Database Management Systems

Course Project

Relational Database Schema Design for Uncertain Database

Group Details

Rohit K

Neerab P

Rahul U

Index

- ## Abstract:

We introduce uncertainty to relational database schema design. Here, Uncertainty is modeled qualitatively by assigning to tuples a degree of possibility with which they occur in a relation, and assigning to functional dependencies a degree of certainty which reflects to which tuples they apply. Further, A design theory is developed for possibilistic functional dependencies, including efficient axiomatic and algorithmic characterizations of their implication problem. Naturally, the possibility degrees of tuples result in a scale of different degrees of data redundancy, caused by the possibilistic functional dependencies.

Scaled versions of the classical syntactic Boyce-Codd and Third Normal Forms are established and semantically justified in terms of avoiding data redundancy of different degrees. Therefore, Possibilistic functional dependencies do not just enable designers to control the level of data integrity targeted but also to balance the classical trade-off between query and update efficiency.

- ## Title / Problem Statement:

Relational Database Schema Design for Uncertain Data.

- ## Introduction:

Relational databases were developed for applications with certain data, such as accounting, inventory, and payroll. Modern applications such as information extraction, data integration and cleaning, require techniques for uncertain data.

Two trends can be observed in this research field:
Queries are the dominant focus point,
and uncertainty is mostly modeled quantitatively as probabilistic data.

In classical schema design, update inefficiencies are avoided by removing data redundancy caused by functional dependencies (FDs). Intuitively, if data is uncertain, then so is any redundancy that results from this data. The more possible it is for data redundancy to occur, the more FDs can cause this redundancy, and the harder the normalization effort will be to remove that redundancy. In other words, the removal of

redundancy from less possible data requires normalization for a smaller number of FDs. This is great news as data that are less possible are intuitively subject to more updates. Therefore, most of the frequent updates can be supported efficiently with less normalization effort. In turn, less normalization results in better query efficiency. Our findings will establish a practical and precise framework that enables database designers to take full advantage of this intuitive impact of uncertainty.

Therefore, the impact we expect uncertainty to have on schema design is that the more frequent updates are likely to be processed efficiently with less normalization effort. This means that more queries can be processed efficiently, as less joins are required. Uncertainty results in a greater variety of normalized schema designs from which the best match for the target workload of updates and queries can be chosen. In other words, uncertainty can be understood as an effective mechanism to better control the classical trade-off between update and query efficiency.

- Contributions:

Our contributions are as follows:

1. We formalize uncertainty by assigning degrees of possibilities to tuples in a database.
2. We define possibilistic functional dependencies (PFDs) as classical functional dependencies (FDs) with a degree of certainty, derived from the possibility degree of the smallest possible world in which it is violated.
3. We establish a full design theory for PFDs, including axiomatic and algorithmic characterizations of their associated implication problem, as well as algorithms for computing covers and the maximum certainty degrees by which PFDs are implied.
4. We apply the design theory to establish a new normalization framework, including scaled versions of Boyce-Codd (BCNF) and Third normal forms (3NF), their semantic justification in terms of dependency-preservation and removal of data redundancy, as well as normalization algorithms.

Our techniques provide database designers with a full scale of normalized schemata, from which better-informed selections of the final schema can be made.

We envision two main use-case scenarios:

i) An organization that is primarily focused on data integrity may choose the FDs that apply to the target degree of certainty. In this case, our algorithms compute the normal forms that exactly meet that target.

ii) For an organization that is primarily focused on efficiently processing its workload in terms of queries and updates, a database designer can use our techniques to derive the normal forms that best match the workload. In that case, it is also clear what level of data integrity is achieved. Therefore, our findings show that certainty degrees of FDs provide an efficient mechanism to not only control the degree of data integrity targeted, but also to derive designs that match the workload of the target database by balancing the trade-off between query and update efficiency. The selection of different certainty degrees βi results in different decomposition's, for example, βi-3NF for $i = 1; : : : ; k$. In particular, our findings can be used to also justify classically de-normalized schema in terms of permitting data redundancy which is caused by FDs whose degree of certainty is smaller than that targeted.

Extensive experiments with our algorithms confirm their efficiency in practice, and provide new insight into classical normalization trade-offs. For example, we provide first empirical evidence that there almost always is a fair chance that 3NF synthesis results in an optimal decomposition, defined as a lossless, dependency-preserving BCNF decomposition, while there is hardly ever a reasonable chance that the BCNF decomposition algorithm results in a dependency-preserving decomposition.

