GBC КМС SQL Аналитические функции

Домашнее задание

# Введение

В этом документе – задачи для тех, кто прошел курс «КМС SQL Аналитические функции». Презентацию по курсу см. на странице [Основы баз данных и SQL](https://wiki.glowbyteconsulting.com/pages/viewpage.action?pageId=60194989).

# Задачи

## Переходы задач внутри команды

**Предыстория:** есть несколько команд разработки и набор их задач. Каждая задача в момент времени находится у одного разработчика и может быть передана внутри команды или другой команде. И, допустим, нам в будущем хочется собрать статистику, как часто задачи передавались между командами.

**Данные на входе** – таблица TASK\_HISTORY:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ROW\_ID** | **TASK\_ID** | **ASSIGNEE** | **TEAM** | **START\_DATE** | **END\_DATE** |
| 1 | 1 | Саша из A | А | 01.01.2019 | 02.01.2019 |
| 2 | 1 | Катя из А | А | 02.01.2019 | 03.01.2019 |
| 3 | 1 | Вася из Б | Б | 03.01.2019 | 05.01.2019 |
| 4 | 1 | Саша из А | А | 05.01.2019 | 15.01.2019 |
| 5 | 1 | Петя из Б | Б | 15.01.2019 | 30.01.2019 |
| 6 | 1 | Вася из Б | Б | 30.01.2019 | 31.01.2019 |
| 7 | 1 | Петя из Б | Б | 31.01.2019 | 01.02.2019 |
| 8 | 2 | Саша из A | А | 01.01.2019 | 02.01.2019 |
| 9 | 2 | Катя из А | А | 02.01.2019 | 03.01.2019 |
| 10 | 2 | Саша из А | А | 03.01.2019 | 05.01.2019 |
| 11 | 2 | Вася из Б | Б | 05.01.2019 | 15.01.2019 |

Это история того, как разработчики передавали друг другу задачи. Здесь:

* ROW\_ID – уникальный номер строки в таблице
* TASK\_ID – номер задачи
* ASSIGNEE – сотрудник, которому передали в работу задачу
* TEAM – команда сотрудника
* START\_DATE, END\_DATE – период времени, пока задача была у сотрудника в работе.

Для статистики того, как задачи передавали между командами, нам неинтересны последовательные переходы задачи внутри одного отдела. И мы хотим убрать такие переходы из данных.

**Что нужно сделать:**

1. Написать запрос SELECT и в нем вычислить столбец SWITCH\_NUM – счетчик, который отсчитывает последовательные переходы задачи внутри одной команды:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ROW\_ID** | **TASK\_ID** | **ASSIGNEE** | **TEAM** | **START\_DATE** | **END\_DATE** | **SWITCH\_NUM** |
| 1 | 1 | Саша из A | А | 01.01.2019 | 02.01.2019 | 1 |
| 2 | 1 | Катя из А | А | 02.01.2019 | 03.01.2019 | 2 |
| 3 | 1 | Вася из Б | Б | 03.01.2019 | 05.01.2019 | 1 |
| 4 | 1 | Саша из А | А | 05.01.2019 | 15.01.2019 | 1 |
| 5 | 1 | Петя из Б | Б | 15.01.2019 | 30.01.2019 | 1 |
| 6 | 1 | Вася из Б | Б | 30.01.2019 | 31.01.2019 | 2 |
| 7 | 1 | Петя из Б | Б | 31.01.2019 | 01.02.2019 | 3 |
| 8 | 2 | Саша из A | А | 01.01.2019 | 02.01.2019 | 1 |
| 9 | 2 | Катя из А | А | 02.01.2019 | 03.01.2019 | 2 |
| 10 | 2 | Саша из А | А | 03.01.2019 | 05.01.2019 | 3 |
| 11 | 2 | Вася из Б | Б | 05.01.2019 | 15.01.2019 | 1 |

1. С помощью предыдущего SELECT написать итоговый запрос DELETE, который удалит из TASK\_HISTORY все последовательные переходы задач в рамках отдела. Т.е. все строки, где SWITCH\_NUM > 1.

## Версионный список задач.

**Предыстория:** есть команда разработчиков и их задачи в Jira. Допустим, мы планируем собрать статистику по тому, на ком и когда были какие задачи.

