И.И. Семенова

SQL СТАНДАРТ В СУБД MS SQL SERVER, ORACLE, VFP И ACCESS: МАНИПУЛИРОВАНИЕ ДАННЫМИ

Учебное пособие

Омск • 2008

Федеральное агентство по образованию
Сибирская государственная автомобильно-дорожная академия
(СибАДИ)

И.И. Семенова

SQL СТАНДАРТ В СУБД MS SQL SERVER, ORACLE, VFP И ACCESS:

МАНИПУЛИРОВАНИЕ ДАННЫМИ

Учебное пособие

Омск
Издательство СибАДИ
2008

УДК 681.3.06
ББК 31.965
С 30

Рецензенты:

канд. техн. наук, доцент кафедры АСОИУ, В.Н. Цыганенко, ОмГТУ
канд. техн. наук, доцент кафедры СС иИБ, В.Г. Осипов, ОмГТУ

Рекомендовано редакционно-издательским советом академии в качестве учебно-практического издания по дисциплине «Системы управления базами данных» для студентов специальностей «Прикладная информатика в экономике», «Автоматизированные системы обработки информации и управления», «Комплексное обеспечение информационной безопасности автоматизированных систем».

Семенова И.И.
С 30 SQL стандарт в СУБД MS SQL SERVER, ORACLE, VFP И ACCESS: манипулирование данными.– Омск: Изд-во СибАДИ, 2008. – 57 с.

ISBN 978 – 5 – 93204 – 422 – 3

Основной целью создания данного сборника стала необходимость закрепления навыков построения команд в стандарте SQL для различных предметных областей с учетом особенностей программирования в СУБД MS SQL Server, Oracle, VFP, Access, у студентов высших учебных заведений, изучающих дисциплины “Базы данных” и “Системы управления базами данных”. Данный сборник будет полезен студентам, обучающимся на специальностях «Прикладная информатика в экономике», «Автоматизированные системы обработки информации и управления», «Комплексное обеспечение информационной безопасности автоматизированных систем».

Табл. 16. Ил. 22 . Библиогр.: 18 назв.

ISBN 978 – 5 – 93204 – 422 – 3

Оглавление

1. РАБОТА С БАЗОЙ ДАННЫХ В КОМАНДАХ SQL

Прежде чем перейти к самостоятельному выполнению заданий по вариантам, рассмотрим примеры построения команд для обработки данных в реляционной базе данных с использованием SQL стандарта. В качестве демонстрационной базы данных используем фрагмент, описывающий предметную область «Заработная плата».

1.1. Пример базы данных

Представленная на рисунке 1 упрощенная структура БД (табл. 1 – 3) позволяет вести учет сотрудников, работающих на предприятии, а также хранить все выданные заработные платы с их расшифровкой по доходным и расходным статьям (например, оклад, подоходный налог и т.д.).

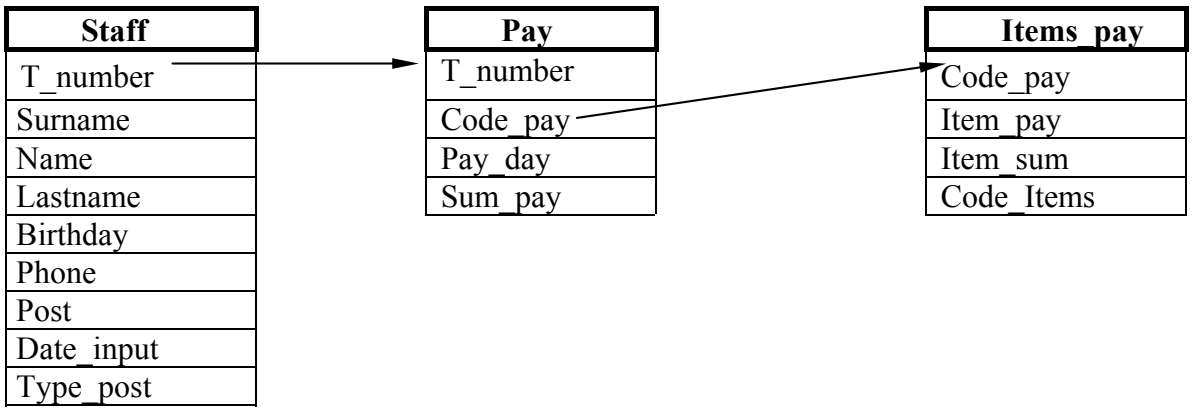

Рис. 1. Фрагмент базы данных «Заработная плата»

Связь между таблицами осуществляется с помощью следующих пар полей с типом связи «один-ко-многим» соответственно:

1. Staff.T_number- Pay.T_number.
2. Pay.Code_pay - Items_pay.Code_pay.

Таблица 1

Список сотрудников (таблица Staff)

Название поля	Тип поля	Описание поля
T_number	Integer	Табельный номер сотрудника (уникальный)
Surname	Character	Фамилия сотрудника
Name	Character	Имя сотрудника
Lastname	Character	Отчество сотрудника
Birthday	Date	Дата рождения сотрудника
Phone	Numeric	Контактный телефон сотрудника
Post	Character	Должность сотрудника
Type_post	Character	Тип сотрудника (ИТР, служащий, рабочий)
Date_input	Date	Дата устройства на работу

Таблица 2

Таблица учета выданной зарплаты (таблица Pay)

Название поля	Тип поля	Описание поля
T_number	Integer	Табельный номер сотрудника, получающего зарплату
Code_pay	Integer	Код выданной зарплаты (уникальный)
Pay_day	Date	Дата выдачи зарплаты
Sum_pay	Numeric	Общая сумма зарплаты на руки

Таблица 3

Таблица расшифровки каждой зарплаты по статьям (таблица Items_pay)

Название поля	Тип поля	Описание поля
Code_pay	Integer	Код выданной зарплаты
Item_pay	Character	Название статьи, по которой начисляют зарплату (как доход, так и расход)
Item_sum	Numeric	Сумма на получение или на вычет из зарплаты
Code_Items	Integer	Ключевое поле таблицы

В таблицах 4 – 6 представлен пример заполнения данными таблиц, которые будут использоваться для демонстрации результатов выполнения запросов.