● Methodology:

❖ PFDS:

We introduce possibilistic FDs (pFDs) as classical FDs with a degree of certainty. In the same way FDs are fundamental to relational database design, pFDs will play a fundamental role to schema design for uncertain data. The certainty with which an FD holds in an uncertain relation corresponds to the possibility of the smallest possible world in which the FD is violated. Therefore, similar to a scale S of possibility degrees for tuples we use a scale ST of certainty degrees (c-degrees) for FDs. We commonly use sub-scripted versions of the Greek letter β to denote c-degrees associated with FDs. Formally, the correspondence between p-degrees in S and the c-degrees in ST can be defined by the mapping αi --> $\beta k+2-i$ for $i = 1,...,k + 1$. Formally then, the certainty $Cr(X$ --> $Y)$ with which the FD $X ! Y$ holds in the uncertain relation r is the top degree $\beta 1$ whenever $X ! Y$ is satisfied by rk, or otherwise the minimum amongst the c-degrees $\beta(k+2-i)$ that correspond to possible worlds ri in which X --> Y is violated, that is,
$Cr(X$ --> $Y)$
=

β 1	*, if* r_k *satisfies* X --> Y

min ($\frac{}{B(k+2-i)}$)	, *otherwise.*

Definition 1. *A* possibilistic FD *(pFD) over a p-schema* (*R; S*) *is an expression* (*X ! Y;β*) *where X; Y ⊆ R and β belongs to ST . A p-relation* (*r; Possr*) *over* (*R; S*) *is said to* satisfy *the pFD* (*X ! Y; β*) *if and only if Cr*(*X --> Y*) ≥ *β.*

◇ Scaled design theory:

■ β-Cuts :

For a set Σ of pFDs on some p-schema (*R; S*) and c-degree *β belongs to ST* where *β* > *βk*+1, let

$$\Sigma_\beta = \{X \to Y \mid (X \to Y, \beta') \in \Sigma \text{ and } \beta' \geq \beta\}$$

be the *β-cut* of Σ.

The major strength of our framework is engraved in the following theorem:

THEOREM 1. *Let* $\Sigma \cup \{(X \to Y, \beta)\}$ *be a set of pFDs over a p-schema* $(R, \mathcal{S})$ *where* $\beta > \beta_{k+1}$. *Then* $\Sigma \models (X \to Y, \beta)$ *if and only if* $\Sigma_\beta \models X \to Y$. □

◇ Scaled Amstrong's axioms:

$$\frac{}{(XY \to Y, \beta)}\ (\text{reflexivity}, \mathcal{R}) \qquad \frac{(X \to Y, \beta)}{(X \to XY, \beta)}\ (\text{extension}, \mathcal{E})$$

$$\frac{(X \to Y, \beta) \quad (Y \to Z, \beta)}{(X \to Z, \beta)}\ (\text{transitivity}, \mathcal{T})$$

$$\frac{}{(X \to Y, \beta_{k+1})}\ (\text{bottom}, \mathcal{B}) \qquad \frac{(X \to Y, \beta)}{(X \to Y, \beta')}\ \beta' < \beta\ (\text{weakening}, \mathcal{W})$$

Table 2: Axiomatization $\mathfrak{P} = \{\mathcal{R}, \mathcal{E}, \mathcal{T}, \mathcal{B}, \mathcal{W}\}$ of pFDs

This sound and complete axiomatization can be used to solve the implication problem.

◇ Scaled Normal Forms:

In p-relations, different data can occur with different pdegrees. Therefore, data redundancy may occur with different p-degrees, too. Intuitively, the smaller the p-degree for which data redundancy is to be eliminated, the smaller the normalization effort will be. We will introduce notions of
data redundancy that are tailored to the p-degree of tuples in which they occur. This results in a whole range of semantic normal forms by which data redundancy of growing p-degrees are eliminated. We then characterize each of these semantic normal forms by a corresponding syntactic normal form, and establish strong correspondences with BCNF and 3NF in relational databases.

◇ Scaled Data Redundancy:

DEFINITION 3. *A p-schema $(R, \mathcal{S})$ is in α_i-Redundancy-Free Normal Form (α_i-RFNF) for a set Σ of pFDs over $(R, \mathcal{S})$ iff there do not exist a p-relation $(r, Poss_r)$ over $(R, \mathcal{S})$ that satisfies Σ, an attribute $A \in R$, and a tuple $t \in r_i$ such that $t(A)$ is α_i-redundant.* □

THEOREM 4. *$(R, \mathcal{S})$ is in α_i-RFNF for Σ if and only if R is in RFNF for Σ_{α_i}.*

◇ Scaled Normalization:

■ Scaled BCNF Decomposition:

THEOREM 10. *Let $(X \to Y, \beta_i)$ be a pFD that satisfies the p-relation $(r, Poss_r)$ over the p-schema $(R, \mathcal{S})$. Then $r_{k+1-i} = r_{k+1-i}[XY] \bowtie r_{k+1-i}[X(R-Y)]$, that is, the possible world r_{k+1-i} of r is the lossless join of its projections on XY and $X(R-Y)$.*

Therefore, an α_{k+1-i}-lossless BCNF decomposition for a pFD set Σ can be obtained by performing a classical lossless BCNF decomposition for the β_i-cut Σ_{β_i} of Σ. This suggests a simple lossless BCNF decomposition strategy.

