

Минобрнауки России

федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
«Санкт-Петербургский государственный технологический институт (технический университет)»

Кафедра систем автоматизированного проектирования и управления

**Иванов А.А.
Жадановская Н.П.**



Базы данных

Контрольные работы

Учебное пособие
для студентов заочной формы обучения

Санкт-Петербург
2016

УДК 681.3.066

А.А. Иванов Базы данных. Контрольные работы : учеб. пособие для студентов заочной формы обучения / А.А. Иванов, Н.П. Жадановская. – СПб. : СПбГТИ(ТУ), 2016. – 47 с.

В учебное пособие включены задания для выполнения контрольных работ, предназначенных для изучения языков запросов, используемых в большинстве современных систем управления базами данных.

Контрольные работы включают практические задания, при выполнении которых необходимо применить теоретические знания, полученные при изучении курса «Базы данных», в частности раздела «Языки запросов».

Учебное пособие предназначено для студентов очной и заочной формы обучения по направлению подготовки 09.03.01 «Информатика и вычислительная техника».

Учебное пособие может также использоваться для студентов очной и заочной формы обучения по направлению подготовки 15.03.04 «Автоматизация технологических процессов».

Ил. 1, табл. 1, библиогр. назв. 8

Рецензенты:

Утверждено на заседании учебно-методической комиссии факультета информатики и управления

Рекомендовано к изданию РИСо СПбГТИ(ТУ)

© СПбГТИ(ТУ), 2016.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	Ошибка! Закладка не определена.
Контрольная работа №1. ЗАПРОСЫ НА ВЫБОРКУ ДАННЫХ.	<u>7</u>
Контрольная работа №2. ЗАПРОСЫ НА МОДИФИКАЦИЮ ДАННЫХ.	<u>22</u>
Вариант16. ПРИМЕРЫ ВЫПОЛНЕНИЯ КОНТРОЛЬНЫХ РАБОТ	34
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	<u>45</u>
ПРИЛОЖЕНИЕ А Структура учебной базы данных «Борей».....	<u>46</u>

ВВЕДЕНИЕ

Сегодня существует большое количество различных систем управления базами данных (СУБД). Это обуславливает значительные функциональные различия этих систем.

В данном учебном пособии представлены задания на выполнение контрольных работ по изучению структурированного языка запросов (SQL) .

В учебном пособии составлены две контрольные работы, включающие в себя содержание работы, варианты заданий и примеры выполнения.

Каждая контрольная работа содержит три части: теоретическую, тестовые вопросы и практическую, связанную с анализом кода на SQL и разработку собственного текста запроса заданного функционала.

Тестовые задания должны показать качество проработки теоретических вопросов.

Студент самостоятельно выбирает задания контрольной работы. Номер варианта соответствует номеру первой буквы фамилии студента согласно таблице 1.

Таблица 1 - Распределение вариантов заданий

Первая буква фамилии студента	Номер варианта	Первая буква фамилии студента	Номер варианта
А, Б	1	Р, С	9
В, Г	2	Т, У	10
Д, Е	3	Ф, Х	11
Е, Ж	4	Ц, Ч	12
З, И	5	Ш, Щ	13
К, Л	6	Э, Ю	14
М, Н	7	Я	15
О, П	8	Образец	16

Студенту необходимо представить отчёт о выполненных контрольных работах в распечатанном виде и в электронном виде на любом носителе информации. Отчёт должен включать: титульный лист, условие задачи (либо поставленный вопрос) и предложенное решение поставленных задач (либо развернутый ответ на поставленный теоретический вопрос). На титульном листе отчёта о выполнении контрольных работ необходимо указать фамилию,

имя и отчество студента, номер учебной группы, номер контрольной работы, номер варианта.

Содержание контрольных работ:

✓ **КОНТРОЛЬНАЯ РАБОТА №1**

Тема: Запросы на выборку данных

✓ **КОНТРОЛЬНАЯ РАБОТА №2**

Тема: Запросы на модификацию данных

Условия задач одинаковы для всех вариантов контрольной работы и приведены ниже.

В качестве учебной базы данных выступает база «Борей» (см. приложение А).

Содержание работы:

Задание №1 – Аналитический обзор

Напишите ответ на поставленный вопрос. Ответ на вопросы желательно давать в развернутом виде. Вы должны показать глубину ваших знаний в ваших ответах. Эти знания помогут также выполнить остальные части контрольной работы.

Задание №2 – Тестовые задания

2.1 Вставьте пропущенное слово

2.2 Выберите правильные ответы (ответ м.б. не один) на предложенное тестовое задание

Задание №3 – Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если ошибка найдена, то поясните, в чем она заключается. Если же запрос верный, то приведите результат выполнения этого запроса.

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия.

В **Примере решения** (Вариант 16) дано более подробное руководство для выполнения этой контрольной работы.

Приступая к выполнению контрольных работ, рекомендуется ознакомиться со следующими методическими материалами:

А.А. Иванов Базы данных. Базовый курс : учеб. пособие для студентов заочной формы обучения / А.А. Иванов, Н.П. Жадановская. – СПб. : СПбГТИ(ТУ), 2016. – 101 с.

Далее представлены варианты заданий контрольных работ №1 и №2

Перечень основной и дополнительной учебной литературы, рекомендуемой для освоения дисциплины :

основная:

- 1 Дьюсон, Р. SQL Server 2008 для начинающих разработчиков / Р. Дьюсон ; пер. с англ. – СПб. : БХВ-Петербург, 2009. – 688 с.
- 2 Норенков, И. П. Автоматизированные информационные системы : учеб. пособие / И. П. Норенков. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2011. – 342 с.
- 3 Советов, Б. Я. Базы данных. Теория и практика / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. – 2-е изд., стер. – М. : Юрайт., 2012. – 463 с.
- 4 Советов, Б. Я. Информационные технологии : учеб. для вузов / Б. Я. Советов, В. В. Цехановский. – 3-е изд., стер. – М. : Высш. шк., 2006. – 263 с.

дополнительная:

- 1 Дейтел, П. Как программировать на Visual C# 2012 : Включая работу в Windows 7 и Windows 8 / П. Дейтел, Х. Дейтел. - 5-е изд. - М. ; СПб. ; Н. Новгород : Питер, 2014. - 864 с.
- 2 Кляйн, К. Е. SQL. Справочник / К. Е. Кляйн, Д. Кляйн, Б. Хант ; пер. с англ. – 3-е изд. – М. ; СПб. : Символ-Плюс, 2010. – 651 с.
- 3 Кузнецов, С. Д. Базы данных. Модели и языки / С. Д. Кузнецов. – СПб. : Бином-Пресс, 2008. – 720 с.
- 4 Культин, Н. Б. C++ Builder / Н. Б. Культин. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2008. – 463 с.
- 5 Культин, Н. Б. Microsoft Visual C# в задачах и примерах / Н. Б. Культин. – СПб. : БХВ-Петербург, 2009. – 309 с.
- 6 Малыгина, М. П. Базы данных : основы, проектирование, использование / М. П. Малыгина. – 2-е изд. – СПб. : БХВ-Петербург, 2006. – 517 с.
- 7 Петкович, Д. Microsoft SQL Server 2008. Руководство для начинающих / Д. Петкович ; пер. с англ. – СПб. : БХВ-Петербург, 2009. – 730 с.
- 8 Хотек, М. Microsoft SQL Server 2008. Реализация и обслуживание : пер. с англ. / М. Хотек. - М. : Рус. ред., 2012. - 553 с.

вспомогательная:

- 9 Карпова, Т. С. Базы данных : модели, разработка, реализация : учебник / Т. С. Карпова. – СПб. : Питер, 2001. – 303 с.
- 10 Системы управления базами данных и знаний / А. Н. Наумов, А. М. Вендров, В. К. Иванов. – М. : Финансы и статистика, 1991. – 348 с.
- 11 Хомоненко, А. Д. Базы данных : учебник для высших учебных заведений / А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев ; под ред. А. Д. Хомоненко. — 6-е изд. доп. — Санкт-Петербург : КОРОНА -Век, 2011.—

1 КОНТРОЛЬНАЯ РАБОТА №1

Контрольная работа 1: Вариант 1

1 Аналитическая часть

Реализация языка запросов в СУБД, разработанных в России.

2 Тестовые задания

2.1 В СУБД предусмотрены правила, обеспечивающие данных, что позволяет минимизировать избыточность данных и гарантировать их непротиворечивость.

2.2 Как выбрать все записи из таблицы "Persons", где значение поля "LastName" в алфавитном порядке находится между значениями "Ivanov" и "Petrov"?

- SELECT * FROM Persons WHERE LastName>'Ivanov' AND LastName<'Petrov'
- SELECT LastName>'Ivanov' AND LastName<'Petrov' FROM Persons
- SELECT * FROM Persons WHERE LastName BETWEEN 'Ivanov' AND 'Petrov'

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Клиенты.Название, Заказы.КодЗаказа, Доставка.Название,  
Заказы.ДатаРазмещения, Заказы.ДатаРазмещения  
FROM Клиенты LEFT JOIN (Доставка RIGHT JOIN Заказы ON  
Доставка.КодДоставки = Заказы.Доставка) ON Клиенты.КодКлиента =  
Заказы.КодКлиента  
ORDER BY Клиенты. Название, Заказы.КодЗаказа;
```

```
SELECT Типы.Категория, Count(Товары.КодТовара) AS [Count-КодТовара],  
Max(Товары.Цена) AS [Max-Цена], Типы.Описание,  
Min(Товары.МинимальныйЗапас) AS [Min-МинимальныйЗапас]  
FROM Типы LEFT JOIN Товары ON Типы.КодТипа = Товары.КодТипа  
GROUP BY Типы.Категория, Типы.Описание;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:
Вывести список сотрудников вида:

Фамилия, Имя, Дата найма, количество заказов данного сотрудника.

