ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Secrecy-Protection Accuracy-Restrained Access Control Mechanism Over Relational Data

Shaik Abdul Saleem
Department of CSE,
Nimra College of Engineering & Technology, Vijayawada.

Sayeed Yasin, Ph.D
Department of CSE,
Nimra College of Engineering & Technology, Vijayawada.

G.Minni
Assistant Professor,
Department of CSE,
Nimra College of Engineering & Technology, Vijayawada.

Abstract:

Access control mechanisms protect sensitive information from unauthorized users. However, when sensitive information is shared without presence of Privacy Protection Mechanism (PPM), an authorized user can still compromise the privacy of a person leading to identity disclosure. A PPM can use suppression and generalization of relational data to anonymize and satisfy privacy requirements, e.g., k-anonymity and l-diversity, against identity and attribute disclosure. However, privacy is achieved at the cost of precision of authorized information. In this paper, we propose an accuracy-constrained privacy-preserving access control framework. The access control policies define selection predicates available to roles while the privacy requirement is to satisfy the k-anonymity or l-diversity. An additional constraint that needs to be satisfied by the PPM is the imprecision bound for each selection predicate. The techniques for workload-aware anonymization for selection predicates have been discussed in the literature. However, to the best of our knowledge, the problem of satisfying the accuracy constraints for multiple roles has not been studied before. In our formulation of the aforementioned problem, we propose heuristics for anonymization algorithms and show empirically that the proposed approach satisfies imprecision bounds for more permissions and has lower total imprecision than the current state of the art.

Index Terms:

Access control, privacy, k-anonymity, query evaluation.

1.INTRODUCTION:

Different organizations collect and analyze consumer data toimprove their quality of services. We use Access Control Mechanisms(ACM) to provide only authorized information to users.

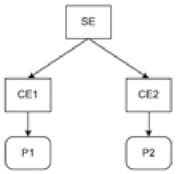
However, sensitive information canstill be misused by authorized users to compromise the privacyof consumers. The concept of privacy-preservation forsensitive data can require the enforcement of privacy policiesor the protection against identity disclosure by satisfyingsome privacy requirements. In this paper, weinvestigate privacy-preservation from the anonymityaspect. The sensitive information, even after the removal ofidentifying attributes, is still susceptible to linking attacksby the authorized users. This problem has been studiedextensively in the area of micro data publishing and privacydefinitions, e.g., k-anonymity , l-diversity, andvariance diversity. Anonymization algorithms use suppression and generalization of records to satisfy privacy requirements with minimal distortion of micro data. The anonymity techniques can be used with an access controlmechanism to ensure both security and privacy of the sensitiveinformation. The privacy is achieved at the cost of accuracyand imprecision is introduced in the authorizedinformation under an access control policy.

We use the concept of imprecision bound for eachpermission to define a threshold on the amount ofimprecision that can be tolerated. Existing workload awareanonymization techniques, minimize theimprecision aggregate for all queries and the imprecisionadded to each permission/query in the anonym zedmicro data is not known. Making the privacy requirementmore stringent (e.g., increasing the value of k or l)results in additional imprecision for queries. However,the problem of satisfying accuracy constraints for individualpermissions in a policy/workload has not beenstudied before. The heuristics proposed in this paper for accuracy-constrained privacy-preserving access control are also relevant in the context of workload-aware anonymization. In this paper the focus is on a static relational table that is anonym zed only once. To exemplify our approach, role-based access control is assumed. However, the concept of accuracy constraints for permissions can be applied to any privacy-preserving security policy, e.g., discretionary access control.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Example 1 (Motivating Scenario). Syndromic surveillance systems are used at the state and federal levels to detect and monitor threats to public health. The department of health in a state collects the emergency department data (age, gender, location, time of arrival,symptoms, etc.) from county hospitals daily. Generally,each daily update consists of a static instance that isclassified into syndrome categories by the departmentof health. Then, the surveillance data is anonym zedand shared with departments of health at each county.

Role	Designation
SE	State Epidemiologist
CE1	County 1 Epidemiologist
CE2	County 2 Epidemiologist



Permission	Authorized Query Predicate
P1	Location = County 1 ∧ Age = 15-65 ∧ Syndrome = Influenza
P2	Location = County 2 ∧ Age = 15-65 ∧ Syndrome = Influenza

Fig. 1. Access control policy.

