электронный научно-технический журнал
ИНЖЕНЕРНЫЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51036. ISSN 2307-0595

Проблемы объединения реляционных таблиц

 # 07, июль 2014

Брешенков А. В., Белоус В. В., Белошицкий Д. А.

УДК: 681.3.07

Россия, МГТУ им. Баумана
Breshenkov@rambler.ru

В работах [1-3] обоснована актуальность проблемы преобразования заполненных нереляционных таблиц в реляционные таблицы, сформулированы задачи преобразования, намечены пути решения отдельных задач. Здесь рассматривается одна из этих задач – задача объединения импортированных таблиц в базах данных.

Если не рассматривать множество специфических особенностей конкретных БД, необходимость объединения содержимого таблиц может возникнуть в двух случаях. В первом случае данные поступают из нескольких источников в центр и там они сводятся в одну таблицу для дальнейшего анализа и обработки. Во втором случае, когда создается БД на основе существующей информации табличного вида, необходимо выявить сходные по структуре и смысловому содержанию таблицы и объединить их в одну.

В соответствии с положениями реляционной алгебры объединение двух отношений есть множество всех кортежей, принадлежащих каждому из исходных отношений [4]. Другими словами в результате объединения двух таблиц создается третья таблица, которая включает в себя все записи 1-й таблицы и недостающие записи 2-й таблицы. Исходные отношения должны быть совместимы по объединению. Отношения называются совместимыми по объединению, если они базируются на одном и том же числе одних и тех же доменов (столбцов).

В качестве иллюстрации операции объединения используем отношения, представленные таблицами Таблица 1, Таблица 2 и Таблица 3. В Таблице 1 и Таблице 2 приведены операнды операции отношение, в Таблице 3 приведен результат выполнения этой операции.

Таблица 1

ФАМИЛИЯ	ГОД РОЖДЕНИЯ	ГОРОД
Чугунов	1955	Ногинск
Конев	1958	Козельск
Дребизова	1959	Моршанск
Караваев	1957	Семикино
Попова	1951	Ледово

Таблица 2

ФАМИЛИЯ	ГОД РОЖДЕНИЯ	ГОРОД
Харченко	1954	Киев
Умуралиев	1954	Астана
Комлев	1958	Москва
Мялицына	1959	Москва
Попова	1951	Ледово
Чугунов	1955	Ногинск

Таблица 3

ФАМИЛИЯ	ГОД РОЖДЕНИЯ	ГОРОД
Чугунов	1955	Ногинск
Конев	1958	Козельск
Деребизова	1959	Моршанск
Караваев	1957	Семикино
Попова	1951	Ледово
Харченко	1954	Киев
Умуралиев	1954	Астана
Комлев	1958	Москва
Мялицына	1959	Москва

Из анализа Таблицы 1 и Таблицы 2 нетрудно заметить, что операнды операции объединения совместимы. Действительно, они состоят из одних и тех же столбцов – заголовки столбцов одинаковые, содержимое одноименных столбцов совпадают по типу. Результаты объединения, как видно из табл. 3., представляют собой все записи 1-й таблицы и недостающие записи 2-й таблицы. Действительно, т.к. первая и последняя запись 1-й таблицы присутствуют и в 1-й и во 2-й таблице, то в 3-й таблице (результирующей) они встречаются единожды.

Смысловое содержание приведенного примера может быть таким. В таблицах – источниках приведены списки участников конференции. Некоторые участники по какой-либо причине зарегистрировались у двух регистраторов. В базе данных необходимо сохранить данные обо всех участниках конференции без дублирования записей. Применение оператора объединения для таблиц двух регистраторов и позволяет получить нужную таблицу. Если участников конференции регистрировало N регистраторов, то оператор объединения отношений необходимо выполнить N -1 раз. Причем в качестве 1-го операнда при первой итерации объединения выступает 1-я таблица, а в качестве 2-го операнда - 2-я таблица. При второй итерации объединения в качестве 1-го операнда выступает результат выполнения предыдущего объединения, а в качестве 2-го операнда - 3-я таблица и т.д.

