Розглянуто принципи шифрування даних у реляційній базі даних для підтримки конфіденційності й збереження вірогідності при моделюванні поводження інформаційної системи. Описано механізм кодування строкових даних таблиці й запитів до інформаційної системи

Ключові слова: реляційна база даних, моделювання, шифрування

Рассмотрены принципы шифрования данных в реляционной базе данных для поддержания конфиденциальности и сохранения достоверности при моделировании поведения информационной системы. Описан механизм кодирования строковых данных таблицы и запросов к информационной системе

Ключевые слова: реляционная база данных, моделирование, шифрование

The principles of data encryption in a relational database for maintaining confidentiality and maintaining the reliability of the modeling behavior of an information system are considered. The mechanism of encoding of string data tables and query to the information system is described

Keywords: relational database, modeling, encryption

УДК 004.67

МЕТОД ШИФРОВАНИЯ СТРОК В ИМИТАЦИОННОЙ МОДЕЛИ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

А.Б. Кунгурцев
Кандидат технических наук, профессор*
Контактный тел.: (0482) 68-03-02
E-mail: abkun@te.net.ua
С.Л. Зиноватная
Кандидат технических наук, доцент*
Контактный тел.: 067-939-00-94
E -mail: svzino@rambler.ru
Аль Абдо Мунзер
Аспирант*
Контактный тел.: 093-156-26-43
E-mail: munther1427@yahoo.com
*Кафедра системного программного обеспечения
Одесский национальный политехнический университет
пр. Шевченко, 1, Одесса, Украина, 65044

Введение

Моделирование – мощное средство проверки эффективности различных настроек и реструктуризации, выполняемых для реляционной базы данных (РБД) с целью повышения производительности информационной системы (ИС). Однако не всегда подобные мероприятия выполняются тем обслуживающим персоналом, который имеет доступ к информации, содержащейся в БД. Кроме того, в ряде случаев могут быть приглашены специалисты из других организаций. В таких условиях необходимо применять специальные меры для сохранения секретности данных.

Известные способы шифрования [1] не могут быть применены для решения данной задачи, ввиду специфических требований к зашифрованным данным в модели РБД, в частности сохранения исходного распределения значений данных. В работах [2, 3] предложен специальный способ шифрования для числовых данных.

В данной работе представлен метод шифрования строковых данных, который позволяет проводить достоверные исследования с применением имитационной модели РБД.

Постановка задачи

Пусть некоторое поле p таблицы T РБД имеет строковый тип. Требования к шифрованию таких данных следующие:

1) в результате шифрования строк должны быть получены строки; в этом случае сохраняется возможность использовать в модели реальные обращения к РБД и значительно усложняется дешифровка данных;

2) одинаковые строки поля p должны в результате шифрования получить одинаковые коды, что позволит сохранить «семантику» запросов к модели;

3) при шифровании нужно использовать только те символы, которые присутствуют в исходных данных; это поддерживает «правдоподобие» зашифрованных данных;

4) в зашифрованных данных не должно быть строк с длиной, превышающей максимальную длину строки исходных данных; это позволяет сохранить исходный формат данных;

5) в зашифрованных данных не должно быть строк с длиной, которая меньше чем минимальная длина строки исходных данных; в этом случае сохраняется «правдоподобие» зашифрованных данных;

6) нельзя выходить за диапазон (в лексикографическом смысле) исходных данных;

7) необходимо предусмотреть возможность введения специальных ограничений на позиции определенных символов в зашифрованных данных (например, первый символ – только буква или только заглавная буква, некоторый знак – только в конце строки и т. д.).

Метод шифрования заключается в замене исходных строк значениями, сгенерированными таким образом, чтобы удовлетворить изложенные выше требования.

Определение возможного количества генерируемых строк

Прежде всего, необходимо определить максимальное количество строк Nl, которые могут быть сгенерированы для замены (кодирования) оригинальных строк поля p. Если Nl значительно превышает количество исходных данных, то, во-первых, обеспечивается высокая надежность кодирования, а, во-вторых, предоставляется возможность в широком диапазоне изменять количество данных в модели РБД. В противном случае следует применить другие алгоритмы шифрования.

