

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА



Институт радиоэлектроники и информационных технологий

## ОТЧЕТ

по лабораторной работе №2  
«Изучение основ T-SQL»

по дисциплине

Базы данных

РУКОВОДИТЕЛЬ:

\_\_\_\_\_

(подпись)

профессор каф. ВСТ

Мисевич П. В.

(фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_

(подпись)

Сухоруков В.А.

(фамилия, и.,о.)

19-B-2

(шифр группы)

Работа защищена «\_\_» \_\_\_\_\_

С оценкой \_\_\_\_\_

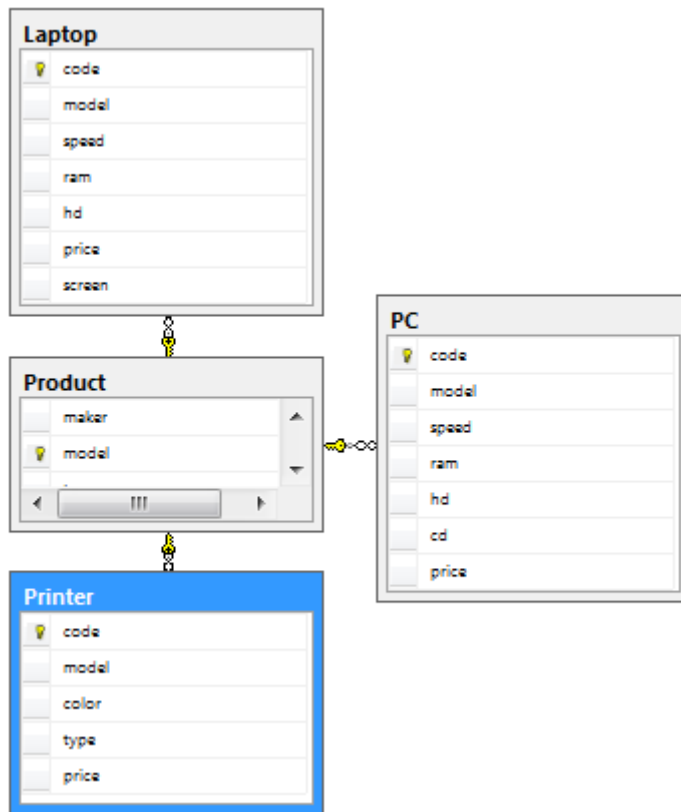
Нижний Новгород 2021

## Цель

Изучение операторов языка T-SQL и их применение на тестовой БД.

### Протокол применения операторов языка T-SQL:

- 1) База данных computers состоит из таблиц Laptop, Product, PC и Printer.



Содержание таблицы PC:

| code | model | speed | ram | hd | cd  | price    |
|------|-------|-------|-----|----|-----|----------|
| 1    | 1232  | 500   | 64  | 5  | 12x | 600,0000 |
| 2    | 1121  | 750   | 128 | 14 | 40x | 850,0000 |
| 3    | 1233  | 500   | 64  | 5  | 12x | 600,0000 |
| 4    | 1121  | 600   | 128 | 14 | 40x | 850,0000 |
| 5    | 1121  | 600   | 128 | 8  | 40x | 850,0000 |
| 6    | 1233  | 750   | 128 | 20 | 50x | 950,0000 |
| 7    | 1232  | 500   | 32  | 10 | 12x | 400,0000 |
| 8    | 1232  | 450   | 64  | 8  | 24x | 350,0000 |
| 9    | 1232  | 450   | 32  | 10 | 24x | 350,0000 |
| 10   | 1260  | 500   | 32  | 10 | 12x | 350,0000 |
| 11   | 1233  | 900   | 128 | 40 | 40x | 980,0000 |
| 12   | 1233  | 800   | 128 | 20 | 50x | 970,0000 |

Оператор **SELECT** осуществляет выборку из базы данных и имеет наиболее сложную структуру среди всех операторов языка SQL.

