Theory of Normalization contd.

Part I

More Normalization

- Multi-Valued Dependencies (MVD)
  - 4th Normal Form
- Join Dependencies (JD)
  - 5th Normal Form
- Inclusion Dependencies
- Template Dependencies
-
  - Domain-Key Normal Form

Still More Problems

PVL (Unnormalized)

PRESIDENT	VP	LOSER
Clinton	Gore	Bush, Perot
Bush	Quayle	Dukakis
Reagan	Bush	Carter, Mondale
Carter	Mondale	Ford
Ford	Rockefeller	
Nixon	Agnew, Ford	Humphrey, Walace, McGovern

PVL (Normalized)

PRESIDENT	VP	LOSER
Clinton	Gore	Bush
Clinton	Gore	Perot
Bush	Quayle	Dukakis
Reagan	Bush	Carter
Reagan	Bush	Mondale
Carter	Mondale	Ford
Ford	Rockefeller	
Nixon	Agnew	Humphrey
Nixon	Agnew	Walace
Nixon	Agnew	McGovern
Nixon	Ford	Humphrey
Nixon	Ford	Walace
Nixon	Ford	McGovern

Better - Why?

PV

PRESIDENT	VP
Clinton	Gore
Bush	Quayle
Reagan	Bush
Carter	Mondale
Ford	Rockefeller
Nixon	Agnew
Nixon	Ford

PL

PRESIDENT	LOSER
Clinton	Bush
Clinton	Perot
Bush	Dukakis
Reagan	Carter
Reagan	Mondale
Carter	Ford
Ford	
Nixon	Humphrey
Nixon	Walace
Nixon	McGovern

- ***If*** tuples
  - <P, V1, L1> and
  - <P, V2, L2> both appear
- ***then*** tuples
  - <P, V1, L2> and
  - <P, V2, L1> both appear.

For example
<Nixon, Agnew, Humphrey>
<Nixon, Ford, Wallace >

Must Have

<Nixon, Agnew, Wallace >
<Nixon, Ford, Humphrey >

Multivalued Dependency (MVD)

PRESIDENT →→ VP

PRESIDENT →→ LOSER

Given R(A, B, C)
R.A →→ R.B
attribute R.B is **multidependent** on attribute R.A

R.A **multidetermines** attribute R.B if the set of B-values matching a given (A-value, C-value) pair in R depends only on the A-value and is independent of the C-value.

As usual A, B, and C may be composite.

Theorem
R.A →→ R.B
if and only if
R.A →→ R.C

We write
R.A →→ R.B | R.C

For example:
President →→ VP | Loser

Note that FD implies MVD.

Theorem.
The relation R(A, B, C) can be *non-loss decomposed* into its two projections R_1(A,B) and R_2(B, C)
iff
the multivalued dependency
A →→ B|C
holds in R.

Fourth Normal Form (4NF)
For every *nontrivial* multivalued dependency A →→ B in F^+, A is a superkey for R.

Theorem.
Any relation can be *non-loss decomposed* into an equivalent collection of *4NF* relations.

Theorem.
If R is in *4NF*, then it is also in *BCNF*.

Yet More Problems
or
5th Normal Form

ACP

AGENT	COMPANY	PRODUCT
Smith	Ford	car
Smith	Ford	truck
Smith	GM	car
Smith	GM	truck
Jones	Ford	car

AGENTS represent COMPANIES
COMPANIES make PRODUCTS
AGENTS sell PRODUCTS

Rule:

If an AGENT sells a certain PRODUCT and that AGENT represents a company making that PRODUCT
then
the AGENT must sell the PRODUCT for that COMPANY.

ACP

AGENT	COMPANY	PRODUCT
Smith	Ford	car
Smith	Ford	truck
Smith	GM	car
Smith	GM	truck
Jones	Ford	car

AC

AGENT	COMPANY
Smith	Ford
Smith	GM
Jones	Ford

AP

AGENT	PRODUCT
Smith	car
Smith	truck
Jones	car

CP

COMPANY	PRODUCT
Ford	car
Ford	truck
GM	car
GM	truck

ACP = AC * AP * CP

Join Dependency (JD)

A relation $R(R_1, R_2, \ldots, R_n)$ satisfies the **join dependency**
$JD(R_1, R_2, \ldots, R_n)$
if it is the join of its projections on
$R_1, R_2, \ldots, R_n$
where $R_1, R_2, \ldots, R_n$ are subsets of the set of attributes of R.

Note that a multi-valued dependency is a special case of a join dependency (n=2).

A relation schema R is in fifth normal form (5NF) or project-join normal form (PJNF) with respect to a set of functional, multi-valued and join dependencies if, for every nontrivial join dependency, JD(R_1, R_2, …, R_n) in F^+, every R_i is a superkey of R.

