



BITS Pilani
Hyderabad Campus

Database Design & Applications (SS ZG 518)

Dr.R.Gururaj
CS&IS Dept.

Relational Data Model & Relational Constraints (Ch.5)



Content

1. What is Relational model
2. Characteristics
3. Relational constrains
4. Representation of schemas
5. Operations on relations and constraints

Relational Model



Edgar Codd proposed Relational Data Model in 1970.
It is a *representational* or *implementation* data model.

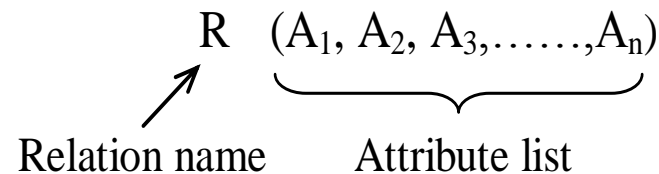
Using this representational (or implementation) model we represent a database as *collection of relations*.

The notion of *relation* here is different from the notion of *relationship* used in ER modeling.

Relation is the main construct for representing data in relational model.

Every relation consists of a relation schema and Relation instance.

Relation Schema is denoted by



The number of columns in a relation is known as its *degree* or *arity*.

Relation instance or Relation State (r) of R (thought of as a table)

Each *row* in the table represents a collection of related data.

Each row contains facts about some entity of same entity-set.

$$R = (A_1, A_2, A_3, \dots, A_n)$$

$r(R)$ is a set of n tuples in R

$$r = \{t_1, t_2, t_3, \dots, t_n\}$$

r is an instance of R each t is a tuple and is a ordered list of values.

$t = (v_1, v_2, \dots, v_n)$ where v_i is an element of domain of A_i

Entities of each type/set are stored as rows in a single relation.

Hence in general, a *relation* corresponds to a single entity type in ER diagram.

In some cases a relationship between two entities can have some specific attributes which can be captured in a relation (table).

A row is called a *tuple*.

The columns of the table represent attributes of that entity type.

The column header is known as *attribute* or *field*.

Data type or *format* of an attribute: is the format of data for that attribute. Ex. Character strings, numeric, alphanumeric etc.

Values that can appear in a column for any row is called the *domain* of that attribute.

Relational Database Schema is denoted by

$$\begin{array}{ccc} & \nearrow & \\ \text{Database} & & S = \{R_1, R_2, \dots, R_n\} \\ \text{name} & & \underbrace{\hspace{10em}} \\ & & \text{Relations in the} \\ & & \text{database (tables)} \end{array}$$

Attribute A of relation R is accessed by notation- $R.A$.

Ex: *Student (name, age, branch)*. Here *Student* is the relation name.
Student.age - denotes *age* attribute of *Student* relation.

Characteristics of a Relation:

- ❖ Ordering of tuples is not significant.
- ❖ Ordering of values in a tuple is important.
- ❖ Values in a tuple under each column must be atomic (simple & single).

