Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

Efficient Management Of Data In Uncertain And Probabilistic Databases

Mr. Vivek V. Kheradkar[1] , Dr. S. K. Shirgave[2]

[1]Assistant Professor,Department of Computer Science and Engineering, D.K.T.E's Soceity Textile & Engineering Institute, Ichalkarnji, Maharashtra, India.

[2]Professor,Department of Computer Science and Engineering, D.K.T.E's Soceity Textile & Engineering Institute, Ichalkarnji,Maharashtra, India.

Abstract – Databases are used to store information related to entities that are part of the real world. The Database management system is used to store, retrieve, and update the information. There are two categories of databases, namely certain and Uncertain. The certain databases are those databases where if the value of a particular attribute is unknown, then it will be stored as Null value. The uncertain database is associated with the following: Data, Uncertainty about the value, and presence. Uncertain databases are those where if a definite value for a particular attribute is not known then it will be stored with the help of a confidence value associated with alternatives. There are real-world applications where there is a need to store uncertain values of their attributes. There are many uncertain and probabilistic database management systems like TRIO, ULDB. Each defines its own management system for uncertain data and has its own advantages for particular applications and has some drawbacks. The main focus of this paper is towards a new, efficient way of managing uncertain and probabilistic databases. It shows the way of handling different types of queries on uncertain databases. It provides an efficient way of getting better performance in the case of handling simple queries, joins queries, multiple aggregations, multiple queries, and subqueries..

Keywords – Certain Data, Uncertain database, probabilistic database, lineage.

1. Introduction

In today's world, data is either uncertain or certain. Most of the data are uncertain data. Uncertainty in data is present due to data collected from resources being unreliable, unsure about its presence and confidence value. Knowledge about data collected in the world cannot indicate its presence and confidence value correctly. This uncertainty about the presence and value might result from unreliable information resources which take faulty instrument data or input form resources which might take filled data incorrectly. The success of any software system mainly depends on data on which system will work. Due to this uncertainty in the data, this result in the error in the system such as fault generation, wrong result, transmission errors, and unexpected result, delay in processing transaction activities and imperfections of the software. Therefore, the uncertainty in data is the most important thing which cannot be avoided and which becomes a threat in the achievement of the success of software systems. The unavoidable information-gathering methods that require an appropriate

procedure, estimation, or judgment. There are many domains such as data integration, scientific and experimental data, data automatic extraction from textual data, data from the physical sources that have found increasing use of large amounts of uncertain data. In traditional and conventional databases, including relational databases, it allows correct, fixed, deterministic, and certain data. So, there is no option and provision for handling uncertain data in traditional relational databases. Also, no option for ignorance of such uncertainty generated in reliable applications. To deal with such an application, handling uncertain data becomes a must and important thing.

The broad category of current research in handling uncertain data can be classified into two main areas: The first is the organization of uncertain data. The main task in this area is to handle uncertain data, store it with a new structure, and use the proper organization of uncertain data for different applications. The second area is the management of uncertain data where traditional data management techniques are adopted to deal with uncertain data, such as query processing based on the insert, update, delete, retrieval from single as well as join processing, aggregate query processing, indexing, and data integration.

The main aim of this paper has three parts: the first is surveying on different existing uncertain data management, its advantages, and issues in handling uncertain databases. The second is an introduction to the main model and concepts of handling uncertainty in the database. The third is how the system will handle different types of queries such as query processing based on the insert, update, delete, retrieval from a single as well as join processing, aggregate query processing, indexing, and data integration of this new uncertain database management system.

To complete the study, we surveyed current uncertain database management systems such as Trio [1], May BMS [2], Mysti Q [7], Orion [8], Bayes-Store [9], MCDB [10]The issues in the above existing uncertain management system are most of discussing and covered by Aggrawal et al in [21]. Other additional issues such as security and information leakage and representation formalisms are discussed and updated in the introduction and survey paper by [31]. All concerning issues in this existing system are covered through two survey paper et al[21] and [31], but the main focus of this paper is towards the survey of the existing uncertain management system and how a new system handles uncertain data and querying in the probabilistic and uncertain databases.

This paper is organized as follows: Section first covers surveying on different existing uncertain data management and its advantages. The second section briefly describes the introduction to the main model and some key concepts of handling uncertainty in the database. Section third shows how the system will handle different types of queries such as query processing based on the insert, update, delete, retrieval from a single as well as join processing, aggregate query processing, indexing, and data integration of this new uncertain database management system. Section four contains Experimental results and Section five contains the conclusion and summary.

1.1 Existing System

There many similar terms that represent similarity of meaning and can be used interchangeably for uncertainty such as vagueness, uncertainty, ambiguity, imprecision, and inconsistency etc. But these terms have their own meaning. In daily life, there are different meanings and use of this term. From a database point of view, uncertainty refers to the data or the presence of data that cannot be assured about its value and presence. Vagueness represents a range data item that does not have a clear determination of its exact value. For example, when saying that a particular entity is long without specifying its exact length. Ambiguity means the multiple paths represent the same way of a data item.

That represents incomplete data items. For example, it may not be specified whether the entity is measured in cm or mm. Imprecision means not precise, and it refers to the level of exactness by which data represents. For example, the entity color is red or orange. Finally, Inconsistency happens when having differing items that are conflicting in nature. For example, the entity is greater than 20 cm and the entity greater than or equal 32 cm.

The uncertain and probabilistic database is a management system that handles the data item in the database with some amount of confidence, or its value may be approximate. Value is caught by tuples that may have a few conceivable esteems, which represents certainty regarding every option. For instance, if a witness saw a vehicle that was a Tata with certainty 0.5, a Suzuki with certainty 0.3, or a Ford with certainty 0.2, the locating yields one tuple in the table Saw with three options for the saw car. Moreover, the nearness of tuples might be doubtful, again with alternatively indicated certainty. For instance, another witness may have 0.6 certainties that she saw a wrong doing vehicle, yet in the event that she saw one, it was unquestionably an Acura. In view of option tuple qualities and confidences, each ULDB (Uncertainty-Lineage Database) [1] provides to various conceivable occasions of a database.

