DATABASE MANAGEMENT SYSTEM

(DBMS)

CONTENT

- **Relational Data Model**
- Object-Relational Data Models
- Database Design

Relational Data model →

→ flexible than the hierarchical & N/w

→ RDBMS is based on relational model developed by EF Codd. A Relational ~~model developed~~ DB represents all Data in the DB & as simple as two Dimensional table called Relations that is just equivalent to files

→ 3 operations are used to Develop useful set of Data → ① Selection
② Projection
③ join

① selection → It Retrieves certain Records from table based on the user specified criteria

② Projection → It extracts fields from a table.
→ permitting the user to Create new table that contain only the required Information.

③ join → It combines the Data from the two table based on a common Column, providing the user with more Information that is available in individual tables

3 Major Component of Relational model :->

① Data structure -> The set of Relations & set of Domains that Define how the Data can be presented.

② Data Integrity -> Integrity rules that Define the procedure to protect Data.

③ Data Manipulation -> operation that can be performed on Data.

Characteristics => ① Data in orderly manner in terms of rows & column called Relation.

② Value are scalar -> one value at row/column.

③ All operations are performed on an entire Relation & result is an entire Relation, a concept known as closure.

Terminologies =>

① Tuple -> Each Row of Data is a tuple.

② Cardinality -> Number of tuple.

③ Degree -> Number of Attribute.
↳ Each column in tuple.

④ Domains -> Set of all values that attribute contain.
It is Different from Data type
Data type is physical while Domain logical.

Number → Datatype.

Age → Domain.

Advantage =>

① Simple → Designers are free from Data Storage & Concentrate on logical view.

② Structural Independence → Changes in DB structure Don't affect Data Access.

③ Better query Capability → It uses 4GL like SQL that makes adhoc queries a Reality

NOTION OF RELATION

A table is said to be a relation, if it satisfies following properties: -

- It is column homogeneous.

  All items in a column are of the same kind.

- Each column is atomic.

  Each item is an integer or a character string.

NOTION OF RELATION

- All rows are distinct.

  No two rows may be identical in every column.

- The ordering of rows is immaterial(Not Important).

- The ordering of columns is immaterial and they are assigned distinct names.

NOTE: the first and third properties holds normally for any table. The rest are specific to the relational model.

S#	P#	Sc
10	1	Delhi
10	2	Delhi
11	1	Mumbai
11	2	Mumbai

A valid relation

S#	P#	City
11	1	Delhi
11	1	Delhi

Invalid relation

Two rows are not distinct.

Name	Child
Robert	Johnny,12-04-1985

Invalid relation

Child field is not atomic.

Customer – name	Security-number	Address	City
Williams	321-12-3123	Downhill	Banglore
Rama	321-12-3122	Downhill	Banglore, Hyderabad
Jaya	321-14-4562	Model Town	Delhi
Jones	321-12-3123	MG Road	Madras
Smith	321-14-9012	Main town	Calcutta
Jaya	321-14-4562	Model Town	Delhi

Identify whether the given relation is valid or invalid. Justify reasons in support.

A Relation

- Domain is the set of values over which the relation is constructed

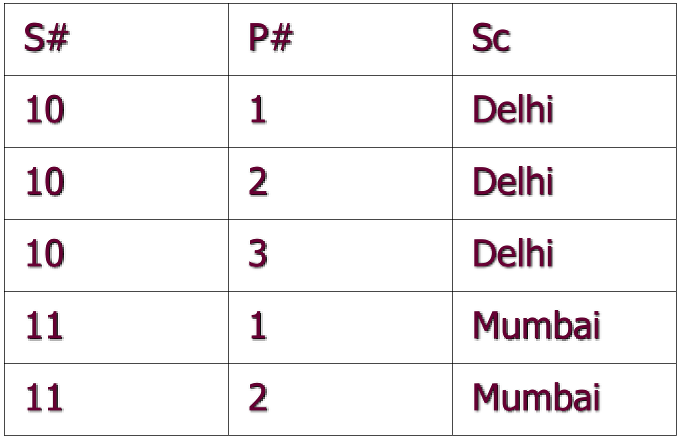
  integer and character strings

- Given n-domains **(D_1 , D_2 , ….., D_n)** , relation **R** is constructed as

$$\mathbf{R(D_1, D_2, \ldots., D_n) \subseteq X\,(D_1, D_2, \ldots\ldots, D_n)}$$

- Degree of relation R is **n** or it is a **n-ary** since it is defined over n domains (D_1 , D_2 , ….., D_n)

- A ternary relation :



S#	P#	Sc
10	1	Delhi
10	2	Delhi
10	3	Delhi
11	1	Mumbai
11	2	Mumbai

Relation Definition and Relation

- Definition of relation gives a name to the relation and specifies the attributes over which it is built.

