1. The Relational Model

Stéphane Bressan

January 22, 2015

This lecture is based on material by Professor Ling Tok Wang.

CS 4221: Database Design

The Relational Model

Ling Tok Wang
National University of Singapore

1

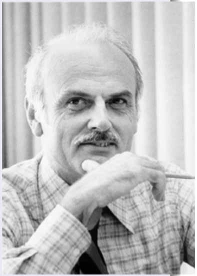
CS4221: The Relational Model

`https://www.comp.nus.edu.sg/`

`~lingtw/cs4221/rm.pdf`

Content

"Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). "
A Relational Model of Data for Large Shared Data banks [CACM 1970], by Edgar F. Codd



Example

The database of a manufacturing company contains information about parts and projects. For each part, the part number, part name, part description, quantity-on-hand, and quantity-on-order are recorded. For each project, the project number, project name and project description are recorded. Whenever a project makes use of a certain part, the quantity of that part committed to the given project is also recorded.

Many different ways of representing in the hierarchical model.

```
PART
part#
name
description
quantity-on-hand
quantity-on-order
PROJECT
    project#
    name
    description
    quantity-committed
```

```
PROJECT
project#
name
description
PART
    part#
    name
    description
    quantity-on-hand
    quantity-on-order
    quantity-committed
```

```
PART
part#
name
description
quantity-on-hand
quantity-on-order

PROJECT
project#
name
description

COMMIT
part#
project#
quantity-committed
```

The Relational Design Question

How many tables? What tables? How many columns in each table? What columns?

But Also

What Integrity Constraints?

Integrity Constraints in SQL

- PRIMARY KEY
- UNIQUE
- NOT NULL
- FOREIGN KEY
- CHECK

Part

Part#	PartName	Desc	QuantityHand	QuantityOrder
35212	nut	FISH HEAD BOLT	10000	560
23212	bolt	CAGE NUT	24366	123
6653	screw	Pan Head Screw	123	5000
...				

Project

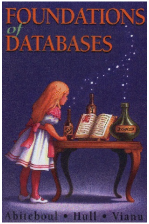
Project#	ProjectName	Desc
101	Bicyle	Build a bicycle with side car
203	Electric Car	Build an electric car that runs on solar power
...		

Commit

Project#	Part#	Quantity
35212	203	500
23212	101	232
6653	101	65
...		

Readings

- Codd, E.F.. "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM 13 (6): 377-387, (1970).
- Fillat A., Kraning L. "Generalized Organization of Large Data-bases; A Set-Theoretic Approach to Relations". MIT-LCS-TR-070 (1970). [OPTIONAL]
- Abiteboul S.,Hull R. and Vianu V."Foundations of Databases", `http://webdam.inria.fr/Alice/pdfs/all.pdf`



Motivation

Codd motivates the relational model by the inadequacy of hierarchical and network models with respect to:

- the lack of data independence;
- and the poor management of data inconsistencies.

Data Independence

- Ordering dependence;
- Indexing dependence;
- Access Path Dependence.

Data Consistency

- Structural constraint;
- Logical constraint.

Part

Part#	PartName	Desc	QuantityHand	QuantityOrder
35212	nut	FISH HEAD BOLT	10000	560
23212	bolt	CAGE NUT	24366	123
6653	screw	Pan Head Screw	123	5000
...				

Project

Project#	ProjectName	Desc
101	Bicyle	Build a bicycle with side car
203	Electric Car	Build an electric car that runs on solar power
...		

Commit

Project#	Part#	Quantity
35212	203	500
23212	101	232
6653	101	65
...		

Take				
Student#	Course#	S-name	C-desc	Mark
95001	CS1101	Tan CK	Programming	75
95023	CS1101	Lee SL	Programming	58
95023	CS2103	Tan CK	D.S. and Alg.	64
...				