An uncertain relation is a multiset of rows. Each row is composed of one or more alternatives. Each alternative is a regular row associated with a probabilistic confidence value in the range of [0,1]. In a single row, if $\sum$ is the sum of the confidence values of all alternatives, then $0 \leq \sum \leq 1$. If $\sum < 1$, then the entire x-row may not exist, whose confidence is $(1 - \sum)$ [1][2]. An Uncertain database represents a set of possible instances [3][4].

There are many existing systems has already developed models which deals with handling uncertain and probabilistic databases like Trio [1][2], May BMS [6], Mysti Q [7], Orion [8], BAYESSTORE [9], MCDB [10], Pr DB [11], and so on all are from 2008 to 2013.

Trio [1][2][5] is a robust model that supports uncertain data and information lineage, alongside the standard highlights of a relational DBMS. The Trio gives both an API and a full-featured graphical UI. Tri QL defines the semantics of any type of relational query over a ULDB. Tri QL uses the same syntax as SQL. The Trio system is built entirely on top of a conventional relational DBMS. ULDBs are represented in relational tables. In this, Tri QL queries and commands are rewritten automatically into SQL commands. Consider a relation Person (Time, Name) that lists the names of people that in the contact may be regularly visited or rarely visited peopless. Suppose, at the particular time, the first person visited who is regularly in contact with you. So, clearly that person is Amit and at another time, one other person visited who is from rarely visible. This person looks most like Bob (60% chance), but may also be Charl (30%) or Dary (10%). We represent the relation as follows; "||" separates alternatives [5].

ID	Person (Time, Name)
11	(1, Amit) :1.0
12	(2, Bob) :0.6 || (2, Charl) :0.3 || (2, Dary) :0.1

Fig 1: Relation Person (Time, Name)

This above relation person represents data by TRIO [1] model representation. The sum of the confidence values of all alternatives in a tuple must be at most 1; if the sum is less than 1 (say, 0:8), then there is a 20% chance that none of the alternatives is present in the relation.

MayBMS [6] is front-line probabilistic database organization systems that utilize the characteristics of past database inquiry for achieving adaptability. Mysti Q [7] is a middleware: altogether, upgrades the querying capacity of an average social DBMS by enabling a variety of approximations to be added to normal SQL queries.

Orion [8] locally manages uncertain information displayed as arbitrary joint probability distributions. BAYESSTORE [9] is a model built on the principle of handling statistical models and probabilistic inference tools as first-class citizens of the database system. MCDB [10] is a model to handle uncertain data that can easily handle arbitrary joint probability distributions over discrete or continuous attributes, arbitrarily complex SQL queries, and arbitrary functions of the query-result distribution such as means, variances, and quantiles.

The Prob View [11] framework has been built up that enables the client to pick an alternate system for every administrator, per session or even per inquiry.

Lineage [12] is important in uncertain data management because it can be used for finding out which part of data contributes to a result and computing the probability of the result. Nonetheless, the existing works represent an uncertain tuple as a set of tuples that can be stored in a relational table. Lineage [13] can derive each tuple in the table, with which one can only find out the tuples rather than specific attributes that contribute to the result. If uncertain tuples have multiple uncertain attributes, for a result tuple with low probability, users cannot know which attribute is the main cause of it.

Drawbacks of these existing models are data redundancy, space overhead, and more time for query processing. To overcome these drawbacks, there is scope to develop a new model to handle data from uncertain and probabilistic databases. The proposed system in this paper concentrates on a new model for handling uncertain and probabilistic data with efficient techniques.

1.2 Proposed System

In order to address drawbacks of the existing models, we propose Relational Cross.

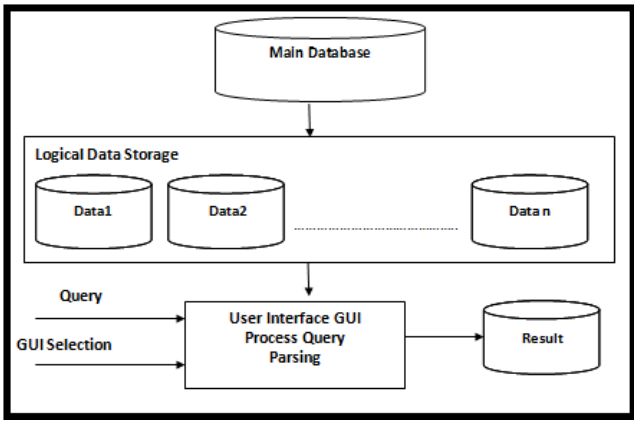


Figure 2: System for Relational Cross model

The Figure 2 represents system design for the Relational Cross model for handling data in the uncertain and probabilistic database.

The System consists of four phases- Translation phase, Execution phase, Post processing phase, and Result phase.

In the translation phase, input is GUI based on parameter selection or uncertain queries. This phase translates GUI-BASED parameter selection or uncertain queries to SQL queries. During translation, it

creates a parse tree and multiple SQL statements. In the Execution phase, from the multiple execution plans, best plan will be use for efficient execution. Cursors and different views are created. It gets results according to parameter selection or input query. In the post-processing phase, it processes the data to get relevant data from an uncertain database. During processing, it applies the constraints or format display on retrieved uncertain data. Get a confidence query if required. In a result phase, if it is stored data then stored in the table. If it is transient data, then retrieve data at runtime.

2. Methodology

Relational Cross model for handling data an in uncertain and probabilistic database divided into five modules –Input model, Translation model, Execution model, Post processing model, and Result model. To get user results, the Relational Cross model takes input from the input module. Relational Cross model actually uses the Translation model, Execution model, and Post processing model for processing data. Finally, the Relational Cross model produces results from the result module.

The Figure 3 represent module methodology consists of the design of all the internal modules. It shows how all the internal modules works together in the Relational Cross model for handling data in an uncertain and probabilistic database.

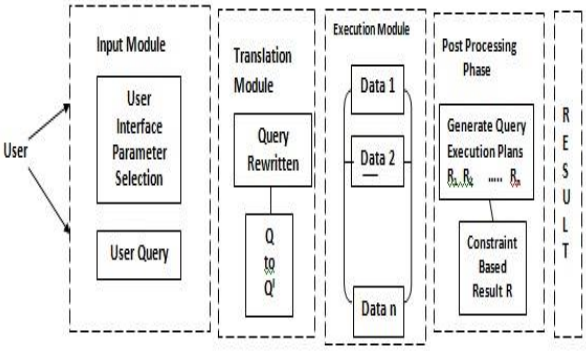


Figure 3: Module Methodology

Sections The Figure 3 represent module methodology consists of the design of all the internal modules. It shows how all the internal modules works together in the Relational Cross model for handling data in an uncertain and probabilistic database.