Список отсортировать по дате найма по количеству заказов сотрудника по убыванию и по дате найма по возрастанию.

Контрольная работа 1: Вариант 2

1 Аналитическая часть

Общий формат оператора выборки в SQL.

2 Тестовые задания

2.1 Команда вместе со своими многочисленными опциями и предложениями используется для формирования запросов к реляционной базе данных.

2.2 Как выбрать все записи из таблицы "Persons", где значение поля "FirstName" равно "Ivan"?

- SELECT [all] FROM Persons WHERE FirstName LIKE 'Ivan'
- SELECT * FROM Persons WHERE FirstName<>'Ivan'
- SELECT * FROM Persons WHERE FirstName='Ivan'
- SELECT [all] FROM Persons WHERE FirstName='Ivan'

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Товары.Марка, Типы.Категория, Поставщики.Название,  
Товары.НаСкладе, MAX (Товары.МинимальныйЗапас)  
FROM Поставщики RIGHT JOIN (Типы RIGHT JOIN Товары ON  
Типы.КодТипа = Товары.КодТипа) ON Поставщики.КодПоставщика =  
Товары.КодПоставщика  
WHERE (((Товары.Цена)<5000))
```

```
SELECT Доставка.Название, Доставка.Телефон, Заказы.КодЗаказа,  
Заказано.Цена, Заказано.Скидка, [Заказано]![Цена]*(1-[Заказано]![Скидка]) AS  
Итоговая_цена  
FROM (Доставка INNER JOIN Заказы ON Доставка.КодДоставки =  
Заказы.Доставка) INNER JOIN Заказано;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:
Вывести список следующего вида:

Код Заказа, Категория товара, Марка товара, стоимость данного товара в заказе.

В список включить товары, поставки которых не прекращены и цена складского запаса которых меньше 15000.

Список отсортировать по категории товара по возрастанию и по марке товара по убыванию.

Контрольная работа 1: Вариант 3

1 Аналитическая часть

Скалярные функции СУБД ЛИНТЕР. Запросы без таблиц.

2 Тестовые задания

2.1 Язык запросов это язык, т.е. он предоставляет пользователю возможность определять, что необходимо выполнить, без необходимости указывать, как это нужно выполнять.

2.2 Как выбрать все записи из таблицы "Persons", где значение поля "FirstName" начинается с буквы "a"?

- SELECT * FROM Persons WHERE FirstName='%a%'
- SELECT * FROM Persons WHERE FirstName LIKE '%a'
- SELECT * FROM Persons WHERE FirstName LIKE 'a%'
- SELECT * FROM Persons WHERE FirstName='a'

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Товары.Марка, Типы.Категория, Поставщики.Название,  
Товары.НаСкладе, Товары.МинимальныйЗапас  
FROM Поставщики RIGHT JOIN (Типы RIGHT JOIN Товары ON  
Типы.КодТипа = Товары.КодТипа) ON Поставщики.КодПоставщика =  
Товары.КодПоставщика  
WHERE (((Товары.Цена)<15000))  
ORDER BY Товары.НаСкладе DESC , Товары.МинимальныйЗапас;
```

```
SELECT Типы.Категория, Count(Товары.КодТовара) AS [Count-КодТовара],  
Max(Товары.Цена) AS [Max-Цена], Типы.Описание,  
Min(Товары.МинимальныйЗапас) AS [Min-МинимальныйЗапас]  
FROM Типы LEFT JOIN Товары ON Типы.КодТипа = Товары.КодТипа  
GROUP BY Типы.Категория, Типы.Описание;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:
Вывести список заказов с минимальной датой размещения.

Список следующего вида:

Код заказа, Адрес получателя, Название службы доставки, Стоимость доставки, Стоимость доставки с НДС (18%).

Для заказов у которых не определена служба доставки вывести в Название службы доставки null.

Контрольная работа 1: Вариант 4

1 Аналитическая часть

Запрос. Языки запросов. Характеристика языка QBE.

2 Тестовые задания

2.1 Важнейшим достоинством применения БД в информационных системах является обеспечение данных от прикладных программ

2.2 Как выбрать все записи из таблицы "Persons", где значение поля "FirstName" равно "Ivan" и "LastName" равно "Petr"?

- SELECT * FROM Persons WHERE FirstName<>'Ivan' AND LastName<>'Petr'
- SELECT * FROM Persons WHERE FirstName='Ivan' AND LastName='Petr'
- SELECT FirstName='Ivan', LastName='Petr' FROM Persons

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Товары.Марка, Типы.Категория, Поставщики.Название,  
Товары.НаСкладе, MAX (Товары.МинимальныйЗапас)  
FROM Поставщики RIGHT JOIN (Типы RIGHT JOIN Товары ON  
Типы.КодТипа = Товары.КодТипа) ON Поставщики.КодПоставщика =  
Товары.КодПоставщика  
GROUP BY Товары.Марка, Типы.Категория, Поставщики.Название
```

```
SELECT Товары.Марка, Товары.Цена, Типы.Категория,  
Товары.ЕдиницаИзмерения, Товары.НаСкладе, Заказано.КодЗаказа,  
Заказано.Количество, Заказано.Цена  
FROM (Типы RIGHT JOIN Товары ON Типы.КодТипа = Товары.КодТипа)  
LEFT JOIN Заказано ON Товары.КодТовара = Заказано.КодТовара  
ORDER BY 2 DESC;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать список клиентов, имеющих не более двух заказов (включая клиентов не имеющих заказов), следующего вида:

Название, Город, Телефон, Число заказов клиента.

Список отсортировать по названию клиентов по убыванию.

Контрольная работа 1: Вариант 5

1 Аналитическая часть

Коррелированные вложенные и иерархические подзапросы.

2 Тестовые задания

2.1 Управление безопасностью. СУБД создает систему безопасности, которая обеспечивает защиту пользователей и данных внутри СУБД

2.2 Как выбрать колонку с названием "FirstName" из таблицы "Persons"?

- SELECT Persons.FirstName
- SELECT FirstName FROM Persons
- EXTRACT FirstName FROM Persons

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Доставка.КодДоставки, Доставка.Название, Sum(Заказы.КодКлиента)
AS Sum-КодКлиента
FROM (Доставка INNER JOIN Заказы ON Доставка.КодДоставки =
Заказы.Доставка) INNER JOIN Заказано ON Заказы.КодЗаказа =
Заказано.КодЗаказа
GROUP BY Доставка.КодДоставки, Доставка.Название;
```

```
SELECT Клиенты.КодКлиента, Клиенты.Название,
Sum(Заказы.СтоимостьДоставки) AS Sum-СтоимостьДоставки,
Min(Заказы.ДатаИсполнения) AS Min-ДатаИсполнения, Max(Заказано.Цена) AS
Max-Цена
FROM Клиенты LEFT JOIN (Заказы LEFT JOIN Заказано ON Заказы.КодЗаказа
= Заказано.КодЗаказа) ON Клиенты.КодКлиента = Заказы.КодКлиента
GROUP BY Клиенты.КодКлиента, Клиенты.Название DESC
ORDER BY Клиенты.КодКлиента DESC , Sum(Заказы.СтоимостьДоставки);
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать список заказов, в которых цена товара в заказе ниже цены, указанной в таблице «Товары». Список имеет следующий вид:

Номер заказа, Наименование товара, Цена товара в заказе, Цена товара в таблице «Товары».

Контрольная работа 1: Вариант 6

1 Аналитическая часть

Использование агрегирующих функций без фразы GROUP BY.

2 Тестовые задания

2.1 Для запроса, включающего информацию из нескольких таблиц должны использоваться:

- Агрегирующие функции
- Соединения
- Функции преобразования

2.2 Как вывести количество записей, хранящихся в таблице "Persons"?

- SELECT COUNT() FROM Persons
- SELECT COLUMNS(*) FROM Persons
- SELECT COUNT(*) FROM Persons
- SELECT COLUMNS() FROM Persons

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Доставка.КодДоставки, Доставка.Название, Заказы.КодКлиента,  
Заказано.Количество  
FROM Доставка, Заказано, Заказы;
```

```
SELECT Доставка.КодДоставки, Доставка.Название, Sum(Заказы.КодКлиента)  
AS Sum-КодКлиента  
FROM (Доставка INNER JOIN Заказы ON Доставка.КодДоставки =  
Заказы.Доставка) INNER JOIN Заказано ON Заказы.КодЗаказа =  
Заказано.КодЗаказа  
GROUP BY Доставка.КодДоставки;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать список заказов, в котором присутствует товар с минимальной ценой следующего вида:

Код заказа, Дата исполнения, Стоимость доставки.

Список отсортировать по Стоимости доставки по возрастанию.

Контрольная работа 1: Вариант 7

1 Аналитическая часть

Неопределенные значения. Вовлечение неопределенного значения в SQL-запросы

2 Тестовые задания

2.1 Язык определения данных (Data Definition Language, DDL) позволяет создавать и изменять структуру объектов базы данных, например, создавать и таблицы.

2.2 Как выбрать все записи из таблицы "Persons", упорядоченных по полю "FirstName" в обратном порядке?