Relation Definition

Customer(Customer-name, Date-of-birth, Address)

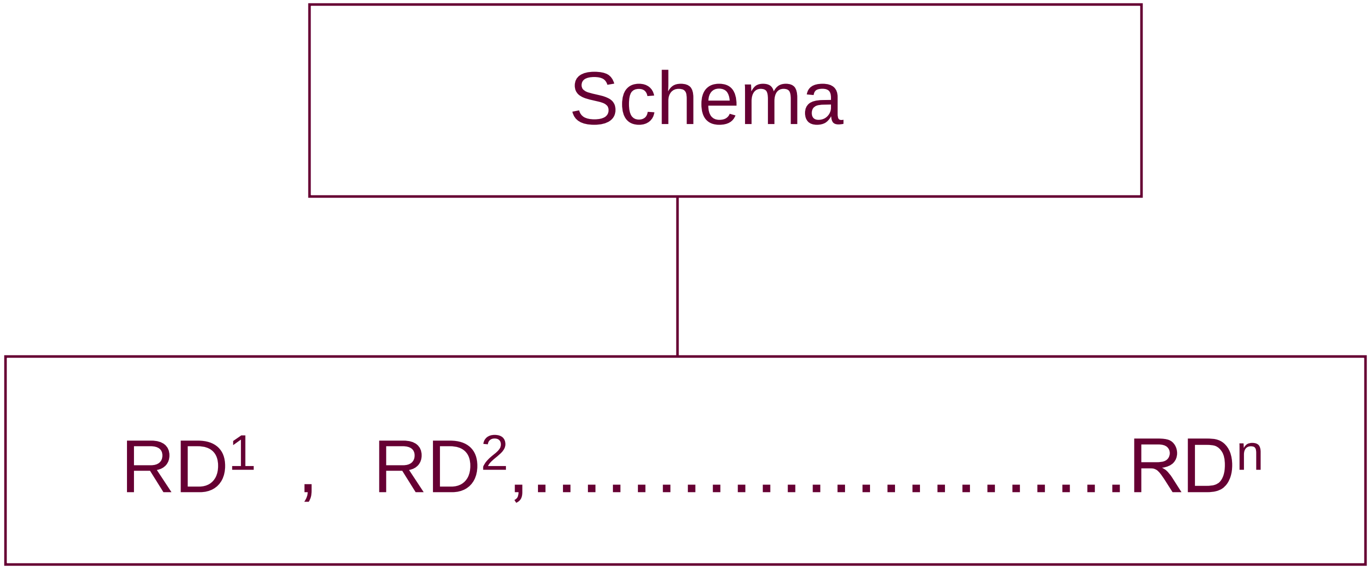
- Relation is a set of tuples which constitutes it at a given instant of time

Customer-name	Date-of-Birth	Address
john	12-04-78	Delhi
Harry	22-02-78	Goa

Relation may change with time while its definition remains same.

Relational Schema

A relational schema is a collection of relation definitions



Schema

RD^1 , RD^2,.......................RD^n

Relational Schema does not change over time.

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.

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 though 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.

FORMAL DEFINITIONS

Tuple-

- 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.

FORMAL DEFINITIONS

Domain-

- 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.