`SELECT * FROM PC`

осуществляет выборку всех записей из объекта БД с именем PC.

|    | code | model | speed | ram | hd | cd  | price  |
|----|------|-------|-------|-----|----|-----|--------|
| 1  | 1    | 1232  | 500   | 64  | 5  | 12x | 600,00 |
| 2  | 2    | 1121  | 750   | 128 | 14 | 40x | 850,00 |
| 3  | 3    | 1233  | 500   | 64  | 5  | 12x | 600,00 |
| 4  | 4    | 1121  | 600   | 128 | 14 | 40x | 850,00 |
| 5  | 5    | 1121  | 600   | 128 | 8  | 40x | 850,00 |
| 6  | 6    | 1233  | 750   | 128 | 20 | 50x | 950,00 |
| 7  | 7    | 1232  | 500   | 32  | 10 | 12x | 400,00 |
| 8  | 8    | 1232  | 450   | 64  | 8  | 24x | 350,00 |
| 9  | 9    | 1232  | 450   | 32  | 10 | 24x | 350,00 |
| 10 | 10   | 1260  | 500   | 32  | 10 | 12x | 350,00 |
| 11 | 11   | 1233  | 900   | 128 | 40 | 40x | 980,00 |
| 12 | 12   | 1233  | 800   | 128 | 20 | 50x | 970,00 |

`SELECT DISTINCT speed, ram`

`FROM PC`

`ORDERBY ram DESC;`

`ORDERBY ram DESC` упорядочивает результирующий набор по объему оперативной памяти в порядке убывания. `DISTINT` показывает только уникальные строки.

|   | speed | ram |
|---|-------|-----|
| 1 | 600   | 128 |
| 2 | 750   | 128 |
| 3 | 800   | 128 |
| 4 | 900   | 128 |
| 5 | 450   | 64  |
| 6 | 500   | 64  |
| 7 | 450   | 32  |
| 8 | 500   | 32  |

2) Оператор **INSERT** вставляет новые записи в таблицу.

`INSERT INTO Product (type, model, maker)`

`VALUES ('PC', 1157, 'B');`

добавит в таблицу модель ПК 1157 производителя B.

До добавления записи:

| maker | model | type    |
|-------|-------|---------|
| B     | 1121  | PC      |
| A     | 1232  | PC      |
| A     | 1233  | PC      |
| E     | 1260  | PC      |
| A     | 1276  | Printer |

После добавления записи:

|   |      |         |
|---|------|---------|
| B | 1121 | PC      |
| B | 1157 | PC      |
| A | 1232 | PC      |
| A | 1233 | PC      |
| E | 1260 | PC      |
| A | 1276 | Printer |

Создавая таблицу в столбцах можно задать значения по умолчанию:

```
CREATE TABLE product_D (
maker char (1) NULL,
model varchar (4) NULL,
type varchar (7) NOT NULL DEFAULT 'PC' );
```

В столбцы с значением по умолчанию задавать данные не обязательно:

```
INSERT INTO Product_D (model, maker)
VALUES (1157, 'B');
```

| maker | model | type |
|-------|-------|------|
| B     | 1157  | PC   |

3) Оператор **UPDATE** изменяет имеющиеся данные в таблице.

Содержание таблицы Laptop:

| code | model | speed | ram | hd | price     | screen |
|------|-------|-------|-----|----|-----------|--------|
| 1    | 1298  | 350   | 32  | 4  | 700,0000  | 11     |
| 2    | 1321  | 500   | 64  | 8  | 970,0000  | 12     |
| 3    | 1750  | 750   | 128 | 12 | 1200,0000 | 14     |
| 4    | 1298  | 600   | 64  | 10 | 1050,0000 | 15     |
| 5    | 1752  | 750   | 128 | 10 | 1150,0000 | 14     |
| 6    | 1298  | 450   | 64  | 10 | 950,0000  | 12     |

Можем уменьшить все цены портативных компьютеров на 10 процентов с помощью следующего оператора:

UPDATE Laptop

SET price = price\*0.9;

| code | model | speed | ram | hd | price     | screen |
|------|-------|-------|-----|----|-----------|--------|
| 1    | 1298  | 350   | 32  | 4  | 630,0000  | 11     |
| 2    | 1321  | 500   | 64  | 8  | 873,0000  | 12     |
| 3    | 1750  | 750   | 128 | 12 | 1080,0000 | 14     |
| 4    | 1298  | 600   | 64  | 10 | 945,0000  | 15     |
| 5    | 1752  | 750   | 128 | 10 | 1035,0000 | 14     |
| 6    | 1298  | 450   | 64  | 10 | 855,0000  | 12     |

Если нужно поставить жесткие диски объемом 20 Гбайт на портативные компьютеры с памятью менее 128 Мбайт и 40 гигабайтные — на остальные портативные компьютеры, то можно написать такой запрос (CASE позволяет изменять данные в зависимости от столбца):

UPDATE Laptop

SET hd = CASE

WHEN ram < 128

THEN 20

ELSE 40

END;

| code | model | speed | ram | hd | price     | screen |
|------|-------|-------|-----|----|-----------|--------|
| 1    | 1298  | 350   | 32  | 20 | 630,0000  | 11     |
| 2    | 1321  | 500   | 64  | 20 | 873,0000  | 12     |
| 3    | 1750  | 750   | 128 | 40 | 1080,0000 | 14     |
| 4    | 1298  | 600   | 64  | 20 | 945,0000  | 15     |
| 5    | 1752  | 750   | 128 | 40 | 1035,0000 | 14     |
| 6    | 1298  | 450   | 64  | 20 | 855,0000  | 12     |

4) Оператор **DELETE** удаляет строки из временных или постоянных базовых таблиц, представлений или курсоров.