Таблица 4

Пример заполнения таблицы Staff

T_number	Surname	Name	Lastname	Birthday	Phone	Post	Type_post	Date_input
1	Иванов	Иван	Петрович	12.01.1971	124563	Бухгалтер	Служащий	12.04.2000
2	Сидоров	Василий	Михайлович	14.06.1954	451263	Начальник отдела кадров	ИТР	14.11.1999
3	Васильков	Петр	Аркадьевич	14.06.1981	145236	Специалист отдела кадров	Служащий	30.11.2000
67	Артемьев	Иван	Васильевич	05.12.1970	365462	Главный инженер	ИТР	10.02.1998
4	Соянов	Савел	Игнатьевич	15.05.1981	121212	Строитель	Рабочий	25.06.1980
11	Ушаков	Виктор	Семенович	30.05.1970	156462	Бухгалтер	Служащий	18.11.2003
15	Иванова	Анна	Михайловна	12.03.1960	145214	Строитель	Рабочий	12.11.1979

Таблица 5

Пример заполнения таблицы Pay (фрагмент)

T_number	Code_pay	Pay_day	Sum_pay
1	1	01.01.2003	2544.00
1	2	01.02.2003	4521.00

Окончание табл. 5

T_number	Code_pay	Pay_day	Sum_pay
1	3	01.03.2003	12542.00
2	4	01.01.2003	1452.00
2	5	01.02.2003	2145.00
2	6	01.03.2003	2135.00
3	7	01.01.2003	4511.00
3	8	01.02.2003	1542.00
3	9	01.03.2003	1542.00
4	10	01.03.2003	2456.00

Таблица 6

Пример заполнения таблицы Items_pay (фрагмент)

Code_pay	Item_pay	Item_sum	Code_Items
1	Премия	124.00	1
1	Налог	-451.00	2
1	Оклад	1457.00	3
1	Поощрение	4512.00	4
1	Оплата учебы	145.00	5
2	Оклад	4656.00	6
2	Налог	-415.00	7
2	Поощрение	326.00	8
3	Оклад	1654.00	9
3	Премия квартальная	1213.00	10
10	За бездетность	-154.00	11
10	Оклад	1456.00	12
10	Премия разовая	1245.00	13
10	Налог подоходный	-452.00	14

Рассмотрим возможности программного создания описанного фрагмента базы данных в разных СУБД.

В SQL Server:

```
CREATE DATABASE DB_pay
/*создание БД с названием  DB_pay, команда выполняется отдельно от остальных, так как занимает некоторое время*/
USE DB_pay
/*сделать активной БД с названием  DB_pay*/

CREATE TABLE Staff(T_number INT IDENTITY(1,1) PRIMARY KEY, Surname CHAR(25), Name CHAR(25), Lastname CHAR(25), Birthday
```

SMALLDATETIME, Phone Numeric(13,0), Post CHAR(30), Type_post CHAR(8) DEFAULT 'Служащий', Date_input SMALLDATETIME DEFAULT Getdate())

CREATE TABLE Pay(T_number INT FOREIGN KEY REFERENCES Staff(T_number) ON UPDATE CASCADE, Code_pay INT IDENTITY(1,1) PRIMARY KEY, Pay_day SMALLDATETIME DEFAULT Getdate(), Sum_pay Numeric(8,2))

CREATE TABLE Items_pay(Code_pay INT FOREIGN KEY REFERENCES Pay(Code_pay), Item_pay CHAR(20) DEFAULT 'Оклад', Item_sum Numeric(8,2), Code_Items BIGINT IDENTITY(1,1) PRIMARY KEY)

В ORACLE:

```
/*Вводим набор операторов для создания администратора создаваемой БД*/
    CREATE USER "ADMIN_PAY"
       PROFILE "DEFAULT"
       IDENTIFIED BY "P@ssw0rd"
       DEFAULT TABLESPACE "USERS"
       TEMPORARY TABLESPACE "TEMP"
       ACCOUNT UNLOCK;

    GRANT "CONNECT" TO "ADMIN_PAY" WITH ADMIN OPTION;
    GRANT "DBA" TO "ADMIN_PAY" WITH ADMIN OPTION;
    GRANT "EXP_FULL_DATABASE" TO "ADMIN_PAY" WITH ADMIN OPTION;

    /*Теперь приступаем к созданию табличного пространства программно*/
CREATE TABLESPACE "DB_PAY"
   LOGGING
   DATAFILE 'C:\ORACLE\ORADATA\ORCL\DB_PAY.dbf' SIZE 5M EXTENT
   MANAGEMENT LOCAL;

/*Теперь переопределяем ранее созданного пользователя ADMIN_PAY на работу только в табличном пространстве DB_PAY*/
ALTER USER "ADMIN_PAY"  DEFAULT TABLESPACE "DB_PAY";

    CREATE TABLE ADMIN_PAY.Staff (T_number NUMBER(5), CONSTRAINT "ID_STAFF" PRIMARY KEY(T_number) USING INDEX TABLESPACE "DB_PAY", Surname CHAR(25), Name CHAR(25), Lastname
```

CHAR(25), Birthday DATE, Phone NUMBER(13,0), Post CHAR(30), Type_post CHAR(8) DEFAULT 'Служащий', Date_input DATE DEFAULT Sysdate) TABLESPACE "DB_PAY";

CREATE TABLE ADMIN_PAY.Pay(T_number NUMBER(5), CONSTRAINT "ID_STAFF_FK" FOREIGN KEY(T_number) REFERENCES ADMIN_PAY.Staff(T_number), Code_pay NUMBER(6), CONSTRAINT "ID_PAY" PRIMARY KEY(Code_pay) USING INDEX TABLESPACE "DB_PAY", Pay_day DATE DEFAULT Sysdate, Sum_pay NUMBER(8,2)) TABLESPACE "DB_PAY";

CREATE TABLE ADMIN_PAY.Items_pay(Code_pay NUMBER(6), CONSTRAINT "ID_PAY_FK" FOREIGN KEY(Code_pay) REFERENCES ADMIN_PAY.Pay(Code_pay), Item_pay CHAR(20) DEFAULT 'Оклад', Item_sum Numeric(8,2), Code_Items NUMBER(8) PRIMARY KEY USING INDEX TABLESPACE "DB_PAY") TABLESPACE "DB_PAY";

--создание последовательности для каждого ключевого поля
CREATE SEQUENCE ADMIN_PAY.ID_STAFF_SEQ INCREMENT BY 1 START WITH 1 MAXVALUE 99999 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

CREATE SEQUENCE ADMIN_PAY.ID_PAY_SEQ INCREMENT BY 1 START WITH 1 MAXVALUE 999999 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

CREATE SEQUENCE ADMIN_PAY.ID_ITEM_SEQ INCREMENT BY 1 START WITH 1 MAXVALUE 99999999 MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

1.2. Упражнения с использованием операторов обработки данных SQL

Сортировка.

