Database Management Systems

Vijaya Saradhi

IIT Guwahati

Thu, 16th Jan 2020

Data Model

Definition

Collection of high level data description constructs that hide many low-level storage details

Data Model

Categories

- Conceptual data models: provide concepts that are close to the way users perceive data
- Low-level/Physical data models: details how data is stored on computer storage media

Conceptual Data Model

About

- They use concepts such as entities, attributes and relationships
- An entity represents a real-world object or a concept. That has physical existence or conceptual thing
- An attribute represents some property that describes an entity
- A relationship among two or more entities represents an association among entities
- A popular high-level conceptual data model is Entity-Relationship model

Physical data models

About

- Describe how data is stored as files
- Specifies record formats, record ordering, access paths
- access path is a structure that makes the search for a record(s) efficient
- An index is an example of an access path

Entities and Their Attributes

Definition

An entity is a thing in the real world with an independent existence

Entities and Their Attributes

Definition

An entity is a thing in the real world with an independent existence

Examples

- An entity may be an object with physical existence
- Example: Person, car, employee, student etc.
- It may be an object with conceptual existence
- Example: a job, course etc.

Attributes

Definition

Each entity has attributes that describe it.

Attributes

Definition

Each entity has attributes that describe it.

Explanation

- Example: Employee's description:
  - Has name
  - Has address
  - Has age
  - Has home phone number

Attribute - value

Values

- Each attribute takes some value
- The attribute-value that describe each entity becomes major part of data stored in the database
- Entity Employee has attributes name, address, age, home_phone
- Their values for a particular entity are ('smith', '2311 BSBE, IIT Guwahati', 27, '0361 258 0001')

Types of Attributes

Various Types

- Simple attributes
- Composite attributes
- Single valued
- Multivalued
- Stored
- Derived
- Complex

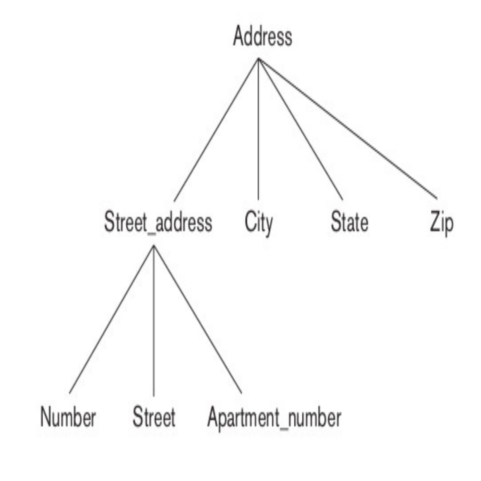
Simple vs Composite

Comparison

- Composite attributes can be divided into smaller subparts
- Each subpart represent more basic attributes
- Each of them have independent meaning
- They form hierarchy of attributes
- Example: Address: line 1, line 2, State, Pin
- Simple (atomic) attributes are not divisible further

Composite - Hierarchy

Hierarchy



Single-Valued vs Multivalued

Comparison

- Most attributes have single value for a particular entity
- Such attributes are called single-valued
- Example: Age
- When attribute have multiple values;
- Example: Degree of a person holds (BTech, MTech, MBA)
- It may so happen that multivalued attributes takes no value at times

Stored vs Derived

Comparison

- When two or more attribute values are related
- Example: `Age` and `birth_date`
- Age is derivable from birth_date
- birth_date is called stored attributed
- Age is called derived attribute

Complex Attributes

What is complex attribute?

- When nested attributes involving composite and multivalued attributes are used
- Arbitrary nesting can be used
- Notation:
  - Composite attributes are represented within ()
  - Multivalued attributes are represented withing {}
- Example: {Address_phone ({Phone(Area_code, Phone_number)}, Address(Street_address(Number, Street, Apartment_number), City, State, Zip)) }

NULL values

Notion of NULL

- Particular entity may not have an applicable value for an attribute
- Example: Apartment_number not applicable for individual houses
- Example: College_degree apply to only people with college degrees
- Such instances, NULL values are used to store data

NULL values

Meaning

- We do not know land line number of ‘Smith’.
- NULL here may mean this field not applicable
- In case ‘Smith’ has land line number but we do not know, it can be recorded as unknown
  - missing When it is known attribute value is exists but is missing
  - When the attribute values is not known Not known - land line of ‘Smith’ (exists but we do not know)

Attributes Classification

Classification

- Identifier
- Category
- Quantifier
- A text item

Identifier

Description

- Exists purely to identify entity instances
- Do not imply any property of instances
- Example: Order number, product code, batch number, etc.