- SELECT * FROM Persons SORT 'FirstName' DESC
- SELECT * FROM Persons SORT BY 'FirstName' DESC
- SELECT * FROM Persons ORDER BY FirstName DESC
- SELECT * FROM Persons ORDER FirstName DESC

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT COUNT(Доставка.КодДоставки), COUNT(Доставка.Название),  
MIN(Заказы.КодКлиента), Заказано.Количество  
FROM Доставка, Заказано, Заказы;
```

```
SELECT Доставка.КодДоставки, Доставка.Название, Sum(Заказы.КодКлиента)  
AS Sum-КодКлиента  
FROM (Доставка INNER JOIN Заказы ON Доставка.КодДоставки =  
Заказы.Доставка) INNER JOIN Заказано ON Заказы.КодЗаказа =  
Заказано.КодЗаказа  
GROUP BY Доставка.КодДоставки, Доставка.Название;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать **список всех городов, в которых расположены клиенты, исключив дубликаты.**

Итоговый список отсортировать в алфавитном порядке.

Кроме того, посчитать и вывести количество различных городов (то есть число записей в списке).

Контрольная работа 1: Вариант 8

1 Аналитическая часть

Предложение INSERT. Форматы. Вставка множества записей.

2 Тестовые задания

2.1 NULL-значение в целочисленном поле означает:

- Отсутствие значения
- 0
- "0"

2.2 Как выбрать все поля из таблицы "Persons"?

- SELECT *.Persons
- SELECT Persons
- SELECT [all] FROM Persons
- SELECT * FROM Persons

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

SELECT Клиенты.Название, Заказы.ДатаРазмещения, Заказы.ДатаНазначения, Поставщики.Название, Товары.Марка, Заказано.Количество
FROM Поставщики LEFT JOIN (Товары INNER JOIN ((Клиенты RIGHT JOIN Заказы ON Клиенты.КодКлиента = Заказы.КодКлиента) INNER JOIN Заказано ON Заказы.КодЗаказа = Заказано.КодЗаказа) ON Товары.КодТовара = Заказано.КодТовара) ON Поставщики.КодПоставщика = Товары.КодПоставщика
ORDER BY Клиенты.Название, Заказы.ДатаРазмещения DESC , Поставщики.Название DESC , Товары.Марка;

SELECT Типы.Категория, Count(Товары.КодТовара) AS [Count-КодТовара], Max(Товары.Цена) AS [Max-Цена], Типы.Описание, Min(Товары.МинимальныйЗапас) AS [Min-МинимальныйЗапас]
FROM Типы LEFT JOIN Товары ON Типы.КодТипа = Товары.КодТипа
GROUP BY Типы.Категория, Типы.Описание;
ORDER BY Типы.Категория DESC;

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:
Сформировать список товаров вида:

Марка, Категория, Минимальный Запас, стоимость остатков на складе, поставки которых прекращены.

Список отсортировать по величине минимального запаса по возрастанию.

Контрольная работа 1: Вариант 9

1 Аналитическая часть

Реализация реляционных операторов с помощью средств языка SQL.

2 Тестовые задания

2.2 С помощью предложения можно реализовать любую операцию реляционной алгебры.

2.1 Как выбрать все записи из таблицы "Persons", упорядоченных по полю "FirstName" в обратном порядке?

- SELECT * FROM Persons SORT 'FirstName' DESC
- SELECT * FROM Persons SORT BY 'FirstName' DESC
- SELECT * FROM Persons ORDER BY FirstName DESC
- SELECT * FROM Persons ORDER FirstName DESC

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Клиенты.КодКлиента, Клиенты.Название,  
Sum(Заказы.СтоимостьДоставки) AS Sum-СтоимостьДоставки,  
Min(Заказы.ДатаИсполнения) AS Min-ДатаИсполнения, Max(Заказано.Цена) AS  
Max-Цена  
FROM Клиенты LEFT JOIN (Заказы LEFT JOIN Заказано ON Заказы.КодЗаказа  
= Заказано.КодЗаказа) ON Клиенты.КодКлиента = Заказы.КодКлиента  
GROUP BY Клиенты.Название  
ORDER BY Клиенты.КодКлиента DESC , Sum(Заказы.СтоимостьДоставки);
```

```
SELECT Доставка.Название, Доставка.Телефон, Заказы.КодЗаказа,  
Заказано.Цена, Заказано.Скидка, Заказано.Цена*(1-Заказано.Скидка) AS  
Итоговая_цена  
FROM (Доставка INNER JOIN Заказы ON Доставка.КодДоставки =  
Заказы.Доставка) INNER JOIN Заказано ON Заказы.КодЗаказа =  
Заказано.КодЗаказа;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать список товаров, которые вошли в заказ с минимальным кодом, вида:

Марка, Цена, Ожидается, На складе.

Список отсортировать по цене по убыванию

Контрольная работа 1: Вариант 10

1 Аналитическая часть

Запрос. Языки запросов. Характеристика языка SQL.

2 Тестовые задания

2.1 2.1 При выполнении оператора вставки список столбцов может всегда не указываться:

- Если значения могут распределяться по столбцам таблицы в произвольном порядке
- Если количество значений равно количеству столбцов
- Если значения обеспечены по всем столбцам в порядке их расположения в таблице

2. 2 Оператор, позволяющий указать имена исходных таблиц, участвующих в формировании выборки, - это:

- WHERE;
- FROM;
- SELECT;
- LIKE.

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку.

Если же запрос верный, то приведите результат выполнения этого запроса.

SELECT Доставка.КодДоставки, Доставка.Название, Sum(Заказы.КодКлиента)

AS Sum-КодКлиента

FROM (Доставка **INNER JOIN** Заказы **ON** Доставка.КодДоставки =

Заказы.Доставка) **INNER JOIN** Заказано **ON** Заказы.КодЗаказа =

Заказано.КодЗаказа

SELECT Клиенты.КодКлиента, Клиенты.Название,

Sum(Заказы.СтоимостьДоставки) **AS** Sum-СтоимостьДоставки,

Min(Заказы.ДатаИсполнения) **AS** Min-ДатаИсполнения, Max(Заказано.Цена) **AS**

Max-Цена

FROM Клиенты **LEFT JOIN** (Заказы **LEFT JOIN** Заказано **ON** Заказы.КодЗаказа =
Заказано.КодЗаказа) **ON** Клиенты.КодКлиента = Заказы.КодКлиента

GROUP BY Клиенты.КодКлиента, Клиенты.Название;

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать список поставщиков, чьи товары поставлялись клиенту разметившему заказ с минимальным кодом, вида:

Наименование, адрес, телефон.

Контрольная работа 1: Вариант 11

1 Аналитическая часть

Конструкция GROUP BY. Агрегирующие функции.

2 Тестовые задания

2.1 Реляционной СУБД российского рынка является

2.2 Как выбрать все записи из таблицы "Persons", где значение поля "LastName" в алфавитном порядке находится между значениями "Ivanov" и "Petrov"?

- SELECT * FROM Persons WHERE LastName>'Ivanov' AND LastName<'Petrov'
- SELECT LastName>'Ivanov' AND LastName<'Petrov' FROM Persons
- SELECT * FROM Persons WHERE LastName BETWEEN 'Ivanov' AND 'Petrov'

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

SELECT Клиенты.Название, Заказы.ДатаРазмещения, Заказы.ДатаНазначения, Поставщики.Название, Товары.Марка, Заказано.Количество
FROM Поставщики INNER JOIN (Товары INNER JOIN ((Клиенты INNER JOIN Заказы ON Клиенты.КодКлиента = Заказы.КодКлиента) INNER JOIN Заказано ON Заказы.КодЗаказа = Заказано.КодЗаказа) ON Товары.КодТовара = Заказано.КодТовара) ON Поставщики.КодПоставщика = Товары.КодПоставщика
ORDER BY Клиенты.Название, Заказы.ДатаРазмещения DESC , Поставщики.Название DESC , Товары.Марка;

ELECT Товары.Марка, Товары.Цена, Типы.Категория, Товары.ЕдиницаИзмерения, Товары.НаСкладе, Заказано.КодЗаказа, Заказано.Количество, Заказано.Цена
FROM (Типы RIGHT JOIN Товары ON Типы.КодТипа = Товары.КодТипа) LEFT JOIN Заказано ON Товары.КодТовара = Заказано.КодТовара
ORDER BY Товары.Марка DESC , Товары.Цена, Типы.Категория;

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать список следующего вида:

Фамилия сотрудника, Общее количество обслуженных им заказов, Последняя дата исполненного заказа, Максимальная стоимость доставки среди всех заказов, обслуженных сотрудником.

При формировании учесть сотрудников, не имеющих обслуженных заказов, а также однофамильцев.

Контрольная работа 1: Вариант 12

1 Аналитическая часть

Коррелированные вложенные подзапросы.

2 Тестовые задания

2.1 Некоррелированные подзапросы оцениваются:

- Начиная с внутреннего запроса
- Начиная с внешнего запроса

2.1 Как выбрать все записи из таблицы "Persons", где значение поля "LastName" равно "Petrov"?

- SELECT [all] FROM Persons WHERE LastName LIKE 'Petrov'
- SELECT * FROM Persons WHERE LastName<>'Petrov'
- SELECT * FROM Persons WHERE LastName='Petrov'
- SELECT [all] FROM Persons WHERE L:astName='Petrov'

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Товары.Марка, Товары.Цена, Типы.Категория,  
Товары.ЕдиницаИзмерения, Товары.НаСкладе, Заказано.КодЗаказа,  
Заказано.Количество, Заказано.Цена  
FROM (Типы RIGHT JOIN Товары) LEFT JOIN Заказано ON  
Товары.КодТовара = Заказано.КодТовара  
ORDER BY 2 DESC;
```

```
SELECT 2, 3, 4  
from Типы  
order by 2,3;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:
Сформировать список заказов вида:

Код заказа, код клиента, дата размещения заказа, суммарная стоимость заказа.