1. Вывести все сведения о сотрудниках из таблицы Staff и отсортировать результат по табельному номеру:

– в VFP, MS SQL Server, Access:

SELECT * FROM Staff ORDER BY T_number

– в Oracle:

SELECT * FROM ADMIN_PAY.Staff ORDER BY T_number;

SELECT – ключевое слово, обозначающее начало SQL запроса, за ним обычно следует перечень полей, информация из которых помещается в результат выполнения запроса.

* – условное обозначение, которое позволит помещать в результат запроса информацию из всех полей таблицы, в которой осуществляется поиск (в некоторых СУБД используется ключевое слово ALL).

FROM – ключевое слово, после которого указывается имя источника/ов данных (если источников несколько, то они разделяются запятыми) для выполнения запроса.

При выполнении запроса в MS SQL Server убедитесь, что БД DB_Pay активна, или выполните команду USE DB_Pay.

При выполнении запроса в Oracle перед именем таблицы обязательно указывается имя схемы, к которой принадлежит таблица. В данном примере ADMIN_PAY.

2.Вывести список фамилий, имен, отчеств сотрудников, их должности, отсортировать результат по названиям должностей по возрастанию и по фамилиям по убыванию:

– в VFP, MS SQL Server, Access:

SELECT Surname, Name, Lastname, Post FROM Staff ORDER BY Post ASC, Surname DESC

– в Oracle:

SELECT Surname, Name, Lastname, Post FROM ADMIN_PAY.Staff ORDER BY Post ASC, Surname DESC;

ORDER BY – сортирует результаты запроса на основании данных, содержащихся в одном или нескольких столбцах, по умолчанию сортировка выполняется по возрастанию. Если это предложение не указано, результаты запроса не будут отсортированы.

ASC – сортировка данных по возрастанию значений поля, после которого стоит ключевое слово ASC.

DESC – сортировка данных по убыванию значений поля, после которого стоит ключевое слово DESC.

Если сортировка выполняется по нескольким полям, то порядок сортировки следующий:

– выполняется сортировка строк по первому указанному полю;

– внутри групп повторяющихся значений первого поля выполняется сортировка строк по второму полю и т.д.

3.Выбрать из таблицы Pay табельные номера сотрудников и даты получения зарплат и отсортировать результат по дате по убыванию (рис. 2):

– в VFP, MS SQL Server, Access:

SELECT T_number, Pay_day FROM Pay ORDER BY Pay_day DESC

– в Oracle:

SELECT T_number, Pay_day FROM ADMIN_PAY.Pay ORDER BY Pay_day DESC;

T_Number	Pay_day
1	01.03.2003
2	01.03.2003
3	01.03.2003
4	01.03.2003
1	01.02.2003
2	01.02.2003
3	01.02.2003
1	01.01.2003
2	01.01.2003
3	01.01.2003

Рис. 2. Сортировка
по дате

Изменение порядка следования полей.

4.Вывести все сведения о сотрудниках из таблицы Staff таким образом, чтобы в результате порядок столбцов был следующим: Name, Lastname, Surname, Post, Date_input, Phone, Birthday, T_number, Type_post (рис. 3):

– в VFP, MS SQL Server, Access:

SELECT Name, Lastname, Surname, Post, Date_input, Phone, Birthday, T_number, Type_post FROM Staff

– в Oracle:

SELECT Name, Lastname, Surname, Post, Date_input, Phone, Birthday, T_number, Type_post FROM ADMIN_PAY.Staff;

Surname	Post	Date_input	Phone	Birthday	T_number	Type_post
Иванов	Бухгалтер	12.04.2000	124563	12.01.1971	1	Служащий
Сидоров	Начальник отдела кадров	14.11.1999	451263	14.06.1954	2	ИТР
Васильков	Специалист отдела кадров	30.11.2000	145236	14.06.1981	3	Служащий
Артемьев	Главный инженер	10.02.1998	365462	05.12.1970	67	ИТР
Соянов	Строитель	25.06.1980	121212	15.05.1981	4	Рабочий
Ушаков	Бухгалтер	18.11.2003	156462	30.05.1970	11	Служащий
Иванова	Строитель	12.11.1979	145214	12.03.1940	15	Рабочий

Рис. 3. Результат запроса изменения порядка полей

5.Выбрать все поля из таблицы Pay таким образом, чтобы в результате порядок столбцов был следующим: Sum_pay, Pay_day, T_number, Code_pay:

– в VFP, MS SQL Server, Access:

SELECT Sum_pay, Pay_day, T_number, Code_pay FROM Pay

– в Oracle:

SELECT Sum_pay, Pay_day, T_number, Code_pay FROM ADMIN_PAY.Pay;

Выбор некоторых полей из двух (трех) таблиц.

6.Вывести список фамилий, имен, отчеств сотрудников (поля Surname, Name, Lastname), а также значения их заработных плат (поле Sum_pay) и даты получения (поле Pay_day):

– в VFP, MS SQL Server, Access:

SELECT Surname, Name, Lastname, Sum_pay, Pay_day FROM Staff, Pay WHERE Staff.T_number = Pay.T_number

– в Oracle:

SELECT Surname, Name, Lastname, Sum_pay, Pay_day FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE Staff.T_number = Pay.T_number;

WHERE – предложение WHERE показывает, что в результаты запроса следует включать только некоторые строки. Для отбора строк, включаемых в результаты запроса, используется условие поиска, которое строится как логическое выражение, состоящее из одного или нескольких условий, объединенных логическими конструкциями типа AND или OR.