ACP

AGENT	COMPANY	PRODUCT
Smith	Ford	car
Smith	Ford	truck
Smith	GM	car
Smith	GM	truck
Jones	Ford	car
Brown	Ford	car
Brown	GM	car
Brown	Toyota	car
Brown	Toyota	bus

AC

AGENT	COMPANY
Smith	Ford
Smith	GM
Jones	Ford
Brown	Ford
Brown	GM
Brown	Toyota

AP

AGENT	PRODUCT
Smith	car
Smith	truck
Jones	car
Brown	car
Brown	bus

CP

COMPANY	PRODUCT
Ford	car
Ford	truck
GM	car
GM	truck
Toyota	car
Toyota	bus

AC*AP

AGENT	COMPANY	PRODUCT
Smith	Ford	car
Smith	Ford	truck
Smith	GM	car
Smith	GM	truck
Jones	Ford	car
Brown	Ford	bus
Brown	Ford	car
Brown	GM	bus
Brown	GM	car
Brown	Toyota	car
Brown	Toyota	bus

AP*CP

AGENT	COMPANY	PRODUCT
Smith	Ford	car
Smith	Ford	truck
Smith	GM	car
Smith	GM	truck
Smith	Toyota	car
Jones	Ford	car
Jones	GM	car
Jones	Toyota	car
Brown	Ford	car
Brown	GM	car
Brown	Toyota	car
Brown	Toyota	bus

AC*CP	AGENT	COMPANY	PRODUCT
	Smith	Ford	car
	Smith	Ford	truck
	Smith	GM	car
	Smith	GM	truck
	Jones	Ford	car
	Jones	Ford	truck
	Brown	Ford	car
	Brown	Ford	truck
	Brown	GM	car
	Brown	GM	truck
	Brown	Toyota	car

Inclusion Dependencies

- Interrelational constraints
- Generalization of
  - referential integrity constraints and
  - class/subclass constraints
- No normal forms have been developed.
- See Section 15.4.

Other Dependencies and Normal Forms

- Template Dependencies
  - See Section 13.5.1
- Domain-Key Normal Form
  - See Section 13.5.2
- Neither of practical use.

Database Design Process

Part II

Database Design Process

- Requirements collection and analysis
- Conceptual DB design
- Choice of DBMS
- Data Model Mapping
- Physical Database Design
- Database System Implementation and Tuning

Requirements Collection and Analysis

- Identify major application areas and user groups.
- Study and analyze applications and uses.
- Identify the planned use of information, types of transactions, and frequency of use [make common case fast]
- Evaluate real customers

Conceptual DB Design

- Conceptual Schema Design
  - Identify basic structure, semantics, interrelationships and constraints, independent of DBMS.
  - Expressive, Simple, Graphical, formal model desired. (EER)
- Conceptual Transaction Design
  - Identify their Input/Output and Functional Behavior
    - Update oriented, read-only, mixed transactions (TPC)

Conceptual Schema Design

- Application/User Centric
  - One-Shot approach
    - Combine all users/applications and get one common design.
  - View integrated approach
    - Start separately and slowly integrate
    - Have to handle conflicts
      - Name, Type, Domain, Constraints
- Strategies
  - Top Down (specialization)
  - Bottom Up (generalization)
  - Inside-Out (restricted bottom-up)
  - Mixed (mixture)

Conceptual Transaction Design

- Analyze database queries and transactions for files that will be accessed, attributes involved & files updated.
- Analyze expected frequency and invocation of transactions
- Analyze time constraints of queries
- Analyze expected frequecies of update operations
- Analyze the uniqueness constraints on attributes
  - Can improve query optimization performance

Choice of DBMS

- Issues
  - Acquisition and Maintenance Costs
  - Creation and Conversion costs
  - Personnel and Training costs
  - Operating Costs
  - Data-centered Issues
    - Data complexity
    - Performance Requirements
      - Data volume and influx
      - Transactional Properties

Data Model Mapping

- System Independent Mapping
  - Already done in step 2
- Tailoring schemas to specific Mapping.
  - May have to adjust schemas to special features/advantages of the specific DBMS.
    - Crucial for performance

Physical Database Design

- Goal to choose specific storage and access structures to achive good performance. Large choice of indexing schemes, clustering schemes & hashing algorithms.
  - Response time
    - Depends also on transaction characteristics and system load.
  - Space Utilization
  - Estimated Transaction Throughput Requirements

Implementation & Tuning

- Goals
  - To make applications run faster
  - Lower response times of queries/transactions.
  - Improve throughput of transactions
- Use statistics
  - Storage, I/O, Q/T, Index
- Tuning Indexes
  - Use specialized trees for certain indexes
  - Rebuild/re-order indices based on statistics.
- Tuning DB design
  - Consider Horizontal/Vertical Partitioning
  - Reduce normalization level if required
    - Say 2 attributes in different relations are often used together. Increase redundancy for performance.
- Tuning Queries
  - Query optimizations (re-ordering etc.)