An access control policy is given in Fig. 1 that allowsthe roles to access the tuples under the authorizedpredicate, e.g., Role CE1 can access tuples under PermissionP1. The epidemiologists at the state andcounty level suggest community containment measures,e.g., isolation or quarantine according to thenumber of persons infected in case of a flu outbreak.According to the population density in a county, anepidemiologist can advise isolation if the number ofpersons reported with influenza are greater than 1,000and quarantine if that number is greater than 3,000 ina single day. The anonymization adds imprecision tothe query results and the imprecision bound for eachquery ensures that the results are within the tolerancerequired. If the imprecision bounds are not satisfiedthen unnecessary false alarms are generated due to thehigh rate of false positives.

2 BACKGROUND:

Here, In this section, role-based access control and privacydefinitions based on anonymity are over-viewed. Query evaluation semantics, imprecision, and the Selection Mondrian algorithm are briefly explained.Given a relation T = {A1,A2, . . .,An}, where Ai is anattribute, T_ is the anonym zed version of the relation T.We assume that T is a static relational table.

2.1 Access Control for Relational Data:

Fine-grained access control for relational data allows todefine tuple-level permissions, e.g., Oracle VPD andSQL. For evaluating user queries, most approachesassume a Truman model. In this model, a user query ismodified by the access control mechanism and only theauthorized tuples are returned. Column level access controlallows queries to execute on the authorized column of therelational data only. Cell level access control forrelational data is implemented by replacing the unauthorizedcell values by NULL values. Role-based Access Control (RBAC) allows defining permissionson objects based on roles in an organization. AnRBAC policy configuration is composed of a set of Users(U), a set of Roles (R), and a set of Permissions (P). For therelational RBAC model, we assume that the selection predicateson the QI attributes define a permission. UA is auser-to-role (U X R) assignment relation and PA is a roleto-permission (R X P) assignment relation. A role hierarchy(RH) defines an inheritance relationship among roles and isa partial order on roles (R X R). Each permission definesa hyper-rectangle in the tuple space and all the tuplesenclosed by this hyper-rectangle are authorized to the roleassigned to the permission. In practice, when a userassigned to a role executes a query, the tuples satisfying theconjunction of the query predicate and the permission arereturned.

2.2 Anonymity Definitions:

Definition 1 (Equivalence Class (EC)). An equivalence class is a set of tuples having the same QI attribute values.

Definition 2 (k-anonymity Property). A table T_ satisfies the k-anonymity property if each equivalence class has k or moretuples.

Definition 3 (Query Imprecision). Query Imprecision is defined as the difference between the number of tuples returned by a query evaluated on an anonym zed relation T* and the number of tuples for the same query on the original relation T. The imprecision for query   is denoted by

imp_{Q_i},

$$imp_{Q_i} = |Q_i(T^*)| - |Q_i(T)|, \text{ where}$$
$$|Q_i(T^*)| = \sum_{EC \text{ overlaps } Q_i} |EC|.$$

k-anonymity is prone to homogeneity attacks when the sensitive value for all the tuples in an equivalence class is the same. To counter this shortcoming, l-diversity has been proposed and requires that each equivalence class of T* contain at least l distinct values of the sensitive attribute.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

For sensitive numeric attributes, an l-diverse equivalence class can still leak information if the numeric values are close to each other. For such cases, variance diversity has been proposed that requires the variance of each equivalence class to be greater than a given variance diversity parameter.

	QI_1	QI_2	S_1
ID	Age	Zip	Disease
1	5	15	Flu
2	15	25	Fever
3	28	28	Diarrhea
4	25	15	Fever
5	22	28	Flu
6	32	35	Fever
7	38	32	Flu
8	35	25	Diarrhea

(a) Sensitive table

	QI_1	QI_2	S_1
ID	Age	Zip	Disease
1	0-20	10-30	Flu
2	0-20	10-30	Fever
3	20-30	10-30	Diarrhea
4	20-30	10-30	Fever
5	20-30	10-30	Flu
6	30-40	20-40	Fever
7	30-40	20-40	Flu
8	30-40	20-40	Diarrhea

(b) 2-anonymous Table

Fig. 2. Generalization for *k*-anonymity and *l*-diversity.