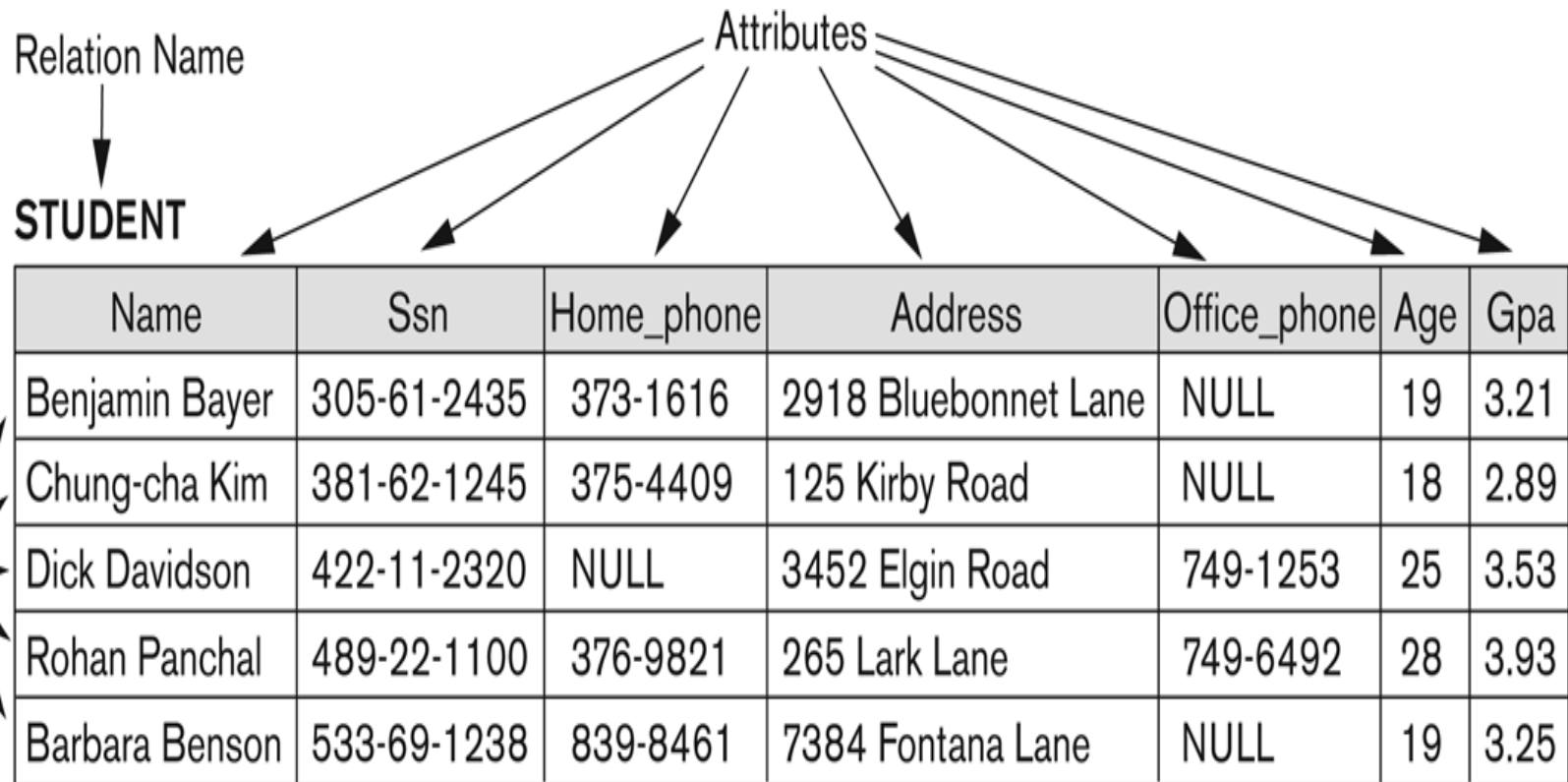


Figure 5.1

The attributes and tuples of a relation STUDENT.

Relational Model Terminology



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

Relational Constraints

Constraints are restrictions on data of a relation.

❑ *Domain level Constrains* – Format of data Ex.
Character numeric etc.

Semantic – Not

NULL etc.