7.Вывести табельные номера, даты получения зарплаты и ее расклад по статьям, результат отсортировать по табельному номеру сотрудника (рис. 4):

– в VFP, MS SQL Server, Access:

SELECT T_number, Pay_day, Item_pay, Item_sum FROM Pay, Items_pay WHERE Pay.Code_pay = Items_pay.Code_pay ORDER BY T_number

– в Oracle:

SELECT T_number, Pay_day, Item_pay, Item_sum FROM ADMIN_PAY.Pay, ADMIN_PAY.Items_pay WHERE Pay.Code_pay = Items_pay.Code_pay ORDER BY T_number;

T_Number	Pay_day	Item_pay	Item_sum
1	01.01.2003	Премия	124.00
1	01.01.2003	Налог	-451.00
1	01.01.2003	Оклад	1457.00
1	01.01.2003	Поощрение	4512.00
1	01.01.2003	Оплата учебы	145.00
1	01.02.2003	Оклад	4656.00
1	01.02.2003	Налог	-415.00
1	01.02.2003	Поощрение	326.00
1	01.03.2003	Оклад	1654.00
1	01.03.2003	Премия квартальная	1213.00
4	01.03.2003	За бездетность	-154.00
4	01.03.2003	Оклад	1456.00
4	01.03.2003	Премия разовая	1245.00
4	01.03.2003	Налог подоходный	-452.00

Рис. 4. Результат запроса с выбором полей из таблиц

8. Вывести список фамилий и табельных номеров сотрудников, а также значения их заработных плат и даты получения с раскладкой каждой зарплаты по статьям:

– в VFP, MS SQL Server, Access:

SELECT Surname, Staff.T_number, Sum_pay, Pay_day, Item_pay, Item_sum FROM Staff, Pay, Items_pay WHERE (Staff.T_number = Pay.T_number) AND (Pay.Code_pay = Items_pay.Code_pay)

– в Oracle:

SELECT Surname, Staff.T_number, Sum_pay, Pay_day, Item_pay, Item_sum FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay, ADMIN_PAY.Items_pay WHERE (Staff.T_number = Pay.T_number) AND (Pay.Code_pay = Items_pay.Code_pay);

Если в запросе участвует несколько таблиц и в них встречаются поля с одинаковыми названиями, то обязательно рядом с полем указывать название таблицы, из которой берется поле. Например: Staff.T_number

AND – "логическое И", выполняет роль объединения двух условий и возвращает результат ИСТИНА, оба условия также возвращают результат ИСТИНА. В результат запроса помещаются только те строки, которые соответствуют условиям=ИСТИНА, записанным после ключевого слова WHERE.

Связь таблиц в запросе – несмотря на то, что в базе данных установлены связи между таблицами, при построении запроса нужно также указать правила связи между таблицами.

Самый простой способ связать таблицы: в условии WHERE указать условия равенства полей связи пары таблиц; если нужно объединить три и более таблиц, то нужно перечислить пары полей связи и объединить их "логическими И", как показано в примере.

Условие совпадения.

9.Вывести список сотрудников с должностью, название которой начинается на ‘главный’:

– в MS SQL Server, Access (регистр текста не важен, но пробелы слева и количество символов сравниваемых текстов значимы):

SELECT Surname, Name, Lastname, Post FROM Staff WHERE Post = 'главный'

Результат выполнения запроса не будет содержать ни одной строки, так как в соответствии с примером заполнения (табл. 4) нет строк с должностью 'главный' или 'Главный'. Так же будет выглядеть результат запроса в VFP с условием точного совпадения.

– в VFP с условием точного совпадения:

SET ANSI ON &&условие точного совпадения

SELECT Surname, Name, Lastname, Post FROM Staff WHERE Post = 'главный'

– в VFP с условием неточного совпадения (рис. 5):

SET ANSI OFF &&условие неточного совпадения

SELECT Surname, Name, Lastname, Post FROM Staff WHERE Post = 'главный'

Surname	Name	Lastname	Post
Артемьев	Иван	Васильевич	Главный инженер

Рис. 5. Результат запроса с условием неточного совпадения

– в Oracle (регистр текста, пробелы слева и количество символов сравниваемых текстов значимы):

SELECT Surname, Name, Lastname, Post FROM ADMIN_PAY.Staff WHERE Post = 'главный';

' … ' – для выделения в запросе строковых значений используются одинарные кавычки. Например, ‘главный’.

Точное несовпадение значений одного из полей.

10. Вывести список сотрудников и их должности, которые не являются служащими:

– в VFP, MS SQL Server, Access:

SELECT Surname, Name, Lastname, Post FROM Staff WHERE NOT(Type_post = 'служащий')

В MS SQL Server, если в некоторых строках в столбце Type_post будет значение NULL, то эти строки не попадут в результат выполнения запроса, хотя логика функции NOT( ) должна вернуть истину.

– в Oracle:

SELECT Surname, Name, Lastname, Post FROM ADMIN_PAY.Staff WHERE NOT(Type_post = 'служащий');

В Oracle в отличие от MS SQL Server, если в некоторых строках в столбце Type_post будет значение NULL, эти строки попадут в результат выполнения запроса.

NOT( ) – функция "логического НЕ". В примере если условие в скобках вернет ИСТИНУ, то функция NOT( ) изменит его на противоположное ЛОЖЬ и в результат строка помещена не будет. Поэтому в запросе будут выбраны только те работники, которые не являются служащими.

11. Вывести список сотрудников, которые не являются бухгалтерами, и их даты поступления на работу (рис. 6):

– в VFP, MS SQL Server, Access:

SELECT Name, Lastname, Surname, Date_input FROM Staff WHERE NOT(Post = 'бухгалтер')

– в Oracle:

SELECT Name, Lastname, Surname, Date_input FROM ADMIN_PAY.Staff WHERE NOT(Post = 'бухгалтер');

Name	Lastname	Surname	Date_input
Василий	Михайлович	Сидоров	14.11.1999
Петр	Аркадьевич	Васильков	30.11.2000
Иван	Васильевич	Артемьев	10.02.1998
Савел	Игнатьевич	Соянов	25.06.1980
Анна	Михайловна	Иванова	12.11.1979

Рис. 6. Список сотрудников, не являющихся бухгалтерами

Выбор записей по диапазону значений (Between).