Запрос, который необходимо выполнить для объединения 2-х таблиц, выглядит следующим образом:

INSERT INTO Список1 (Фамилия, [Год рождения], Город)

SELECT Список2.Фамилия, Список2.[Год рождения], Список2.Город FROM Список2;

Здесь с помощью конструкции “SELECT Список2.Фамилия, Список2.[Год рождения], Список2.Город FROM Список2” из таблицы “Список2” выбираются значения трех полей. Посредством конструкции “INSERT INTO Список1 (Фамилия, [Год рождения], Город)” значения выбранных полей добавляются в соответствующие столбцы таблицы “Список1”. На рис. 1 приведено содержимое таблицы “Список1” после выполнения запроса.

Список1 : таблица

Фамилия	Год рождения	Город
Чугунов	1955	Ногинск
Конев	1958	Козельск
Деребизова	1959	Моршанск
Караваев	1957	Семикино
Попова	1951	Ледово
Харченко	1954	Киев
Умуралиев	1954	Астана
Комлев	1958	Москва
Мялицына	1959	Москва
Попова	1951	Ледово
Чугунов	1955	Ногинск

Рис. 1. Результат выполнения запроса на добавление

Как видно из рисунка, в таблице имеет место дублирование записей. Поэтому для исключения дублирования необходимо выполнить еще один запрос вида:

SELECT DISTINCT Список1.* INTO Список_общий

FROM Список1;

Здесь посредством конструкций "SELECT DISTINCT Список1.*” и ”FROM Список1” выбираются все значения полей таблицы “Список1”. Посредством режима ”DISTINCT” из выбранного списка исключаются повторяющиеся записи, а посредством конструкции ”INTO Список_общий” выбранные записи помещаются в новую таблицу ”Список_общий”. В результате выполнения этого запроса сформируется таблица ”Список_общий”, которая имеет вид рис. 2.

Список_общий : таблица

Фамилия	Год рождения	Город
Деребизова	1959	Моршанск
Караваев	1957	Семикино
Комлев	1958	Москва
Конев	1958	Козельск
Мялицына	1959	Москва
Попова	1951	Ледово
Умуралиев	1954	Астана
Харченко	1954	Киев
Чугунов	1955	Ногинск

Рис. 2. Таблица без дублирования записей

Как видно из рисунка, нужный результат достигнут – сформирован общий список, а дублирование записей исключено.

В рассмотренном примере, который полностью удовлетворяет условию совместимости по объединению, проблем в процессе формирования сводной таблицы практически не возникает. Это видно из сказанного выше.

В реальных ситуациях дело нередко обстоит несколько сложнее, и возникают проблемы, которые необходимо решать. Рассмотрим эти ситуации в порядке возрастания сложности.

Исходные таблицы по своей природе удовлетворяют требованиям совместимости, а по форме – нет.

Такого рода ситуации возникают когда:

- заголовки одинаковых по смыслу столбцов у объединяемых таблиц отличаются;
- порядок столбцов первой таблицы – операнда не совпадает с порядком столбцов второй таблицы – операнда.

В формате Microsoft Access таблица приведена на рис. 3.

Список0 : таблица

Фамилия	Год рождения	Город
Чугунов	1955	Ногинск
Конев	1958	Козельск
Деребизова	1959	Моршанск
Караваев	1957	Семикино
Попова	1951	Ледово

Рис. 3. Первый операнд операции объединения в формате Microsoft Access

В качестве 2-го операнда используем таблицу, приведенную в формате Microsoft Access на рис. 4.

Список3 : таблица

Откуда прибыл	Участник конференции	Дата рождения
Киев	Харченко	1954
Астана	Умуралиев	1954
Москва	Комлев	1958
Москва	Мялицына	1959
Ледово	Попова	1951
Ногинск	Чугунов	1955

Рис. 4. Второй операнд операции объединения в формате Microsoft Access

Как видно из рис. 4, заголовки 2-го операнда не совпадают с заголовками 1-го операнда. Кроме того, порядок столбцов 2-го операнда не совпадает с порядком столбцов 1-го операнда. Однако визуальный анализ содержимого 1-го и 2-го операндов (обеих таблиц) позволяет сделать вывод о том, что структуры этих таблиц совпадают, и каждому столбцу 1-й таблицы находится соответствующий столбец 2-й таблицы. В связи с этим администратор БД может принять решение о том, в каком порядке содержимое столбцов 2-й таблицы добавлять к содержимому 1-й таблицы. Свое решение администратор может отразить в бланке запроса на добавление. Соответствующий бланк запроса в системе Microsoft Access приведен на рис. 5.