The table in Fig. 2a does not satisfy k-anonymitybecause knowing the age and zip code of a person allowsassociating a disease to that person. The table in Fig. 2b isa 2-anonymous and 2-diverse version of table in Fig. 2a.The ID attribute is removed in the anonym zed table andis shown only for identification of tuples. Here, for anycombination of selection predicates on the zip code andage attributes, there are at least two tuples in each equivalenceclass. In Section 4, algorithms are presented for k-anonymityonly. However, the experiments are performedfor both l-diversity and variance diversity usingthe proposed heuristics for partitioning.

2.3 Top Down Selection Mondrian(TDSM):

The objective of TDSM is to minimize the total imprecision for all queries while the imprecision bounds for queries have not been considered. TDSM starts with the whole tuple space as one partitionand then partitions are recursively divided till the timenew partitions meet the privacy requirement. To divide apartition, two decisions need to be made, i) Choosing asplit value along each dimension, and ii) Choosing adimension along which to split. In the TDSM algorithm[5], the split value is chosen along the median and thenthe dimension is selected along which the sum of imprecisionfor all queries is minimum.

3ANONYMIZATION WITH IMPRECISION BOUNDS

3.1 Definitions

Definition (Query Imprecision Slack). The query imprecision slack, denoted by sQi for a Query, say Qi, is defined as the difference between the query imprecision bound and the actual query imprecision.

$$s_{Q_i} = \begin{cases} B_{Q_i} - imp_{Q_i}, & \text{if } imp_{Q_i} \leq B_{Q_i}, \\ 0, & \text{otherwise.} \end{cases}$$

Definition (Query Imprecision Bound).The query imprecision bound, denoted by BQi , is the total imprecision acceptable for a query predicate Qi and is preset by the access control administrator.

Example 3. Assume two range queries as given in Fig. 3. The queries are the shaded rectangles with solid lines while the partitions are the regions enclosed by rectangles with dashed lines. The imprecision bounds forQueries Q1 and Q2 are preset to 2 and 0. The partitioninggiven in Fig. 2b does not satisfy the imprecision bounds.However, the partitioning given in Fig. 3 satisfies thebounds for Queries Q1 and Q2 as the imprecision for Q1and Q2 is 2 and 0, respectively.

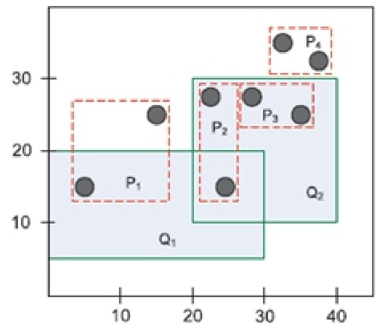


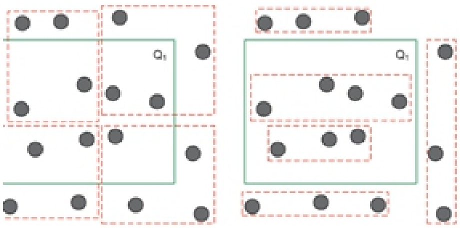
Fig. 3. Anonymization satisfying imprecision bounds.

Definition (Query Cut). A query cut is defined as the splitting of a partition along the query interval values. For a query cut using Query Qi, both the start

of the query interval $(a_j^{Q_i})$ *and the end of the query interval* $(b_j^{Q_i})$ *are considered to split a partition along the* j^{th} *dimension.*

Example 4. A comparison of median cut and query cut isgiven in Fig. 4 for 3-anonymity. The rectangle with solidlines represents Query Q1. While, the rectangles withdotted lines represent partitions. In Fig. 4a the tuples arepartitioned according to the median cut and even afterdividing the tuple space into four partitions there is noreduction in imprecision for the Query Q1. However, forquery cuts in Fig. 4b the imprecision is reduced to zeroas partitions are either non-overlapping or fully enclosedinside the query region.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal



(a) Median cut (b) Query cut

Fig. 4. Comparison of median and query cut.