Definition

Let us consider

- the countably infinite set $\mathcal{R}$ is a set of relations (relation names),
- the countably infinite set $\mathcal{A}$ is a set of attributes (attribute names) such that $\mathcal{R} \cap \mathcal{A} = \emptyset$, and,
- the set $\mathcal{D}$ is the domain (set of atomic values).
- If attributes need different domains the function *Dom* on $\mathcal{A}$ ($Dom : \mathcal{A} \mapsto 2^{\mathcal{D}}$) defines the domain of an attribute $A \in \mathcal{A}$: $Dom(A) \subset dom$.

$$TAKE \in \mathcal{R}$$

$$\{Student\#, Course\#, \text{S-name}, \text{C-desc}, Mark\} \subset \mathcal{A}$$

Definition

The structure of a table is given by the relation name and a finite set of attributes.
We assume that there exists a function from the set of relation names to the set of finite subsets of attribute names.

$$sort : \mathcal{R} \mapsto 2^{\mathcal{A}}_{finite}$$

$sort(R)$ is the schema of the relation. We write $R = sort(R)$.

$$sort(TAKE) = \{Student\#, Course\#, \text{S-name}, \text{C-desc}, Mark\}$$

$$TAKE = \{Student\#, Course\#, \text{S-name}, \text{C-desc}, Mark\}$$

Definition

The arity (or degree) of a relation name, R, is the number of its attributes.

$$arity(R) = \mid sort(R) \mid$$

TAKE is quinary.

$$arity(TAKE) = 5$$

degree 0= nullary, degree 1 = unary, degree 2 = binary, degree 3 = ternary ...
Do we have degree 0 relations in SQL?

Original Definition by Codd

The tabular representation of relations is a convenient (vizualization), practical (implementation) but not essential part (design and query) of the relational model.

1. Each row represents an n-tuple of R.
2. The ordering of rows is immaterial.
3. All rows are distinct.
4. The ordering of columns is significant-it corresponds to the ordering $S_1, S_2, \cdots, S_n$ of the domains on which R is defined.
5. The significance of each column is partially conveyed by labeling it with the name of the corresponding domain.

Original Definition by Codd

Given the non-necessarily distinct sets of atomic (i.e. non-decomposable) elements $S_1, S_2, \cdots, S_n$, a first normal form (1NF) relation R on these sets if it is a subset of the Cartesian (cross) product of these sets.

$$R \subset S_1 \times S_2 \times \cdots \times S_n$$

Domain

We refer to S_i as the n^{th} domain.

T-uples

R is a set of (ordered) n-tuples (t-uples)

$$< e_1, e2, \cdots, e_n > \in S_1 \times S_2 \times \cdots \times S_n$$

T-uples Constructor

R is a set of (ordered) n-tuples (t-uples) $< . >$ is the t-uple constructir symbol.

Duplicates

A set does not contain duplicate elements:

$$\{a, a, b\} = \{a, b\}$$

In the definition we gave the relation instances differ from the tables as they do not contain duplicate elements.

Order

A set is not ordered:

$$\{a, b\} = \{b, a\}$$

The order of t-uples in relations and rows in tables is irrelevant.

Order

Components of a t-uple in the unnamed view are ordered.

$$< Andrew, Jackson >$$

Andrew is the first name and Jackson is the family name.

Codd's Definition: Unnamed View

Under the unnamed view a tuple is is an element of the Cartesian product of the domain(s).

$$t_1 \in dom \times dom \times dom \times dom \times dom$$

$$t_1 =< 95001, CS1101, TanCK, Programming, 75 >$$

Named View

Under the named view a tuple is a functions mapping an attribute to a value in the domain of the attribute.

$$t_1 : R \mapsto dom$$

$$t_1(Student\#) = 95001$$

Conventional View

Under the conventional view a relation instance of a relation schema $R[U]$ (over the attributes U) is a finite set $I(R)$ of tuples.

Logic Porgramming View

Under the logic programming view a relation instance of a relation schema $R[U]$ (over the attributes U) is a finite set of facts over R.

A database.

