Module 2
Relational Model

Teknik Informatika Fakultas Teknik
Universitas Dr. Soetomo Surabaya

Module 2 - Motivation

- The Relational Model introduced in 1970, has been the turning point for modern database systems
- The Relational Model is the basis for several commercial Database Management Systems, for example, Oracle, IBM's DB2, Microsoft Access
- The Relational Model has a sound theoretical basis

Module 2 - Contents

The Relational Data Model

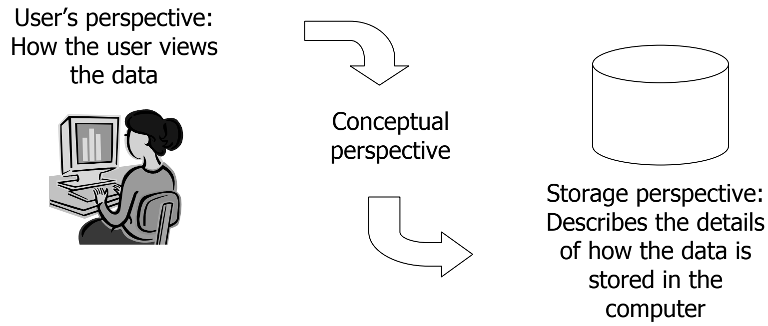
- Introductory Concepts
  - Data Model
  - Schemas and Instances
- The Relational Model
  - Basic Concepts
  - Relational Schemas and Instances
  - Characteristics of relations
- Relational database design and Constraints

Data Model

- An abstract view of data that excludes many details that are either too complex or not of interest to the users
- Comprised of logical concepts, e.g., objects or entities, their properties and their interrelationships
- May be easier for most users to understand

Categories of Data Model

- The data may be modeled from different perspectives



Types of Data Models

- Relational Data Model ⇐ Focus of this lecture
  - Most commonly used model for traditional applications, and therefore basis for many commercially successful DBMSs
- Legacy Data Models
  - Mostly used in the past, for example Network and Hierarchical Data Models
- Object Data Models
  - New paradigm, in response to applications with complex data and the success of object oriented concepts in software engineering

Schemas and Instances

- Most data models have the concept of “schema” and “instance”
- Schema
  - A Schema is the *meta-data*, or *data describing data*
  - Schema is specified during database design, and is not expected to change frequently
- Instance
  - An Instance is the data in the database at a particular time
  - Instances are created during data updates and change frequently

Module 2 - Contents

The Relational Data Model

- Introductory Concepts
  - Data Model
  - Schemas and Instances
- The Relational Model
  - Basic Concepts
  - Relational Schemas and Instances
  - Characteristics of relations
- Relational database design and Constraints

Relational Data Model

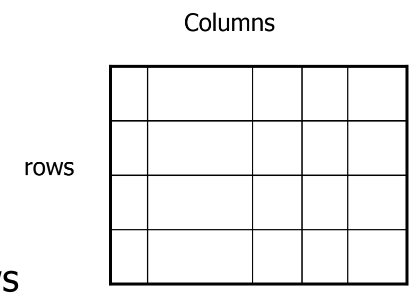
- Introduced by E.F. Codd in 1970
- Many DBMS products based on this model (but few *completely* implement it)
- Based on sound theoretical foundation
- Simple and uniform data structure : ***Relation***

Relational Model Concepts

- Relations
- Domains
- Attributes
- Tuples

Relations

- A Relation is the main construct for representing data in the Relational Model
- Informally, a relation
  – is a set of records
  – is similar to a table with columns and rows



Relations, not Tables

- The term Table is used interchangeably with Relation
  – Every relation is a table
  – Every table is not necessarily a relation !
- Relations have specific properties, based on the mathematical set theory

City: Brisbane		Product	Year: 1998			
Region	**Suburb**		**Qtr 1**	**Qtr 2**	**Qtr 3**	**Qtr 4**
South	Algeter	Disk	32	243	23	246
South	Calam Vale	Labels	4232	65	865	768
West	Taringa	Envelops	3242	543	4554	454
North	McDowell	Toners	23	465	24	432
South	Sunny Bank	Ribbons	324	65	56	657
West	Indooroopill y	Disk	234	6786	324	

Not a Relation !