Category

Description

Stores single value

- Can only hold one of the defined set of values
- Example: Product type, credit rating, payment method, delivery status, etc.

Quantifier

Description

Can perform arithmetic operations

- Attribute on which arithemtic can be performed
- Comparisons can be performed
- Example: Order quantity, order date, Unit price, Discount rate, etc

Text Item

Description

Can hold any string of characters that the user may choose to enter

Identifier

Details

IDs may be of three types

- System generated
- Administrator generated
- Externally definied identifiers

System Generated

Examples

- Order numbers (no human intervention)
- Account numbers, RD number, FD number, mobile number, etc..
- Generated in sequence without any specific requirement of the sequence generation
- Can be numeric and non-numeric

Administrator Generated

Examples

- Only suitable for relatively low-volume entity classes
- Department codes, product codes, class room numbers, course codes etc
- Can be numeric or non-numeric
- Administrator have mechanism to create new identifiers

Externally Defined

Examples

- Defined by external party
- Often by national or international standards authority
- Country codes (telephone numbers)
- Currency codes
- State codes
- Pin codes
- Codes externally defined but administrator generated for postal department

Identifiers

Role

- Used in many instances of operations
- Used as constraints
- Uniquely identifying entities

Categories

Details

- Typically administrator defined
- Some times externally defined
- Examples: {'Cash', 'CoD', 'Credit Card', 'Net Banking', 'Debit Card'}
- Grades etc.

Quantifiers

Details

Manifests in many forms

Count vehicle count, employee count, student count etc.

Dimension Answer questions like How long?, how high?, how wide?, how heavy?. A unit must follow the number

Currency amount answers questions of the form How much? and specifies an amount of money (Unit price, payment amount, etc)

Factor Dimensionless quantity: Interest rate, hourly rate, etc.

Specific Time Point answers questions of the form when?: Order date, arrival date etc.

Recurrent Time Point : RD deposit date, subscription renewal time, fee payment date etc

Interval answers questions like: loan repayment period, EMI installments periods etc.

Location answers questions like: Where?

Attribute Domains

Details

- Each simple attribute is associated with a domain of values
- Example: age attribute - between 16 and 70
- Example: Name string formed using alphabet {a, b, c, ..., z, SPACE}
- Domain is not represented in ER diagram

Attribute Domains

Mathematical definition

An attribute A of an entity set E whose domain is V is defined as function: $A : E \rightarrow P(V)$ where $P(V)$ is the power set

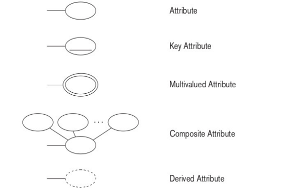
Attribute Domains

Discussion

- Value of attribute A for entity e is referred as $A(e)$
- Definition of domain covers single-valued and multivalued attributes
- NULL is represented by empty set
- Composite attribute A, the domain of V is the power set of Cartesian product of $P(V_1), P(V_2), \cdots, P(V_n)$;
  $V = P(P(V_1) \times P(V_2) \times \cdots \times P(V_n))$

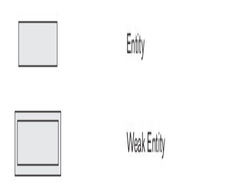
Attributes

ER Notations



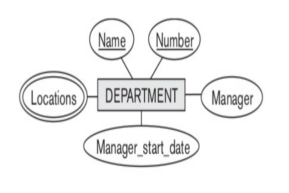
Entity

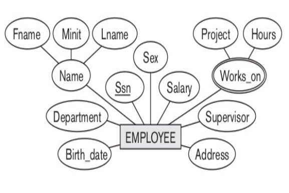
ER Notations



Entity Attributes

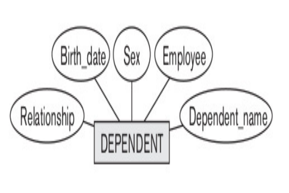
Department and Employee

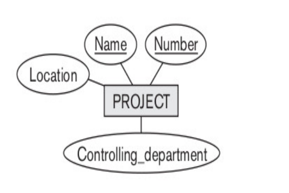




Entity Attributes

Dependent and Project





Entity Types, Entity Sets

Entity Type - Definition

A collection of entities that have same attributes. It describes schema
Example: {Employee, Company}

Entity Types, Entity Sets

Entity Type - Definition

A collection of entities that have same attributes. It describes schema
Example: {Employee, Company}