2.1 MODULE DESCRIPTION:

2.1.1 Input Model:

The Relational Cross model is designed for handling data in uncertain and probabilistic databases. This model works on data received from the user. The input model gives input data on which the Relational Cross model works. The model accepts properly formatted data with some restrictions. In the input model, there are two modes provided by which the user will enter data. The first mode is a GUI-BASED parameter. In GUI-BASED, user interface is provide for the user. The user interface has different menus for users for interaction such as table creation, insert data, delete data, view data, and select required data by selection.

An uncertain relation is a multiset of rows. Each row is composed of one or more alternatives. Each alternative is a regular row associated with a probabilistic confidence value in the range of [0,1]. So,

while creating an uncertain table, a form is provided which asks information such as table name, a number of attributes, each defining attribute associated with the attribute name, data type, the checkbox for certain or uncertain, and probability. In a single row, if $\sum$ is the sum of the confidence values of all alternatives, then $0 \leq \sum \leq 1$. In the probability column, the constraint is applied about value checks between 0 and 1, and the sum of the probability of alternatives of the same row should be 1.

The second mode is the user query. In this, the user will enter a query for interaction such as table creation, insert data, and delete data, view data, and select required data by selection. For this mode, the user must know some essential information about databases such as supported format query based on Relational Cross model, table name, and attribute name.

2.1.2 Translation Module:

The user wants to perform the operation such as table creation, insert data, delete data, view data, and select the required data by selection on uncertain and probabilistic databases. But actually, the Relational Cross model maintains this database is certain for storing value. So, there is a need for conversion from input operation on the uncertain and probabilistic database to the corresponding to the operation on a certain database which is maintained by the Relational Cross model.

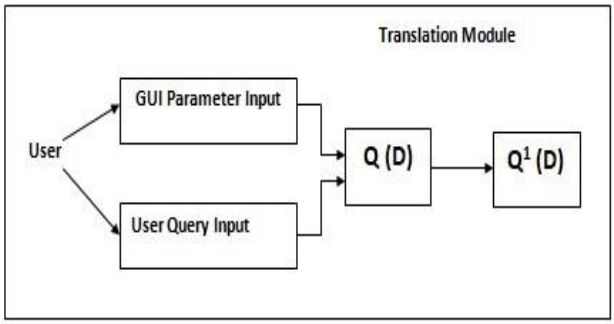


Figure 4: Translation Module

The input module provides an input operation requests on uncertain and probabilistic databases to the Translation module. Translation module of Relational Cross model which translates user query Q on the uncertain and probabilistic database (represented as Q(D) in figure 4) into Q1 on a certain database (represented as Q(D) in figure 4). In this module, query rewritten is performed which translates query Q to Q1. During rewriting, the Relational Cross model selects an appropriate corresponding certain table for given uncertain and probabilistic tables. Relational Cross model generates rewritten query Q1 from Q with the addition of rule generated from the execution module. The Relational Cross model makes use of Translation and execution module to generate Q1 from Q, where part of rewritten is performed by Translation module and the rest by the execution module.

2.1.3 Execution Module:

The execution module of the Relational Cross model is also performed part of query rewriting. In the execution module, the query is modified with an appropriate data item from the database. This phase replaces uncertain and probabilistic data items D from query Q1 (represented as Q1 (D) in figure 5) by certain data item D1, D2 …….., Dn (represented as Q1 (D1, D2 …….., Dn) in figure 5). So, the transformation of Query on D to modified with a query on D1, D2 …….., Dn.

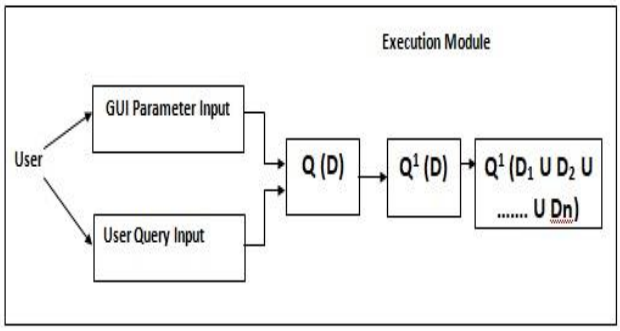
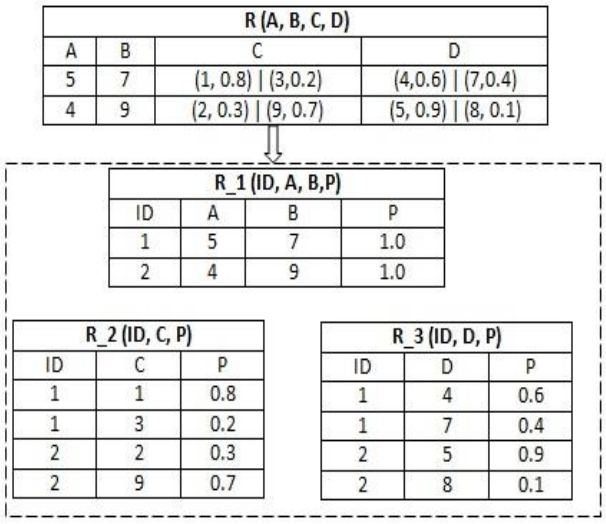


Figure 5: Execution Module

In Figure 5, D represents uncertain and probabilistic data items and data items D1, D2 ……., Dn represents its corresponding certain data items. When a query is rewritten, at that time Relational Cross will select appropriate certain data items from the relational database.

Relational Cross model manages uncertain data by storing it into relational certain data. A Relational Cross model performs how an uncertain table is stored in certain tables. An uncertain table R in Relational Cross model consists of two parts: certain attributes and uncertain attributes. A certain relation represents certain attributes. A certain relation represents all certain attributes with an actual certain value. The probability of a certain attribute value is always 1. So, a certain relation consists of a certain attribute with its actual value and its existence probability equals to 1 since it is certain. In this model, a unique ID is generated for each certain tuple. Each uncertain attribute has value and probability pair.