Контрольная работа 1: Вариант 13

1 Аналитическая часть

Запросы, использующие EXISTS.

2 Тестовые задания

2.2 Фактически любой запрос, который выражается через IN, может быть альтернативным образом сформулирован также с помощью

2.1 Как выбрать все записи из таблицы "Persons", упорядоченных по полю "LastName" в обратном порядке?

- SELECT * FROM Persons SORT 'LastName' DESC
- SELECT * FROM Persons SORT BY 'LastName' DESC
- SELECT * FROM Persons ORDER BY LastName DESC
- SELECT * FROM Persons ORDER LastName DESC

3.1 Определите, содержат ли предложенные тексты запросов ошибку.

Если же запрос верный, то приведите результат выполнения этого запроса.

SELECT Типы.Категория, Count(Товары.КодТовара) AS [Count-КодТовара],
Max(Товары.Цена) AS [Max-Цена], Типы.Описание,
Min(Товары.МинимальныйЗапас) AS [Min-МинимальныйЗапас]
FROM Типы LEFT JOIN Товары ON Типы.КодТипа = Товары.КодТипа
GROUP BY Типы.Категория, Типы.Описание;
ORDER BY 3 DESC, 1;

SELECT Типы.Категория, Товары.Марка, Товары.ЕдиницаИзмерения,
Товары.НаСкладе, Товары.ПоставкиПрекращены
FROM Типы INNER JOIN Товары ON Типы.КодТипа=Товары.КодТипа
WHERE (((Товары.ПоставкиПрекращены)<>Yes))
ORDER BY Типы.Категория, Товары.Марка;

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать

список поставщиков

тех товаров, поставки которых прекращены. Список должен содержать все атрибуты поставщика, а также не содержать дубликатов.

Список отсортировать по названию поставщика в порядке, обратном алфавитному.

Контрольная работа 1: Вариант 14

1 Аналитическая часть

Соединение таблицы со своей копией.

2 Тестовые задания

2.1 Именно возможность включения одного предложения внутрь другого послужила мотивировкой использования прилагательного "структурированный" или "структурированный" в названии языка SQL.

2.1 Как выбрать все записи из таблицы "Persons", где значение поля "LastName" в алфавитном порядке находится между значениями "Ivanov" и "Petrov"?

- SELECT * FROM Persons WHERE LastName>'Ivanov' AND LastName<'Petrov'
- SELECT LastName>'Ivanov' AND LastName<'Petrov' FROM Persons
- SELECT * FROM Persons WHERE LastName BETWEEN 'Ivanov' AND 'Petrov'

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Заказано.КодЗаказа, Заказано.КодТовара, Товары.Марка,  
Заказано.Цена, Заказано.Количество, Заказано.Скидка,  
(Заказано.Цена*[Количество]*(1-[Скидка])/100)*100 AS ОтпускнаяЦена  
FROM Товары INNER JOIN Заказано ON  
Товары.КодТовара=Заказано.КодТовара  
ORDER BY Заказано.КодЗаказа;
```

```
SELECT Типы.КодТипа, Типы.Категория, Товары.Марка, Sum([Сведения о  
заказах].ОтпускнаяЦена) AS ПродажиТовара  
FROM Типы INNER JOIN (Товары INNER JOIN (Заказы INNER JOIN  
[Сведения о заказах] ON Заказы.КодЗаказа=[Сведения о заказах].КодЗаказа)  
ON Товары.КодТовара=[Сведения о заказах].КодТовара) ON  
Типы.КодТипа=Товары.КодТипа  
WHERE (((Заказы.ДатаРазмещения) Between '1/1/1997' And '12/31/1997'))  
GROUP BY Типы.КодТипа, Типы.Категория, Товары.Марка  
ORDER BY Товары.Марка;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать **список поставщиков тех товаров, поставки которых не прекращены.**

Список должен содержать все атрибуты поставщика, а также не содержать дубликатов.

Список отсортировать по названию поставщика в алфавитном порядке.

Контрольная работа 1: Вариант 15

1 Аналитическая часть

Вложенные подзапросы.

2 Тестовые задания

2.1 SQL обладает великолепным механизмом для одновременной или последовательной обработки данных из нескольких таблиц. В нем реализованы возможности "соединять" или "объединять" несколько таблиц и так называемые "вложенные подзапросы".

2.2 Как выбрать все записи из таблицы "Persons", где значение поля "LastName" равно "Petrov"?

- SELECT [all] FROM Persons WHERE LastName LIKE 'Petrov'
- SELECT * FROM Persons WHERE LastName<>'Petrov'
- SELECT * FROM Persons WHERE LastName='Petrov'
- SELECT [all] FROM Persons WHERE L:astName='Petrov'

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Типы.КодТипа, Типы.Категория, Товары.Марка, Sum([Сведения о
заказах].ОтпускнаяЦена) AS ПродажиТовара
FROM Типы INNER JOIN (Товары INNER JOIN (Заказы INNER JOIN
[Сведения о заказах] ON Заказы.КодЗаказа=[Сведения о заказах].КодЗаказа)
ON Товары.КодТовара=[Сведения о заказах].КодТовара) ON
Типы.КодТипа=Товары.КодТипа
WHERE (((Заказы.ДатаРазмещения) Between '1/1/1997' And '12/31/1997'))
GROUP BY Типы.КодТипа, Типы.Категория, Товары.Марка
ORDER BY 3;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:
Сформировать список клиентов вида:

**Название, ОбращатьсяК, Должность, Адрес, Город, Количество
сотрудников, обслуживавших заказы этого клиента).**

Список отсортировать по названию города в алфавитном порядке, и по названию клиента в обратном алфавитном порядке.

КОНТРОЛЬНАЯ РАБОТА №2

Контрольная работа 2: Вариант 1

1 Аналитическая часть

Фраза WHERE в SQL-запросах на модификацию данных. Использование фразы WHERE.

2 Тестовые задания

2.1 Какая SQL команда используется для выборки данных из базы?

- EXTRACT
- GET
- SELECT
- OPEN

2.2 Как изменить значение "Ivanov" на "Stepanov" в колонке "LastName", таблицы Persons?

- UPDATE Persons SET LastName='Ivanov' INTO LastName='Stepanov'
- MODIFY Persons SET LastName='Ivanov' INTO LastName='Stepanov'
- MODIFY Persons SET LastName='Stepanov' WHERE LastName='Ivanov'
- UPDATE Persons SET LastName='Stepanov' WHERE LastName='Ivanov'

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
INSERT INTO Доставка ( КодДоставки, Название, Телефон )
SELECT Доставка.КодДоставки, Доставка.КодДоставки, Доставка.Телефон
FROM Доп_Доставка
WHERE (((Доставка.КодДоставки)>10000) AND ((Доставка.КодДоставки) Is
Not Null));
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Увеличить на 50% цену всех товаров, поставляемых не российскими поставщиками, в случае если стоимость складского остатка меньше 10000.

Контрольная работа 2: Вариант 2

1 Аналитическая часть

Предложение INSERT. Форматы. Особенности применения.

2 Тестовые задания

2.1 Какая SQL команда используется для обновления данных в базе?

- MODIFY
- SAVE

- UPDATE
- SAVE AS

2.2 Удаление определенной записи в таблице:

- DELETE...FROM;
- DELETE...COLUMN...FROM;
- DELETE...FROM...WHERE;
- DELETE...DROP...WHERE.

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

DELETE

FROM Заказано

WHERE (((Заказано.Скидка)>10) AND ((Заказано.Количество)<1500));

GROUP BY Заказано.Скидка

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать список

(Название, Город, Адрес, Телефон)

российских поставщиков товаров с наибольшим значением Минимального запаса. Список отсортировать по Названию. Список записать в новую таблицу с именем «Т1»

Контрольная работа 2: Вариант 3

1 Аналитическая часть

Типы команд языков запросов.

2 Тестовые задания

2.1 Какая SQL команда используется для удаления данных из базы?

- COLLAPSE
- DELETE
- REMOVE

2.2 Как удалить записи, где значение поля "FirstName" равно "Ivan"?

- DELETE FROM Persons WHERE FirstName = 'Ivan'
- DELETE ROW FirstName='Ivan' FROM Persons
- DELETE FirstName='Ivan' FROM Persons

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

UPDATE Товары SET Товары.Цена = Цена*0,98
WHERE (((Товары.МинимальныйЗапас < 1500 Or Товары.НаСкладе = 1000)
And Товары.ЕдиницаИзмерения<>'Л'));

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать список заказов, доставкой которых занята фирма «Фирма1». Список должен иметь вид:

код заказа, дата размещения, код сотрудника, код клиента.

Список отсортировать по коду заказа по убыванию.

Список записать в таблицу с именем «Т1», предполагая, что такая таблица существует

Контрольная работа 2: Вариант 4

1 Аналитическая часть

Вложенные подзапросы. Виды. Особенности использования вложенных подзапросов в предложениях модификации данных.