PROBLEM: Scaled BCNF Decomposition	
INPUT:	P-Relation Schema $(R, \mathcal{S})$ Set Σ of pFDs over $(R, \mathcal{S})$ Certainty degree $\beta_i \in \mathcal{S}^T$
OUTPUT:	α_{k+1-i}-lossless BCNF decomposition of $(R, \mathcal{S})$ for Σ
METHOD:	Perform a lossless BCNF decomposition of R for Σ_{β_i}

Theorem 7 *(R; S) is in β-BCNF with respect to a set Σ if and only if R is in BCNF with respect to Σβ.*

Theorem 10 (*R; S*) *is in β-3NF with respect to a set Σ if and only if R is in 3NF with respect to Σβ.*

■ Scaled 3NF Synthesis:

DEFINITION 8. *A* β_i*-dependency-preserving,* α_{k+1-i}*-lossless 3NF decomposition of a p-schema* $(R, \{\alpha_1, \ldots, \alpha_{k+1}\})$ *for the pFD set* Σ *is a dependency-preserving, lossless 3NF decomposition of* R *for* Σ_{β_i}. □

Due to Theorem 10 a β_i-dependency-preserving, α_{k+1-i}-lossless 3NF synthesis for a pFD set Σ can simply be obtained by performing a classical dependency-preserving lossless 3NF synthesis for the β_i-cut Σ_{β_i} of Σ.

PROBLEM: Scaled 3NF Synthesis	
INPUT:	P-relation schema $(R, \mathcal{S})$ Set Σ of pFDs over $(R, \mathcal{S})$ Certainty degree $\beta_i \in \mathcal{S}^T$
OUTPUT:	β_i-dependency-preserving, α_{k+1-i}-lossless 3NF decomposition of $(R, \mathcal{S})$ for Σ
METHOD:	Perform a dependency-preserving, lossless 3NF synthesis of R for Σ_{β_i}

3NF synthesis guarantees dependency-preservation, but cannot guarantee the elimination of all data redundancy in terms of FDs. Recently, Third normal form was shown to pay the smallest possible price, in terms of data redundancy, for achieving dependency-preservation. We will now equip our schema design framework for uncertain data with
an appropriate 3NF synthesis strategy .

- Algorithms:

Algorithm 1 Closure Computation

```
1: procedure CLOSURE(R, X, Σ_β)
2:     Closure ← X;
3:     FDList ← List of X → Y ∈ Σ_β;
4:     repeat
5:         OldClosure ← Closure;
6:         Remove Closure from LHS of FDs in FDList;
7:         for all ∅ → Y in FDList do
8:             Closure ← Closure ∪ Y;
9:             FDList ← FDList − {∅ → Y};
10:        end for
11:    until Closure=OldClosure or FDList=[ ]
12:    return(Closure);
13: end procedure
```

Algorithm 2 Non-redundant Cover

```
1: procedure NR-COVER(R, Σ)
2:     for all σ = (X → A, β) ∈ Σ do
3:         if A ∈ CLOSURE(R, X, (Σ − {σ})_β) then
4:             Σ ← Σ − {σ};
5:         end if
6:     end for
7:     return(Σ);
8: end procedure
```

Algorithm 3 Canonical Cover

```
1: procedure CAN-COVER(R, Σ)
2:     for all σ = (X → A, β) ∈ Σ do
3:         Z ← X;
4:         for all B ∈ Z do
5:             if A ∈ CLOSURE(R, Z − {B}, Σ_β) then
6:                 Z ← Z − {B};
7:             end if
8:         end for
9:         Σ ← (Σ − {σ}) ∪ {(Z → A, β)};
10:    end for
11:    Σ ← NR-COVER(R, Σ);
12:    while (X → Y, β), (X → Z, β) ∈ Σ do
13:        Σ ← Σ − {(X → Y, β), (X → Z, β)};
14:        Σ ← Σ ∪ {(X → YZ, β)};
15:    end while
16:    return(Σ);
17: end procedure
```

Input:

- Relational Schema(R) , Possibility Degree (or) Certainty degree Scale(S), set of possibilistic functional dependencies.