R (A, B, C, D)

A	B	C	D
5	7	(1, 0.8) | (3,0.2)	(4,0.6) | (7,0.4)
4	9	(2, 0.3) | (9, 0.7)	(5, 0.9) | (8, 0.1)

R_1 (ID, A, B,P)

ID	A	B	P
1	5	7	1.0
2	4	9	1.0

R_2 (ID, C, P)

ID	C	P
1	1	0.8
1	3	0.2
2	2	0.3
2	9	0.7

R_3 (ID, D, P)

ID	D	P
1	4	0.6
1	7	0.4
2	5	0.9
2	8	0.1

Figure 6:RelationalCross Model Data Management

Relational Cross model would store each uncertain attribute in separate relation. Since ID of each uncertain tuple corresponds to a certain one tuple probability, the ID of a certain tuple in certain relation can be seen as the ID of uncertain tuples in each relation. Figure 6 shows how the Relational Cross model manages uncertain table. The information is stored by corresponding certain tables. In Figure 6, uncertain table R contains certain attributes A and B, whereas uncertain attributes C and D. This R is represented by an uncertain relation. The Relational Cross model stores this uncertain table into

certain tables. The Relational Cross model manages uncertain table R into R_1, R_2, and R_3 as follows,

1. First, it creates a table of the certain attribute as one table name as R_1 with certain attribute A and B. In which, it adds column ID which contains automatically generated ID for each certain tuple and also adds column probability say P, which is set as 1 for each tuple since it is certain. In figure 6, corresponding first table R_1 contain four attribute ID, A, B, and P.
2. Uncertain Relation R contains two uncertain attributes as C and D. Second, it creates a table of one uncertain as R_2 with an uncertain attribute C. In which, it adds column ID which is referenced column ID from a certain table R_1. R_2 contains the ID column which represents the value from referenced a certain table R_1. The ID of each uncertain tuple corresponds to a certain one tuple probability; the ID of a certain tuple in a certain relation can be seen as the ID of uncertain tuples in this relation. Also, it contains column probability say P, which is its existence probability. In figure 6, corresponding first table R_2 contain three attribute ID, C, and P.
3. Thirdly, as Uncertain Relation R contains one more uncertain attribute D. Same as the previous scenario, it creates a table of another one uncertain as R_3 with uncertain attribute D. In which, it adds column ID which is referenced column ID from certain table R_1. R_3 contains the ID column which represents the value from referenced certain table R_1. The ID of each uncertain tuple corresponds to a certain one tuple probability; the ID of a certain tuple in certain relation can be seen as the ID of uncertain tuples in this relation. Also, it contains column probability say P, which is its existence probability. In figure 6, the corresponding first table R_3 contains three attribute ID, D, and P.

Figure 6, shows the Relational Cross model manage uncertain table R (A, B, C, D) is stored by corresponding certain tables R_1 (ID, A, B, P), R_2 (ID, C, P) and R_3 (ID, D, P) where R_1 represent table for certain attribute and R_2 and R_3 respectively represented tables for uncertain attribute C and D..

2.1.4 Post Processing Module:

In the Execution module of the Relational Cross model, final query rewriting is performed. So, the final query is generated and executed from the database. A result is generated from query execution. The output result is from a certain attribute. So, there is a need to apply some procedures and constraints to get the appropriate result as per the user understanding. Post processing performed this task. In the Post-processing phase, it processes the data to get relevant data from an uncertain database. Post processing will display output results in a format as per user requirements. During processing, it applies data processing format display and the constraints on retrieving certain data. After the post processing procedure, the result will be generated which will display formatted uncertain data. Get a confidence query if required. In a result phase, if it is stored data, then stored in the table. If it is transient data, then retrieve data at runtime.

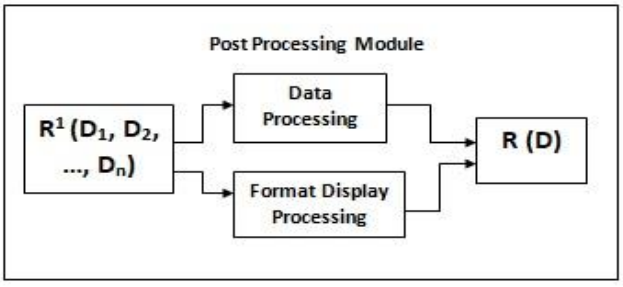


Figure 7: Post Processing Module

Figure 7 shows the Relational Cross model post processing phase. It shows the output result R1(D1, D2, …, Dn) is from a certain database. On this result, R1 post processing activities such as data processing format display and the constraints are applied. After post processing procedure, final result R (D) is generated which displays formatted uncertain data.

3. QUERY PROCESSING

When users retrieve some data from uncertain tables, a set of possible tuples may be returned as a query result due to several uncertain attribute values that existed in uncertain tables. When this possible tuple is generated at that time, it is not sufficient to satisfy users' needs. That generated tuple comes from a combination of the existing uncertain tuple. That uncertain tuple has an existing probability. When a new tuple generated as a result of a query, users want to know the probabilities for each of them and how they were generated from source uncertain tables. As we have mentioned in Section 1.2, we have constructed a working the Relational Cross model to convert the query on uncertain tables to query on certain tables, so any query result is generated by one or more uncertain tuples with probability.

In the following, we define operations on uncertain tables and their corresponding operation on certain tables by the Relational Cross model and the probability of a result p obtained by them.

- **Selection:**

Select (R, condition) -> {Select (R_1 join R_2 join …., R_n with p= P(R_1) ⋀ P(R_2)⋀……. ⋀ P(R_n)), condition}

Where, it represents select operation on R with a given condition and its corresponding select operation on R_1, R_2 …R_n with the same condition. Whereas, p= P(R_1) ⋀ P(R_2) ⋀……. ⋀ P(R_n) is a probability calculation of the probability attributes in R_1, R_2 …R_n corresponding to all the certain and uncertain attributes in R. For an uncertain table R, the selection operation on it actually executes on R_1, R_2…. R_n.