Entity Set - Definition

The collection of entities of a particular entity type
Example: $\{e_1, e_2, \cdots,\}$

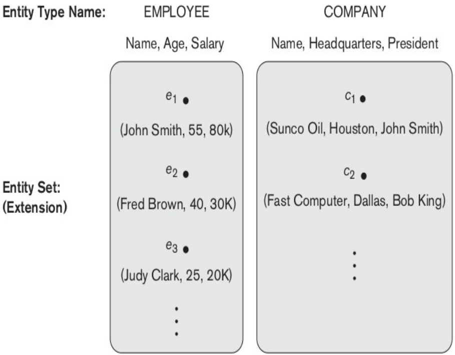
Entity Types, Entity Sets

Entity Type - Definition

A collection of entities that have same attributes. It describes schema
Example: $\{$Employee, Company$\}$

Entity Set - Definition

The collection of entities of a particular entity type
Example: $\{e_1, e_2, \cdots,\}$



Entity Type Name: EMPLOYEE COMPANY

Name, Age, Salary

Name, Headquarters, President

Entity Set: (Extension)

e_1 •
(John Smith, 55, 80k)

e_2 •
(Fred Brown, 40, 30K)

e_3 •
(Judy Clark, 25, 20K)

⋮

c_1 •
(Sunco Oil, Houston, John Smith)

c_2 •
(Fast Computer, Dallas, Bob King)

⋮

Key Attributes

Definition

An entity type having minimal set of attributes whose values are distinct for each individual entity in an entity set.

Key Attributes

Definition

An entity type having minimal set of attributes whose values are distinct for each individual entity in an entity set.

- It stems from practicle considerations
- No two students posses identical roll numbers
- No two employees are assigned identical employee number
- Note that one or more attributes together form a key
- For example, student registers for a course, roll number and course number becomes a key
- This constraint prohibits any two entities having same value for the key attribute at the same time.

Introduction

Informal definitions

- Database is represented as collection of relations
- Each relation resembles a table
- Table contains rows and columns
- Each row represent a collection of related data values
- Every column stand for attributes of the entities
- A row represents a fact that correspond to a real-world entity or a relationship

Notations

Relational Model Terminology

Row	is a tuple
Colum header	is an attribute
Table	is a relation
Data type	corresponding to each column - is the domain

Domain

Definition

A domain D is a set of atomic values. Atomic means each value in the domain is indivisible

- A domain is associated with three things
- A name
- Data type
- Data format

Examples

Example

- CPI - a real value between 0.00 and 10.00

Name	:	cpi
Data type	:	floating point between (0.00 and 10.00)
Format		`ff.ff`

- Employee age

Name	:	`emp_age`
Data type	:	integer value between (15 and 80)
Format		`dd`

- Deparmtnet

Name	:	`dept_name`
Data type	:	From a set of values { CSE, ECE, ME, CE, DD . . . }
Format		`internal format`

Schemas, Instances, & Database State

About

- Description of database is different from database itself
- The description of a database is called database schema
- Schema is specified during database design
- Schema's do not change frequently as opposed to data

Schema - Example

Examples

Student(

Name	Student_number	Class	Major

)

Schema - Example

Examples

Student(

Name	Student_number	Class	Major

)

Examples

Course(

Name	Number	Credit_hours	Department

)

Schema - Example

Examples

Student(

Name	Student_number	Class	Major

)

Examples

Course(

Name	Number	Credit_hours	Department

)

Examples

Prerequisite(

Course_number	Prerequisite_numbe

)

Schema - Example

Examples

Student(

Name	Student_number	Class	Major

)

Examples

Course(

Name	Number	Credit_hours	Department

)

Examples

Prerequisite(

Course_number	Prerequisite_numbe

)

Examples

Section(

section_id	course_number	semester	year	instructor

)

Schema - Example

Examples

Student(	Name	Student_number	Class	Major	)

Examples

Course(	Name	Number	Credit_hours	Department	)

Examples

Prerequisite(	Course_number	Prerequisite_numbe	)

Examples

Section(	section_id	course_number	semester	year	instructor	)

Examples

Grade_Report(	student_number	section_id	grade	)

Schemas

About

- Displays some aspects of a scheme
- Data types, relationship between schemas are not shown
- Constraints are not represented in this scheme
- Not all constraints can be represented in the schema