Пусть Msp представляет собой совокупность строк поля p. Устранив повторяющиеся значения и упорядочив оставшиеся строки, получим множество $Msu = \{S_i\}, \overline{i = 1, k_s}$, где k_s – количество различных значений в поле p.

Будем считать, что символ c принадлежит строке S: $c \in_c S$, если существует такое значение j, что $S[j] = c$, где j – порядковый номер символа в строке.

Множество символов, используемых в поле p, определяется следующим образом

$$Mc = \{c_j \mid \exists c_j \in_c S_i\}, \overline{i = 1, k_s}$$

Будем считать, что каждый символ с представлен своим кодом – целым числом в диапазоне от 1 до $k_c = |Mc|$.

Определение количества строк, которое может быть сгенерировано с использованием символов из Mc, является задачей комбинаторики [4]. Однако необходимо учесть возможный диапазон значений поля p с учетом установленных ограничений в лексикографическом смысле.

Определим «наименьшую» строку S1 в таком контексте. Пусть функция fl определяет количество символов в строке.

При сравнении строк разной длины будем считать, что строка с меньшим значением fl дополняется справа символами с кодом 0. Условие выбора S1 имеет следующий вид

$$S1[j] \le S_i[j], \text{если} \exists k \mid k > j \wedge S1[k] < S_i[k], i = \overline{1, k_s}, j = \overline{1, fl(S1)}$$

Определим «наибольшую» строку S2 в лексикографическом смысле.

$$S2[j] \ge S_i[j], \text{если} \exists k \mid k > j \wedge S2[k] > S_i[k], i = \overline{1, k_s}, j = \overline{1, fl(S2)}$$

Для иллюстрации принципа определения возможного количества строк, расположенных между строками S1и S2, рассмотрим следующий пример.

Пусть текстовое поле содержит три строки «BED», «CCD» и «ACB». Тогда S1= «ACB» и S2= «CCD», Mc={A, B, C, D, E}, $k_c = 5$, символы Mc упорядочены согласно алфавиту. Предположим также, что строки должны иметь одинаковую длину (в примере 3 символа).

Используя в качестве первого символа «A» (см. рис. 1), можно создать следующие строки: ADA, ADB, ADC, ADD, ADE, AEA, AEB, AEC, AED, AEE, ACC, ACD, ACE. Строки AA*, AB* и строка ACA выходят за границы диапазона возможных строк.

Рассмотрим возможное множество строк, которые можно создать, используя первый символ строки S2.

Из рис. 2 следует, что, используя в качестве первого символа «C», можно создать следующие строки: CAA, CAB, CAC, CAD, CAE, CBA, CBB, CBC, CBD, CBE, CCA, CCB, CCC. Строку CCE и группы строк CD* и CE* нельзя создавать, поскольку они выходят за диапазон возможных строк.

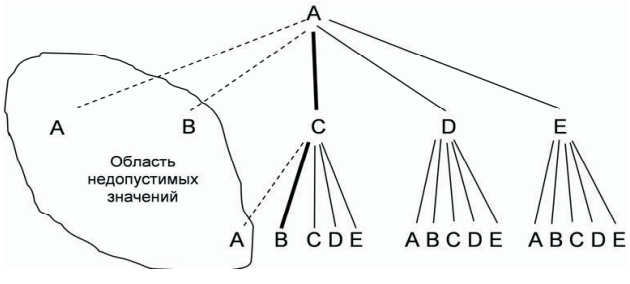

Рис. 1. Строки, которые можно создать с первым символом строки S1

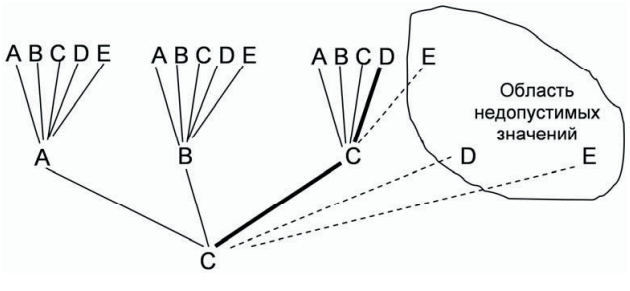

Рис. 2. Строки, которые можно создать с первым символом строки S2

Рассмотрим множество строк, которые можно создать, используя в качестве первого символа любой символ $c \in Mc \mid c > S1[1] \wedge c < S2[1]$. В нашем случае был использован символ «B» (см. рис. 3).