For example, assume that the selection operation on the uncertain table R (A, B, C, D) in Figure6 is σ A=4 AND B>2 AND C<5 AND D>6 (R). For the result "(4, 9, 2, 8)", according to the query execution module, we find all source tuples of matching to the condition given in the select query. Then we find A, B belongs to R_1, C belongs to R_2 and D belongs to R_3 according to the management of the execution module of the Relational Cross model. So, we can determine that an attribute corresponds to which table and obtain its probability value and ID. Probabilistic computation is calculated by attribute expression as P(R_1) ⋀ P(R_2) ⋀ P(R_3) based on the management of the Relational Cross model.

- **Projection:**

Project (R, a) ->{Project (a, R_1 join R_2 join ….., R_n with p= P(R_1) ⋀ P(R_2)⋀…….⋀ P(R_n)), condition }

Where it represents the project operation on R with the required attribute say a, and its corresponding project operation on R_1, R_2 …R_n with the same required attribute and p= P(R_1) ⋀ P(R_2) ⋀…….

Λ P(R_n) is a probability calculation of the probability attributes in R_1, R_2 …R_n corresponding to all the certain and uncertain attributes in R. For uncertain table R, the selection operation on it actually executes on R_1, R_2…. R_n.

For example, assume that the projection operation on the uncertain table R (A, B, C, D) in Figure.6 is π C (R). For the result "C column with tuples (1, 0.8) | (3,0.2) and (2, 0.3) | (9, 0.7)", according to the query execution module, we find all source tuples of a required attribute in projection query. Then we find ID belonging to R_1 and C belonging to R_2 according to the management of the execution module of the Relational Cross model. We can determine that a C attribute corresponds to the R_2 table and obtain its probability value and ID. Probabilistic computation is calculated by attribute expression as P(R_1) Λ P(R_2) based on management of the Relational Cross model.

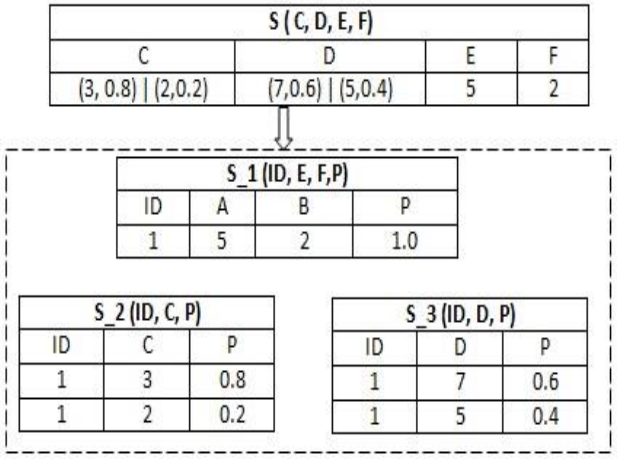


Figure 8: Example of Relational Cross Model Data Management

- **Join:**

Join ({R, S}, {a1, a2}) -> {Join ({ R_1 join R_2 join ….., R_n with p1= P(R_1) Λ P(R_2)Λ…….Λ P(R_n)} , {S_1 join S_2 join ….., S_n with p2= P(S_1) Λ P(S_2)Λ…….Λ P(S_n)}, {a1, a2} with p= p1 V p2)

Where it represents the join operation on an uncertain table R and S with {a1, a2} pair of join attribute and its corresponding join operation on R_1, R_2 …R_n with attribute and p1= P(R_1) Λ P(R_2) Λ…….Λ P(R_n) is a probability calculation of the probability attributes in R_1, R_2 …R_n corresponding to all the certain and uncertain attributes in R. Same join operation for table S with probability p2= P(S_1) Λ P(S_2) Λ…….Λ P(S_n) is a probability calculation of the probability attributes in S_1, S_2 …S_n corresponding to all the certain and uncertain attributes in R . R join with S, with the resulting probability is p= p1 Λ p2.

For example, assume that the join operation on the uncertain table R (A, B, C, D) and S (C, D, E, F) in Figure.6 and figure.8 respectively is R ⋈ S. For the result "tuples (5, 7, 3, 7, 5, 2) and (4, 9, 2, 5, 5, 2)", according to the query execution module, we find all source tuples of a required attribute in a join query of R and S to match attribute C and D. Then it find the ID, C, and D belonging to R_1, R_2, and R_3 respectively according to the management of execution module of Relational Cross model. Same it find the ID, C, and D belonging to S_1, S_2, and S_3 respectively according to the management of

the execution module of Relational Cross model. Then it perform R ⋈ S with join attributes C and D. So, it can determine that C and D attribute corresponds to R_2, R_3 and S_2, and S_3 respectively table and obtain its probability value and ID. Probabilistic computation is calculated by attribute expression as p= { p1 (P(R_1) ⋀ P(R_2) ⋀ P(R_3)) ⋀ p2 (P(S_1) ⋀ P(S_2) ⋀ P(S_3)) } based on management of Relational Cross model.

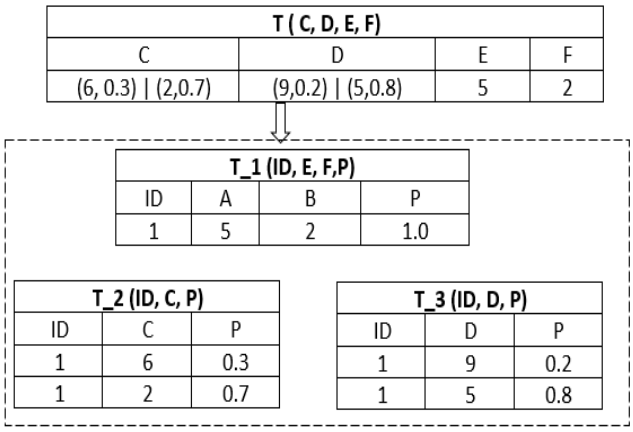


Figure 9: Example of the RelationalCross Model Data Management

- **Union:**

Union ({S, T}) -> {Union ({ S_1 join S_2 join ….., S_n with p1= P(S_1) ⋀ P(S_2)⋀…….⋀ P(S_n)} , {T_1 join T_2 join ….., T_n with p2= P(T_1) ⋀ P(T_2)⋀…….⋀ P(T_n)} with p= p1 ⋁ p2)

Where it represents the union operation on an uncertain table S and T and its corresponding join operation on S_1, S_2 …S_n with attribute and p1= P(S_1) ⋀ P(S_2) ⋀…….⋀ P(S_n) is a probability calculation of the probability attributes in S_1, S_2 …S_n corresponding to all the certain and uncertain attributes in S. Same join operation for table T with probability p2= P(T_1) ⋀ P(T_2) ⋀…….⋀ P(T_n) is a probability calculation of the probability attributes in T_1, T_2 …T_n corresponding to all the certain and uncertain attributes in T. To perform the union operation, the two conditions must be satisfied by the relation in which, union is performed. The condition includes, both relations that have the same arity and domain of the ith attribute of the relations must be the same. The Result of S Union T contains all possible combinations of a tuple without duplicate from S and T with probability is p= p1 ⋁ p2.

