Relational Database Model

Introduced by E.F. Codd (1970)
http://www.acm.org/classics/nov95/

Based on relational algebra and logic developed by

Schröder (1880s)
Charles Peirce (1890s)
Russell and Whitehead (1900s)

Codd's Twelve Rules

1. Information represented at the logical level in tables.
2. Data is determined by table, primary key, and column.
3. Missing information is modeled as null values.
4. Metadata is part of the database.
5. Single language for all tasks in DBMS.
6. Views and tables must change simultaneously.
7. Single operations for retrieve, insert, delete, update.
8. Operations independent of physical storage and access.
9. Database modifiable without affecting applications.
10. Constraints are part of database.
11. DML independent of physical layer (distributed, etc.)
12. Row-processing obeys same rules as set-processing.

Relations

Extensional versus intensional

Extensional Representation:

table of values
rows = records
columns = attributes

Note: tables are ordered, relations are not

Domains

Set of **atomic** values for an attribute

atomic = indivisible
(e.g. IT 240 = IT + 240 is divisible)

Examples
age: {x: x is a valid age}
sex: {male, female}

Physical Level: data type + format

Relation Schema

$R(A_1, A_2, \ldots, A_n)$ **relational schema**

R : Name of Relation
$A_1, A_2, \ldots, A_n$: **Attributes** A_i has **domain** D_i
N : **degree** (**arity**) of R

Movie(movieID, title, genre, length, rating)

dom(movieID) = MovieIDs = {x: x is a number}
dom(title) = Titles = {x: x is a word}
dom(genre) = Genres = {Musical, Horror, …}
dom(length) = Lengths = {x: x is valid time}
dom(rating) = Ratings = {NR, G, PG, PG-13, R, NC-17}

Relational Schemas Example

CUSTOMER(Customer_ID, Customer_Name, City, State, Postal_Code)

ORDER(Order_ID, Order_Date, Customer_ID)

ORDERLINE(Order_ID, Product_ID, Ordered_Quantity)

PRODUCT(Product_ID, Product_Description, Product_Finish, Standard_Price, Product_Line_ID)

More Examples: page 32

Instances

$R(A_1, A_2, \ldots, A_n)$ **relational schema**,
A_i has **domain** D_i

Instance of schema R is a table with data from domains

Example:

Rep(RepNum, LastName, FirstName, Street, City, State, Zip, Commission, Rate)

Rep

RepNum	LastName	FirstName	Street	City	State	Zip	Commission	Rate
20	Kaiser	Valerie	624 Randall	Grove	FL	33321	$20,542.50	0.05
35	Hull	Richard	532 Jackson	Sheldon	FL	33553	$39,216.00	0.07
65	Perez	Juan	1626 Taylor	Fillmore	FL	33336	$23,487.00	0.05

Figure 2-1 page 30

Schema and Relation Examples

STUDENT(Name, SSN, HomePhone, Address, OfficePhone, Age, GPA)

MOVIE

VIDEO

RENTAL

Records

<101, "Thirty-Nine Steps", mystery, 101, R>
<510, "Monkey Business", comedy, null, null>

are possible **records (or tuples)** in

MOVIE(movieID, title, genre, length, rating).

null: value unknown, or attribute does not apply

values **atomic**: no multiple values (first normal form)
(e.g. several genres)
indivisible (name = first name + last name)

What about multiple values?

Orders

OrderNum	OrderDate	PartNum	NumOrdered
21608	10/20/2010	AT94	11
21610	10/20/2010	DR93 DW11	1 1
21613	10/21/2010	KL62	4
21614	10/21/2010	KT03	2
21617	10/23/2010	BV06 CD52	2 4
21619	10/23/2010	DR93	1
21623	10/23/2010	KV29	2

FIGURE 5-5 Sample unnormalized table

Orders

OrderNum	OrderDate	PartNum	NumOrdered	QuotedPrice
21608	10/20/2010	AT94	11	$21.95
21610	10/20/2010	DR93	1	$495.00
21610	10/20/2010	DW11	1	$399.99
21613	10/21/2010	KL62	4	$329.95
21614	10/21/2010	KT03	2	$595.00
21617	10/23/2010	BV06	2	$794.95
21617	10/23/2010	CD52	4	$150.00
21619	10/23/2010	DR93	1	$495.00
21623	10/23/2010	KV29	2	$1290.00

FIGURE 5-7 Sample Orders table

Telephone numbers ?