DELETE FROM Laptop

WHERE screen < 12;

удалит из таблицы Laptop все портативные компьютеры с размером экрана меньше 12 дюймов.

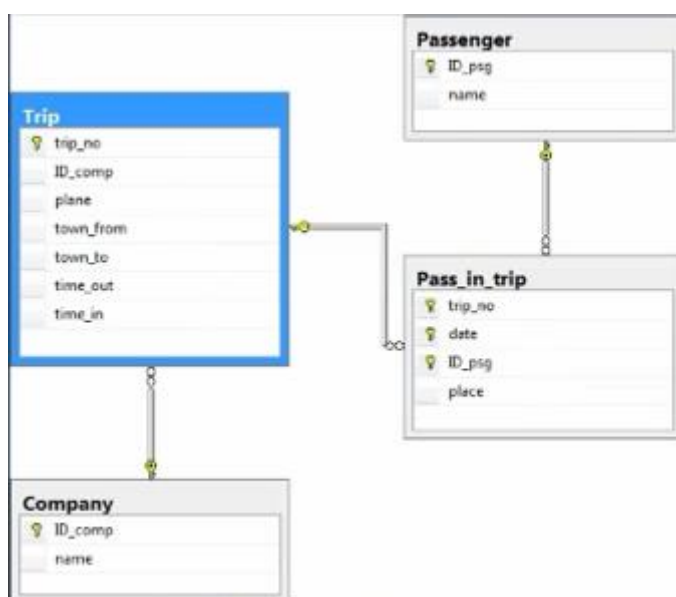
| code | model | speed | ram | hd | price     | screen |
|------|-------|-------|-----|----|-----------|--------|
| 2    | 1321  | 500   | 64  | 20 | 873,0000  | 12     |
| 3    | 1750  | 750   | 128 | 40 | 1080,0000 | 14     |
| 4    | 1298  | 600   | 64  | 20 | 945,0000  | 15     |
| 5    | 1752  | 750   | 128 | 40 | 1035,0000 | 14     |
| 6    | 1298  | 450   | 64  | 20 | 855,0000  | 12     |

**DELETE FROM Product WHERE type = 'pc' AND model NOT IN (SELECT model FROM PC );**

удалит из таблицы Product модели ПК, которых нет в таблице PC.

| maker | model | type    |
|-------|-------|---------|
| B     | 1157  | PC      |
| A     | 1232  | PC      |
| A     | 1233  | PC      |
| E     | 1260  | PC      |
| A     | 1276  | Printer |
| D     | 1288  | Printer |
| A     | 1298  | Laptop  |
| C     | 1321  | Laptop  |
| A     | 1401  | Printer |
| A     | 1408  | Printer |
| D     | 1433  | Printer |
| E     | 1434  | Printer |
| B     | 1750  | Laptop  |
| A     | 1752  | Laptop  |
| E     | 2112  | PC      |
| E     | 2113  | PC      |

5) База данных аеро состоит из таблиц Company, Trip, Passenger и Pass\_in\_trip.



**Datepart** возвращает целое число, представляющее собой указанную аргументом datepart часть заданной вторым аргументом даты (date).

```
SELECT DATEPART (hh, time_out)*60 + DATEPART(mi, time_out)
FROM Trip
WHERE trip_no = 1123
```

Данный запрос выведет время вылета рейса 1123 в минутах.

|   |     |
|---|-----|
| 1 | 980 |
|---|-----|

Функция **DATENAME** (datepart, date) возвращает символьное представление составляющей (datepart) указанной даты (date).

```
SELECT DATENAME(weekday, '20031231') + ', ' + DATENAME(day,
'20031231') + ' ' + DATENAME(month, '20031231') + ' ' +
DATENAME(year, '20031231');
```

|   |                        |
|---|------------------------|
| 1 | среда, 31 Декабрь 2003 |
|---|------------------------|

### Вывод

В результате выполнения лабораторной работы были изучены операторы языка T-SQL и их применение на тестовой БД.