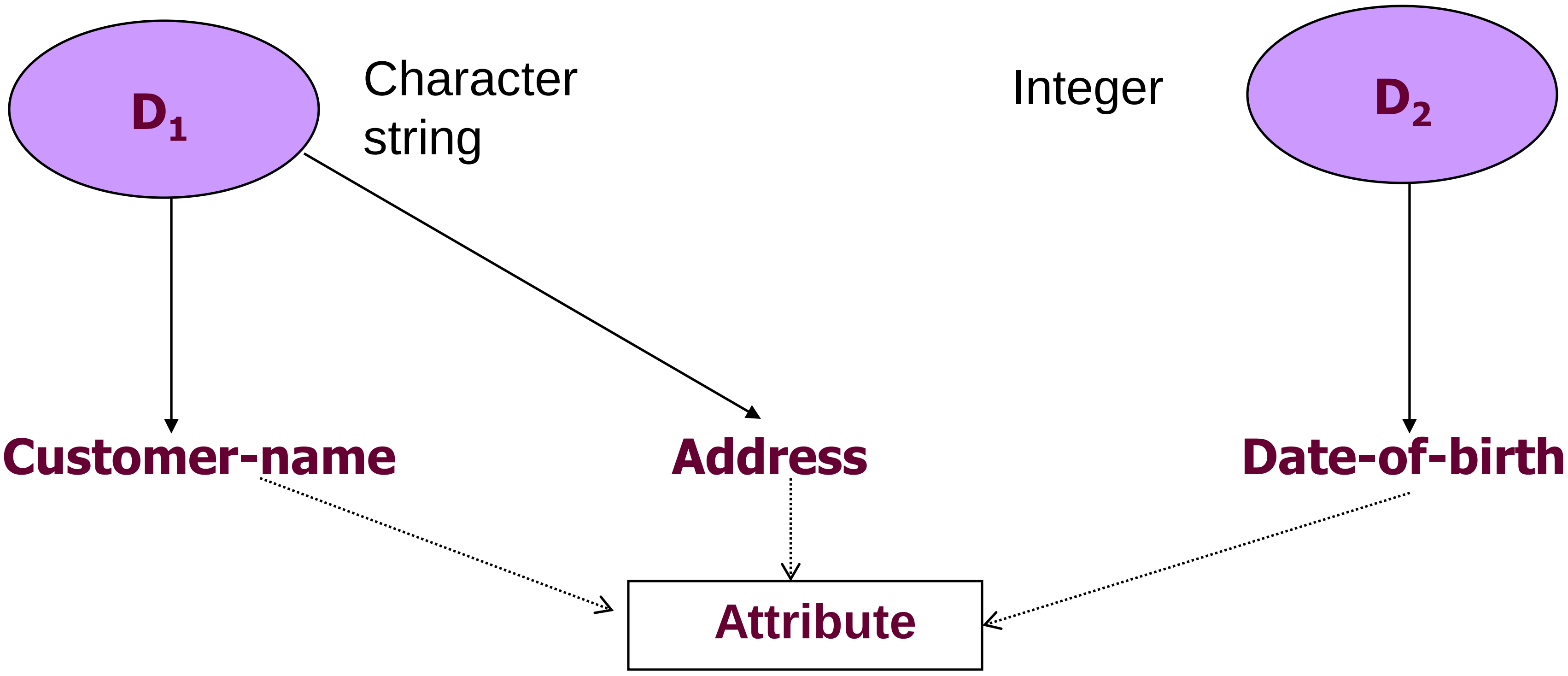
  E.g., Dates have various formats such as monthname, date, year or yyyy-mm-dd, or dd mm,yyyy etc.
- An attribute designates the **role** played by the domain.

  E.g., the domain Date may be used to define attributes “Invoice-date” and “Payment-date”.

