**Лабораторная работа № 9**

**«Инструкция MERGE»**

Время выполнения: 80 минут.

Цель: Изучить принципы слияния таблиц, настройки и параметры инструкции MERGE.

Методические указания.

MERGE – операция в языке T-SQL, при которой происходит обновление, вставка или удаление данных в таблице на основе результатов соединения с данными другой таблицы или SQL запроса. Другими словами, с помощью MERGE можно осуществить слияние двух таблиц, т.е. синхронизировать их.

В операции MERGE происходит объединение по ключевому полю или полям основной таблицы (*в которой и будут происходить все изменения*) с соответствующими полями другой таблицы или результата запроса. В итоге если условие, по которому происходит объединение, истина (WHEN MATCHED), то мы можем выполнить операции обновления или удаления, если условие не истина, т.е. отсутствуют данные (WHEN NOT MATCHED), то мы можем выполнить операцию вставки (*INSERT добавление данных*), также если в основной таблице присутствуют данные, которые отсутствуют в таблице (*или результате запрос*а) источника (WHEN NOT MATCHED BY SOURCE), то мы можем выполнить обновление или удаление таких данных.

В дополнение к основным перечисленным выше условиям можно указывать «*Дополнительные условия поиска*», они указываются через ключевое слово AND.

Синтаксис выглядит так:

MERGE <Основная таблица> AS TARGET

USING <Таблица или запрос источника>

AS SOURCE

ON <Условия объединения>

[ WHEN MATCHED [ AND <Доп. условие> ]

THEN <UPDATE или DELETE>

[ WHEN NOT MATCHED [ AND Доп. условие> ]

THEN <INSERT> ]

[ WHEN NOT MATCHED BY SOURCE [ AND <Доп. условие> ]

THEN <UPDATE или DELETE> ] [ ...n ] ;

Обратите внимание, что в конце инструкции MERGE обязательно ставится точка с запятой.

Например,

|  |
| --- |
| --Заполнение сводной таблицы при помощи инструкций MERGE |
|  |
| --Вставка данных из Группа1 |
| --RunnersData - целевая сводная таблица |
| **MERGE** RunnersData **AS** TARGET |
| --RunnerGroup1 - источник |
| **USING** RunnerGroup1 |
| **AS** SOURCE |
| --Условие соединения |
| **ON** **(**TARGET**.**RunnerID **=** SOURCE**.**RunnerID**)** |
| --В источнике и целевой таблице есть совпадение по условию, |
| --если есть несовпадение по времени забега (NULL или не равно источнику), то обновить его в целевой таблице |
| **WHEN** MATCHED **AND** TARGET**.**RunnerTimeSeconds1 **<>** SOURCE**.**RunnerTimeSeconds **OR** TARGET**.**RunnerTimeSeconds1 **IS** **NULL** |
| **THEN** **UPDATE** **SET** TARGET**.**RunnerTimeSeconds1 **=** SOURCE**.**RunnerTimeSeconds |
| --В целевой таблице нет записи с нужным RunnerID, добавить новую запись |
| **WHEN** **NOT** MATCHED **BY** TARGET |
| **THEN** **INSERT** **(**RunnerID**,** RunnerTimeSeconds1**)** **VALUES** **(**SOURCE**.**RunnerID**,** SOURCE**.**RunnerTimeSeconds**)**  **И т.д. для других таблиц;** |

Например

|  |
| --- |
| --Создание таблицы с лучшими результатами забегов каждого спортсмена |
| **CREATE** **TABLE** RunnerBestTime **(** |
| RunnerID int **NOT** **NULL** **PRIMARY** **KEY,** |
| BestTimeSeconds float **NULL** |
| **);** |
|  |
|  |
| --Вставка данных в RunnerBestTime из RunnersData |
| **MERGE** RunnerBestTime **AS** TARGET |
| **USING** |
| **(SELECT** RunnerID**,** |
| **(** |
| **SELECT** **MIN(**BestRunnerTime**)** |
| **FROM** **(VALUES** **(**RunnerTimeSeconds1**),** **(**RunnerTimeSeconds2**),** **(**RunnerTimeSeconds3**))** **AS** x**(**BestRunnerTime**)** |
| **)** **AS** BestTime |
| **FROM** RunnersData**)** **AS** SOURCE |
| **ON** TARGET**.**RunnerID **=** SOURCE**.**RunnerID |
| **WHEN** MATCHED **AND** TARGET**.**BestTimeSeconds **<>** SOURCE**.**BestTime **OR** TARGET**.**BestTimeSeconds **IS** **NULL** |
| **THEN** **UPDATE** **SET** TARGET**.**BestTimeSeconds **=** SOURCE**.**BestTime |
| **WHEN** **NOT** MATCHED **BY** TARGET |
| **THEN** **INSERT** **(**RunnerID**,** BestTimeSeconds**)** **VALUES** **(**SOURCE**.**RunnerID**,** SOURCE**.**BestTime**);** |