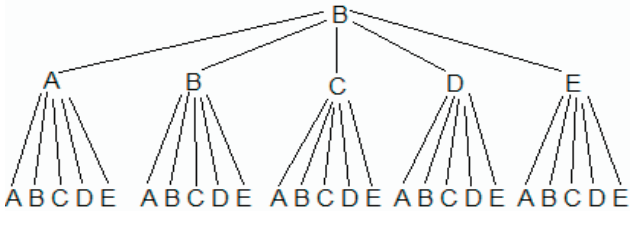

Рис. 3. Строки, состоящие из символов множества Mc, с первым символом «B»

В общем случае, начиная с первого символа строки S1, можно создать Ns1 строк, полагая, что длина строки постоянна и равна l.

$$Ns1 = (k_c - S1[2])\prod_{i=2}^{l} k_c + (k_c - S1[3])\prod_{i=3}^{l} k_c + ... +$$

$$+(k_c - S1[l])$$

или

$$Ns1 = \sum_{i=2}^{l}((k_c - S1[i])\prod_{j=1+i}^{l} k_c) \quad (1)$$

В общем случае, начиная с первого символа строки S2, можно создать Ns2 строк, полагая, что длина строки постоянна и равна l.

$$Ns2 = S2[2]\prod_{i=2}^{l} k_c + S2[3]\prod_{i=3}^{l} k_c + ... + S1[l]$$

или

$$Ns2 = \sum_{i=2}^{l}(S2[i]\prod_{j=1+i}^{l} k_c) \quad (2)$$

Для любого другого первого символа c1 возможное количество строк

$$Nc^1 = \prod_{i=2}^{l} k_c \quad (3)$$

Таким образом, общее количество строк N, которые можно создать в диапазоне от S1 до S2, определяется по формуле

$$N = Ns1 + Ns2 + (S2[1] - S1[1] - 1)\prod_{i=2}^{l} k_c$$

В общем случае, если длины строк могут лежать в диапазоне от l1до l2, то выражения (1) – (3) принимают соответственно следующий вид

$$Ns1l = \sum_{l=l1}^{l2}\sum_{i=2}^{l}((k_c - S1[i])\prod_{j=1+i}^{l} k_c)$$

$$Ns2l = \sum_{l=l1}^{l2}\sum_{i=2}^{l}(S2[i]\prod_{j=1+i}^{l} k_c)$$

$$Nc^1l = \sum_{l=l1}^{l2}\prod_{i=2}^{l} k_c$$

$$Nl = \sum_{l=l1}^{l2}(Ns1 + Ns2 + Nc^1l) \quad (4)$$

Замена оригинальных строк кодами

Согласно с ограничениями, установленными при определении (4) введём понятие упорядоченного множества всех возможных значений, которые могут быть поставлены в соответствие значениям множества Msu : $Mks = \{S_{k1}, S_{k2}, ..., S_{knl}\}$

Суть кодирования будет заключаться в выборе для каждого элемента $S_i \in Msu$ некоторого кода в виде элемента $S_{kj} \in Mks$.

Учитывая обеспечение возможной вставки новых записей в кодируемую таблицу, а также сохранения лексикографического порядка оригинальных и закодированных строк, будем выбирать элементы из множества Mks последовательно с некоторым шагом δ.

На рис. 4 приведена схема кодирования. Здесь Mk – множество выбранных кодов. Пара значений $<S^1, S^2>$, где $S^1 \in Msu$, а $S^2 \in Mk$, и S1 соответствует элемент Mks, отстоящий на δ шагов от S_{k1} в прямом направлении, а S2 соответствует элемент Mks, отстоящий на δ шагов от S_{knl} в обратном направлении. Множество пар $<S^1, S^2>$ образуют таблицу шифрования Tsh.