R	
A	B
a	b
c	d
a	a

S
C
d

Query

Find the A-value in R such that the corresponding B-value in R is a C-value in S.

Unnamed and Conventional (Codd's View)

$$I(R) = \{< a, b >, < c, d >, < a, a >\}$$

$$I(S) = \{< d >\}$$

Domain Relational Calculus

$$\{< X > \mid \exists Y \; (< X, Y > \in R \wedge \; < Y > \in S)\}$$

Named and Conventional

$$I(R) = \{f_1, f_2, f_3\}$$

$$f_1(A) = a, f_2(A) = c, f_3(A) = a, f_1(B) = b, f_2(B) = d, f_3(B) = a$$

$$I(S) = \{g_1\}$$

$$g_1(C) = d$$

SQL

$$SELECT\ R.A\ FROM\ R,\ S\ WHERE\ R.B = S.C$$

Domain Relational Calculus

$$\pi_{R.A}(\sigma_{R.B=S.C}(R \times S))$$

$$\pi_{R.A}(R \bowtie_{R.B=S.C} S)$$

SQL

$$SELECT\ R.A\ FROM\ R,\ S\ WHERE\ R.B = S.C$$

$$SELECT\ R.A\ FROM\ R\ INNER\ JOIN\ S\ ON\ R.B = S.C$$

Named and Logic Programming

$$I = \{I(t_1 \in R) = true, I(t_2 \in R) = true, I(t_3 \in R) = true,$$

$$I(t_1.A = a) = true, I(t_1.B = b) = true,$$

$$I(t_2.A = c) = true, I(t_2.B = d) = true,$$

$$I(t_3.A = a) = true, I(t_3.B = a) = true,$$

$$I(t_4 \in S) = true, I(t_4.C = d) = true, \cdots^{a}\}$$

[a] The rest is false.

Domain T-uple Calculus

$$\{T \mid \exists T_1\, \exists T_2\, (T_1 \in R \wedge T_2 \in S \wedge T_1.B = T_2.B \wedge T.A = T_1.A)\}$$

$$\{< T_1.A > \mid \exists T_1\, \exists T_2\, (T_1 \in R \wedge T_2 \in S \wedge T_1.B = T_2.B)\}$$

$$\{< T_1.A > \mid \exists T_1 \in R\, \exists T_2 \in S\, (T_1.B = T_2.B)\}$$

SQL

$$SELECT\ \ T_1.A\ FROM\ R\ \ T_1,\ \ S\ \ T_2\ WHERE\ \ T_1.B = T_2.C$$

Unnamed and Logic Programming

$$I = \{I(R(a,b)) = true, I(R(c,d)) = true,$$

$$I(R(a,a)) = true, I(S(d)) = true, \cdots{}^{a}\}$$

[a] The rest is false.

Domain Relational Calculus

$$\{< X > \mid \; \exists Y \; (R(X, Y) \land S(Y))\}$$

Datalog

$$Q(X) \leftarrow R(X, Y), S(Y).$$

$$\leftarrow Q(X).$$

Definition

A database schema is a non-empty finite set $\mathcal{R}$ of relation and constraints on these relations.

The Design Question

How many tables? What tables? How many columns in each table? What columns? What Integrity Constraints?

The Universal Relation

Do we need more than one relation?

The Universal Relation

All the data is kept in a single relation whose scheme consists of all attributes. If necessary, null values are used to pad out t-uples

C	T	H	R	S	G
CS101	Deawood	M9	222	Weenie	B+
CS101	Deawood	W9	333	Weenie	B+
CS101	Deawood	F9	222	Weenie	B+
CS101	Deawood	M9	222	Grind	C
CS101	Deawood	W9	333	Grind	C
CS101	Deawood	F9	222	Grind	C

Readings

- Ullman J.,"Principles of Database and Knowledge-base Systems". Volume II (Chapter 17), Computer Science Press (1989).