Relation Components

Relation Name

Attribute Names

STUDENT

Name	StudentNo	Sex	Degree
Joe Smith	606 567 333	M	BSc
A. Brown	321 638 999	F	BInfTch

Tuples

Attribute Values from same Domain

Domains

- A Domain **D** is a set of atomic values
- An atomic value is indivisible (as far as the relational data model is concerned)
- Each domain has a data type or format
- Examples:
  - Domain of Names: {Joe Smith, Alan Yates, Bob Lovell, Jane Austin, ...}
  - Domain of Degrees: {Phd, MSc, BInfTech, BSc, MInfTech, …}

Domain Types

- integers
- real numbers
- fixed or variable length character strings
- date
- time stamp
- currency
- sub-range from a data type, e.g. 1£_ Grade £_ 7
- enumerated data type, e.g. {'Male', 'Female'}

Example Domains

Indonesian telephone numbers

- Values: the set of telephone numbers valid in Indonesia
- Format: the digits "62" followed by 9 digits 0-9

Auto registration numbers

- Values: The set of valid vehicle registration codes (i.e., arbitrary combinations of characters and digits)
- Format: 6 characters (either alpha or digits but no 'Q's allowed)

Attributes

- Each attribute **A** is the name of a role played by some domain **D** in the relation named **R**
- The number of attributes in a relation R is called the **degree of R**
- Example: **StudentNo** is an attribute name

Name	StudentNo	Sex	Degree
Joe Smith	606 567 333	M	BSc
A. Brown	321 638 999	F	BInfTch

Domains / Attribute Restrictions

- Same attribute name does not necessarily imply same domain

Subject

Code	Subject
CSEE	CS182
CSEF	CS180
Law	LA303

Lecturer

Code	Staffmem
CS182	Harrison
CS181	Duke
CS180	Marjanovic

Domains for Subject.CODE and Lecturer.CODE must differ, even though Attribute names are the same

Domains / Attribute Restrictions

- Different attribute name does not necessarily imply different domain

EID	ENAME	ESAL	EADDR	SID
86	**Smith**	**$42K**	**Fry Ave**	**52**
52	**Chen**	**$54K**	**King St.**	**52**

Domains for EID and SID are the same
but the Attribute names must differ

Tuples

- Each Tuple **t** is an ordered list of n values: **t = <v1,v2, … , vn>** where each value vi (1 £ i £ n) is an element of the corresponding domain of attribute Ai or a special value called "null"
- **t** is called an n-tuple
- Example
  (254, John, Smith, $45K, 3453-2543,M)

Relation Components - Review

Relation Name

Attribute Names

STUDENT

Name	StudentNo	Sex	Degree
Joe Smith	606 567 333	M	BSc
A. Brown	321 638 999	F	BInfTch

Tuples

Attribute Values from same Domain

Relation Schema and Instance

- Relation Schema
  - Denoted by R(A1,A2,A3,…,An), includes a relation name R and list of attributes A1, A2, … An
  - Integer *n* is termed "degree of the relation"
  - Example relation schema of degree 4:

    STUDENT(Name, Age, Id, Address)
- Relation Instance
  - A relation instance *r* of the relation schema R, denoted by r(R), is a set of n-tuples r = {t1,t2, …, tm}.

Example Schema and Instance

Schema: description of the relation

Name	StudentNo	Sex	Degree
Joe Smith	606 567 333	M	BSc
A. Brown	321 638 999	F	BInfTch

Instance: data within the relation

Ordering Characteristics

- Ordering of Tuples
- Ordering of Values in a Tuple

Ordering of Tuples

Relations are *sets* of tuples

- Mathematically, elements of a set have no implied order
- Semantically, when reasoning with relations, e.g., when formulating queries, order is irrelevant
- Physically, tuples reside on blocks of secondary storage, which have a partial ordering, hence tuples have an ordering

Ordering of Tuples

Subject-Lecturer-1

Code	Staffmem
CS182	Harrison
CS181	Duke
CS180	Marjanovic
LA303	Skase

Subject-Lecturer-2

Code	Staffmem
LA303	Skase
CS180	Marjanovic
CS181	Duke
CS182	Harrison

Same Relation

Ordering of Values within a Tuple

n-tuple is an *ordered* list of n values

- Syntactically, all tuples in a relation havevalues in the same order
- Semantically,the order chosen is irrelevant, as long as the correspondence between the attributes and the values can be maintained

