**Триггер** – это особый вид хранимой процедуры, предназначенной для обработки событий в БД. Поддерживается два типа триггеров: *DDL-триггеры* и *DML-триггеры*. Для каждого типа определено свое семейство событий, обработку которых триггер этого типа может выполнять.

**DML-**триггеры бывают двух типов: AFTER-триггеры и INSTEAD OF-триггеры. Триггеры типа AFTER исполняются *после* выполнения оператора, вызвавшего соответствующее событие. При этом создаются автоматически две псевдотаблицы INSERTED и DELETED.

Триггер типа INSTEAD OF выполняется *вместо* оператора, вызвавшего соответствующее событие. Выполнение INSTEAD OF триггера предшествует проверке установленных для таблицы ограничений целостности.

Лабораторная работа по базам данных SQL Server Management Studio (SSMS) с использованием триггеров включает в себя работу с псевдотаблицами INSERTED и DELETED, создание AFTER-триггера TRIG\_Tov\_Ins для отслеживания событий INSERT, DELETE, и UPDATE в таблице Товары, а также управление порядком выполнения триггеров.

1. \*\*Описание работы триггера TRIG\_Tov\_Ins:\*\*

- \*\*INSERT:\*\* При вставке новой строки в таблицу Товары, триггер отражает событие INSERT в таблицу TR\_Tov, записывая информацию из псевдотаблицы INSERTED.

- \*\*DELETE:\*\* При удалении строки из таблицы Товары, триггер отражает событие DELETE в таблицу TR\_Tov, записывая информацию из псевдотаблицы DELETED.

- \*\*UPDATE:\*\* При обновлении строки в таблице Товары, триггер отражает событие UPDATE в таблицу TR\_Tov, записывая информацию из обеих псевдотаблиц INSERTED и DELETED.

2. \*\*Пример DML-операторов:\*\*

```sql

insert into Товары(Наименование, Цена, Количество) values('Стол', 140, 20);

delete from Товары where Наименование = 'Стол';

update Товары set Количество = 20 where Наименование = 'Стул';

```

3. \*\*Описание порядка выполнения триггеров:\*\*

- Созданы три триггера (AUD\_AFTER\_UPDA, AUD\_AFTER\_UPDB, AUD\_AFTER\_UPDC) для реагирования на событие UPDATE в таблице Товары.

- Изменен порядок выполнения триггеров с помощью системных процедур SP\_SETTRIGGERORDER.

4. \*\*Инструкция по проверке порядка выполнения триггеров:\*\*

```sql

select t.name, e.type\_desc

from sys.triggers t join sys.trigger\_events e

on t.object\_id = e.object\_id

where OBJECT\_NAME(t.parent\_id) = 'Товары' and e.type\_desc = 'UPDATE';

```

5. \*\*Пример ограничения целостности с использованием триггера:\*\*

```sql

alter table Товары add constraint Цена check(Цена >= 15);

update Товары set Цена = 10 where Наименование = 'Стул';

```

6. \*\*Пример INSTEAD OF-триггера для предотвращения удаления строк:\*\*

```sql

-- INSTEAD OF-триггер

create trigger TRIG\_PreventDelete

on Товары

instead of delete

as

begin

print 'Удаление строк запрещено';

end;

```

7. \*\*Триггер DDL\_PRODAJI для запрета изменений в базе данных ПРОДАЖИ:\*\*

```sql

create trigger DDL\_PRODAJI

on database

for DDL\_DATABASE\_LEVEL\_EVENTS

as

begin

print 'Тип события: ' + @t;

print 'Имя объекта: ' + @t1;

print 'Тип объекта: ' + @t2;

raiserror(N'Операции с таблицей Товары запрещены', 16, 1);

rollback;

end;

```

Примечание: Ваш запрос по изменению порядка выполнения триггеров с использованием системных процедур был воспроизведен в предыдущем контексте.

Эти теоретические основы должны помочь вам лучше понять и описать содержание лабораторной работы.