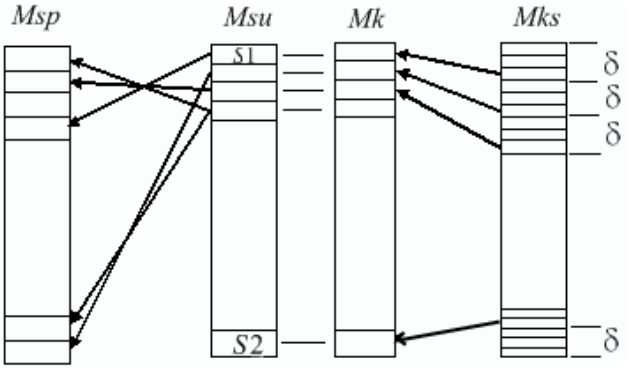

Рис. 4. Схема кодирования строк

Вопрос выбора значений интервала δ является очень важным, поскольку именно этот атрибут кодирования способен обеспечить защиту исходных данных.

Определим средний интервал между закодированными строками:

$$\Delta = Nl / k_s .$$

Δ определяет среднее количество строк, которые можно вставить между двумя существующими строками.

Теперь можно ввести понятие ключа шифрования k, значения которого рекомендуется устанавливать в следующем диапазоне

$$0.5 \le k < 1$$

Предлагаемый диапазон значений k позволяет, с одной стороны, сохранить достаточно большое значение интервала δ, а, с другой стороны, обеспечивает широкие возможности выбора конкретного значения ключа шифрования.

С учётом выбора значения ключа шифрования k величина интервала δ определяется по формуле

$$\delta = \Delta * k$$

Значение ключа шифрования определяется лицом, ответственным за безопасность базы данных (администратором БД). Произвольный выбор ключа в указанном диапазоне позволяет получать различные результаты шифрования. Сложность дешифровки зависит от количества знаков после запятой в ключе. Так, например, при использовании 5 знаков потребуется выполнить до 55000 циклов дешифровки. Если же учесть, что наложенные ограничения на процесс шифрования могут создать для зашифрованных данных «достоверный» вид, то процесс дешифровки существенно усложнится.

Администратор БД может установить постоянное значение ключа k для шифрования всех записей таблицы РБД либо определить ряд диапазонов значений ключа k, что ещё больше усложнит расшифровку закодированных строк.

Процедура определения различных значений ключа k может быть выполнена как «вручную», так и автоматически путем выбора одной из корректирующих функций подготовленных заранее. Для иллюстрации такого принципа изменения ключа используем функцию sin

$$k = k_0 + k_\delta * \sin(x * i + \phi),$$

где k_0 – первоначально выбранное (основное) значение ключа;

k_δ – определяет амплитуду вариативной части ключа;

i – определяет номер кодируемой строки;

x и φ – период повторения и фазовый сдвиг корректирующей функции соответственно.

Шифрование запросов к модели

Для исследования ИС с помощью модели могут быть использованы реальные запросы, записанные ранее в журнале транзакций. Для запросов, использующих поле p, необходимо выполнить процедуру шифрования.

Пусть в запросе используется константное значение S_Q поля p. Если $S_Q \in Ms$, то S_Q заменяется на значение S_k из пары $< S_Q, S_k >$ таблицы Tsh.

Если $S_Q \notin Ms$, то необходимо выполнить шифрование строки S_Q.

Если $S1 \le S_Q \le S2$, то определяется положение S_Q среди ранее зашифрованных строк:

$$(S_{q-1}{}^1 \le S_Q \le S_{q+1}{}^1).$$

С помощью генератора строк определяется количество шагов d от формирования закодированной строки, соответствующей $S_{q-1}{}^1$ до момента формирования закодированной строки, соответствующей $S_{q+1}{}^1$. Далее генератора формирует результирующую строку S_{Qk}, отстоящую от v на d/2 цикла.