For example, assume that the union operation on the uncertain table S (C, D, E, F) and T (C, D, E, F) in Figure.8 and figure.9 respectively is S ∪ T. For the result “tuples (3, 7, 5, 2), (3, 5, 5, 2), (2, 7, 5, 2), (2, 5, 5, 2), (6, 9, 5, 2), (6, 5, 5, 2), and (2, 9, 5, 2)”, according to query execution module, we find all source tuples of a required attribute in union query of S and T with all possible combinations of a tuple without duplicate from S and T match attribute value of C, D, E, and F. Then we find the ID, C, D, E and F belonging to S_1, S_2, and S_3 respectively, according to the management of the execution module of the Relational Cross model. Same we find the ID, C, D, E, and F belonging to T_1, T_2, and T_3 respectively, according to the management of execution module of Relational Cross model. Then we perform S ∪ T with the same attribute C, D, E, and F. So we can determine that C, D, E, and F attribute corresponds to S_1, S_2, S_3 and T_1, T_2 and T_3 respectively table and obtain its probability value and ID. Probabilistic computation is calculated by attribute expression as p= {p1

(P(S_1) ⋀ P(S_2) ⋀ P(S_3)) ⋁ p2 (P(T_1) ⋀ P(T_2) ⋀ P(T_3))} based on management of Relational Cross model.

- **Intersection:**

Intersect ({S, T}) -> {Intersect ({ S_1 join S_2 join ….., S_n with p1= P(S_1) ⋀ P(S_2)⋀…….⋀ P(S_n)} , {T_1 join T_2 join ….., T_n with p2= P(T_1) ⋀ P(T_2)⋀…….⋀ P(T_n)} with p= p1 ⋀ p2)

Where it represents the intersection operation on uncertain table S and T and its corresponding join operation on S_1, S_2 …S_n with attribute and p1= P(S_1) ⋀ P(S_2) ⋀…….⋀ P(S_n) is a probability calculation of the probability attributes in S_1, S_2 …S_n corresponding to all the certain and uncertain attributes in S. Same join operation for table T with probability p2= P(T_1) ⋀ P(T_2) ⋀…….⋀ P(T_n) is a probability calculation of the probability attributes in T_1, T_2 …T_n corresponding to all the certain and uncertain attributes in T. To perform intersection operation, the two conditions must be satisfied by the relation in which intersection is performed. The condition includes, both relations have the same arity, and the domain of ith attribute of the relations must be the same. The Result of S Union T contains tuples that are common from S and T with probability is p= p1 ⋀ p2.

For example, assume that union operation on the uncertain table S (C, D, E, F) and T (C, D, E, F) in Figure.8 and figure.9 respectively is S⋂ T. For the result “tuples (2, 5, 5, 2)”, according to the query execution module, we find all source tuples of a required attribute in intersecting query of S and T contains tuples which are common from S and T match attribute value of C, D, E, and F. Then we find the ID, C, D, E and F belonging to S_1, S_2, and S_3 respectively, according to the management of execution module of Relational Cross model. Same we find the ID, C, D, E, and F belonging to T_1, T_2, and T_3 respectively according to management of the execution module of Relational Cross model. Then we perform S ⋂ T with the same attribute C, D, E, and F. So we can determine that C, D, E, and F attribute corresponds to S_1, S_2, S_3 and T_1, T_2 and T_3 respectively table and obtain its probability value and ID. Probabilistic computation is calculated by attribute expression as p= { p1 (P(S_1) ⋀ P(S_2) ⋀ P(S_3)) ⋀ p2 (P(T_1) ⋀ P(T_2) ⋀ P(T_3)) } based on management of Relational Cross model.

4. EXPERIMENT RESULTS

We conducted experiments for analyzing the performance of the system. There are many simple queries can be used for testing, but it shows very small variants of difference. We have used queries such as multiple aggregations with join and subqueries with join. In this section, we analyze the better performance of the system in the case of different types of queries such as multiple aggregations with join and subqueries with joins, etc. There are two systems used for handling an uncertain database. One, an existing relational model system that handles uncertainty in the table in the single table with repetition of value for each uncertain value and Second, the proposed system the Relational Cross model. We compare the result of the performance of these two systems by type of queries. We report the performance results of the system on real-life data sets.

- **Multiple Aggregation with join:**

The constructed working the Relational Cross model will convert any type of query on uncertain tables to the corresponding queries on a certain table. The query result is generated by one or more uncertain tuples with probability. There are many aggregate functions in SQL such as COUNT, SUM, AVG, MIN, and MAX. In this section, we have used multiple aggregation queries with join for testing the performance of the system.

This part shows results in the analysis of the time required for running multiple aggregation query with join on the existing relational model system and the proposed system Relational Cross model. Figure 10 shows the result of the sample multiple aggregation query with a join. It shows the time required to run multiple aggregation queries with join in milliseconds. It shows the total time required to run multiple aggregates with join on the existing relational model system and Relational Cross model system in milliseconds.

Multiple aggregation query with join	RelationalCross Model System	Relational Model System
Sample Query 1	17	14
Sample Query 2	20	16
Sample Query 3	24	21
Sample Query 4	27	22

Figure 10: Result of multiple aggregation query with join

Figure 11, we show performance of the existing relational model system and Relational Cross model system for multiple aggregation queries with join in milliseconds. It shows the total time required. The X-axis corresponds to different sample queries of multiple aggregates with join and the Y-axis corresponds to the time required in milliseconds. The graph shows the time required to run multiple aggregates with join on the existing relational model system and Relational Cross model system. By observing the graph of Figure 11, it conclude that as the total time required to run multiple aggregates with join on the existing relational model system is more than the time required to run some queries on the Relational Cross model system. The Relational Cross model system provides better performance than the existing relational model system for multiple aggregates with a join query.