12. Вывести список сотрудников и размеры полученных зарплат за период 01.01.2003 по 01.03.2003 (рис. 7):

– в VFP:

SELECT Name, Lastname, Surname, Sum_pay, Pay_day FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day BETWEEN CTOD(‘01.01.2003’) AND CTOD(‘01.03.2003’)

Name	Lastname	Surname	Sum_pay	Pay_day
Иван	Петрович	Иванов	2544.00	01.01.2003
Иван	Петрович	Иванов	4521.00	01.02.2003
Иван	Петрович	Иванов	12542.00	01.03.2003
Василий	Михайлович	Сидоров	1452.00	01.01.2003
Василий	Михайлович	Сидоров	2145.00	01.02.2003
Василий	Михайлович	Сидоров	2135.00	01.03.2003
Петр	Аркадьевич	Васильков	4511.00	01.01.2003
Петр	Аркадьевич	Васильков	1542.00	01.02.2003
Петр	Аркадьевич	Васильков	1542.00	01.03.2003
Савел	Игнатьевич	Соянов	2456.00	01.03.2003

Рис. 7. Выбор по диапазону

– в MS SQL Server:

SELECT Name, Lastname, Surname, Sum_pay FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day BETWEEN '1-JAN-2003' AND '1-MAR-2003'

или

SET DATEFORMAT dmy

SELECT Name, Lastname, Surname, Sum_pay FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day BETWEEN '1-01-2003' AND '1-03-2003'

– в Access:

SELECT Name, Lastname, Surname, Sum_pay FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day BETWEEN #01.01.2003# AND #01.03.2003#

– в Oracle:

SELECT Name, Lastname, Surname, Sum_pay FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day BETWEEN '1-JAN-2003' AND '1-MAR-2003';

или

SELECT Name, Lastname, Surname, Sum_pay FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day

BETWEEN to_date('1-01-2003','dd-mm-yyyy') AND to_date('1-03-2003','dd-mm-yyyy');

BETWEEN – проверка на принадлежность диапазону значений. При этом проверяется, находится ли значение поля между двумя определенными значениями.

Особенности оформления дат в различных СУБД .

В Access дата заключается в решетки # # (формат дд.мм.гггг).

В VFP оформляется как строка и преобразуется в формат даты с помощью функции CTOD( ).

В MS SQL Server можно задать как строку в двойных кавычках в формате дд-мес-гггг (где месяц может быть оформлен как JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC).

13. Вывести список сотрудников, которые были устроены на работу в период с 12.03.2000 по 15.06.2000, и их должности:

– в VFP:

SELECT Name, Lastname, Surname, Post FROM Staff WHERE Date_input BETWEEN CTOD(‘12.03.2000’) AND CTOD(‘15.06.2000’)

– в MS SQL Server:

SELECT Name, Lastname, Surname, Post FROM Staff WHERE Date_input BETWEEN '12-MAR-2000' AND '15-JUN-2000'

– в MS Access:

SELECT Name, Lastname, Surname, Sum_day FROM Staff WHERE Date_input BETWEEN #12.03.2000# AND #15.06.2000#

– в Oracle:

SELECT Name, Lastname, Surname, Post FROM ADMIN_PAY.Staff WHERE Date_input BETWEEN to_date('12-03-2000', 'dd-mm-yyyy') AND to_date('15-06-2000', 'dd-mm-yyyy');

14. Вывести список сотрудников и их телефоны, значения которых находятся в диапазоне с 111111 по 222222:

– в VFP, MS SQL Server, Access:

SELECT Name, Lastname, Surname, Phone FROM Staff WHERE Phone BETWEEN 111111 AND 222222

– в Oracle:

SELECT Name, Lastname, Surname, Phone FROM ADMIN_PAY.Staff WHERE Phone BETWEEN 111111 AND 222222;

15. Вывести список сотрудников, у которых фамилия начинается на одну из букв диапазона ‘Р’ – ‘У’:

– в VFP, MS SQL Server, Access:

SELECT Name, Lastname, Surname FROM Staff WHERE Surname BETWEEN 'Р' AND 'У'

– в Oracle:

SELECT Name, Lastname, Surname FROM ADMIN_PAY.Staff WHERE Surname BETWEEN 'Р' AND 'У';

Выбор записей по диапазону значений (In).

16. Вывести список сотрудников с должностями ‘начальник отдела кадров’, ‘специалист отдела кадров’, ‘операционист отдела кадров’:

– в VFP, MS SQL Server, Access:

SELECT Name, Lastname, Surname, Post FROM Staff WHERE Post IN('начальник отдела кадров', 'специалист отдела кадров', 'операционист отдела кадров')

– в Oracle:

SELECT Name, Lastname, Surname, Post FROM ADMIN_PAY.Staff WHERE Post IN('начальник отдела кадров', 'специалист отдела кадров', 'операционист отдела кадров');

IN( ) – проверка на членство в множестве. Вывести только те строки, у которых значение указанного поля принадлежит указанному множеству, т.е. равно одному из значений, перечисленных в IN( ).

17. Вывести список сотрудников, получающих одну из следующих надбавок к зарплате: ‘премию’, ‘оплату учебы’, ‘поощрение’:

– в VFP, MS SQL Server, Access:

SELECT Name, Lastname, Surname FROM Staff, Pay, Items_pay WHERE (Staff.T_number = Pay.T_number) AND (Pay.Code_pay = Items_pay.Code_pay) AND (Item_pay IN('премия', 'оплата учебы', 'поощрение'))

– в Oracle:

SELECT Name, Lastname, Surname FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay, ADMIN_PAY.Items_pay WHERE (Staff.T_number = Pay.T_number) AND (Pay.Code_pay = Items_pay.Code_pay) AND (Item_pay IN('премия', 'оплата учебы', 'поощрение'));

18. Вывести список сотрудников с табельными номерами 4, 67, 45, 77 (рис. 8):

– в VFP, MS SQL Server, Access:

SELECT Name, Lastname, Surname, T_Number FROM Staff WHERE T_Number IN(4, 67, 45, 77)

– в Oracle:

SELECT Name, Lastname, Surname, T_Number FROM ADMIN_PAY.Staff WHERE T_Number IN(4, 67, 45, 77);

Name	Lastname	Surname	T_number
Иван	Васильевич	Артемьев	67
Савел	Игнатьевич	Соянов	4

Рис. 8. Список сотрудников
по указанным номерам

Выбор записей с использованием Like.