Functional Dependencies I

ORDER_DETAIL(Product, Quantity,
PricePerUnit, Subtotal)

Example

<Soymilk, 3, $3.19, $9.57> in ORDER_DETAIL

(Quantity, PricePerUnit) → Subtotal

Quantity and PricePerUnit determine Subtotal

Functional Dependencies II

A → B (A determines B)

if

setting the values of attribute(s) of A determines the values of B

Example

SID → Name in STUDENT(SID, Name)

Functional Dependencies III

EMPLOYEE(EmpID, FirstName, LastName, Salary, Sex)

STUDENT(Name, SID, HomePhone, Address, OfficePhone, Age, GPA)

PRODUCT(ProductID, Name, Description, PricePerUnit, UnitSize)

CAR(OwnerName, Vehicle#, Engine#, Color)

CUSTOMER(accountID, lastName, firstName, street, city, state, zipcode)

Constraints

- Domain constraints
- Key (uniqueness) constraints
- Entity integrity constraints
- Referential integrity constraints

- Data dependencies (functional dependencies, etc.)

Domain Constraints

Restriction on values of attributes (domain).

Specified as **data-type**: integer, char, etc., or user-defined type

Operations on data-types: +, *, <, =, …

not null constraint for an attribute

Keys

Key: smallest set of attributes that determines all attributes in a relation.
composite: more than one attribute

Example
STUDENT(Name, SID, HomePhone, Address, OfficePhone, Age, GPA)

{SID}: key
{SID, Name, Address}: not key

Key Examples

MOVIE(Title, Year, Director, Length, Rating)

PRODUCT(ProductID, Name, Description, PricePerUnit, UnitSize)

ACTIVITY(StudentID, Activity, Fee)

COURSE(CourseID, Title, Enrolment)

ENROLED(StudentID, CourseID, CourseTitle, Grade)

Candidate Keys

If a relation has more than one key, these keys are called **candidate keys.**

Examples
- EMPLOYEE(EmpID, FirstName, LastName, Salary, Gender)
- DePaul students: peoplesoftID and SSN
- COURSE(Department, Number, Name, Instructor)
- CAR(OwnerName, Vehicle#, Engine#, Color)

One candidate key is declared the **primary key** of the relation (underlined in schema)

Relational Databases and Schemas

Relational Database **Schema**: Collection of Relations
Relational Database **State**: Collection of Instances

ACTIVITIES = {STUDENT, ACTIVITY}

Student	<u>SID</u>	FName	LName
	101	Mark	Spencer
	971	Charles	Loeffler

Activity	<u>StudentID</u>	<u>Activity</u>	Fee
	971	Piano	$20
	971	Swimming	$10

Note: different names (SID, StudentID) for the same concepts

Foreign Key 1

A set of attributes in one relation (R_1) refering to a unique tuple in a second relation (R_2) through R_2's primary key.

Student	<u>SID</u>	FName	LName
	101	Mark	Spencer
	353	Gil	Ryle
	971	Charles	Loeffler

Activity	<u>StudentID</u>	<u>Activity</u>	Fee
	971	Piano	$20
	353	Reading	$5
	971	Swimming	$10

Terminology R_1 referencing relation
R_2 referenced relation

Foreign Key 2

Examples

REGISTRATION = {STUDENT, ENROLMENT, COURSE}
COMPANY = {EMPLOYEE, WORKS_ON, PROJECT}
SUPPLY = {SUPPLIES, SUPPLIER, PART, COMPANY}

Note R1 = R2 is possible

Example

EMPLOYEE (with supervisor)
MOVIE (remakes)

Referential Integrity

declaration of foreign keys in a database schema

STUDENT(SID, FName, LName)
ACTIVITIES(StudentID references STUDENT, Activity, Fee)

or visually, by an arrow from foreign key to primary key

STUDENT(<u>SID</u>, FName, LName)

ACTIVITIES(<u>StudentID, Activity</u>, Fee)

Integrity Constraints

- **Domain Constraints:** declaration of domains
- **Not Null Constraints:** attribute values can not be null
- **Key Constraints:** candidate keys (uniqueness)
- **Entity Integrity Constraint**: primary key is not null
- **Referential Integrity Constraint:** declaring foreign keys

A **valid state** is a database state fulfilling all integrity constraints

Integrity Constraints defined by DDL

Semantic constraints (transitions) later