERENCES Students)
```

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Shero	shero@cs	18	3.2

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Referential Integrity

- All of these are examples of how referential integrity (RI) could potentially be violated.
- Example ways in which RI violations can be handled:

- NO ACTION: reject deletion/update
- CASCADE: also delete/update all tuples that refer to the deleted/updated tuple
- SET DEFAULT: set foreign key value to a pre-determined, special DEFAULT value

```
CREATE TABLE Enrollment
        (sid: CHAR(20),
        cid: CHAR(30),
        grade: CHAR(2),
        PRIMARY KEY (sid, cid),
        FOREIGN KEY (sid)
            REFERENCES Students
                ON DELETE CASCADE
                ON UPDATE NO ACTION)
```

Exercise

- Consider a relational database with many relations
- Consider 3 main kinds of operations that arise in practice:
  - INSERT a tuple into a relation
  - DELETE a tuple from a relation
  - MODIFY/UPDATE a tuple in a relation

- For each of these operations, which integrity constraints can they possibly violate? Examples?