3.3.1 Access Control Enforcement:

The exact tuple values in a relation are replaced by the generalizedvalues after the anonymization. In this case, accesscontrol enforcement over the generalized data needs to bedefined. In this section, we discuss the Relaxed and Strictaccess control enforcement mechanisms over anonymizeddata. The access control enforcement by reference monitorcan be of the following two types:

1.Relaxed. Use overlap semantics to allow access to allpartitions that are overlapping the permission.
2.Strict. Use enclosed semantics to allow access to only those partitions that are fully enclosed by the permission.

Relaxedenforcement violates the authorization predicate by givingaccess to extra tuples but is beneficial for applicationswhere low cost of a false alarm is tolerable as comparedto the risk associated with a missed event. Examplesinclude epidemic surveillance and airport security. Onthe other hand, strict enforcement is suitable for applicationswhere a high risk is associated with a false alarm ascompared to the cost of a missed event.

An example is afalse arrest in case of shoplifting. Here in this paper, we first focus on relaxed enforcement. However the proposed methodsfor anonymization are also valid for strict enforcementbecause the proposed heuristics reduce the overlapbetween partitions and queries.

4 HEURISTICS FOR PARTITIONING
4.1 Top-Down Heuristic 1 (TDH1)

Algorithm 1: TDH1

Input : T, k, Q, and B_{Q_j}
Output: P
1 Initialize Set of Candidate Partitions($CP \leftarrow T$)
2 **for** $(CP_i \in CP)$ **do**
3 Find the set of queries QO that overlap CP_i such that $ic_{CP_i}^{QO_j} > 0$
4 Sort queries QO in increasing order of B_{Q_j}
5 **while** *(feasible cut is not found)* **do**
6 Select query from QO
7 Create query cuts in each dimension
8 Select dimension and cut having least overall imprecision for all queries in Q
9 **if** *(Feasible cut found)* **then**
10 Create new partitions and add to CP
11 **else**
12 Split CP_i recursively along median till anonymity requirement is satisfied
13 Compact new partitions and add to P
14 return (P)

The TDH1 algorithm is listed in Algorithm 1. In thefirst line, the whole tuple space is added to the set of candidatepartitions. In the Lines 3-4, the query overlappingthe candidate partition with least imprecision bound andimprecision greater than zero is selected.

The while loopin Lines 5-8 checks for a feasible split of the partitionalong query intervals. If a feasible cut is found, then theresulting partitions are added to CP. Otherwise, the candidatepartition is checked for median cut in Line 12. Afeasible cut means that each partition resulting from splitshould satisfy the privacy requirement.

4.2 Top-Down Heuristic 2 (TDH2):

In the TDH2,the query bounds are updated as the partitions are added tothe output. This update is carried out by subtracting the value from the imprecision bound of each query,for a Partition, say Pi, that is being added to the output.

Forexample, if a partition of size k has imprecision 5 and 10 forQueries Q1 and Q2 with imprecision bound 100 and 200,then the bounds are changed to 95 and 190, respectively.The best results are achieved if the kd-tree traversal isdepth-first (preorder). Preorder traversal for the kd-treeensures that a given partition is recursively split till the leafnode is reached.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Algorithm 2: TDH2

Input : T, k, Q, and B_{Q_j}
Output: P
1 Initialize Set of Candidate Partitions($CP \leftarrow T$)
2 **for** ($CP_i \in CP$) **do**
    // Depth-first (preorder) traversal
3   Find the set of queries QO that overlap CP_i such that $ic_{CP_i}^{QO_j} > 0$
4   Sort queries QO in increasing order of B_{Q_j}
5   **while** (*feasible cut is not found*) **do**
6   Select query from QO
7   Create query cuts in each dimension
8   Select dimension and cut having least overall imprecision for all queries in Q
9   **if** (*Feasible cut found*) **then**
10   Create new partitions and add to CP
11   **else**
12   Split CP_i recursively along median till anonymity requirement is satisfied
13   Compact new partitions and add to P
14   Update B_{Q_j} according to $ic_{P_i}^{Q_j}$, $\forall Q_j \in Q$
15 **return** (P)