Таблица, в которую предполагается добавить записи, указывается в меню Запрос/Добавление. В нашем случае указана таблица с именем “Список0”. Из рисунка видно, каким образом установлено соответствие между полями таблиц “Список3” и “Список0”. Соответствующий запрос в режиме SQL можно просмотреть с помощью меню Вид/Режим SQL

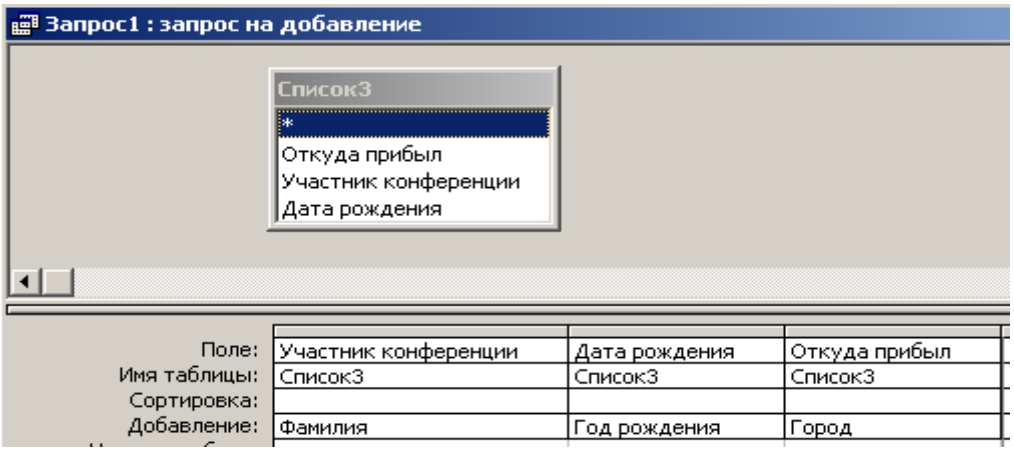

Рис. 5. Бланк запроса на объединение двух таблиц

. Этот запрос, который может быть использован с незначительными модификациями в любой СУБД, выглядит следующим образом:

INSERT INTO Список0 (Фамилия, [Год рождения], Город)

SELECT Список3.[Участник конференции], Список3.[Дата рождения], Список3.[Откуда прибыл]

FROM Список3;

Из анализа запроса можно сделать заключение о том, каким образом установлено соответствие между полями двух таблиц.

Результат выполнения запроса – это таблица “Список0” с добавленными записями. Она приведена на рис. 6.

Список0 : таблица

Фамилия	Год рождения	Город
Чугунов	1955	Ногинск
Конев	1958	Козельск
Деребизова	1959	Моршанск
Караваев	1957	Семикино
Попова	1951	Ледово
Харченко	1954	Киев
Умуралиев	1954	Астана
Комлев	1958	Москва
Мялицына	1959	Москва
Попова	1951	Ледово
Чугунов	1955	Ногинск

Рис. 6. Результат выполнения запроса на объединение

Как видно из рисунка, результат выполнения запроса ничем не отличается от результата выполнения запроса, приведенного на рис. 1. Таким образом, в результате формирования описанного запроса получен требуемый результат.

Как следует из сказанного выше, рассмотренная проблема несложно решается, если администратор БД способен принять решение о том, каким образом установить соответствие между столбцами объединяемых таблиц. Очень часто такое решение не представляет существенных проблем, т.к. изначально, как правило, предполагается объединение таких таблиц, для которых это имеет смысл и суть столбцов очевидна.

Теоретически можно выявить соответствующие столбцы и автоматически. Но для этого нужно с помощью соответствующих программных средств анализировать семантику содержимого столбцов, что сложно и практически неприемлемо.

