The Rise and Fall and Rise of Dependency Theory

Part I: Rise and Fall

Moshe Y. Vardi

Rice University
Primary Keys

E.F. Codd, *A Relational Model of Data for Large Shared Data Banks*, CACM, June 1970

“Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). ... In Section 1, a model based on \(n\)-ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. ... Normally, one domain (or combination of domains) of a given relation has values which uniquely identify each element \((n\text{-tuple})\) of that relation. Such a domain (or combination) is called *primary key*.”
Figure 1: Edgar Frank “Ted” Codd, 1923–2003
Functional Dependencies

E.F. Codd, *Further Normalization of The Data Base Relational Model, IBM Research Report, 1971*

**Definition**: A relation \( r \) on a scheme \( R \) satisfies the *functional dependency* \( X \rightarrow Y \), where \( X, Y \subseteq R \), if, for all \( t_1, t_2 \in r \), we have that \( t_1[X] = t_2[X] \) implies \( t_1[Y] = t_2[Y] \);

**Motivation**: *normalization*

- **Scheme**: Emp, Dept, Mgr
- **FDs**: Emp \( \rightarrow \) Dept, Dept \( \rightarrow \) Mgr
- **Violation of 3rd Normal Form**
- **Decompose to** Emp, Dept and Dept, Mgr.
Theory of Functional Dependencies

The Emergence of Database Theory:

- Codd, 1971: FDs and lossless joins
- Delobel and Casey, 1973: properties of FDs
- Armstrong, 1974: axiomatization of FDs
- Bernstein, 1976: synthesis of 3NF database scheme from FDs
- Fagin, 1976: FDs and propositional logic
- Beeri and Bernstein, 1979: linear-time algorithm for implication of FDs
Dependency Implication

**Definition:** Let $R$ be a relation scheme. A set $\Sigma$ of dependencies implies a dependency $\sigma$, if, for every relation $r$ on $R$, if $r$ satisfies all dependencies in $\Sigma$, then it also satisfies $\sigma$.

**The Implication Problem:** Given set $\Sigma$ of dependencies and a dependency $\sigma$, does $\Sigma$ implies $\sigma$?
Multivalued Dependencies


**Definition:** Let $R$ be a relation scheme, $X, Y \subseteq R$, and $Z = R - (X \cup Y)$. A relation $r$ on $R$ satisfies the *multivalued dependency* (MVD) $X \rightarrow Y$ if, for all $t_1, t_2 \in r$, we have that $t_1[X] = t_2[X]$ implies that there is $t \in R$ with $t[X] = t_1[X] = t_2[X]$, $t[Y] = t_1[Y]$, and $t[Z] = t_2[Z]$. 
MVDs and Lossless Decomposition

Let $R$ be a relation scheme, $X, Y \subseteq R$, and $Z = R - (X \cup Y)$. A relation $r$ on $R$ satisfies $X \rightarrow Y$ iff $r = \pi_{XY}(r) \Join \pi_{XZ}(r)$

Motivation: normalization

- **Scheme**: Emp, Project, Hobby
- **MVD**: Emp $\rightarrow$ Project
- **Violation of 4th Normal Form** (Fagin)
- **Decompose to** Emp, Project and Emp, Hobby.

Beeri, Fagin & Howard, SIGMOD’77: axiomatization and polytime implication problem for FDs and MVDs.

**Example**: $X \rightarrow Y$ implies $X \rightarrow Y$
Join Dependencies


**Definition**: Let \( R \) be a relation scheme, and \( R_1, \ldots, R_k \) such that \( R = R_1 \cup \ldots \cup R_k \). A relation \( r \) on \( R \) satisfies the *join dependency* \( \star[R_1, \ldots, R_k] \) if \( r = \pi_{R_1}(r) \Join \ldots \Join \pi_{R_k}(r) \).

**Motivation**: normalization

- 5th Normal Form [Fagin, SIGMOD’79]
The Chase


To check if $\Sigma$ does *not* implies $\sigma$, search for a candidate counterexample:

- Propose a relation $r$ that violates $\sigma$.
- “Massage” $r$ using the dependencies in $\Sigma$ to ensure that all dependencies in $\Sigma$ are satisfied.
- Check that final $r$ still violates $\sigma$.

**Computational Complexity:**

- In EXPTIME
- NP-hard [Beeri&Vardi, 1980]
Figure 2: Alberto O. Mendelzon, 1951–2005
Embedded Dependencies


**Definition**: Let $R$ be a relation scheme, and $X, Y, Z \subseteq R$. A relation $r$ on $R$ satisfies the *embedded multivalued dependency (EMVD)* $X \rightarrow Y|Z$ if $\pi_{XYZ}(r) = \pi_{XY}(r) \Join \pi_{XZ}(r)$

**Definition**: Let $R$ be a relation scheme, and $R_1, \ldots, R_k$ such that $R_1 \cup \ldots \cup R_k = R' \subseteq R$. A relation $r$ on $R$ satisfies the *embedded join dependency (EJD)* $\star[R_1, \ldots, R_k]$ if $\pi_{R'}(r) = \pi_{R_1}(r) \Join \ldots \Join \pi_{R_k}(r)$.