4.3 Top-Down Heuristic 3 (TDH3):

In the TDH3 algorithm, we modify TDH2 so that the time complexity of can be achieved at the cost of reduced precisionin the query results. Given a partition, TDH3 checks thequery cuts only for the query having the lowest imprecisionbound. Also, the second constraint is that the query cuts arefeasible only in the case when the size ratio of the resultingpartitions is not highly skewed. We use a skew ratio of 1:99for TDH3 as a threshold. If a query cut results in onepartition having a size greater than hundred times theother, then that cut is ignored. TDH3 algorithm is listed inAlgorithm 3. In Line 4 of Algorithm 3, we use only onequery for the candidate cut. In Line 6, the partition size ratiocondition needs to be satisfied for a feasible cut. If a feasible query cut is not found, then the partition is split along themedian as in Line 11.

5 IMPROVING THE NUMBER OF QUERIES SATISFYING THE IMPRECISION BOUNDS:

The algorithm for TDH2 is listed in Algorithm 2. Thereare two differences compared to TDH1. First, the kd-treetraversal for the for loop in Lines 2-14 is preorder. Second,in Line 14, the query bounds are updated as the partitionsare being added to the output (P). The timecomplexity of TDH2 is which is the same asthat of TDH1. In Section 3, the query imprecision slack is defined as thedifference between the query bound and query imprecision.This query imprecision slack can help

Algorithm 3: TDH3

Input : T, k, Q, and B_{Q_j}
Output: P
1 Initialize Set of Candidate Partitions($CP \leftarrow T$)
2 **for** ($CP_i \in CP$) **do**
    // Depth-first (preorder) traversal
3   Find the set of queries QO that overlap CP_i such that $ic_{CP_i}^{QO_j} > 0$
4   Select query from QO with smallest B_{Q_j}
5   Create query cuts in each dimension
6   Reject cuts with skewed partitions
7   Select dimension and cut having least overall imprecision for all queries in Q
8   **if** (*Feasible cut found*) **then**
9   Create new partitions and add to CP
10   **else**
11   Split CP_i recursively along median till anonymity requirement is satisfied
12   Compact new partitions and add to P
13   Update B_{Q_j} according to $ic_{P_i}^{Q_j}$, $\forall Q_j \in Q$
14 **return** (P)

satisfy queriesthat violate the bounds by only a small margin by increasingthe imprecision of the queries having more slack. In repartitioning step, we consider only thefirst two groups of queries that fall within 10 percent and10-25 percent of the bound only and these queries areadded to the Candidate Query set (CQ), while all queriessatisfying the bounds are added to the query set SQ. Theoutput partitions are all the leaf nodes in the kd-tree. Forrepartitioning, we only consider those pairs of partitionsfrom the output that are siblings in the kd-tree and haveimprecision greater than zero for the queries in the candidatequery set. These pairs of partitions are then added tothe candidate partition set for repartitioning. Mergingsuch a pair of sibling leaf nodes ensures that we still get ahyper-rectangle and the merged partition is non-overlappingwith any other output partition. The repartitioning isfirst performed for the set of queries within 10 percent ofthe bound.

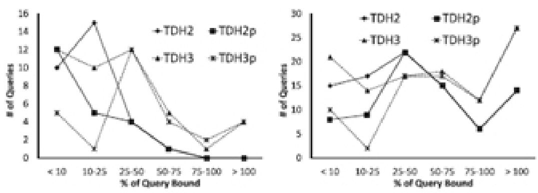
The partitions that are modified are removedfrom the candidate set and then the second group ofqueries is checked. The algorithm for repartitioning islisted as Algorithm 4. In Lines 6-9, we check if a query cutalong any dimension exists that reduces the total imprecisionfor the queries in CQ Set while still satisfying thebounds of the queries in SQ. If such a cut exists, then theold partitions are removed and the new ones are added toOutput P in Lines 11-12. After every iteration, the imprecisionof the queries in Set CQ is checked. If the imprecisionis less than the bound for any query, then as in Line 15that query is moved from Set CQ to SQ. The proposedalgorithm in the experiments satisfies most of the queriesfrom the first group and only a few queries from the secondgroup. This repartitioning step is equivalent to partitioningall the leaf nodes that in the worst case can take time for each candidate query set.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Algorithm 4: Repartitioning