2 Тестовые задания

2.1 Если раздел WHERE в операторе UPDATE отсутствует, то:

- Изменяются все строки таблицы
- Не изменяется ни одна строка таблицы
- Гарантированно изменятся только первая строка таблицы

2.2 При удалении таблицы:

- Автоматически удаляются все ее строки
- Автоматически удаляются все ее индексы

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

INSERT INTO Доставка
SELECT Доставка.КодДоставки, Доставка.КодДоставки, Доставка.Телефон
FROM Доставка
WHERE (((Доставка.КодДоставки)>10000) AND ((Доставка.КодДоставки) Is Not Null));

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Увеличить вдвое цену

всех товаров,

у которых название категории начинается со слова «Винт», либо поставки которых прекращены.

Контрольная работа 2: Вариант 5

1 Аналитическая часть

Предложение UPDATE. Форматы. Особенности применения

2 Тестовые задания

2.1 Какая SQL команда используется для вставки данных в базу?

- ADD RECORD
- INSERT INTO
- ADD NEW
- INSERT NEW

2.2 Столбцы таблиц, по которым производится соединение, должны иметь:

- Одинаковые имена
- Одинаковые типы

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку.

Если же запрос верный, то приведите результат выполнения этого запроса.

UPDATE Сотрудники SET Сотрудники.Обращение = "Sr"

WHERE (((Сотрудники.Должность) Like "Директор%") AND

((Сотрудники.Страна)="Испания")) OR (((Сотрудники.Страна)="Мексика"))

GROUP BY Сотрудники.Должность;

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Вывести **всю информацию о заказах** (то есть все атрибуты заказа), которые обслуживаются сотрудниками, родившимися в период с 01/05/1972 по 31/12/1990

Контрольная работа 2: Вариант 6

1 Аналитическая часть

Фраза HAVING. Особенности. Применение.

2 Тестовые задания

2.1 За удаление полей таблицы отвечает команда:

- ADD;
- DELETE;
- COLUMN;
- DROP.

2.2 Оператор OR срабатывает когда **ХОТЬ ОДНО** условие истинно. AND когда **ВСЕ** условия истинны.

- Ложь
- Истина

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку.
Если же запрос верный, то приведите результат выполнения этого запроса.

UPDATE Сотрудники SET Сотрудники.Обращение = "Sr"
WHERE (((Сотрудники.Должность) Like "Директор%") AND
((Сотрудники.Страна)="Испания")) OR (((Сотрудники.Страна)="Мексика"));

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Вывести всю информацию о товарах, с максимальной величиной ожидаемой поставки (независимо от единиц измерения), а также о товарах, относящихся к категориям «Продукты питания», «Бытовая химия» и «Автозапчасти».

Информацию записать в таблицу “Т1”, предполагая, что такая таблица существует

Контрольная работа 2: Вариант 7

1 Аналитическая часть

Типы команд языков запросов.

2 Тестовые задания

2.1 Удаление определенной записи в таблице:

- DELETE...FROM;
- DELETE...COLUMN...FROM;
- DELETE...FROM...WHERE;
- DELETE....DROP...WHERE.

2.2 Оператор, выполняющий проверку на диапазон значений:

- FROM...TO;
- BETWEEN...AND;
- FROM...AND;
- BETWEEN...TO.

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку.
Если же запрос верный, то приведите результат выполнения этого запроса.

DELETE *
FROM Заказано
WHERE (((Заказано.Скидка)>10) AND ((Заказано.Количество)<1500));
GROUP BY Заказано.Скидка

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать список поставщиков товаров (все атрибуты поставщиков), в описании типов которых отсутствует изображение. В списке исключить дубликаты.

Информацию записать в таблицу «Т1», предполагая, что такая таблица существует

Контрольная работа 2: Вариант 8

1 Аналитическая часть

Создание и уничтожение базовых таблиц.

2 Тестовые задания

2.1 Обновление, замена значений полей записи:

- ADD;
- REFERENCES;
- JOIN;
- UPDATE.

2.2 Оператор IN выполняет:

- Проверку выражения на NULL;
- Проверку выражения на совпадение с любым из элементов списка;
- Сравнение строковых элементов;
- Логическую импликацию выражений.

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку.

Если же запрос верный, то приведите результат выполнения этого запроса.

DELETE Заказано.Цена, Заказано.Скидка

FROM Заказано

WHERE (((Заказано.Цена)>100) AND ((Заказано.Скидка)<5));

Задача 2

Сформировать список заказов, которые обслуживали самые молодые сотрудники (с учетом того, что могут быть сотрудники, родившиеся в один день).

Список записать в новую таблицу с именем «Т1»

Контрольная работа 2: Вариант 9

1 Аналитическая часть

Конструкции IN, BETWEEN, LIKE в запросах на модификацию данных.

2 Тестовые задания

2.1 Какое SQL выражение используется для извлечения различных значений?

- SELECT DIFFERENT
- SELECT DISTINCT
- SELECT UNIQUE

2.2 Оператор, вычисляющий количество выделенных записей в запросе:

- Avg;
- Var;
- Sum;
- Count.

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку.

Если же запрос верный, то приведите результат выполнения этого запроса.

```
DELETE Заказано.Цена, Заказано.Скидка
FROM Заказано
WHERE (((Заказано.Цена)>100) AND ((Заказано.Скидка)<5));
ORDER BY Цена
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать список **всех товаров**, в названии которых встречается слово «гайка». Список включает в себя Марку, Описание категории товара, На Складе, Минимальный Запас.

Список отсортировать по цене по убыванию.

Список записать в новую таблицу с именем «Т1»

Контрольная работа 2: Вариант 10

1 Аналитическая часть

Предложение INSERT. Форматы. Особенности применения

2 Тестовые задания

2.1 Выберите правильный SQL запрос для вставки новой записи в таблицу "Persons".

- INSERT ('Stepan', 'Petr') INTO Persons
- INSERT INTO Persons VALUES ('Stepan', 'Petr')
- INSERT VALUES ('Stepan', 'Petr') INTO Persons

2.2 LIKE - это оператор:

- Выполняющий проверку выражения на NULL;
- Определяющий условия отбора записей;
- Выполняющий сравнение строковых значений;
- Проверяющий логическую эквивалентность выражений

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

DELETE

FROM Заказано

WHERE (((Заказано.Скидка)>10) AND ((Заказано.Количество)<1500));

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Исправить дату исполнения заказов с 05/05/2007 и 10/10/2007 на 01/02/2008.

Контрольная работа 2: Вариант 11

1 Аналитическая часть

Предложение DELETE. Форматы. Особенности применения

2 Тестовые задания

2.1 Какая SQL команда используется для упорядочивания результатов?

- SORT
- ORDER BY
- ORDER
- SORT BY

2.2 Оператор соединения таблиц - это:

- INSERT JOIN;
- CREATE JOIN;
- INNER JOIN;
- SELECT JOIN.

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

UPDATE Товары SET Товары.Цена = Цена*0,98

WHERE

((Товары.МинимальныйЗапас < 1500 Or Товары.НаСкладе = 1000) And Товары.ЕдиницаИзмерения<>'Л'))

GROUP BY Товары.МинимальныйЗапас;

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать единый список сотрудников и клиентов следующего вида:

Фамилия (или ОбращатьсяК для клиентов), страна, город, адрес.

Список записать в новую таблицу с именем «Т2»

Контрольная работа 2: Вариант 12

1 Аналитическая часть

Предложение UPDATE. Форматы. Особенности применения

2 Тестовые задания

2.1 Выберите правильный SQL запрос для вставки новой записи в таблицу "Persons", причём в поле "LastName" вставить значение "Sidorov".

- INSERT INTO Persons (LastName) VALUES ('Sidorov')
- INSERT INTO Persons ('Sidorov') INTO LastName
- INSERT ('Sidorov') INTO Persons (LastName)

2.2 Оператор, формирующий выборку для дополнения и определяющий структуру данных источника передаваемых записей для загрузки в таблицу:

- WHERE;
- HAVING;
- ORDER BY;
- SELECT.

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Типы.КодТипа, Типы.Категория, Товары.Марка, Sum([Сведения о заказах].ОтпускнаяЦена) AS ПродажиТовара
INTO T21
FROM Типы INNER JOIN (Товары INNER JOIN (Заказы INNER JOIN
[Сведения о заказах] ON Заказы.КодЗаказа=[Сведения о заказах].КодЗаказа)
ON Товары.КодТовара=[Сведения о заказах].КодТовара) ON
Типы.КодТипа=Товары.КодТипа
WHERE (((Заказы.ДатаРазмещения) Between '1/1/1997' And '12/31/1997'))
GROUP BY Типы.КодТипа, Типы.Категория, Товары.Марка
ORDER BY 3;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Вывести в новую таблицу с именем «Т1» названия и телефоны фирм-доставщиков, доставлявших заказы клиентов из Москвы, Пскова и Смоленска.

Список не должен содержать дубликатов и должен быть отсортирован по названиям фирм в алфавитном порядке.

Контрольная работа 2: Вариант 13

1 Аналитическая часть

Вложенные подзапросы. Использование вложенных подзапросов в предложениях модификации данных

2 Тестовые задания

2.1 Как изменить значение "Ivanov" на "Stepanov" в колонке "LastName", таблицы Persons?

- UPDATE Persons SET LastName='Ivanov' INTO LastName='Stepanov'
- MODIFY Persons SET LastName='Ivanov' INTO LastName='Stepanov'
- MODIFY Persons SET LastName='Stepanov' WHERE LastName='Ivanov'
- UPDATE Persons SET LastName='Stepanov' WHERE LastName='Ivanov'

2.2 Оператор, определяющий, должен ли подчиненный запрос возвращать какие-либо записи:

- HAVING
- As
- SOME
- EXISTS

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