**Major Difficulty**: Chasing with embedded dependencies does not terminate.
Nov. 1978: Mr. Vardi Goes to Graduate School

- Vardi, June 1979: “Anything left to do in dependency theory?”
- Beeri: “Why don’t you work on the implication problem for embedded dependencies?”
- Vardi, February 1980: “I learned today about undecidability. Perhaps the implication problem for EMVDs is undecidable.”
- Beeri: “We are not doing computability theory. This is database theory. Everything ought to be decidable!”
Dependencies and FOL

J.M. Nicolas, SIGMOD’78: Dependencies can be expressed in *first-order logic*.

**Scheme:** A,B,C,D

- **FD:** A → B
  \[ (\forall x, x_2, x_3, x_4, y_2, y_3, y_4)((r(x, x_2, x_3, x_4) \land r(x, y_2, y_3, y_4)) \rightarrow x_2 = y_2) \]

- **MVD:** A → B
  \[ (\forall x, x_2, x_3, x_4, y_2, y_3, y_4)((r(x, x_2, x_3, x_4) \land r(x, y_2, y_3, y_4)) \rightarrow r(x, x_2, y_3, y_4)) \]

- **EMVD:** A → B—C
  \[ (\forall x, x_2, x_3, x_4, y_2, y_3, y_4)((r(x, x_2, x_3, x_4) \land r(x, y_2, y_3, y_4)) \rightarrow (\exists z)r(x, x_2, y_3, z)) \]
Generalizing Dependencies

Fagin, STOC’80:  *Embedded Implicational Dependencies*:
\[(\forall x_1, x_2, \ldots)(A_1 \land A_2 \ldots) \rightarrow (\exists z_1, z_2, \ldots)(B_1 \land B_2 \ldots))\]

- \(A_i\)'s: relational formulas
- \(B_i\)'s: atomic formulas (relational or equality)
- \(x_i\)'s: *guarded* – occur on right only if they occur on left

Beeri&Vardi, 1980:

- *Tuple-generating dependencies*: \(B_i\)'s – relational formulas
- *Equality-generating dependencies*: \(B_i\)'s – equality formulas
Full and Embedded Dependencies

*Full dependencies*: no existential variables
*Embedded dependencies*: existential variables

Beeri & Vardi, 1980: chase for tgds and egds

- Generally, terminates only for full dependencies.
- Termination for embedded dependencies in special cases.
A Query-Based View of Dependencies

Conjunctive Queries: \((\exists z_1, z_2, \ldots)(B_1 \land B_2 \ldots)\) [Chandra&Merlin, STOC’76]

- Incredibly rich theory!

Definition: A database \(D\) satisfies a conjunctive-query-containment dependency (CQCD) \(Q_1 \subseteq Q_2\), for conjunctive queries \(Q_1\) and \(Q_2\), if \(Q_1(D) \subseteq Q_2(D)\).

Yannakakis&Papadimitriou, FOCS’80: EIDs are equivalent to CQCDs.
The Implication Problem

[Beeri & Vardi, ICALP’81, Chandra, Lewis & Makowsky, STOC’81]:

- Implication of Embedded Dependencies: undecidable
- Implication of Full Dependencies: EXPTIME-complete

Significance:

- First undecidable “database-theoretic” problem.
- First intractable “database-theoretic” problem.

But: Have we generalized too much?
Template Dependencies

Sadri&Ullman, STOC’80:
\((\forall x_1, x_2, \ldots)((A_1 \land A_2 \ldots) \rightarrow (\exists z_1, z_2, \ldots) B))\)

- \(A_i\)'s: relational formulas
- \(B\): relational formulas
- \(x_i\)'s: guarded
- \textit{typedness}: a variable cannot occur in two distinct columns

Undecidability of implication:

- Gurevich&Lewis, PODS'82: Bounded number of atomic formulas
- Vardi, PODS’82: Bounded arity
“Practical” Dependencies

Definition: An inclusion dependency is a CQCD with respect to simple projective queries $\pi_{X_1}$ and $\pi_{X_2}$. [Casanova, Fagin&Papadimitrou, PODS’82]

- Significance: Captures referential integrity

Implication:

- Implication of IDs: PSPACE-complete [Casanova, Fagin&Papadimitrou, PODS’82]
- Implication of IDs and FDs: undecidable [Chandra&Vardi, SICOMP’85, Mitchell, I&C’83]
Rise and fall

Brief History:
• Heyday: early PODSes (e.g., 8 papers in PODS’82)
• Decline and disdain: late 1980s (e.g., 0 papers in PODS’88)

Speculative Explanation:
• Hard to build useful theory for undecidable problems
• Database design proved to be a modeling activity (ER, Chen, 1976)
• Attention shifted to query processing (bread and butter of databases)
• Lack of interest in integrity constraints by SIGMOD community (CHECK constraint added to SQL only in 1989)

But:
• Subsequent rise: Fagin
• Impact on Datalog research: Ullman
What about EMVDs?