❑ *Entity integrity constraints* – Primary key,
unique key

❑ *Referential integrity constraints*– Foreign key

Relational Constraints



□ *Dependencies* –

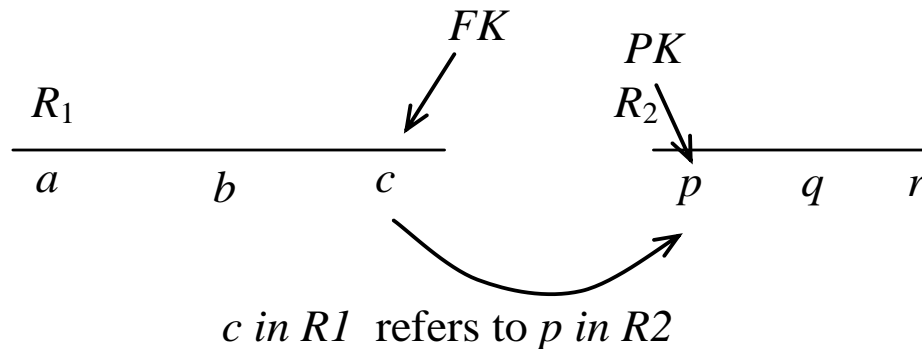
Functional dependency : What attributes value defines the value of another attribute is known as dependency.

This concept is used in database design.

Referential integrity



The **Referential Integrity** constraint is specified between two relations and is used to maintain consistency among the tuples of two relations.



The FK attribute R_1 has the same domain as the primary key attribute of R_2 .

The attribute c in R_1 is said to reference the attribute p in R_2 .

Referential integrity

The value of FK in a tuple t of R_1 either occurs as a value under p in R_2 for some tuple, or is a NULL.

$R_1 \rightarrow$ is known as referencing relation

$R_2 \rightarrow$ is known as referenced relation

Constraints can be specified while defining the structure & also as triggers.

Relational Schema Representation



EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

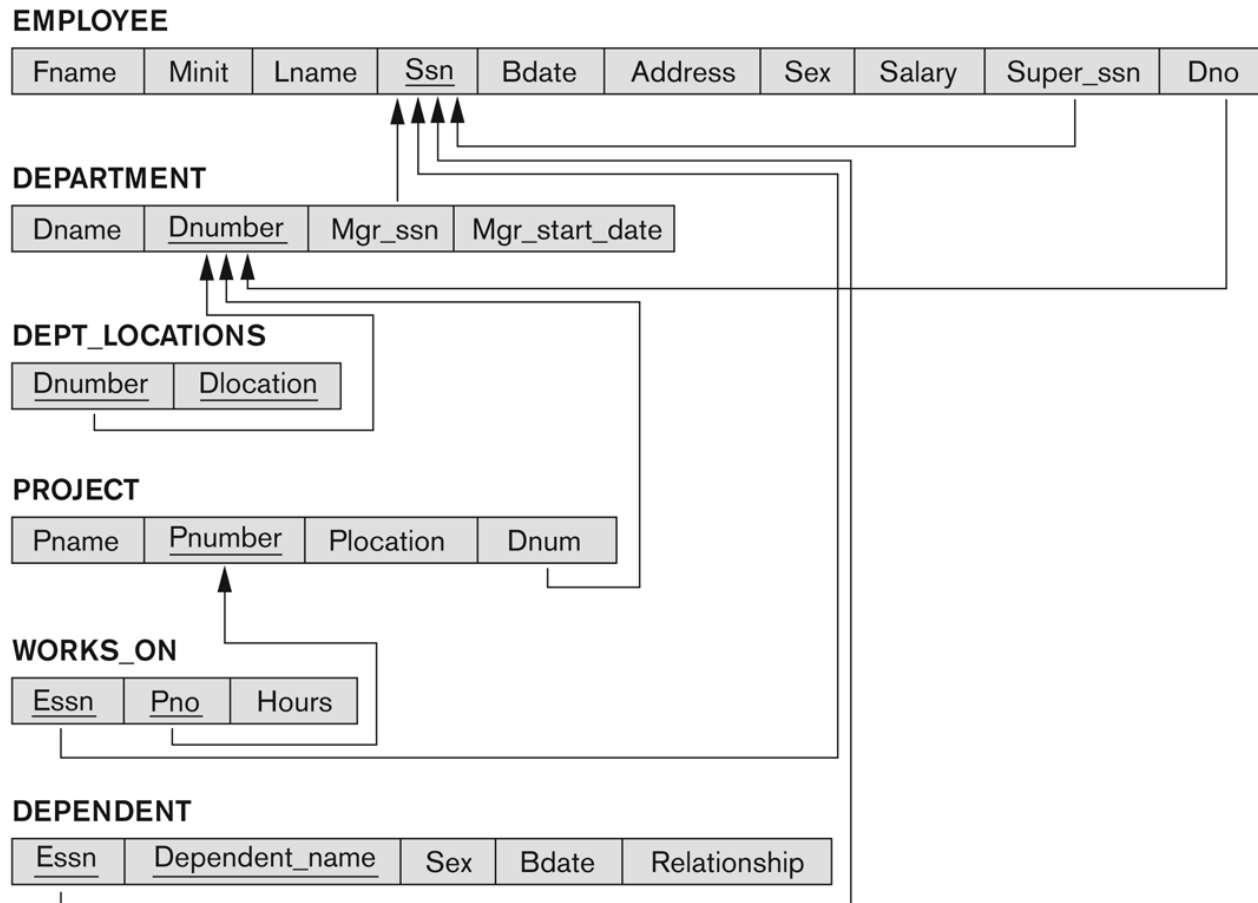
Figure 5.5
Schema diagram for
the COMPANY
relational database
schema.

