

Отчёт по лабораторной работе

Дисциплина: Базы данных

Тема: SQL-программирование: Триггеры, вызовы процедур

Выполнил студент гр. 43501/1

(подпись) Г. А. Жемелев

Преподаватель

(подпись) А. В. Мяснов

“12” декабря 2016 г.

Санкт-Петербург

2016

Цель работы

Познакомиться с возможностями реализации обработки данных на стороне сервера с помощью хранимых процедур и триггеров.

Программа работы

1. Создать два триггера: один триггер для автоматического заполнения ключевого поля, второй триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице
2. Создать триггер в соответствии с индивидуальным заданием, полученным у преподавателя
3. Создать триггер в соответствии с индивидуальным заданием, вызывающий хранимую процедуру
4. Продемонстрировать результаты преподавателю

Ход работы

Триггер для автоматического заполнения ключевого поля при добавлении новой категории:

```
SET TERM ^;
CREATE TRIGGER set_catID BEFORE INSERT ON CATEGORIES
AS
BEGIN
    IF(NEW.ID IS NULL) THEN
        NEW.ID = GEN_ID(Seq_catID, 1);
END^
SET TERM ;^
COMMIT;
```

Триггеры для контроля целостности данных в подчиненной таблице AVAILABILITY при добавлении/удалении записей в таблице GOODS:

```
SET TERM ^;
CREATE TRIGGER ADD_AVAIL AFTER INSERT ON goods AS
    DECLARE VARIABLE storageID INTEGER;
BEGIN
    /* Добавить в availability новый товар в количестве 0 шт на все склады. */
    FOR SELECT ID FROM storages INTO :storageID
    DO
        INSERT INTO availability VALUES (:storageID, NEW.article, 0);
END^

CREATE TRIGGER DEL_AVAIL AFTER DELETE ON goods AS
BEGIN
    /* Удалить из availability удаленный из goods товар. */
```

```

DELETE FROM availability WHERE article = OLD.article;
END^
SET TERM ;^
COMMIT;

```

Индивидуальное задание состоит из следующих пунктов:

1. Увеличить количество товара на складе при поступлении товара.
2. При изменении статуса заказа, если статус заказа равен 4 (CANCELLED), а прежний статус был 3 (COMPLETED) или 2 (READY), то увеличить количество каждого товара из заказа на каком-либо одном складе на количество, указанное в заказе.

Следующий скрипт описывает два триггера, решающих данные задачи:

```

SET TERM ^;
CREATE TRIGGER UPD_AVAIL_ARRIVED AFTER INSERT ON arrivals AS
BEGIN
    /* При поступлении товара увеличить количество товара NEW.article на складе
    NEW.sto_ID на NEW.Quantity */

    UPDATE availability
        SET Quantity = Quantity + NEW.Quantity
        WHERE article = NEW.article AND sto_ID = NEW.sto_ID;
END^

CREATE TRIGGER UPD_AVAIL_RETURNED AFTER UPDATE ON orders AS
    DECLARE VARIABLE goodArticle INTEGER;
    DECLARE VARIABLE goodQuantity INTEGER;
BEGIN
    /* Если статус заказа равен 4 (CANCELLED), а прежний статус был
    3 (COMPLETED) или 2 (READY), то увеличить количество каждого товара
    из заказа на любом складе на количество, указанное в заказе. */

    IF (NEW.Status = 2 AND (OLD.Status = 3 OR OLD.Status = 2)) THEN
        FOR SELECT article, Quantity FROM ORDERS_DISTRIBUTION
            WHERE order_No = NEW.Order_No
            INTO :goodArticle, :goodQuantity
            DO
                UPDATE availability
                    SET Quantity = Quantity + :goodQuantity
                    WHERE article = :goodArticle
                    ROWS 1;
END^
SET TERM ;^
COMMIT;

```

Заметим, что первый триггер не выполняет проверку существования поступившего артикула в базе данных, что может привести к нарушению ограничения целостности внешнего ключа. Поэтому создадим еще один триггер, который будет срабатывать перед добавлением новой записи в таблицу ARRIVALS и проверять, что пришедший товар имеет артикул, присутствующий в базе данных.

Соответствующий код:

```
SET TERM ^;
CREATE EXCEPTION ex_ArticleNotExist 'На склад поступил товар с несуществующим артику-
лом!';

CREATE PROCEDURE checkArticle (chkArt INTEGER) RETURNS (result SMALLINT) AS
BEGIN
    SELECT COUNT(*) FROM goods WHERE Article = :chkArt INTO result;
END^

CREATE TRIGGER CHK_ARRIVED BEFORE INSERT ON arrivals AS
    DECLARE VARIABLE isExist SMALLINT;
BEGIN
    EXECUTE PROCEDURE checkArticle(NEW.article) RETURNING_VALUES :isExist;
    IF (isExist = 0) THEN EXCEPTION ex_ArticleNotExist;
END^
SET TERM ;^
COMMIT;
```

Вызов оператора EXCEPTION не позволит добавить ошибочную запись в таблицу поступлений и выведет информативное сообщение для пользователя.

Выводы

Триггеры представляют собой особые процедуры, которые хранятся и выполняются на стороне сервера. Нельзя вызвать триггер напрямую – его запуск осуществляется автоматически при наступлении события, относящегося к заданной таблице или к базе данных в целом.

Триггер позволяет описать последовательность действий, которые необходимо выполнить до или после некоторого события (триггер для таблицы связывается с событием и фазой), например, проверить допустимость вносимой в таблицу записи и прекратить операцию, если запись не удовлетворит заданным условиям. Для каждой комбинации фаза-событие может быть определено более одного триггера. Порядок, в котором они выполняются, может быть указан явно при создании триггеров.

Механизм триггеров предоставляет широкие возможности для автоматизации работы с базами данных и контроля целостности. В разрабатываемой БД существует таблица (AVAILABILITY) которая заполняется, обновляется и очищается только при помощи триггеров, что позволяет всегда иметь оперативную и достоверную информацию о наличии тех или иных товаров на всех складах.