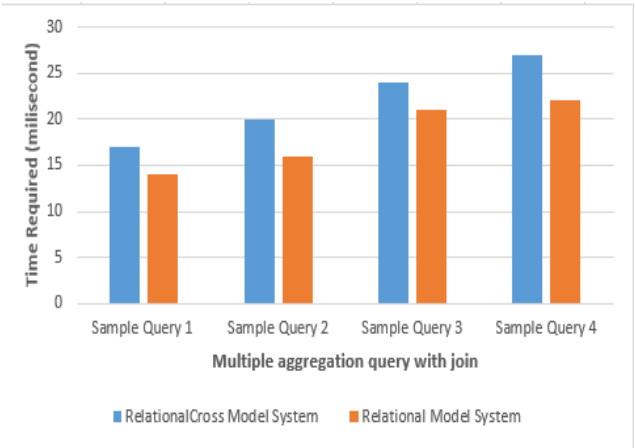


Figure 11: Performance of systems for multiple aggregation query with join

- **Subqueries with join:**

A subquery is a select-from-where expression that is nested within another query. The use of subqueries is to perform tests for set membership, make set comparisons, and determine set cardinality. Based on this, there are many types of subqueries such scalar-subqueries, set membership test sub queries, empty set test subqueries, correlated subqueries, set comparison test subqueries, and uniqueness test subqueries, etc. We have used different types of subqueries with join for testing performance on the system.

This part shows results from the analysis of the time required for running different subqueries with join on the existing relational model system and proposed system Relational Cross model. Figure 12 shows the result of sample subqueries with join. It shows the time required to run subqueries with join in milliseconds. It shows the total time required to run subqueries with join on the existing relational model system and Relational Cross model system in milliseconds.

Sub Queries with join	RelationalCross Model System	Relational Model System
Sample Query 1	10	7
Sample Query 2	14	10
Sample Query 3	13	9
Sample Query 4	8	5

Figure 12: Result of sub queries with join

Figure 13, shows performance of the existing relational model system and Relational Cross model system for subqueries with join in milliseconds. It shows the total time required. The X-axis corresponds to different sample subqueries with join and Y-axis corresponds to the time required in a millisecond. The graph shows the time required to run subqueries with join on the existing relational model system and Relational Cross model system. By observing the graph of figure 13, we can say that as the total time required to run subqueries with join on the existing relational model system is more than the time required to run the same subqueries on the Relational Cross model system. The Relational Cross model system provides better performance than the existing relational model system for subqueries with join.

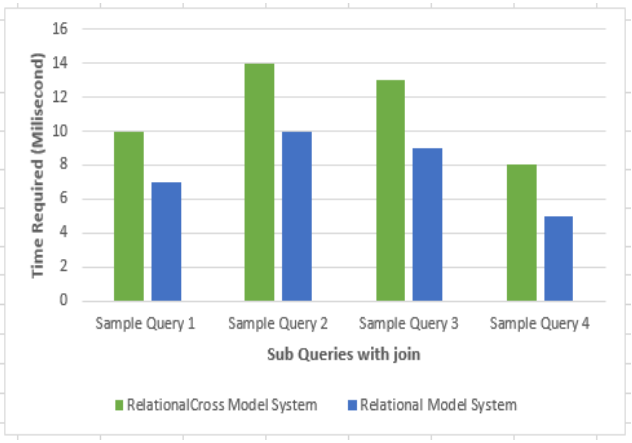


Figure 13: Performance of systems for sub queries with join.

5. Conclusion and future work

The proposed system Relational Cross model provides efficient management of data in uncertain and probabilistic databases. Relational Cross model achieves the efficient management of data through four phases. The translation phase translates GUI based parameter selection or uncertain queries to SQL queries. It also creates a parse tree and multiple SQL statements. The Execution phase, based on multiple execution plans, efficient execution will be performed. Cursors and different views are created. It gets a result, according to parameter selection or input query. The Post processing phase, it processes the data to get relevant data from the uncertain database. During processing, it applies the constraints or format display on retrieved uncertain data. Get a confidence query if required. The final result phase, if it is stored data then stored in the table. If it is transient Data, then retrieve data at runtime. It identifies uncertain and certain data. We provide an efficient way of getting better performance in the case of handling multiple aggregates with join and subqueries with join. We show how the system provides an efficient way and gets better performance through some result analysis. From the above result analysis, we conclude that,

- The Relational Cross model system provides efficient management of data and provides better performance than the existing relational model system for multiple aggregates with a join query.
- The Relational Cross model system provides better performance than existing relational model systems for multiple aggregates with join queries.
- We identify following are some direction to some future work,
- A first obvious extension to our work is a handling system with more complex SQL queries and PL/SQL functions, procedures, trigger, and cursor for better performance.
- Relational Cross model stores uncertain tables into certain tables. Depends on how many uncertain attributes are present in relation, that number of a certain table is used for storing data. This Relational Cross model takes more space for storing such an uncertain table. It needs to Reduce the space usage required for the uncertain database.
- The extension of this work can be executed parallel queries with distributed computing.

References –.

[1]. Agrawal, P., Benjelloun, O., Das, A., Hayworth, C., Nabar, S., & Sugihara, T., et al." Trio: a system fordata, uncertainty, and lineage". Proceedings of the 32nd international conference on Very large data base,(pp. 1151-1154). DOI 10.1.1.108.9426, 2006.

[2]. Widom, J. "Trio A System for integrated management data, accuracy and lineage". Conference on Innovative Data Systems Research. DOI 10.1.1.153.9613, 2005.

[3]. Agrawal, P., &Widom, J. "Continuous uncertainty in trio".Stanford university, Stanford Info Lab Publication Server, (2009).