19. Вывести неповторяющийся список статей в зарплате, которые начинаются на букву ‘н’:

– в MS SQL Server(регистр текста не важен):

SELECT DISTINCT Item_pay FROM Items_pay WHERE Item_pay LIKE 'н%'

– в Access:

SELECT DISTINCT Item_pay FROM Items_pay WHERE Item_pay LIKE “н*”

– в Oracle (регистр текста учитывается):

SELECT DISTINCT Item_pay FROM ADMIN_PAY.Items_pay WHERE Item_pay LIKE 'н%';

LIKE( ) – проверка на соответствие шаблону, где шаблон записывается в двойных кавычках. % или * – подстановочный знак в шаблоне, совпадающий с любой последовательностью из нуля и более символов.
_ или ? – подстановочный знак в шаблоне, совпадающий с одним любым символом на указанном месте. Пример шаблона на Access: "?нар*" – вывести все строки, у которых первый символ любой, далее обязательная последовательность *нар*, конец строки любой.

20. Вывести список сотрудников, отчества которых содержат сочетание букв ‘ва’:

– в VFP, MS SQL Server:

SELECT Name, Lastname, Surname FROM Staff WHERE Lastname LIKE '%ва%'

– в Access:

SELECT Name, Lastname, Surname FROM Staff WHERE Lastname LIKE “*ва*”

– в Oracle:

SELECT Name, Lastname, Surname FROM ADMIN_PAY.Staff WHERE Lastname LIKE '%ва%';

21. Выбрать неповторяющийся список должностей, у которых значение оканчивается на ‘ль’ (рис. 9):

– в VFP, MS SQL Server:

SELECT DISTINCT Post FROM Staff WHERE Post LIKE '%ль'

Post
Строитель

Рис. 9. Результат запроса
с использованием Like

– в Access:

SELECT DISTINCT Post FROM Staff WHERE Post LIKE “*ль”

– в Oracle:

SELECT DISTINCT Post FROM ADMIN_PAY.Staff WHERE Post LIKE '%ль';

Выбор записей по нескольким условиям

22. Вывести всех сотрудников, которые получили зарплату 15.03.2003 в размере от 2000 до 3000 руб.:

– в VFP:

SELECT Name, Lastname, Surname FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day = CTOD(‘15.03.2003’) AND ((Sum_pay>=2000) AND (Sum_pay<3000))

– в MS SQL Server:

SELECT Name, Lastname, Surname FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day = '15-MAR-2003' AND ((Sum_pay>=2000) AND (Sum_pay<3000))

– в Access:

SELECT Name, Lastname, Surname FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day = #15.03.2003# AND ((Sum_pay>=2000) AND (Sum_pay<3000))

– в Oracle:

SELECT Name, Lastname, Surname FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day = '15-MAR-2003' AND ((Sum_pay>=2000) AND (Sum_pay<3000));

или

SELECT Name, Lastname, Surname FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE (Staff.T_number = Pay.T_number) AND Pay_day = to_date('15-MAR-2003', 'dd-mm-yyyy') AND ((Sum_pay>=2000) AND (Sum_pay<3000));

23. Вывести НЕПОВТОРЯЮЩИЙСЯ список табельных номеров и имен сотрудников с табельными номерами 12 – 30 или с зарплатами, превысившими размер 5000 руб.:

– в VFP, MS SQL Server, Access:

SELECT DISTINCT Name, Lastname, Surname, Staff.T_number FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND ((Staff.T_Number BETWEEN 12 AND 30) OR Sum_pay>5000)

– в Oracle:

SELECT DISTINCT Name, Lastname, Surname, Staff.T_number FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE (Staff.T_number = Pay.T_number) AND ((Staff.T_Number BETWEEN 12 AND 30) OR Sum_pay>5000);

24. Вывести список сотрудников с датами рождения 01.01.1950 – 01.01.1960 или табельными номерами из диапазона 10 – 150 (рис. 10):

– в VFP:

SELECT Name, Lastname, Surname, Birthday, T_number FROM Staff WHERE (Birthday BETWEEN CTOD(‘01.01.1950’) AND CTOD(‘01.01.1960’)) OR (T_number>=10 AND T_number<=150)

Name	Lastname	Surname	Birthday	T_number
Василий	Михайлович	Сидоров	14.06.1954	2
Иван	Васильевич	Артемьев	05.12.1970	67
Виктор	Семенович	Ушаков	30.05.1970	11
Анна	Михайловна	Иванова	12.03.1960	15

Рис. 10. Результат запроса с несколькими условиями

– в MS SQL Server:

SELECT Name, Lastname, Surname, Birthday, T_number FROM Staff WHERE (Birthday BETWEEN '01-JAN-1950' AND '01-JAN-1960') OR (T_number>=10 AND T_number<=150)

– в Access:

SELECT Name, Lastname, Surname, Birthday, T_number FROM Staff WHERE (Birthday BETWEEN #01.01.1950# AND #01.01.1960#) OR (T_number>=10 AND T_number<=150)

– в Oracle:

SELECT Name, Lastname, Surname, Birthday, T_number FROM ADMIN_PAY.Staff WHERE (Birthday BETWEEN '01-JAN-1950' AND '01-JAN-1960') OR (T_number>=10 AND T_number<=150);

Многотабличные запросы (выборка из двух таблиц, выборка из трех таблиц с использованием JOIN).