Relational Schema Representation

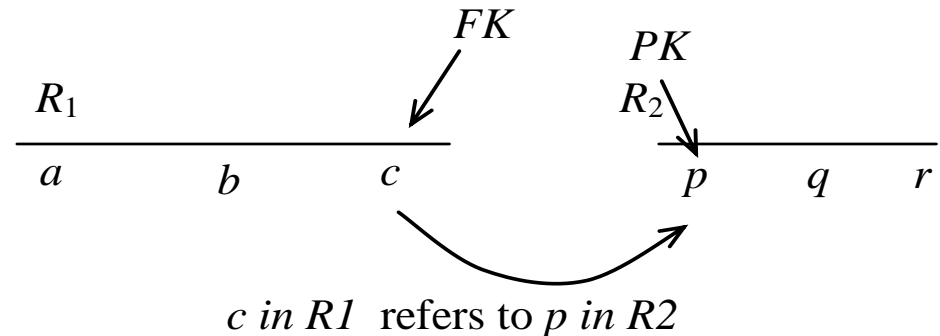


Figure 5.7

Referential integrity constraints displayed on the COMPANY relational database schema.



Actions need to be taken when FK is set , on operations like update, insert, and delete.



If insert a tuple in R_1 where the value for c is not in p of R_2 , then don't allow.

What if a tuple in R_2 is deleted:

Cascade, don't allow, set to default, set to null.

What if update on R_2 's p happens:

Cascade, don't allow, set to default, set to null.



Operations on Relations and constraints



The following table indicates the constraints need to be taken care while performing certain operations on a relation.

Operation on relations	Constraints need to be taken care
Insert	Null, Not Null, PK, unique, FK, format, Domain
Delete	FK
Update	Null, Not Null, PK, unique, FK, domain, and Semantic

Operations on Relations and constraints



Emp

FK

Dept

eid	ename	age	dno
101	kiran	28	10
105	Gopi	32	20
121	Ahmed	29	20
136	John	28	20
138	Mike	30	30
143	Ali	38	10

dnum	dname	dloc
10	Mark	Che
20	Prod	Hyd
30	Acct	Bgl

Cascade
Set to null
Set to default
Restrict/Reject

Operation	Care for constraints
Insert	PK, Unique, null, notnull, format, FK
Update	PK, Unique, null, notnull, format, FK
Delete	No care

Operation	Care for constraints
Insert	PK, Unique, null, notnull, format
Update	PK, Unique, null, notnull, format, FK
Delete	FK

Summary

- ✓ *What are basics of relational model*
- ✓ *Relation instance*
- ✓ *Relational data constraints*
- ✓ *Referential integrity*
- ✓ *Relational scheme representation*
- ✓ *Operations on relations and constraints*