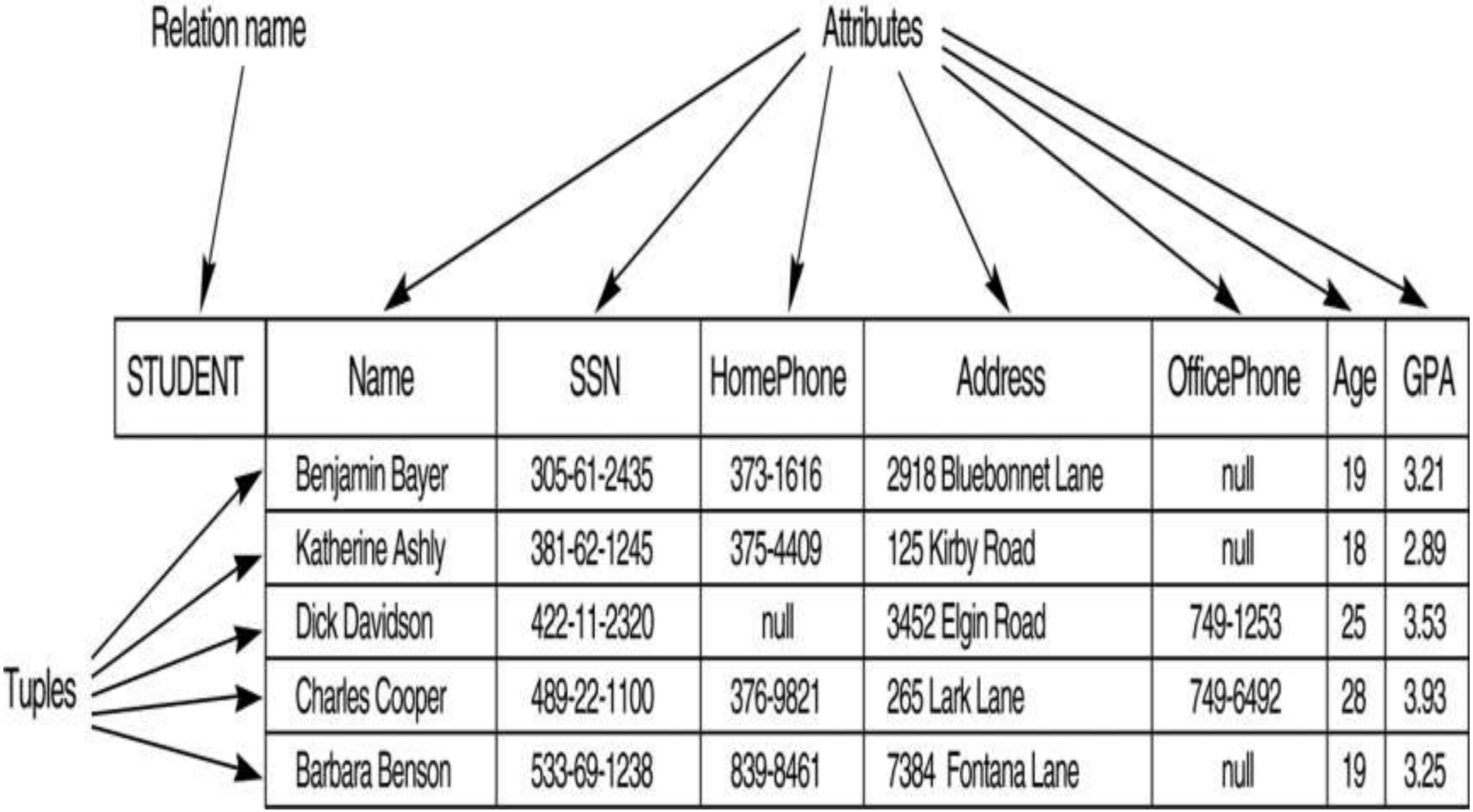
Domains and Attributes

Domain - The set of values on which an attribute is defined

- Domain is concerned with data of type **integer** or **character strings**
- Attribute is the meaning behind the domain



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

FORMAL DEFINITIONS

- 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, ..., A_n)

  $r(R) \subset dom(A_1) \times dom(A_2) \times \ldots \times dom(A_n)$
- 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

FORMAL DEFINITIONS

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

Relational Model Constraints

- The state of whole database will correspond to state of all its relation at a particular point in time. There are many constraints on actual values in a database state.

They are:-

- Inherent Model Constraint
- Explicit Or Schema based constraint
- Application based constraint

Integrity Constraints

Ensures data consistency during modification of database

- **Domain: a homogeneous set of values**
- **Key**
- **Entity Integrity**

On single relations only

Across relations

- **Referential Integrity**

Object-Relational Data Models

- Relational model: flat, “atomic” values
- Object Relational Data Models
  - Extend the relational data model by including object orientation and constructs to deal with added data types.
  - Allow attributes of tuples to have complex types, including non-atomic values such as nested relations.
  - Preserve relational foundations, in particular the declarative access to data, while extending modeling power.
  - Provide upward compatibility with existing relational languages.