Задания на лабораторную работу.

В соревнованиях по бегу участвовало несколько спортсменов. Для соревнования выделили 3 бегущих группы. У спортсмена была возможность бежать в любой из этих групп, при этом можно было бежать 2 или 3 раза, в этом случае учитывается лучший результат. Каждому участвующему спортсмену дается свой номер. Ниже приведены таблицы с результатами по каждой группе с номером атлета и временем, за которое он пробежал дистанцию.

|  |  |
| --- | --- |
| Номер спортсмена | Время, с |
| 371 | 56 |
| 372 | 57 |
| 373 | 58 |
| 374 | 56 |
| 375 | 51 |
| 376 | 62 |
| 377 | 61 |
| 378 | 67 |
| 379 | 48 |
| 380 | 50 |

Таб. 1. Группа 1.

|  |  |
| --- | --- |
| Номер спортсмена | Время, с |
| 372 | 61 |
| 374 | 58 |
| 376 | 57 |
| 378 | 54 |
| 380 | 53 |

Таб. 2. Группа 2.

|  |  |
| --- | --- |
| Номер спортсмена | Время, с |
| 370 | 58 |
| 371 | 57 |
| 372 | 67 |
| 373 | 62 |
| 374 | 55 |
| 375 | 52 |
| 376 | 60 |
| 377 | 58 |
| 378 | 59 |
| 379 | 53 |
| 380 | 54 |

Таб. 3. Группа 3.

1. Создайте в своей базе данных три таблицы и заполните их, аналогично таблицам 1, 2 и 3. Номер спортсмена должен заполнятся не с клавиатуры, а с помощью свойства identity (x, y).
2. Создайте сводную таблицу, в которой будут храниться все данные из первых трех. Добавьте в нее данные с помощью операции MERGE. Незаполненные ячейки должны быть пустые (NULL).

|  |  |  |  |
| --- | --- | --- | --- |
| Номер спортсмена | Время 1, с | Время 2, с | Время 3, с |
| 370 | NULL | NULL | 58 |
| 371 | 56 | NULL | 57 |
| 372 | 57 | 61 | 67 |
| 373 | 58 | NULL | 62 |
| 374 | 56 | 58 | 55 |
| 375 | 51 | NULL | 52 |
| 376 | 62 | 57 | 60 |
| 377 | 61 | NULL | 58 |
| 378 | 67 | 54 | 59 |
| 379 | 48 | NULL | 53 |
| 380 | 50 | 53 | 54 |

Таб. 4. Возможный вид сводной таблицы для п.2.

1. Создайте новую таблицу, в которой будет лучший результат каждого спортсмена. Сортировку выполните с помощью операции MERGE. Обратите внимание, что лучшее время – это минимальное время.

|  |  |
| --- | --- |
| Номер спортсмена | Лучшее время |
| 370 | 58 |
| 371 | 56 |
| 372 | 57 |
| 373 | 58 |
| 374 | 55 |
| 375 | 51 |
| 376 | 57 |
| 377 | 58 |
| 378 | 54 |
| 379 | 48 |
| 380 | 50 |

Таб. 5. Возможный вид таблицы лучших результатов для п.3.

1. Выведите на экран номер победителя и его результат любым удобным способом.

|  |  |
| --- | --- |
| Победитель | Результат |
| 379 | 48 |

Таб. 6 Возможный вид таблицы победителя и его результат для п.4.

1. Аналогично выведите на экран худший результат за все соревнование.
2. Отчет должен содержать все коды и результаты ваших действий.