

Using Relational Databases and SQL for Social Scientific Research: Theory and Practice

Claude Rubinson
Department of Sociology
University of Arizona
October 19, 2009

What Makes a Database Relational?

- Based on relational algebra (set-theory)
- ACID properties
 - Atomicity
 - Transactions are “all or nothing”
 - Consistency
 - Database is always in a consistent state
 - Isolation
 - One transaction doesn't affect another
 - Durability
 - Transactions persist across system crashes

Relational Databases versus Conventional Datasets

- Many tables rather than one
- Rows are unordered
- Columns are unordered
- Relationships among tables (database schema) represent the structure of the data

Relational Database Design as “Casing”

Relational database design should be thought of as the development of an analytic frame. It requires identifying one's unit of analysis (typically represented by the database itself) and units of observation (represented as relations or “relvars”). The database design will define the constituent components of a case, how those components are related to one another, and how cases are related to each other.

Some Terminology

- Tables (Relations, Relvars)
- Rows (Tuples)
- Columns (Attributes)
- Primary Keys

Res

ResUID	LastName	FirstName
1	Marx	Karl
2	Weber	Max
3	Durkheim	Émile
6	Simmel	Georg

Relational Database Design

- Normalization—the process of eliminating redundancy
- Functional dependencies
 - “If I know one attribute, I can determine another”
 - Singular dependencies: $A \rightarrow B$
 - Multivalued dependencies: $A \twoheadrightarrow B$
 - Defines the structure of the data

Functional Dependencies

Res

<u>ResUID</u>	Researcher
1	Marx, K.
2	Weber, M.
3	Durkheim, E.
6	Simmel, G.
7	Mead, M.
8	Lévi-Strauss, C.

Dis

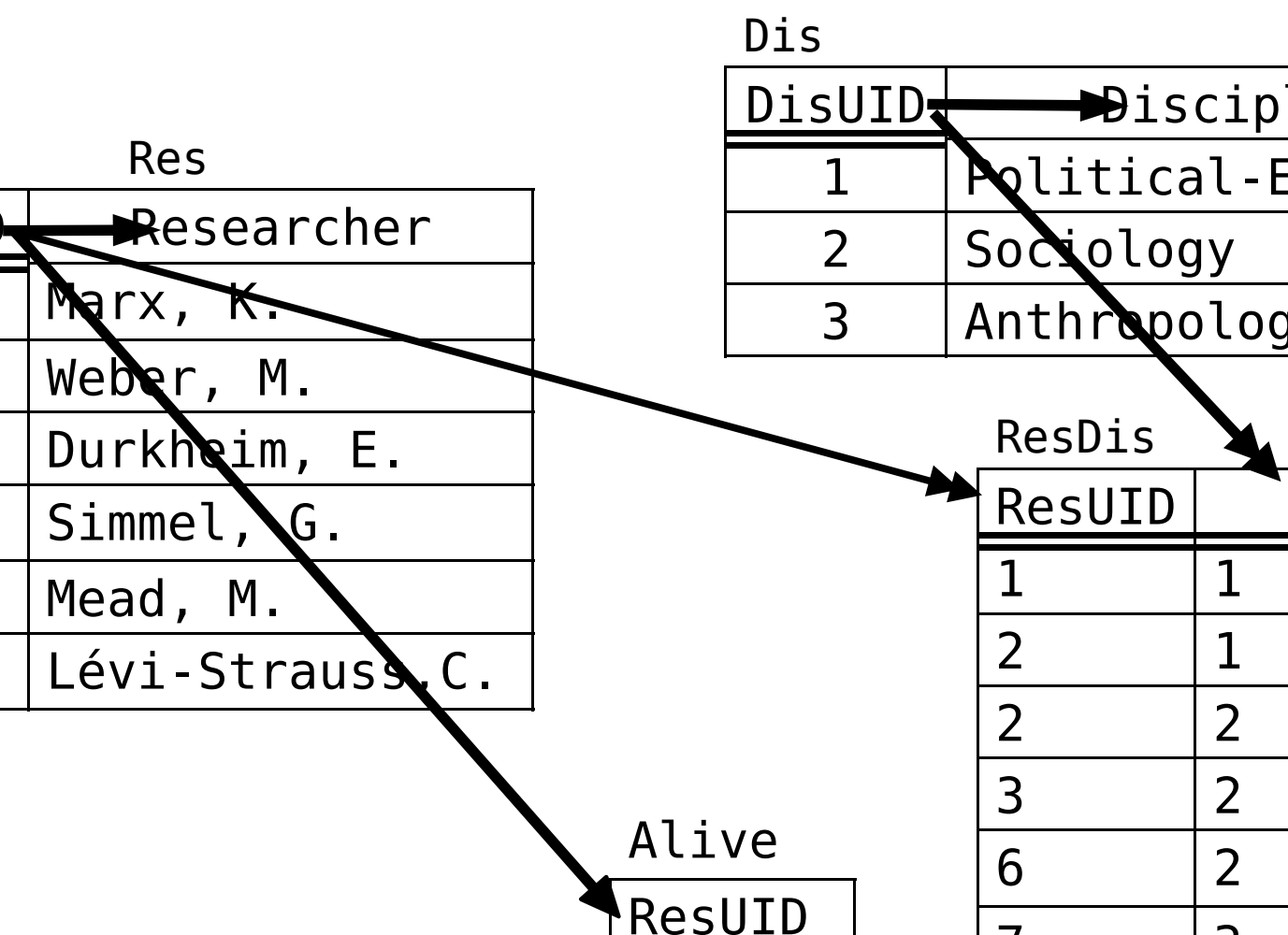
<u>DisUID</u>	Discipline
1	Political-Economy
2	Sociology
3	Anthropology