Ordering of Values within a Tuple

Subject-Lecturer-3

Staffmen	Code
Harrison	CS182
Duke	CS181
Marjanovic	CS180
Skase	LA303

Subject-Lecturer-4

Code	Staffmem
Skase	LA303
Marjanovic	CS180
Duke	CS181
Harrison	CS182

Same Relation

Module 2 - Contents

The Relational Data Model

- Introductory Concepts
  - Data Model
  - Schemas and Instances
- The Relational Model
  - Basic Concepts
  - Relational Schemas and Instances
  - Characteristics of relations
- Relational database design and Constraints

Relational Database Design

- Relational Database
  - A collection of relations with distinct relation names
- Relational Database Schema
  - A collection of schemas for the relations in the database
- Relational Database Design
  - Process of capturing the semantics of an application, and translating it into a relational database schema

Importance of Design

- Poor design can lead to redundant information in tuples and update anomalies
- Can also result in inability to represent information and consequently, loss of information
- Conceptual modeling, and other formal techniques, can assist a designer in obtaining good design characteristics

What are we designing?

Relations

Name

Attribute

STUDENT

Name	StudentNo	Sex	Degree
Joe Smith	606 567 333	M	BSc
A. Brown	321 638 999	F	BInfTch

Constraints
Not Null

Domains
Char, Number, ...

Simple Design Example

- Good database design is not automatic!

Employee Report			
Printing (Darra)			
Dilbert	**40K**	**Taringa**	
Jones	**60K**	**Kenmore**	**(Manager)**
Head Office (City)			
Trump	**65K**	**Auchenflower**	**(Manager)**
Harrison	**78K**	**St.Lucia**	
. . . . . .	. . . . . .	. . . . . .	. . . . . .

Naive Implementation

NAME	SALARY	ADDRESS	DEPT	LOC	MGR
Smith	50k	St.Lucia	Printing	Darra	Jones
Dilbert	40k	Taringa	Printing	Darra	Jones
Jones	60k	Kenmore	Printing	Darra	Jones
Trump	65k	Auchenflower	Head Office	City	Trump
Harrison	78k	St.Lucia	Head Office	City	Trump
...	...	...	...	...	...

- Consider these updates:
  - Printing department moved to Ipswich
  - Smith is new manager of Printing Department

Better Implementation

Relation: Employee

NAME	SALARY	ADDRESS	DEPT
Smith	50k	St.Lucia	Printing
Dilbert	40k	Taringa	Printing
Jones	60k	Kenmore	Printing
Trump	65k	Auchenflower	Head Office
Harrison	78k	St.Lucia	Head Office
…	…	…	…

Relation: Department

DEPT	LOC	MGR
Printing	Darra	Jones
Head Office	City	Trump

- Again consider these updates:
  - Printing department moved to Ipswich
  - Smith is new manager of Printing Department

Implementation of Constraints

- DBMS must enforce constraints :
  - User is not permitted to remove department if employees remain
  - User is not permitted to insert employee without reference to an existing department (in relation Department)
  - User is not permitted to delete an employee who is a manager

Database Integrity Constraints

- Integrity constraints are specified on the database schema
- They must hold on every instance of that schema, as well as on transitions of the schema
- Some integrity constraints are enforced by the DBMS itself
- Some are enforced using procedural logic

Categories of Integrity Constraints

- Static and Dynamic constraints
- Single and Multiple Relation constraints
- Structural and Semantic constraints

Database Integrity Constraints

Restrictions on data that can be specified on a relational database schema

- Structural, Static, Single Relation
  - Domain Constraints
  - Key Constraints
  - Entity Integrity Constraints
- Structural, Static, Multiple Relation
  - Referential Integrity Constraints

Domain Constraints

- A Domain is a set of atomic values
- Each attribute in a relation will belong to some domain
- Data types associated with domains
  - integers, real numbers, fixed or variable length, character strings, date, time stamp, currency, subrange from a data type, e.g. 1£_ Grade £_ 7, enumerated data type, e.g. {'Male', 'Female'}