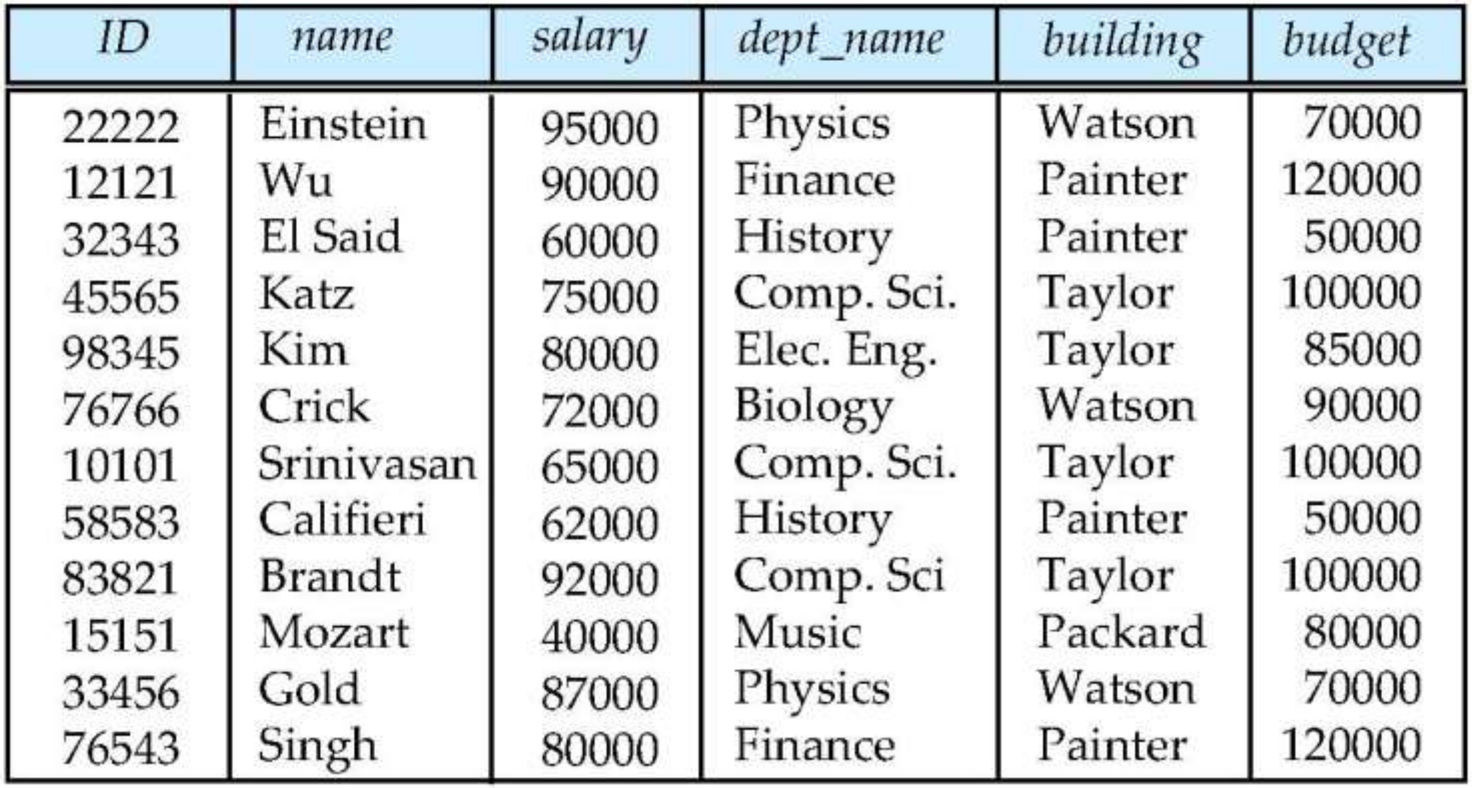
Database Design

The process of designing the general structure of the database:

- Logical Design – Deciding on the database schema. Database design requires that we find a “good” collection of relation schemas.
  - Business decision – What attributes should we record in the database?
  - Computer Science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
- Physical Design – Deciding on the physical layout of the database

Database Design (Cont.)

- Is there any problem with this relation?



ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Design Approaches

- Need to come up with a methodology to ensure that each of the relations in the database is "good"
- Two ways of doing so:
  - Entity Relationship Model
    - Models an enterprise as a collection of *entities* and *relationships*
    - Represented diagrammatically by an *entity-relationship diagram:*
  - Normalization Theory
    - Formalize what designs are bad, and test for them