C	T	H	R	S	G
CS101	Deawood	M9	222	Weenie	B+
CS101	Deawood	W9	333	Weenie	B+
CS101	Deawood	F9	222	Weenie	B+
CS101	Deawood	M9	222	Grind	C
CS101	Deawood	W9	333	Grind	C
CS101	Deawood	F9	222	Grind	C

Find the rooms in which Prof Deawood is teaching.

In the language of Stanford System/U:

RETRIEVE R WHERE T ='Deadwood'

C	T	H	R	S	G
CS101	Deawood	M9	222	Weenie	B+
CS101	Deawood	W9	333	Weenie	B+
CS101	Deawood	F9	222	Weenie	B+
CS101	Deawood	M9	222	Grind	C
CS101	Deawood	W9	333	Grind	C
CS101	Deawood	F9	222	Grind	C

RETRIEVE t1.R WHERE t1.R=t2.R AND t2.C='CS101'

Find the modules using a room used by CS101.

The following database has the following three relations.

$$Supplier(\underline{code}, sname)$$

$$Part(\underline{code}, pname, color)$$

$$Supply(\underline{supplier, part}, price)$$

Universal Relation Assumption: Same Name

Two attributes with the same name correspond to the same attribute in the universal relation, i.e. they are from the same attribute and of the same semantics (same meaning).

Universal Relation Assumption: Different Names

Two attributes with different names from two different relations or from one relation correspond to two different attributes in the universal relation, and have different semantics.

The following database has the following three relations.

$$Supplier(\underline{code}, sname)$$

$$Part(\underline{code}, pname, color)$$

$$Supply(\underline{supplier, part}, price)$$

The database does not satisfy the universal relation assumptions (Why? Bad design of attribute names).

C	T	H	R	S	G
CS101	Deawood	M9	222	Weenie	B+
CS101	Deawood	W9	333	Weenie	B+
CS101	Deawood	F9	222	Weenie	B+
CS101	Deawood	M9	222	Grind	C
CS101	Deawood	W9	333	Grind	C
CS101	Deawood	F9	222	Grind	C

with the underlying scheme $\{CT, CHR, CSG\}$,

Window

The window $[X]$ is a relation with scheme X, where X is the set of attributes mentionned in the query.

Example

The window has schema (R, T).

$$\text{RETRIEVE R WHERE T} = \text{'Deadwood'}$$

Window Function

The window function defines the window from the actual database.

Example

The window function is, for example, the natural join of the minimal set of relations whose scheme includes all the attributes.

with the underlying scheme $\{CT, CHR, CSG\}$,

$$[X] = \pi_{R,T}(CT \bowtie CHR)$$

Design Anomalies

Take					
Student#	Course#	S-name	C-desc	Mark	Text
95001	CS1101	Tan CK	Programming	75	The art of Programming
95023	CS1101	Lee SL	Programming	58	The art of Programming
95023	CS2103	Tan CK	D.S. and Alg.	64	The art of Programming
95001	CS1101	Tan CK	Programming	75	Java
95023	CS1101	Lee SL	Programming	58	Java
...					

Redundant

Exceeding what is necessary or natural; superfluous.

Lecturers and texts are repeated for each student and course.

Anomaly

An inconsistency.

if a new course is created but no students have taken this course, then we cannot enter the information about this course because the use of null values or undefined values in the primary key could cause problem.

- Redundant storage
- insertion/deletion anomaly
- update anomaly

What causes the anomalies?

One process which attempts to remove these undesirable updating anomalies from the relation is called normalization.

$$R1(\underline{STUDENT\#}, \text{S-NAME})$$

$$R2(\underline{COURSE\#}, \text{C-DESCRIPTION})$$

$$R3(\underline{STUDENT\#}, \underline{COURSE\#}, MARK)$$

$$R4(\underline{COURSE\#}, \text{Text})$$

These relations do not have the discussed anomalies.

Underlined attributes indicate a key of the relation.
e.g., attributes $STUDENT\#$ and $COURSE\#$, together, form a key of the relation R3.