Example Domain Constraints

Atomicity of Values in the Tuples

Name	ENo	Salary	Phones
Joe Smith	367-90	30,800	34980987
			36789817
A. Brown	987-87	65,000	39871236
Bob Forrester	884-89	76,000	38490699
			0419998473
			34980876

Non-Atomic Values ←

Person	Lives At
Mike	23 Hawken Dr. St Lucia
Carol, Pate	65 Jephson Str. Toowong

Atomic Values ←

↑ **Non-Atomic Values**

Key Constraints

- All tuples in a relation must be distinct, that is no two tuples can have same values for all attributes ☛ *uniqueness constraint*

Violation of Uniqueness Constraint ? {

Name	StudentNo	Address	Degree
J Smith	321 638 998 23	Milton Rd	BInfTch
J Smith	321 638 998 23	Milton Rd	BInfTch

Notion of a Superkey

- A Superkey is a subset of attributes (SK) of a relation schema R, such that for any two tuples, ti and tj in a relation state r of R ti [SK] ^{1} tj[SK]
- Every relation has *at least one* superkey – the set of all its attributes
- Superkey can have *redundant attributes*, that is, by removing some attributes, the uniqueness constraint is still maintained

Example Superkey

• Superkey for the Relation STUDENT
– (Name, StudentNo, Sex, Degree)
– (Name, StudentNo, Sex)
– (Name, StudentNo, Degree)
– (StudentNo, Degree)
– ...

Which of these attributes are redundant ?
Is (StudentNo) the smallest set of attributes that uniquely identify a tuple in the relation STUDENT ?

STUDENT

Name	StudentNo	Sex	Degree
Joe Smith	321 638 998 23	M	BSc
A. Brown	321 638 998 24	F	BInfTch

Notion of a Key

- K is a key in a relation schema R if
  - K is a Superkey of R, and
  - removing any attribute from K leaves a set of attributes K', where K' is not a superkey of R, that is, K' does NOT maintain the uniqueness constraint
- Key is a minimal Superkey
  - smallest set of attributes that uniquely identify a tuple (StudentNo) **is** the smallest set of attributes that uniquely identify a tuple in the relation STUDENT

Example Key

- **Possible Keys for the Relation WORKS-IN**

– (DEPT) **Not a Superkey, many employees work in the same department**

– (EMPNO) **Not a Superkey, the same employee of the same department will work on several days**

– (DEPT, EMPNO) **Not a Superkey, one employee will work several times**

– (DEPT, EMPNO, DATE, HOURS) **Superkey, but not a key since removing Hours, still maintains uniqueness**

– (DEPT, EMPNO, DATE) **Correct Key for relation WORKS-IN**

WORKS-IN

Dept	EmpNo	Date	Hours
Printing	606	1/1/99	3
Printing	607	1/1/99	6
Printing	606	2/1/99	9
Sales	319	1/1/99	6
Sales	321	1/1/99	9

Example Key

- **Determining the Key for the Relation ENROLMENT**

– (STNO) **Not a Superkey, one student will enroll in many subjects**

– (SUB-CODE) **Not a Superkey, one subject will have many students**

– (STNO, SUB-CODE) **Correct Key for relation ENROLMENT ?**

- **What if a student who failed (or dropped out of) a subject, enrolls in the subject again, in the next semester … Is this key sufficient to record that data ?**

ENROLMENT

STNO	SUB-CODE	SEMESTER	STATUS
S3457890	CS181	1/99	Pass
S9875604	CS181	1/99	Fail
S8901265	CS386	2/99	Dropped
S3457890	CS182	2/99	Pass
S3457890	CS271	1/00	Pass

Characteristics of Keys

- Value of key attributes uniquely identify a tuple in a relation
- Key constraints hold on every relation instance
  - – Name cannot always be used as key
- A schema may have more than one key
  - – Each is called a "**candidate**" key
  - – One is selected as the "**primary**" key

Name	<u>StudentNo</u>	Sex	Degree

Entity Integrity Constraint

- No primary key can be null
  - How would you distinguish between John Smith and Janet Smith if the primary key was missing?