Если $S_Q < S1$, то S_{Qk} генерируется как строка, отстоящая от S_{k1} на половину расстояния до строки S^2, соответствующей S1, в прямом направлении.

Если $S_Q > S2$, то S_{Qk} генерируется как строка, отстоящая от S_{knl} на половину расстояния до строки S^2, соответствующей S2, в обратном направлении.

Новая пара $< S_Q, S_{Qk} >$ записывается в таблицу Tsh. При необходимости корректируются значения S1 и S2.

Схема взаимодействия элементов метода шифрования представлены на рис. 5.

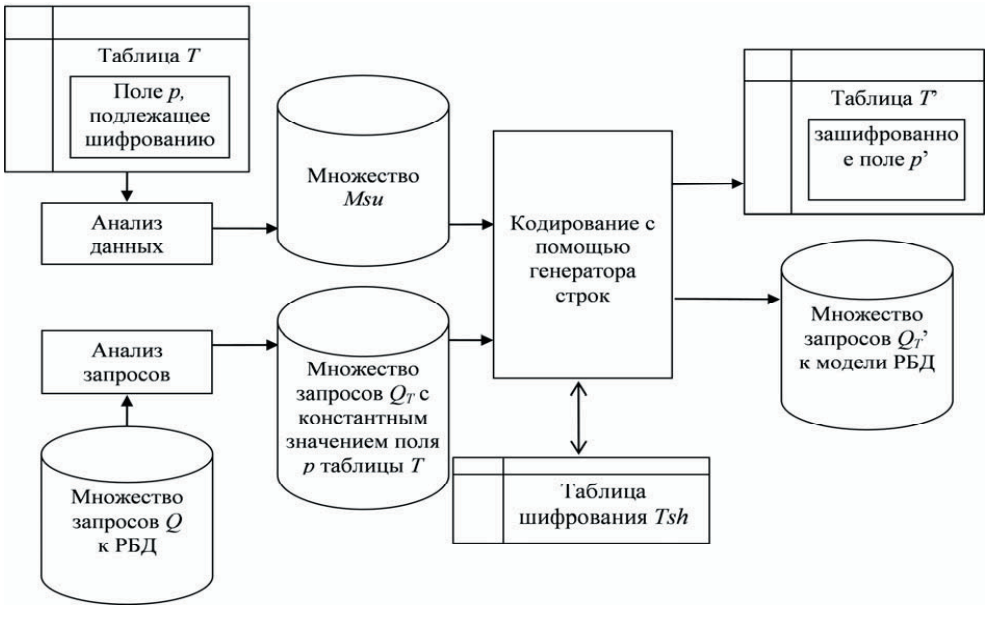

Рис. 5. Схема шифрования строк

Выводы

Предложенный метод шифрования строковых данных позволяет удовлетворить все ограничения, накладываемые на построение имитационной модели РБД, и обеспечивает достаточно надёжное кодирование информации. Поскольку метод не предназначен для ИС, содержащих особо секретные данные, можно ожидать, что стоимость дешифровки во многих случаях превысит стоимость имеющихся данных.

Литература

1. Панасенко, С.П. Алгоритмы шифрования. Специальный справочник [Текст] / С.П. Панасенко – СПб. : БХВ-Петербург, 2009. – 576 с.
2. Kungurtsev, A.B. Ensuring data confidentiality in relational database modeling [Текст] / A.B. Kungurtsev, S.L. Zinovatnaya, Al Abdo Munzer // Компютърни науки и технологии. Технически университет – Варна, 2009. – 1/2009, година VII. – С. 57 – 61.
3. Поточняк, Я.В. Шифрування даних у моделях реляційної бази даних на рівні таблиць [Текст] / Я.В. Поточняк, Мунзер Аль Абдо. // Матеріали 45 наукової конференції молодих дослідників ОНПУ «Сучасні інформаційні технології та телекомунікаційні мережі». – Одеса, 2010. – С. 18 – 19.
4. Липский, В. Комбинаторика для программистов [Текст] / В. Липский– М. : Мир, 1988. – 213 с.