25. Вывести список сотрудников, получающих одну из следующих надбавок к зарплате: ‘премию’, ‘оплату учебы’, ‘поощрение’:

– в VFP, MS SQL Server, Access:

SELECT Name, Lastname, Surname FROM Staff INNER JOIN Pay INNER JOIN Items_pay ON Pay.Code_pay = Items_pay.Code_pay ON Staff.T_number = Pay.T_number WHERE Item_pay IN('премия', 'оплата учебы', 'поощрение')

или

SELECT Name, Lastname, Surname FROM (Staff INNER JOIN Pay ON Staff.T_number = Pay.T_number) INNER JOIN Items_pay ON Pay.Code_pay = Items_pay.Code_pay WHERE Item_pay IN('премия', 'оплата учебы', 'поощрение')

– в Oracle:

SELECT Name, Lastname, Surname FROM (ADMIN_PAY.Staff INNER JOIN ADMIN_PAY.Pay ON Staff.T_number = Pay.T_number) INNER JOIN ADMIN_PAY.Items_pay ON Pay.Code_pay = Items_pay.Code_pay WHERE Item_pay IN('премия', 'оплата учебы', 'поощрение');

INNER JOIN создает объединение пары таблиц, из которого выбираются только те записи, которые содержат совпадающие значения в полях связи, указанных после ключевого слова ON.

LEFT JOIN создает объединение пары таблиц, из которого выбираются все записи из левой таблицы, а также записи из правой таблицы, значения поля связи которой совпадают со значениями поля связи левой таблицы.

RIGHT JOIN создает объединение пары таблиц, из которой выбираются все записи из правой таблицы, а также записи из левой таблицы, значения поля связи которой совпадают со значениями поля связи правой таблицы.

ON – ключевое слово, после которого указывается условие связи пары таблиц.

26. Вывести неповторяющийся список всех сотрудников, у которых размер зарплаты составил от 2000 до 3000 руб. (рис. 11):

– в VFP, MS SQL Server, Access:

SELECT DISTINCT Name, Lastname, Surname FROM Staff INNER JOIN Pay ON Staff.T_number = Pay.T_number WHERE (Sum_pay>=2000) AND (Sum_pay<3000)

– в Oracle:

SELECT DISTINCT Name, Lastname, Surname FROM ADMIN_PAY.Staff INNER JOIN ADMIN_PAY.Pay ON Staff.T_number = Pay.T_number WHERE (Sum_pay>=2000) AND (Sum_pay<3000);

Name	Lastname	Surname
Василий	Михайлович	Сидоров
Иван	Петрович	Иванов
Савел	Игнатьевич	Соянов

Рис. 11. Результат
многотабличного запроса

27. Вывести коды зарплат, в которых была статья вычетов ‘за бездетность’:

– в VFP, MS SQL Server, Access:

SELECT Pay.Code_pay FROM Pay INNER JOIN Items_pay ON Pay.Code_pay = Items_pay.Code_pay WHERE Item_pay = 'за бездетность'

– в Oracle:

SELECT Pay.Code_pay FROM ADMIN_PAY.Pay INNER JOIN ADMIN_PAY.Items_pay ON Pay.Code_pay = Items_pay.Code_pay WHERE Item_pay = 'за бездетность';

28. Вывести неповторяющийся список всех сотрудников, в которых была в зарплате статья вычетов ‘за бездетность’:

– в VFP, MS SQL Server, Access:

SELECT DISTINCT Name, Lastname, Surname FROM Staff INNER JOIN Pay INNER JOIN Items_pay ON Pay.Code_pay = Items_pay.Code_pay ON Staff.T_number = Pay.T_number WHERE Item_pay = 'за бездетность'

– в Oracle:

SELECT DISTINCT Name, Lastname, Surname FROM ADMIN_PAY.Staff INNER JOIN ADMIN_PAY.Pay INNER JOIN ADMIN_PAY.Items_pay ON Pay.Code_pay = Items_pay.Code_pay ON Staff.T_number = Pay.T_number WHERE Item_pay = 'за бездетность';

Вычисления.

29. Вывести список сотрудников, должности и срок их работы в годах с сортировкой по уменьшению стажа (рис. 12):

– в VFP, Access:

SELECT Name, Lastname, Surname, Post, (Date() - Date_input)/365.25 FROM Staff ORDER BY Date_input

– в MS SQL Server:

SELECT Name, Lastname, Surname, Post, CAST((GetDate() - Date_input) AS Bigint)/365.25 FROM Staff ORDER BY Date_input

– в Oracle:

SELECT Name, Lastname, Surname, Post, (SysDate - Date_input)/365.25 FROM ADMIN_PAY.Staff ORDER BY Date_input;

Name	Lastname	Surname	Post	Exp_5
Анна	Михайловна	Иванова	Строитель	25.1061
Савел	Игнатьевич	Соянов	Строитель	24.4873
Иван	Васильевич	Артемьев	Главный инженер	6.8583
Василий	Михайлович	Сидоров	Начальник отдела кадров	5.1006
Иван	Петрович	Иванов	Бухгалтер	4.6899
Петр	Аркадьевич	Васильков	Специалист отдела кадров	4.0548
Виктор	Семенович	Ушаков	Бухгалтер	1.0897

Рис. 12. Результат запроса с вычислением

30. Вывести список сотрудников, у которых еще не было дня рождения в текущем году, а также вывести количество дней до их дней рождения в текущем году:

– в VFP:

SET DATE TO GERMAN

&& необходима для установки даты в формате дд.мм.гг

SELECT Name, Lastname, Surname, Post, Birthday, CTOD(str(day(Birthday))+'.'+str(month(Birthday))+'.'+str(YEAR(Date())))-DATE() FROM Staff Where CTOD (str(day(Birthday)) + ' . ' + str (month(Birthday)) + ' . ' + str (YEAR(Date())))-DATE()) >0

– в MS SQL SERVER:

SET DATEFORMAT dmy --дата в формате дд.мм.гггг

SELECT Name, Lastname, Surname, Post, Birthday,

DATEDIFF(day, getdate(), CAST(str(day(Birthday))+ '.' + str(month(Birthday)) + '.' +str(YEAR(GetDate())) AS datetime)) AS [Дней до дня рождения] FROM Staff Where DATEDIFF(day, getdate(), CAST(str(day(Birthday)) + '.' + str(month(Birthday)) + '.' + str(YEAR(GetDate())) AS datetime))>0

– в Oracle:

SELECT Surname, TO_NUMBER(TO_DATE((to_char(Birthday, 'dd')||'.'||to_char(Birthday, 'mm')||'.'||to_char(Sysdate,'yyyy')), 'DD-MM-YYYY')-SYSDATE) FROM ADMIN_PAY.Staff WHERE TO_NUMBER(TO_DATE((to_char(Birthday,'dd')||'.'||to_char(Birthday,'mm')||'.'||to_char(Sysdate,'yyyy')), 'DD-MM-YYYY')-SYSDATE) >0;