```
SELECT Заказы.КодКлиента, Заказы.КодСотрудника,  
Заказы.ДатаРазмещения,  
Заказано.КодТовара, Заказано.Цена, Заказано.Количество  
INTO T1  
FROM Заказы  
INNER JOIN Заказано  
ON Заказы.КодЗаказа =  
Заказано.КодЗаказа;
```

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Сформировать список сотрудников, не обслуживавших ни один заказ.

Вид списка:

Фамилия, Должность, Страна, Город.

Список отсортировать по дате рождения по убыванию.

Контрольная работа 2: Вариант 14

1 Аналитическая часть

Модификация базовых таблиц с помощью языков запросов.

2 Тестовые задания

2.1 Как удалить записи, где значение поля "FirstName" равно "Ivan"?

- DELETE FROM Persons WHERE FirstName = 'Ivan'
- DELETE ROW FirstName='Ivan' FROM Persons
- DELETE FirstName='Ivan' FROM Persons

2.2 Оператор, определяющий поля, по которым выполняется упорядочение выходных данных, порядок их следования соответствует старшинству ключей сортировки:

- GROUP BY;
- WITH OWNERACCESS OPTION;
- HAVING;
- ORDER BY.

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

DELETE

FROM Сотрудники

WHERE (((Сотрудники.КодСотрудника)>100000) AND ((Сотрудники.Город)="Москва") AND ((Сотрудники.Примечания) Is Not Null)) OR (((Сотрудники.КодСотрудника)<150) AND ((Сотрудники.Город)="Псков"));

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Установить величину скидки 32% на все товары, заказанные (по дате размещения) между 01/01/2003 и 05/07/2005

Контрольная работа 2: Вариант 15

1 Аналитическая часть

Создание и уничтожение базовых таблиц.

2 Тестовые задания

2.1 Записи из двух таблиц объединяются, если:

- Поля этих таблиц имеют одинаковый тип данных;
- Все записи полей имеют одинаковую длину;
- Имена полей таблиц совпадают;
- Поля содержат одинаковые значения.

2.1 Команда FOREIGN KEY используется для:

- Связи с внутренними таблицами;
- Связи ссылок на внутренние таблицы;
- Связи с внешними таблицами;
- Связи ссылок на внешние таблицы.

3 Практические задания

3.1 Определите, содержат ли предложенные тексты запросов ошибку. Если же запрос верный, то приведите результат выполнения этого запроса.

DELETE

FROM Сотрудники
WHERE (((Сотрудники.КодСотрудника)>100000) AND
((Сотрудники.Город)="Москва") AND ((Сотрудники.Примечания) Is Not Null))
OR (((Сотрудники.КодСотрудника)<150) AND ((Сотрудники.Город)="Псков"))
ORDER BY 1, 3;

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия:

Составить SQL-запрос, увеличивающий на 50% цену всех товаров, поставляемых не российскими поставщиками, в случае если стоимость складского остатка меньше 10000

3 ПРИМЕРЫ ВЫПОЛНЕНИЯ РАБОТ

Контрольная работа №1 Вариант 16

1 Аналитическая часть

Реализация языка запросов в СУБД, разработанных в России.

4 РЕАЛИЗАЦИЯ ЯЗЫКА ЗАПРОСОВ В СУБД, РАЗРАБОТАННЫХ В РОССИИ

В настоящий момент основными представителями рынка российских СУБД являются системы «Ника», «Паллада» и «ЛИНТЕР».

Система "Паллада" представляет из себя защищенную объектно-ориентированную СУБД, предназначенную для АСУ вооруженных сил мобильного базирования. "Паллада" функционирует в среде ОС МСВС и ОЛИВИЯ.

СУБД "Ника", разработанная на основе советской системы "ИНЕС" принадлежит сейчас компании Cognitive Technologies. Эта иерархическая СУБД встроена в другие продукты компании. "Ника" является основой для системы документооборота "Евфрат-Документооборот" и информационно-справочной системы "НИКА-музей". При этом компанией СУБД не распространяется в отдельном пакете.

Как можно видеть реляционной СУБД российского рынка является «ЛИНТЕР». Рассмотрим подробнее реализацию языка запросов этой СУБД. При реализации языка SQL в СУБД ЛИНТЕР за основу был взят стандарт ANSI/ISO SQL-92, но в него были внесены некоторые элементы, не специфицированные в стандарте.

4.1 Скалярные функции СУБД ЛИНТЕР

Скалярные функции предназначены для работы со значениями столбцов и/или константами в выражениях. Их допустимо использовать в любых операторах SQL в тех местах, где синтаксическая конструкция позволяет использовать выражение.

Все функции SQL можно разделить на две группы:

- встроенные функции;
- пользовательские функции.

Встроенные функции являются неотъемлемой частью транслятора языка БД SQL СУБД ЛИНТЕР и не зависят от конкретного экземпляра базы данных.

В качестве пользовательских функций выступают хранимые процедуры, возвращающие скалярное значение. Описание пользовательских функций хранятся в базе данных, поэтому обращение к ним возможно только в конкретном экземпляре базы.

При переносе приложения, содержащего пользовательские функции, на другую БД, все используемые функции должны быть перенесены в эту базу. В текущей версии тело пользовательской функции не может содержать SQL-запросы. Вызов встроенных функций и других ранее определенных пользовательских функций допустим. В текущей версии пользовательские функции доступны всем. В следующих версиях владелец функции сможет ограничивать доступ к ней для определенной категории пользователей.

Спецификация пользовательской функции включает:

- имя владельца функции;
- имя функции;
- список входных параметров и их тип.

В БД может быть несколько функций с одинаковым именем, но созданные разными владельцами. Если при вызове имя владельца явно не указано, то вызывается функция с именем владельца, от имени которого выполняется приложение.

Функция идентифицируется ее схемой- владельцем, именем функции, количеством и типом параметров. Это называется прототипом функции, который должен быть уникальным в экземпляре БД.

Запрещены все строковые функции для типов UNICODE, кроме LENGTH, конкатенации (со значением такого же типа), GETSTR и ей подобных. Тип результата при конкатенации приводится к типу первого аргумента.

Подробное описание всех скалярных функций (их достаточно много) можно легко найти в документации. Здесь же мы ограничимся простым разделением их на основные разделы:

- строковые/символьные функции (конкатенация, поиск подстроки, перевод к верхнему/нижнему регистру и т.п.);
 - арифметические функции (абсолютное значение, разного рода округления, тригонометрические и обратные тригонометрические функции, и т.п.);
 - функции преобразования шестнадцатеричных значений;
 - функции преобразования (к символьному, числовому типам, к типу дата);
 - функции дампа поля (пословного, побайтного и даже побитного);
 - функция получения возможностей доступа;
 - функции для работы с BLOB-данными;
 - функции для работы с данными типа дата (выделение дня недели, месяца, года, работа с часовыми поясами и т.п.);
 - функции для работы с данными типа EXTFILE (получение имени, размера файла и т.п.);
- и некоторые другие.

4.2 Запросы без таблиц

Некоторые запросы, которые могут потребоваться приложению, не связаны собственно с данными базы, с данными таблиц.

Типичный пример – получение текущего времени сервера, имя текущего пользователя, что в некоторых случаях достаточно удобно.

Для подобных случаев, когда данные не связаны с конкретной таблицей, в SQL диалект ЛИНТЕР (начиная с версии 5.9) введены SELECT запросы без предложения FROM.

Для общности эти в подобные запросы добавлена возможность использования произвольных выражений:

SELECT <Выражение>;

Здесь<Выражение>может состоять из констант и предопределённых переменных – SYSDATE (текущее время сервера) и USER (имя текущего пользователя).

**SELECT ((26.77*3.14)/2 +1)*((26.77*3.14)/2 -1), USER, SYSDATE;
| 1765.42843521 | SYSTEM | 29.05.2003:10:37:01.00 |**

Аналогичные запросы можно было делать и в более ранних версиях (в ЛИНТЕР-ВС, ЛИНТЕР версии 5.7 и 5.8), но только с использованием таблиц, даже если их информация не используется в запросе. Это усложняло и замедляло обработку подобных запросов, кроме того требовались множественные проверки по доступу и т.п. Кроме того, ответ выдавался ровно столько раз, сколько записей в используемой таблице, например:

**SELECT ((26.77*3.14)/2 +1)*((26.77*3.14)/2 -1), USER, SYSDATE
FROM \$\$\$SYSRL;
| 1765.42843521 | SYSTEM | 29.05.2003:10:37:01.00 |
| 1765.42843521 | SYSTEM | 29.05.2003:10:37:01.00 |
| 1765.42843521 | SYSTEM | 29.05.2003:10:37:02.00 |**

4.3 Иерархические запросы

Достаточно часто в приложениях встречаются запросы типа «выдать всех руководителей Петрова В.В.», при этом имеются в виду не только непосредственные руководители, но так же и руководители руководителей, и все кто как-либо, пусть даже косвенно, может управлять этим работником.

Другим примером может служить поиск всех поддиректорий данной директории. Для решения подобных задач в ЛИНТЕР введена конструкция START WITH...CONNECT BY:

**SELECT <Список столбцов>
FROM <Имя таблицы>**

[WHERE <Спецификатор выборки>]
[START WITH <Начало>]
[CONNECT BY <Связь>]
[<Спецификатор сортировки>]

, где <Начало> – логическое выражение, определяющее записи - начала иерархии (высший уровень), а <Связь> – логическое выражение, связывающие данный уровень иерархии со следующим уровнем иерархии;

При этом:

- Для того чтобы использовать одни и те же столбцы таблицы в одних и тех же на различных уровнях используется ключевое слово PRIOR.
- Ключевое слово PRIOR используется в предикатах типа =, <>, >, <, >=, <=, заданными в конструкции CONNECT BY и может стоять перед любой из частей предиката (левой, правой, перед обеими).
- Если слово PRIOR задано перед левой/правой частью предиката, то все обращения к столбцам в этой части предиката считаются столбцами предыдущего экземпляра таблицы (данными предыдущего уровня), иначе - столбцами текущего экземпляра таблицы (данными текущего уровня).
- Типы данных в предикатах <Связи> должны быть совместимыми.
- Условия WHERE, START и CONNECT в иерархических запросах можно задавать в произвольном порядке.
- В <Списке столбцов> недопустимо использование псевдостолбцов ROWID и ROWTIME.
- В <Список столбцов> система по умолчанию добавляет столбец LEVEL (тип INTEGER), данные которого это номер уровня записи (ответа) в иерархии.
- Столбец LEVEL можно использовать в предикатах WHERE-предложения, в <Спецификаторе выборки>, если WHERE-предложение стоит после предложения START WITH, либо после предложения CONNECT BY.
- Столбец LEVEL можно использовать так же в <Спецификаторе сортировки>.
- Если <Спецификатор сортировки> не задан, то записи (ответа) будут располагаться в таком порядке, чтобы все записи, подчиненные в иерархии некоторой записи, следовали за ней.
- Условие, содержащееся в конструкции START WITH, определяет данные первого (наивысшего) уровня иерархии.
- Условие, содержащееся в конструкции CONNECT, определяет связи между записями предыдущего и последующего уровней.
- Для запросов с CONNECT BY можно подавать команды позиционного обновления и удаления (WHERE CURRENT OF).

Например:

```
SELECT *  
FROM "Служащие"
```

```
START WITH "Должность" = 'РУКОВОДИТЕЛЬ'  
CONNECT BY PRIOR "Таб_Номер" = "Таб_Номер_Руковод";  
SELECT Directory_Name  
FROM Directories  
START WITH Directory_Name = 'Е:\ЛИНТЕР_Курсы'  
CONNECT BY PRIOR Directory_ID = Main_Directory_ID  
WHERE LEVEL > 1  
ORDER BY LEVEL;
```

2 Тестовые задания

2.1 Команда вместе со своими многочисленными опциями и предложениями используется для формирования запросов к реляционной базе данных.

Ответ - SELECT

Реализация в SQL концепции операций, ориентированных на табличное представление данных, позволила создать компактный язык с небольшим набором предложений. Язык SQL может использоваться как для выполнения запросов к данным, так и для построения прикладных программ.

Основные категории команд языка SQL предназначены для выполнения различных функций, включая построение объектов базы данных и манипулирование ими, начальную загрузку данных в таблицы, обновление и удаление существующей информации, выполнение запросов к базе данных, управление доступом к ней и ее общее администрирование.

Основные категории команд языка SQL:

- DDL – язык определения данных;
- DML – язык манипулирования данными;
- DQL – язык запросов;
- DCL – язык управления данными;
- команды администрирования данных;
- команды управления транзакциями

Выборка данных (DQL)

Язык запросов DQL наиболее известен пользователям реляционной базы данных, несмотря на то, что он включает одну команду SELECT. Эта команда вместе со своими многочисленными опциями и предложениями используется для формирования запросов к реляционной базе данных.

2.2 Оператор, позволяющий указать имена исходных таблиц, участвующих в формировании выборки, - это:

- GET
- OPEN
- EXTRACT
- SELECT

Ответ - SELECT

Все запросы на получение практически любого количества данных из одной или нескольких таблиц выполняются с помощью единственного предложения SELECT. В общем случае результатом реализации предложения SELECT является другая таблица.

3 Практические задания

3.1 Содержат ли тексты запросов ошибку? Если содержат, то в чем она заключается? Если запрос верный, что будет результатом выполнения запроса?

SELECT

Клиенты.Название,
Заказы.КодЗаказа,
Доставка.Название,
Заказы.ДатаРазмещения

FROM Клиенты

LEFT JOIN (Доставка

RIGHT JOIN Заказы **ON** Доставка.КодДоставки = Заказы.Доставка)

ON Клиенты.КодКлиента = Заказы.КодКлиента

ORDER BY Клиенты.Название, Заказы.КодЗаказа;

Запрос содержит ошибки. Правильный запрос будет выглядеть так:

SELECT

Клиенты.Название,
Заказы.КодЗаказа,
Доставка.Название,
Заказы.ДатаРазмещения

FROM Заказы

LEFT JOIN Доставка

ON Доставка.КодДоставки = Заказы.Доставка

LEFT JOIN Заказы **ON** Клиенты.КодКлиента = Заказы.КодКлиента

ORDER BY Клиенты.Название, Заказы.КодЗаказа;

Запрос получает список размещенных заказов из таблицы Заказы, подставляет в них названия контрагентов из таблицы Клиенты и вид доставки из таблицы Доставка, сортируя все это дело по контрагенту и коду заказа.

```
SELECT  
Типы.Категория,  
Count(Товары.КодТовара) AS [Count-  
КодТовара],  
Max(Товары.Цена) AS [Max-Цена],  
Типы.Описание,  
Min(Товары.МинимальныйЗапас) AS [Min-МинимальныйЗапас]  
FROM Типы  
LEFT JOIN Товары ON Типы.КодТипа = Товары.КодТипа  
GROUP BY Типы.Категория, Типы.Описание;
```

Запрос составлен верно. Он получает список категорий товаров, для каждой из которых рассчитывает количество товаров в категории, максимальную цену товара и наименьший минимальный запас.

3.2 Приведите текст SQL запроса, выполняющие заданные требования.

Вывести список сотрудников вида
Фамилия, Имя,Дата найма,количество заказов данного сотрудника.

Список отсортировать по дате найма по количеству заказов сотрудника по убыванию и по дате найма по возрастанию.

```
SELECT  
Сотрудники.Фамилия,  
Сотрудники.Имя,  
Сотрудники.ДатаНайма AS [ДатаНайма],  
Count(Заказы.КодЗаказа) AS [КоличествоЗаказов]  
FROM Сотрудники  
LEFT JOIN Заказы ON Сотрудники.КодСотрудника =  
Заказы.КодСотрудника  
GROUP BY  
Сотрудники.Фамилия,  
Сотрудники.Имя,  
Сотрудники.ДатаНайма,  
Сотрудники.КодСотрудника,  
Заказы.КодСотрудника  
ORDER BY ДатаНайма DESC, КоличествоЗаказов DESC, ДатаНайма  
ASC;
```


Контрольная работа №2 Вариант 16

1 Аналитическая часть

**Фраза WHERE в SQL-запросах на модификацию данных.
Использование фразы WHERE.**

Директива UPDATE имеет два формата. Первый из них:

```
UPDATE {базовая таблица | представление}  
SET столбец = значение [, столбец = значение] ...  
[WHERE фраза]
```

где значение — это столбец|выражение|константа|переменная и может включать столбцы лишь из обновляемой таблицы, т.е. значение одного из столбцов модифицируемой таблицы может заменяться на значение ее другого столбца или выражения, содержащего значения нескольких ее столбцов, включая изменяемый.

При отсутствии WHERE фразы обновляются значения указанных столбцов во всех строках модифицируемой таблицы. WHERE фраза позволяет сократить число обновляемых строк, указывая условия их отбора.

Второй формат описывает предложение, позволяющее производить обновление значений модифицируемой таблицы по значениям столбцов из других таблиц. К сожалению в ряде СУБД эти форматы отличаются друг от друга и от стандарта. Для примера приведем один из таких форматов:

```
UPDATE {базовая таблица | представление}  
SET столбец = значение [, столбец = значение] ...  
FROM {базовая таблица | представление} [псевдоним],  
{базовая таблица | представление} [псевдоним]  
[, {базовая таблица | представление}  
[псевдоним]] ...  
[WHERE фраза]
```

Здесь перечень таблиц фразы FROM содержит имя модифицируемой таблицы и тех таблиц, значения столбцов которых используются для обновления. При этом, естественно, таблицы должны быть связаны между собой в WHERE фразе, которая, кроме того, служит для указания условий отбора обновляемых строк модифицируемой таблицы.

В значениях, находящихся в правых частях равенств фразы SET, следует уточнять имена используемых столбцов, предваряя их именем таблицы (псевдонима).

Обновление единственной записи

Изменить количество и наименование товара с кодом P_CODE = 123457 в таблице PRODUCT:

UPDATE PRODUCT SET P_NAME = 'Шариковая ручка', P_QUANTITY = 50 WHERE P_CODE = 123457;

2 Тестовые задания

2.1 За удаление полей таблицы отвечает команда:

- ADD;
- DELETE;
- COLUMN;
- DROP.

Ответ **DELETE**

Примечание:

Основные категории команд языка SQL:

- DDL – язык определения данных;
- DML – язык манипулирования данными;
- DQL – язык запросов;
- DCL – язык управления данными;
- команды администрирования данных;
- команды управления транзакциями

Манипулирование данными (DML)

Язык манипулирования данными (Data Manipulation Language, DML) используется для манипулирования информацией внутри объектов реляционной базы данных посредством трех основных команд: INSERT, UPDATE, DELETE.

Модификация данных может выполняться с помощью предложений DELETE (удалить), INSERT (вставить) и UPDATE (обновить). Подобно предложению SELECT они могут оперировать как базовыми таблицами, так и представлениями. Однако по ряду причин не все представления являются обновляемыми.

Предложение DELETE имеет формат

DELETE FROM базовая таблица | представление [WHERE фраза];

и позволяет удалить содержимое всех строк указанной таблицы (при отсутствии WHERE фразы) или тех ее строк, которые выделяются WHERE фразой.

2.2 Оператор OR срабатывает когда ХОТЬ ОДНО условие истинно. AND когда ВСЕ условия истинны.

- Ложь
- Истина

Ответ **-ИСТИНА**

Примечание:

Фраза WHERE включает набор условий для отбора строк:

WHERE [NOT] WHERE_условие [{AND|OR} [NOT] WHERE_условие]...

где WHERE_условие – одна из следующих конструкций:

значение { = | <> | != | > | >= | !> | < | <= | !< } expression | string_expression [NOT] LIKE string_expression [ESCAPE 'escape_character'] | expression [NOT] BETWEEN expression AND expression | expression IS [NOT] NULL | CONTAINS ({ column | * } , '<contains_search_condition >') | FREETEXT ({ column | * } , 'freetext_string') | expression [NOT] IN (subquery | expression [,...n]) | expression { = | <> | != | > | >= | !> | < | <= | !< } { ALL | SOME | ANY } (subquery) | EXISTS (subquery) }

- AND - когда должны удовлетворяться оба разделяемых с помощью AND условия;
- OR - когда должно удовлетворяться одно из разделяемых с помощью OR условий;
- AND NOT - когда должно удовлетворяться первое условие и не должно второе;
- OR NOT - когда или должно удовлетворяться первое условие или не должно удовлетворяться второе, причем существует приоритет AND над OR (сначала выполняются все операции AND и только после этого операции OR).

3 Практические задания

3.1_Содержит ли текст запроса ошибку? Если содержит, то в чем она заключается? Если запрос верный, что произойдет в результате выполнения запроса?

```
INSERT INTO Доставка ( КодДоставки, Название, Телефон )  
SELECT  
Доставка.КодДоставки,  
Доставка.Название,  
Доставка.Телефон  
FROM Доп_Доставка  
WHERE (((Доставка.КодДоставки) > 10000)  
AND ((Доставка.КодДоставки) Is Not Null));
```

Запрос составлен верно. Хотя, условия несколько излишни — второе поглощается первым. Но это не ошибка запроса, скорее ошибка оптимизации.

В результате выполнения запроса из таблицы Доп_Доставка будут выбраны все записи в которых КодДоставки заполнен и превышает 10000. Затем выбранные записи будут добавлены в таблицу Доставка с заполнением полей КодДоставки, Название, Телефон.

3.2 Приведите тексты SQL-запросов, выполняющих заданные действия.

Увеличить на 50% цену всех товаров, поставляемых не российскими поставщиками, в случае если стоимость складского остатка меньше 10000.

```
UPDATE Товары, Поставщики  
SET Товары.Цена = Товары.Цена * 1.5  
WHERE (Товары.НаСкладе * Товары.Цена < 10000)  
AND (Товары.КодПоставщика = Поставщики.КодПоставщика)  
AND (Поставщики.Страна <> «Россия»)
```

В зависимости от версии и реализации СУБД возможен также такой вариант:

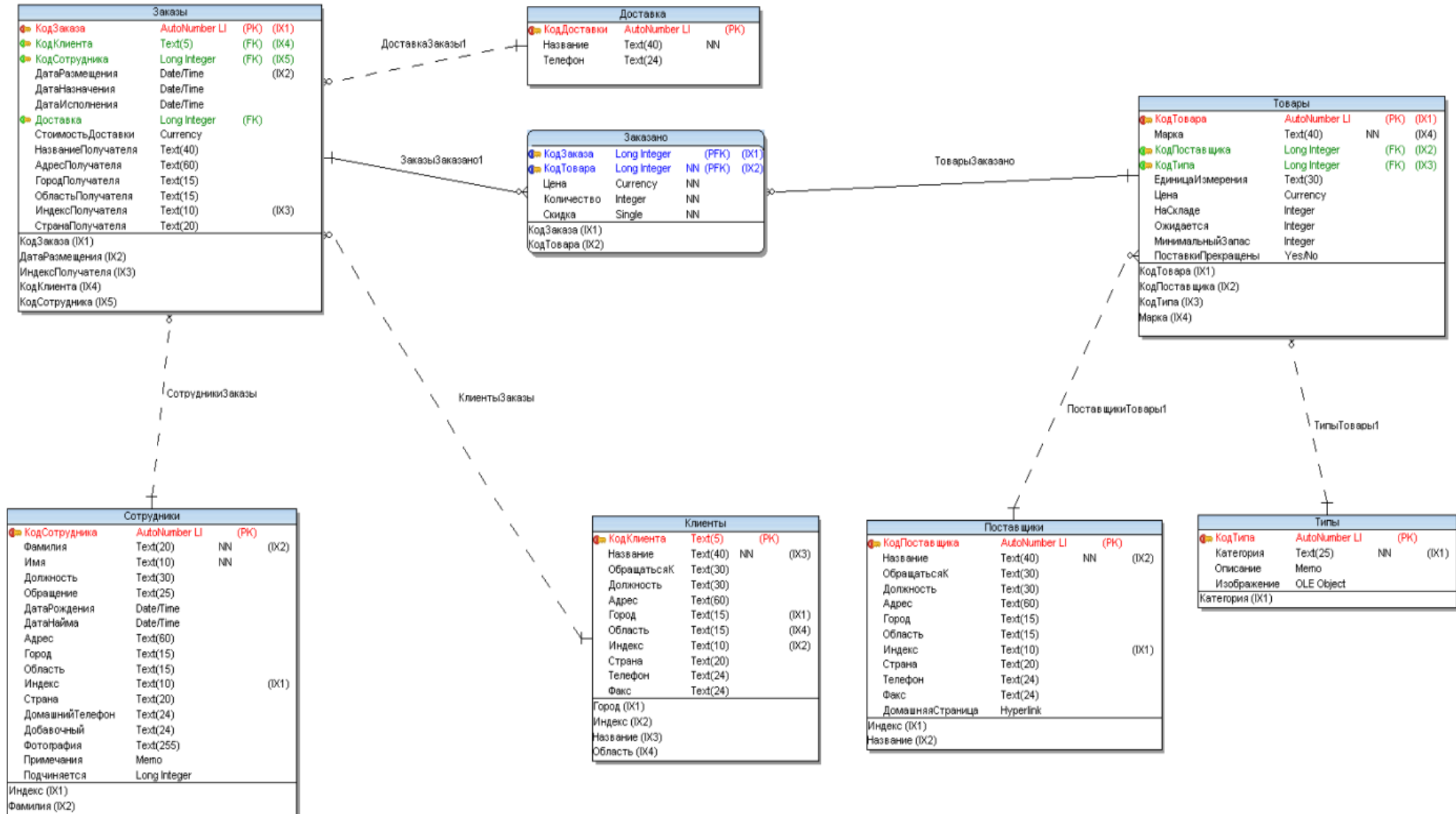
```
UPDATE Товары  
SET Товары.Цена = Товары.Цена * 1.5  
FROM Товары AS Товары  
LEFT JOIN Поставщики  
ON Товары.КодПоставщика = Поставщики.КодПоставщика  
WHERE (Товары.НаСкладе * Товары.Цена < 10000)  
AND (Поставщики.Страна <> «Россия»)
```

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1 А.А. Иванов Базы данных. Базовый курс : учеб. пособие для студентов заочной формы обучения / А.А. Иванов, Н.П. Жадановская. – СПб. : СПбГТИ(ТУ), 2016. – 101 с.
- 2 Кляйн, К. Е. SQL. Справочник / К. Е. Кляйн, Д. Кляйн, Б. Хант ; пер. с англ. – 3-е изд. – М. ; СПб. : Символ-Плюс, 2010. – 651 с.
- 3 Кузнецов, С. Д. Базы данных. Модели и языки / С. Д. Кузнецов. – СПб. : Бинном-Пресс, 2008. – 720 с.
- 4 Малыхина, М. П. Базы данных : основы, проектирование, использование / М. П. Малыхина. – 2-е изд. – СПб. : БХВ-Петербург, 2006. – 517 с.
- 5 Норенков, И. П. Автоматизированные информационные системы : учеб. пособие / И. П. Норенков. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2011. – 342 с.
- 6 Советов, Б. Я. Базы данных. Теория и практика / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. – 2-е изд., стер. – М. : Юрайт., 2012. – 463 с.
- 7 Советов, Б. Я. Информационные технологии : учеб. для вузов / Б. Я. Советов, В. В. Цехановский. – 3-е изд., стер. – М. : Высш. шк., 2006. – 263 с.

Приложение 1

Структура учебной базы данных «Борей»



Кафедра систем автоматизированного проектирования и управления

БАЗЫ ДАННЫХ. ЯЗЫКИ ЗАПРОСОВ.
Контрольные работы

Учебное пособие
для студентов заочной формы обучения

Иванов Александр Александрович
Жадановская Наталия Павловна

Отпечатано с оригинал макета. Формат 60x90 ¹/₁₆
Печ. л. 7,5. Тираж 100 экз. Заказ №

федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
«Санкт-Петербургский государственный технологический институт (технический университет)»

190013, г. Санкт-Петербург, Московский пр., д. 26