Input : T, k, Q, P, and B_q
Output: P
Initialize SQ, CQ, and CP
Add $q \in Q$ satisfying bound to SQ
Add $q \in Q$ violating bound by 10% to Candidate Query set(CQ)
Add all sibling leaf node pairs having $\sum_{q \in CQ}(ic^{q_j}_{P_i} + ic^{q_j}_{P_{i+1}}) > 0$ to Candidate Partition(CP)
for ($CP_i \in CP$) **do**
  Merge the first pair CP_i and CP_{i+1}
  Select q from CQ with the least imprecision greater than the imprecision bound
  Create the candidate cuts in each dimension
  Select the cut and the dimension satisfying all $q \in SQ$ with the minimum imprecision $\forall q \in CQ$
  if *(feasible cut found)* **then**
    Remove CP_i and CP_{i+1} from CP and P
    Add new partitions to P
    for ($q \in CQ$) **do**
      if ($Imp_q < B_q$) **then**
        Remove q from CQ and add to SQ
return (P)

In the experiments, we set the value of k to 5 and 7with a query imprecision bound of 30 percent of thequery size. The results for repartitioning are given inFig. 18. TDH2p and TDH3p are the results after therepartitioning step. Observe that most of the queries inthe 10 percent group have been satisfied, while for the10-25 percent group, some of these have been satisfiedwhile the others have moved into the first group. Repartitioningof the other groups of queries reduces the totalimprecision but the gains in terms of having morequeries satisfying bounds are not worthwhile.



(a) $k = 5$ and Bound = 30%   (b) $k = 7$ and Bound = 30%

Fig. 5 Improvements after repartitioning for k-anonymity for theCensus data set.

5 EXTENSION FOR PROPOSED SYSTEM:

In this extended model, a user query ismodified by the access control mechanism and only theauthorized tuples are returned.Since Cell level access control is proving a best solution than column level access, along with Column level access we also implemented cell level access control and incremental data for our system. If given a relation T = {A1,A2, . . .,An, . . A2n}, where Ai is anattribute, T* is the anonym zed version of the relation T. We assume that T is a static relational table. Her on this relation T, we performed cell level access control over it.Cell level access control forrelational data is implemented by replacing the unauthorizedcell values by NULL values.

8 CONCLUSIONS:

Our Systemis a combination of secrecy protection and accuracy restrainedmechanisms. The access control mechanism allows onlyauthorized query predicates on sensitive data. The secrecy protectionmodule anonymizes the data to meet privacyrequirements and imprecision constraints on predicates setby the access control mechanism. We formulate this interactionas the problem of k-anonymous Partitioning with ImprecisionBounds (k-PIB). We give hardness results for the k-PIBproblem and present heuristics for partitioning the data tothe satisfy the privacy constraints and the imprecisionbounds. Along with this , we also implemented Cell level access control over incremental data of a relational data.

REFERENCES:

[1] E. Bertino and R. Sandhu, “Database Security-Concepts,Approaches, and Challenges,” IEEE Trans. Dependable and Secure
Computing, vol. 2, no. 1, pp. 2-19, Jan.-Mar. 2005.

[2] P. Samarati, “Protecting Respondents’ Identities in MicrodataRelease,” IEEE Trans. Knowledge and Data Eng., vol. 13, no. 6, pp. 1010-1027, Nov. 2001.

[3] B. Fung, K. Wang, R. Chen, and P. Yu, “Privacy-Preserving DataPublishing: A Survey of Recent Developments,” ACM ComputingSurveys, vol. 42, no. 4, article 14, 2010.

[4] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam,“L-Diversity: Privacy Beyond k-anonymity,” ACM Trans.Knowledge Discovery from Data, vol. 1, no. 1, article 3, 2007.

[5] K. LeFevre, D. DeWitt, and R. Ramakrishnan, “Workload-AwareAnonymization Techniques for Large-Scale Datasets,” ACMTrans. Database Systems, vol. 33, no. 3, pp. 1-47, 2008.