Alive

<u>ResUID</u>
8

ResDis

<u>ResUID</u>	<u>DisUID</u>
1	1
2	1
2	2
3	2
6	2
7	3
8	3



Normalization

- The process of eliminating redundancy
- Done wrong, the database will be difficult to maintain and information will be difficult or impossible to retrieve. Even worse, incorrect information may be retrieved.
- Fundamentally, a casing process

An Example of Normalization: Non-normalized Database

Teacher	Class	Student
Smith	Econ 101	John
Smith	Econ 101	Mary
Smith	Econ 101	Jane
Smith	Econ 201	Jane
Jones	Art Hist 101	Mary
Jones	Art Hist 101	Smith

An Example of Normalization: Normalized Database

People

PersonUID	Person
1	Smith
2	Jones
3	John
4	Mary
5	Jane
6	James

Classes

ClassUID	Class
1	Econ 101
2	Econ 201
3	Art Hist 101
4	Soc 101

ClassTeacher

ClassUID	PersonUID
1	1
2	1
3	2

ClassStudents

ClassUID	PersonUID
1	3
1	4
1	5
2	5
3	4
3	1

Constraints

- Check constraints
- Unique constraints
- Primary Key constraints
- Foreign Key constraints

The Problem of Missing Data

- The NULL marker
- 2VL versus 3VL
 - Boolean (fuzzy-set) algebra
 - True = 1.0
 - Unknown = 0.5
 - False = 0.0
- **NULL != NULL**

a	b	a AND b	a OR b	NOT a
T	T	T	T	F
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

a	b	a AND b	a OR b	NOT a
T	T	T	T	F
T	U	U	T	F
T	F	F	T	F
U	T	U	T	U
U	U	U	U	U
U	F	F	U	U
F	T	F	T	T
F	U	F	U	T
F	F	F	F	T

What is SQL?

- Structured Query Language
- Pronounced “Ess Que El” or “Sequel”
- Standardized, English-like language for interacting with Relational Database Management Systems (RDBMS)
- Set (technically, “Bag”) based
- Declarative Language (similar to SAS or Stata)

Data Definition Language (DDL)

```
CREATE TABLE Respondents (  
    respUID serial PRIMARY KEY,  
    name text NOT NULL,  
    age integer CHECK (age >= 18),  
    soc integer REFERENCES Occupys (soc)  
);
```

```
DROP TABLE Respondents;
```

Data Definition Language (DDL)

```
ALTER TABLE Respondents  
ADD COLUMN birthday date;
```

```
ALTER TABLE Respondents  
DROP COLUMN age;
```

Data Manipulation Language (DML)

```
INSERT INTO Respondents (name, age)
VALUES ('Smith', 30), ('Jones', 45),
('Clark', 22);
```

```
UPDATE Respondents SET age = 30
WHERE name='Clark';
```

```
DELETE FROM Respondents
WHERE age >= 30;
```

Query Language

```
SELECT * FROM Respondents;
```

```
SELECT name FROM Respondents  
WHERE age >= 25  
ORDER BY name;
```

```
SELECT soc, avg(age),  
FROM Respondents  
WHERE age >=25  
GROUP BY soc;
```

Query Language

```
SELECT Occupys.OccupationName,  
       avg(Respondents.income),  
       count(*) as N  
FROM Respondents, Occupys  
WHERE Respondents.soc=Occupys.soc  
GROUP BY Respondents.soc;  
HAVING count(Respondents.name) > 5;
```

Types of Joins

- Cross Join (Cartesian Product)

```
SELECT *  
FROM table1, table2;
```

- Inner Join

```
SELECT *  
FROM table1, table2  
WHERE table1.joincol=table2.joincol;  
  
SELECT *  
FROM table1 INNER JOIN table2  
ON (table1.joincol=table2.joincol);
```

Types of Joins

- Outer Joins Create NULLs

```
SELECT *  
FROM table1 LEFT JOIN table2  
    ON (table1.joincol=table2.joincol);
```

```
SELECT *  
FROM table1 RIGHT JOIN table2  
    ON (table1.joincol=table2.joincol);
```

```
SELECT *  
FROM table2 FULL JOIN table2  
    ON (table1.joincol=table2.joincol);
```

Review of RDBMSes

- Oracle, MS SQL Server
 - Industry standards
 - Cost prohibitive for academic use
- MS Access, OpenOffice.org Base
 - Graphical
 - User friendly
 - Inexpensive
 - Slow/Not scalable
 - OOo Base can act as frontend for MySQL, PostgreSQL

Review of RDBMSes

- MySQL
 - Open-source
 - Fast
 - Lots of newbie friendly documentation
- PostgreSQL
 - Open-source
 - Strict(er) adherence to relational model
 - High signal:noise ratio on mailing lists, discussion groups, etc.
 - Very thorough documentation

Using RDBMSes for Social Research

- Important aspect of casing
- Basic mathematical and statistical functions are built in
- Advanced statistical functions are available via extensions or can be programmed via popular programming languages

Calling RDBMSes from Statistical Software

- R
 - Extremely strong support for any RDBMS via RODBC
 - RDBMS-specific support via RMySQL, RPostgreSQL, ROracle, etc
 - PL/R embeds R within PostgreSQL
- Stata
 - Supports any RDBMS via odbc
- SAS
 - Supports any RDBMS via odbc driver
 - PROC SQL permits SQL to be used in place of the DATA step; also useful for basic analysis

Recommended Resources

- *SQL for Smarties* by Joe Celko
- *Database Modeling & Design* by Toby J. Teory
- PostgreSQL Online Documentation at <http://www.postgresql.org/docs/>
- *Developing Time-Oriented Database Applications in SQL* by Richard T. Snodgrass
- *An Introduction to Database Systems* by Chris Date
- *Databases in Depth* by Chris Date