For β-BCNF :

- For each $beta_i$(pfds of one world) :
- Compute canonical cover of pfds (R_1)
    - Define D (empty set of relations), LS (empty set of pfds)
    - Check whether pfds in cover is in bcnf form or not
    - For each pfd in Σ:
        - If (violates bcnf) :
            - Append given pfd to LS.
    - do
        - For every pfd in LS:
            - Perform an (α k+1-i)lossless BCNF decomposition for a pFD set Σ i.e, by performing a classical lossless BCNF decomposition for the βi-cut Σβi of Σ.
            - Append obtained decomposition's to D,remove given pfd from LS and recursively check if D is in bcnf
    - While (LS is not empty)
    - Remove pfds which are there in LS from R_1.
    - Output (R_1 + D(Set of decompositions))

For β-3NF :

- For each $beta_i$(pfds of one world) :
    - Compute canonical cover of pfds (R_1)
    - Define D (empty set of relations), LS (empty set of pfds)
    - Check whether pfds in cover is in 3-NF form or not
    - For each pfd in Σ:
        - If (violates 3-NF) :
            - Append given pfd to LS.
    - Do
        - For every pfd in LS:
            - Perform (α k+1-i)-lossless, (βi)-dependency-preserving 3NF synthesis for a pFD set Σ i.e, by performing a classical lossless 3NF synthesis for Σβi.
            - Append obtained decomposition's to D,remove given pfd from LS and recursively check if D is in 3nf
    - While (LS is not empty)
    - Remove pfds which are there in LS from R_1.
    - Output (R_1 + D).

- Results:

	α_{k+1-i}-lossless	Free from α_{k+1-i} data redundancy	β_i-dependency-preserving
β_i-BCNF	✔	✔	
β_i-3NF	✔		✔

Table 3: Achievements of β-BCNF and β-3NF

- Conclusion / Future Work:

We developed a full framework to design relational schemata for applications with uncertain data. The fact that FDs are closed downwards, i.e. hold on every sub-relation of a relation that satisfies them, enabled us to introduce different levels of data redundancy. We established a customized segmentation of the well-known Boyce-Codd and Third normal forms, extending all of their properties to each segment, Table 3.

Certainty degrees of FDs provide a mechanism for organizations to control the levels of data integrity, data losslessness, query efficiency, and update efficiency.

Several problems warrant future research. As relational schema design for certain data depends on the identification of FDs that are meaningful for the given application, schema design for uncertain data depends on the identification of meaningful pFDs. This motivates research about Armstrong relations in the possibilistic setting. Further normal forms should be customized to their use for uncertain data, including 4NF as well as Inclusion Dependency normal form. Recently, the relational normalization toolbox has been extended to incomplete relations, It would certainly be interesting to develop a normalization framework for possibilistic incomplete relations.

- References:

[1] L. Antova, C. Koch, and D. Olteanu. 10(106) worlds and beyond: efficient representation and processing of incomplete information. *VLDB J.*, 18(5):1021{1040, 2009.

[2] M. Arenas and L. Libkin. An information-theoretic approach to normal forms for relational and XML data. *J. ACM*, 52(2):246{283, 2005.

[3] W. W. Armstrong. Dependency structures of data base relationships. In *IFIP Congress*, pages 580{583, 1974.

[4] O. Benjelloun, A. D. Sarma, A. Y. Halevy, M. Theobald, and J. Widom. Databases with uncertainty and lineage. *VLDB J.*, 17(2):243{264, 2008.

[5] O. Benjelloun, A. D. Sarma, C. Hayworth, and J. Widom. An introduction to ULDBs and the Trio system. *IEEE Data Eng. Bull.*, 29(1):5{16, 2006.

[6] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377{387, 1970.

[7] E. F. Codd. Further normalization of the database relational model. In *Courant Computer Science Symposia 6: Data Base Systems*, pages 33{64, 1972.

[8] A. D. Sarma, O. Benjelloun, A. Y. Halevy, S. U. Nabar, and J. Widom. Representing uncertain data: models, properties, and algorithms. *VLDB J.*, 18(5):989{1019, 2009.

[9] A. D. Sarma, J. D. Ullman, and J. Widom. Schema design for uncertain databases. In *Proceedings of the 3rd Alberto Mendelzon International Workshop on Foundations of Data Management*, volume 450 of *CEUR Workshop Proceedings*, 2009.

[10] D. Suciu, D. Olteanu, C. R´e, and C. Koch. *Probabilistic Databases*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[11] D.-M. Tsou and P. C. Fischer. Decomposition of a relation scheme into Boyce-Codd normal form. *SIGACT News*, 14(3):23{29, 1982.