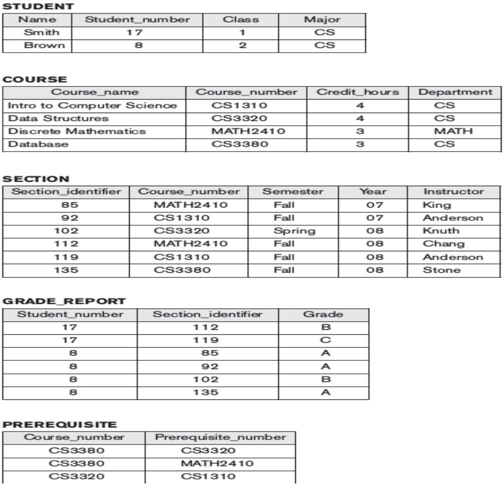
Schemas and databases

About

- When we define a new database, we specify its database schema only to the DBMS
- Immediately after this database state is empty
- We get to initial state when database is populated with some data.
- Every update operation leads to a different database state
- At any point database has a *current state*
- DBMS is responsible for ensuring that every state is a valid state

Database - at a particular moment of time

database state or snapshot or set of occurances or instances



STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Database State

- When a new database is defined state of database is empty
- When database is populated with initial data, initial state is then present
- Every modification operation on the database yields a new state - current state
- Database is in valid state when the current state satisfies structure and constraints

Relation Schema

Definition

- Used to describe a relation R
- Denoted by $R(A_1, A_2, \cdots, A_n)$.
- Where $A_1, A_2, \cdots, A_n$ are list of attributes that describe relation
- Each attribute A_i is the name of role
- Each role has a domain D denoted by $\texttt{dom}(A_i)$
- Degree of relation is the number of attributes of its relation schema

Relation Schema

Examples

Student(

Name	Student_number	Class	Major

)

Relation Schema

Examples

Student(Name: `string`, Student_number: `int`, Class: `string`, Major: $\{1^{st}$ `year`, 2^{nd} `year`, 3^{rd} `year`, 4^{th} `year`$\})$

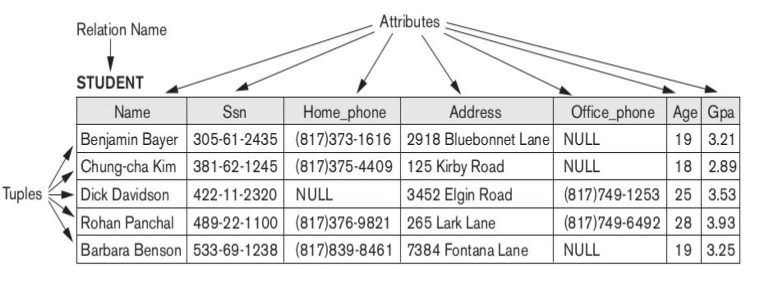
Relation

Definition

- A relation r of the relation schema $R(A_1, A_2, \cdots, A_n)$
- Denoted by $r(R)$
- Is a set of $n-$ tuples
- $r = \{t_1, t_2, \cdots, t_n\}$
- Each `n-tuple` is an ordered list of n values $t = \langle v_1, v_2, \cdots, v_n \rangle$
- i^{th} value in the tuple t corresponds to attribute A_i is denoted as $t[A_i]$ or $t.A_i$

Relation

Illustrative Figure


Relation Name
Attributes
Tuples

STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Relation Schema

Formal Definition

A relation $r(R)$ is a mathematical relation of degree n on domains $\mathtt{dom}(A_1)$, $\mathtt{dom}(A_2)$, $\cdots$, $\mathtt{dom}(A_n)$.

Formal Definition

$$r(R) \subset \mathtt{dom}(A_1) \times \mathtt{dom}(A_2) \times \cdots \times \mathtt{dom}(A_n)$$

Possible number of tuples

Possible number

The total numbr of tuples the relation has: $|\texttt{dom}(A_1)|$ $(\times)$ $|\texttt{dom}(A_2)|$ $(\times \cdots \times)$ $|\texttt{dom}(A_n)|$

Characteristics of Relations

Characteristics

Ordering of Tuples

- Relation is a set of tuples
- Elements of a set have no order among them
- Relation is not sensitive to the ordering of the tuples
- Tuple ordering is not part of relation definition

Ordering of values within a tuple

- Relation of $n - tuple$ is an ordered list of n values
- So the ordering of values in a tuple and hence of attributes in a schema is important
- However, the order of attributes and their values is not that important
- As long as the correspondence between attributes and values is maintained

Characteristics of Relations

Characteristics

Values and NULLs in the Tuples