**На входе** есть следующие данные:

1. Таблица TASK – список задач команды:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TASK\_ID** | **LAST\_STATUS** | **LAST\_ASSIGNEE** | **CREATED\_DATE** | **CLOSED\_DATE** |
| 1 | Закрыто | Коля | 01.01.2019 | 01.02.2019 |
| 2 | Закрыто | Петя | 02.01.2019 | 02.02.2019 |
| 3 | Новое | Саша | 01.02.2019 | NULL |

Содержит поля:

* + TASK\_ID – номер задачи
  + LAST\_STATUS – последний статус по этой задаче
  + LAST\_ASSIGNEE – последний сотрудник, который работал над задачей
  + CREATED\_DATE – дата создания задачи
  + CLOSED\_DATE – дата закрытия задачи

1. Таблица TASK\_ASSIGNEE\_HISTORY – история переходов задач между сотрудниками:

|  |  |  |  |
| --- | --- | --- | --- |
| **TASK\_ID** | **OLD\_ASSIGNEE** | **NEW\_ASSIGNEE** | **CHANGE\_DATE** |
| 1 | NULL | Саша | 02.01.2019 |
| 1 | Саша | Петя | 15.01.2019 |
| 1 | Петя | Коля | 31.01.2019 |
| 3 | Петя | Коля | 02.02.2019 |
| 3 | Коля | Петя | 03.02.2019 |
| 3 | Петя | Саша | 04.02.2019 |

Содержит поля:

* + TASK\_ID – номер задачи
  + OLD\_ASSIGNEE – предыдущий сотрудник
  + NEW\_ASSIGNEE – следующий сотрудник
  + CHANGE\_DATE – дата, когда сменился сотрудник

1. Таблица TASK\_STATUS\_HISTORY – история того, как менялись статусы задач:

|  |  |  |  |
| --- | --- | --- | --- |
| **TASK\_ID** | **OLD\_STATUS** | **NEW\_STATUS** | **CHANGE\_DATE** |
| 1 | Новое | Оценка | 03.01.2019 |
| 1 | Оценка | Разработка | 05.01.2019 |
| 1 | Разработка | Тестирование | 15.01.2019 |
| 1 | Тестирование | Внедрение | 30.01.2019 |
| 1 | Внедрение | Закрыто | 01.02.2019 |
| 2 | Новое | Оценка | 03.01.2019 |
| 2 | Оценка | Разработка | 10.01.2019 |
| 2 | Разработка | Тестирование | 20.01.2019 |
| 2 | Тестирование | Внедрение | 01.02.2019 |
| 2 | Внедрение | Закрыто | 02.02.2019 |

Содержит поля:

* + TASK\_ID – номер задачи
  + OLD\_STATUS – предыдущий статус задачи
  + NEW\_STATUS – новый статус задачи
  + CHANGE\_DATE – дата, когда поменяли статус

**Что нужно сделать:**

1. Написать запрос, который объединит таблицы TASK\_ASSIGNEE\_HISTORY и TASK в единую версионную таблицу TASK\_ASSIGNEE\_HISTORY\_VERSIONED:

|  |  |  |  |
| --- | --- | --- | --- |
| **TASK\_ID** | **ASSIGNEE** | **START\_DATE** | **END\_DATE** |
| 1 | NULL | 01.01.2019 | 02.01.2019 |
| 1 | Саша | 02.01.2019 | 15.01.2019 |
| 1 | Петя | 15.01.2019 | 31.01.2019 |
| 1 | Коля | 31.01.2019 | 01.02.2019 |
| 2 | Петя | 02.01.2019 | 02.02.2019 |
| 3 | Петя | 01.02.2019 | 02.02.2019 |
| 3 | Коля | 02.02.2019 | 03.02.2019 |
| 3 | Петя | 03.02.2019 | 04.02.2019 |
| 3 | Саша | 04.02.2019 | NULL |

1. Аналогично с TASK\_STATUS\_HISTORY – создать по ней TASK\_STATUS\_HISTORY\_VERSIONED.
2. Объединить историю изменений статусов и переходов между сотрудниками в единую версионную таблицу **TASK\_HISTORY**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TASK\_ID** | **STATUS** | **ASSIGNEE** | **START\_DATE** | **END\_DATE** |
| 1 | Новое | NULL | 01.01.2019 | 02.01.2019 |
| 1 | Новое | Саша | 02.01.2019 | 03.01.2019 |
| 1 | Оценка | Саша | 03.01.2019 | 05.01.2019 |
| 1 | Разработка | Саша | 05.01.2019 | 15.01.2019 |
| 1 | Тестирование | Петя | 15.01.2019 | 30.01.2019 |
| 1 | Внедрение | Петя | 30.01.2019 | 31.01.2019 |
| 1 | Внедрение | Коля | 31.01.2019 | 01.02.2019 |
| 1 | Закрыто | Коля | 01.02.2019 | 01.02.2019 |
| 2 | Новое | Петя | 02.01.2019 | 03.01.2019 |
| 2 | Оценка | Петя | 03.01.2019 | 10.01.2019 |
| 2 | Разработка | Петя | 10.01.2019 | 20.01.2019 |
| 2 | Тестирование | Петя | 20.01.2019 | 01.02.2019 |
| 2 | Внедрение | Петя | 01.02.2019 | 02.02.2019 |

Эта задача еще называется «пересечение версионностей» – см. аналогичную задачу 2 с [Учебные задачи по SAS](https://wiki.glowbyteconsulting.com/pages/viewpage.action?pageId=5243316).