StudentNo	Name	Age
?	Smith	18
?	Smith	18

Referential Integrity Constraint

- Key and Entity Integrity constraints are specified on individual relations
- Referential Integrity constraints are specified between two relations

EMP [Eno, Name, Salary, SuperEno, DeptNo]

↓

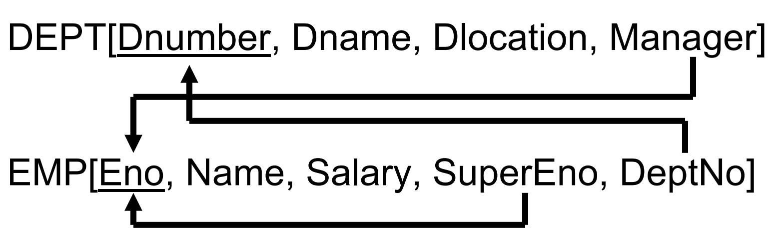
DEPT [Dnumber, Dname, Dlocation]

Foreign Keys

- A set of attributes FK in relation schema R1 is a foreign key if
  – the attributes of FK have the same domain as the the primary key attributes PK of another schema R2
  – t1[FK] = t2[PK]   or   t1[FK] is null
- FK is said to “reference” PK

Example Foreign Key

- Every employee is assigned to one department
  – DeptNo is a foreign key of EMP referencing DEPT
- Every Department has a Manager. The manager is also an employee
  – Manager is a foreign key of DEPT referencing EMP
- Every Employee has a supervisor who is also an employee
  – SuperEno is a foreign key of EMP referencing EMP



Other Integrity Constraints

- General class of constraints that cannot be enforced by the preceding constraints
- Semantic Constraints
  - “The salary of an employee should not exceed the employee’s supervisor’s salary”
  - “The maximum number of hours that an employee can work on a project is 56”
- Transition Constraints
  - “The salary of an employee can only increase”
- Often implemented in a constraint specification language (SQL3) using *triggers* and *assertions*

Database Schema and Instance

- Relational Database Schema:
  - set of relational schemas {R1, R2, ..., Rn}
  - set of integrity constraints
- Relational Database Instance:
  - set of relation instances {r1, r2, ..., rn} such that each ri is an instance of Rj
  - the ri relations satisfy the integrity constraints

**Relational Database =
schema + (current) instances**

Designing a Relational DB

- Deciding which attributes belong in each relation
- Choosing appropriate names for each relation
- Specifying domains and data types for the attributes
- Identifying candidate keys and selecting a primary key for each relation
- Specifying all foreign keys

Constraints and Operations

- Enforcement of integrity constraints ensures that the database remains consistent
- Changes to the database must not violate integrity constraints (leave the database in an inconsistent state)
- If a Database update is submitted to the DBMS that would violate integrity, it must be rejected

Constraints & Insertion

‘323456789’, ‘J Smith’, 18 →

StudentNo	Name	Age

- Integrity constraints can be violated by inserting a new tuple
  - Student with StudentNo already exists
  - Age is ‘old’ instead of 18
- The insert can be rejected, or the reason for rejection corrected

Constraints & Deletion

← delete student ‘323456789’

StudentNo	Name	Age
323456789	J Smith	18

- Referential integrity can be violated if the tuple being deleted is referenced by Foreign Keys from other tuples
  - deleting a DEPT while there are still EMPloyees working in that DEPT
- The deletion can be rejected, cascaded or the referencing attribute values can be modified

Constraints & Modification

change '323456789' age to 19 →

StudentNo	Name	Age
323456789	J Smith	18

- Non key values
  – domain check
- Primary key
  – similar to performing a delete and an insert
- Foreign key
  – DBMS must ensure new value refers to existing tuple in referenced relation

Module 2 - Review

Key Words : Relation, Attribute, Domain, Tuple, Schema, Instance, Constraint

Key points:

- Defining a database includes the specification of the schema
- Each update to the database causes a new database state
- The DBMS ensures that every database instance (or state) is valid/legal, and that every state transition is also legal

Recommended Readings

Elmasri & Navathe

Chapter 7 (7.1 - 7.3)

Next ...

Module 3

Entity Relationship Model