Исходные таблицы удовлетворяют требованиям совместимости, результирующую таблицу необходимо обновлять.

Ситуации такого рода возникают в том случае, если в центральной БД аккумулируются данные, поступающие из различных источников. При этом формат БД центра и форматы БД источников заранее оговорены и совпадают. В этом случае проблема совместимости не стоит. Однако возникают проблемы двух типов.

Проблема первого типа связана с тем, что наряду с новыми записями в центр передаются и те записи, которые уже ранее были переданы и добавлены в таблицы центральной БД.

В этом случае, если не принять специальных мер, в центральной БД возможно дублирование записей, что в конечном итоге приводит к искажению информации, противоречивости БД. Передавать же только те данные, которые не передавались ранее затруднительно, а часто и невозможно. Во-первых, механизм отслеживания хронологии импортированных данных из БД регионов нетривиален, а во-вторых, нередко имеется необходимость передачи уже импортированных данных, т.к. эти записи с данными могут быть обновлены, например, изменена стадия выполнения проекта.

Проблема второго типа возникает в связи с тем, что нередко в центр передаются обновленные записи таблиц, которые ранее в центр уже передавались. В связи с этим возникает необходимость обновления всех записей в центре, которые повторно экспортированы из регионов и которые в регионах были изменены. Проблема несколько упрощается в связи с тем, что, как показывает опыт работы с такого рода механизмом передачи и обработки данных, число обновляемых полей невелико и их состав регламентирован.

Исходные таблицы частично удовлетворяют требованиям совместимости.

Ситуация такого рода чаще всего возникает, когда осуществляется проектирование баз данных на основе использования существующей информации табличного вида. Например, возникает необходимость проектирования БД на основе использования набора заполненных электронных таблиц.

Нередко в наборе электронных таблиц можно выявить группы таблиц, в которых значительная часть атрибутов совпадает. Такое положение вещей может быть обусловлено различными причинами. В частности, столбцы могут быть продублированы по ошибке, одинаковые столбцы в разных таблицах используются из соображений удобства визуального анализа данных в рамках одной таблицы, могут быть и другие причины. В этом случае имеет место дублирование данных. Если это иногда приемлемо и даже оправданно в электронных таблицах, то в БД дублирование информации недопустимо. Дублирование БД приводит к противоречивости БД, ее избыточности. В связи с этим при проектировании БД на основе использования существующей информации табличного вида необходимо решить две проблемы.

Первая проблема связана с выявлением таблиц, в которых значительная часть атрибутов совпадает. В рамках этой проблемы необходимо убедиться не только в том, что имеются совпадающие заголовки таблиц в разных таблицах, но и в том, что эти совпадения неслучайны и суть этих совпадающих заголовков одинакова.

Вторая проблема решается тогда, когда выявлены таблицы с одинаковыми по смыслу атрибутами. Тогда необходимо выявленные таблицы объединить.

В общем случае состав атрибутов даже в таблицах, в которых имеются совпадения атрибутов, может быть различный и тогда задача объединения таблиц усложняется по сравнению с рассмотренными выше примерами.

Список литературы.

1. Брешенков А.В. Неформальная постановка проблемы преобразования информации табличного вида в файлы баз данных. Сб. трудов АУ МВД России "Актуальные вопросы технологий в деятельности органов внутренних дел". - М.: 2004. – 20 с.
2. Брешенков А.В., Бараков Д.Д. Вопросы преобразования электронных таблиц в таблицы реляционных баз данных. Современные информационные технологии. Сб. трудов каф. ИУ-6, посвященный 175-летию МГТУ им. Н.Э. Баумана. - М.: Эликс +, 2004.-5 с.
3. Брешенков А.В., Бараков Д.Д. Методика назначения ключевых полей в заполненных реляционных таблицах. Современные информационные технологии. Сб. трудов каф. ИУ-6, посвященный 175-летию МГТУ им. Н.Э. Баумана. - М.: Эликс +, 2005. – 16 с.
4. Дейт К., Дж. Введение в системы баз данных. 8-е изд.: Пер. с англ.- М.: Вильямс, 2005. - 1328 с.