[4]. M. Mutsuzaki, M. Theobald, A. de Keijzer, J. Widom, P. Agrawal,O. Benjelloun, A. Das Sarma, R. Murthy, and T. Sugihara, "Trio-One: Layering Uncertainty and Lineage on a Conventional DBMS(Demo)," Proc. Conf. Innovative Data Systems Research (CIDR),pp. 269-274, 2007.

[5]. Trio Online Resources: TriQL Language Language Manual, Online Demo and Open-Source Distribution, http://www.infolab.stanford.edu/trio, 2009..

[6]. Huang, J., Antova, L., Koch, C., & Olteanu, D. "May BMS: A Probabilistic Database Management System".Proceedings of the 2009 SIGMOD International Conference on Management of Data (pp. 1071-1074). DOI 10.1145/1559845.1559984,2009.

[7]. Boulos, J., Dalvi, N., Mandhani, B., Mathur, S., Chris, R., &Suciu, D. "Mystiq: a system for finding more answers by using probabilities". Proceedings of the International Conference on Management of Data, (pp. 891-893). DOI 10.1145/1066157.1066277, 2005.

[8]. R. Cheng and S. Prabhakar. "ORION: Managing uncertain(sensor) databases. In SIGMOD Record issue onSensor Technology, December 2003.

[9]. Wang, D. Z., Michelakis, E., Garofalakis, M., & Hellerstein, J. M. Bayesstore: managing large, uncertaindata repositories with probabilistic graphical models". Very Large Database (pp.340-351). DOI 10.1.1.140.6348, 2008.

[10]. Jampani, R., Xu, F., Wu, M., Perez, L. L., Jermaine, C., & Haas, P. J. "MCDB: a montecarlo approach to managing uncertain data". Proceedings of the SIGMOD International Conference on Management of Data,(pp. 687-700). DOI 10.1145/1376616.1376686, 2008.

[11]. Laks V. S. Lakshmanan, Nicola Leone, Robert Ross and V. S. Subrahmanian, "Prob View: A Flexible Probabilistic Database System". ACM Transactions on Database Systems, Vol. 22, No. 3, Pages 419–469,1997.

[12]. O. Benjelloun, A. Das Sarma, A.Y. Halevy, and J. Widom,"ULDBs: Databases with Uncertainty and Lineage," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 953-964, 2006.

[13]. Raghotham Murthy, Robert Ikeda and Jennifer Widom, "Making Aggregation Work in Uncertain and Probabilistic Databases", IEEE Transactions on knowledge and data engineering, Aug, 2011.

[14]. D. Barbara, H. Garcia-Molina, and D. Porter, "The Management of Probabilistic Data", IEEE Transactions on Knowledge and Data Engineering, 1992.

[15]. S. McClean, B. Scotney, and M. Shapcott, "Aggregation of Imprecise and Uncertain Information in Databases", IEEE Transactions on knowledge and data engineering, Nov./Dec.2001.

[16]. L. Chen and A. Dobra, "Efficient Processing of Aggregates in Probabilistic Databases", Technical Report REP-2008-454, Univ. Of Florida,2008.

[17]. N.N. Dalvi and D. Suciu, "Efficient Query Evaluation on Probabilistic Databases", Proc. Int'l Conference on Very Large Data Bases (VLDB), 2007.

[18]. Anish Das Sarma, Martin Theobald, and Jennifer Widom, "Exploiting Lineage for Confidence Computation in Uncertain and Probabilistic Databases", 2009.

[19]. PL/SQL User's Guide and Reference

[20]. V. V. Kheradkar and U.L. Kulkarni, "Efficient retrieval of aggregate data in Uncertain and Probabilistic Databases "International Journal of Engineering Trends & Technology (IJETT), Paper No-IJETT-V8P276 Volume 8 Number 8 February 2014 issue ISSN: 2231-5381 Page no-443 to 449.

[21]. Aggrwal, C. C., "A survey of uncertain data algorithms and applications". IEEE Transactions onKnowledge and Data Engineering (pp.609-623). http://dx.doi.org/10.1109/TKDE.2008.190 , 2009.

[22]. Agrawal, P., & Widom, J., "Confidence-aware join in large uncertain database". Retrieved from Inhttp://dbpuds.stanford.edu/pub/2007-14, 2007.

[23]. Dhandore, K., & Ragha, L., "Performance Evaluation of Decision Trees for Uncertain Data Mining" International Journal of Emerging Trend and Technology in computer science, 3(6), 2014.

[24]. Li, J. W. "Range Queries on Uncertain Data". Springer (pp.326-337).http://dx.doi.org/10.1007/978-3-319-13075-0_26 2014.

[25]. Lian, X., & Chen, L. "Probabilistic Top-k Queries in Uncertain Database." Information Sciences, 2013.

[26]. Afrati, F. N., & Vasilakopoulos, "A. Managing Lineage and Uncertainty under a data exchange setting". In Proceedings of the 4th international conference on Scalable uncertainty management (pp. 28-41).http://dx.doi.org/10.1007/978-3-642-15951-0_9, 2010.

[27]. Antova, L., Jansen, T., Koch, C., & Olteanu, D. (2008). "Fast and simple relational processing of uncertain data". Proceedings of the 2008 IEEE 24th International Conference on Data Engineering (pp. 983-992).http://dx.doi.org/10.1109/ICDE.2008.4497507.

[28]. Chothia, T., Kawamoto, Y., Novakovic, C., & Parker, D. "Probabilistic Point-to-Point Information Leakage." In Computer Security Foundations Symposium (pp.193-205).http://dx.doi.org/10.1109/CSF.2013.20, 2013.

[29]. Prerana Rajaram Jalgaonkar, Samidha Chavan, Kirthi Guptam "Study of Aggregation Work in Uncertain and Probabilistic Databases" International Journal of Engineering Science and Computing, May 2016.

[30]. Nguyen Hoa and Tran Duc Hieu, "A Probabilistic Relational Data Model for Uncertain Information", IEEE Third International Conference on Information Science and Technology, March 23-25, 2013

[31]. Nermin Abdelhakim Othman, Ahmed Sharaf Eldin, Doaa Saad Elzanfaly, "Enhancing Aggregation over Uncertain Databases", 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications.

[32]. M. Yang, H. Wang, H. Chen, and W. Ku, "Querying uncertain data with aggregate constraints," SIGMOD Conference, pp. 817-828, 2011.