31. Вывести список всех сотрудников, их табельные номера, даты и суммы получения зарплаты на руки и зарплаты, если бы у них не брали налог ‘за бездетность’:

– в VFP, MS SQL Server, Access:

SELECT Staff.T_number, Name, Surname, Pay_day, Sum_pay, (Sum_pay-Item_sum) FROM Staff INNER JOIN Pay INNER JOIN Items_pay ON Pay.Code_pay = Items_pay.Code_pay ON Staff.T_number = Pay.T_number WHERE Item_pay = 'за бездетность'

– в Oracle:

SELECT Staff.T_number, Name, Surname, Pay_day, Sum_pay, (Sum_pay-Item_sum) FROM ADMIN_PAY.Staff INNER JOIN ADMIN_PAY.Pay INNER JOIN ADMIN_PAY.Items_pay ON Pay.Code_pay = Items_pay.Code_pay ON Staff.T_number = Pay.T_number WHERE Item_pay = 'за бездетность';

В формуле запроса стоит минус, т.к. в таблице значения налогов хранятся как отрицательные числа.

Вычисление итоговых значений с использованием агрегатных функций.

32. Вывести среднюю зарплату, которая когда-либо выдавалась на предприятии:

– в VFP, MS SQL Server, Access:

SELECT AVG(Sum_pay) FROM Pay

– в Oracle:

SELECT AVG(Sum_pay) FROM ADMIN_PAY.Pay;

AVG( ) – функция вычисляет среднее всех значений, содержащихся в столбце. COUNT( ) – функция подсчитывает количество значений, содержащихся в столбце. COUNT(*) – функция подсчитывает количество строк в таблице результатов запроса. MAX( ) – функция находит наибольшее среди всех значений, содержащихся в столбце. MIN( ) – функция находит наименьшее среди всех значений, содержащихся в столбце. SUM( ) – функция вычисляет сумму всех значений, содержащихся в столбце.

33. Вывести список сотрудников и суммарную зарплату каждого:

– в VFP, MS SQL Server, Access:

SELECT RTRIM(Name)+' '+RTRIM(Lastname)+' '+Surname, Staff.T_number, SUM(Sum_pay) FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) GROUP BY Staff.T_number, RTRIM(Name) + ' ' + RTRIM(Lastname)+' '+Surname

– в Oracle:

SELECT RTRIM(Name)+' '+RTRIM(Lastname)+' '+Surname, Staff.T_number, SUM(Sum_pay) FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE (Staff.T_number = Pay.T_number) GROUP BY Staff.T_number, RTRIM(Name) + ' ' + RTRIM(Lastname)+' '+Surname;

GROUP BY позволяет создавать итоговый запрос. Обычный запрос включает в результат по одной строке для каждой строки из базы данных. Итоговый запрос, напротив, вначале группирует строки базы данных по определенному признаку, а затем включает в результаты запроса одну итоговую строку для каждой группы.

Предложение GROUP BY позволяет вести расчет итогов внутри каждой группы, в данном случае расчет суммарной зарплаты каждого сотрудника. Если бы мы не использовали GROUP BY, то в результате получили бы сумму зарплат всех сотрудников без разбиения по сотрудникам.

HAVING позволяет выводить не все результаты группировки, а только те, которые удовлетворяют указанному условию. После конструкции HAVING можно указывать только условия на агрегатные функции.

34. Вывести среднюю зарплату каждого сотрудника за прошедший год, у которых она получилась больше 10000:

– в VFP:

SELECT Name, Staff.T_number, AVG(Sum_pay) FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND (Pay_day BETWEEN CTOD('01.01.2002') AND CTOD('31.12.2002')) GROUP BY Staff.T_number, Name HAVING AVG(Sum_pay)>10000

– в MS SQL Server:

SELECT Name, Staff.T_number, AVG(Sum_pay) FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND (Pay_day BETWEEN '01-JAN-2002' AND '31-DEC-2002') GROUP BY Staff.T_number, Name HAVING AVG(Sum_pay)>10000

– в Access:

SELECT Name, Staff.T_number, AVG(Sum_pay) FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND (Pay_day BETWEEN

#01.01.2002# AND #31.12.2002#) GROUP BY Staff.T_number, Name HAVING AVG(Sum_pay)>10000

– в Oracle:

SELECT Name, Staff.T_number, AVG(Sum_pay) FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE (Staff.T_number = Pay.T_number) AND (Pay_day BETWEEN '01-JAN-2002' AND '31-DEC-2002') GROUP BY Staff.T_number, Name HAVING AVG(Sum_pay)>10000;

35. Вывести количество сотрудников по каждой должности, в которой работают меньше 5 сотрудников:

– в VFP, MS SQL Server, Access:

SELECT Post, Count(T_number) FROM Staff GROUP BY Post HAVING Count(T_number)<5

– в Oracle:

SELECT Post, Count(T_number) FROM ADMIN_PAY.Staff GROUP BY Post HAVING Count(T_number)<5;

36. Вывести дату устройства на работу самого первого и последнего сотрудников (рис. 13):

– в VFP, MS SQL Server, Access:

SELECT Min(Date_input), Max(Date_input) FROM Staff

– в Oracle:

SELECT Min(Date_input), Max(Date_input) FROM ADMIN_PAY.Staff;

Min_date_input	Max_date_input
12.11.1979	18.11.2003

Рис. 13. Итоговые значения

Изменение наименований полей.

37. Вывести список ФИО сотрудников, который поместить в поле с названием ФИО, и суммарную зарплату каждого, которую поместить в поле с названием Itog:

– в MS SQL Server:

SELECT Name+Lastname+Surname AS [Ф.И.О.], Staff.T_number, SUM(Sum_pay) AS Itog FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) GROUP BY Staff.T_number, Name+Lastname+Surname

– в Oracle:

SELECT Name+Lastname+Surname AS "Ф.И.О.", Staff.T_number, SUM(Sum_pay) AS Itog FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE (Staff.T_number = Pay.T_number) Group by Staff.T_number, Name+Lastname+Surname;

AS – ключевое слово, назначающее полю или выражению альтернативное название, которое будет отражено в результате запроса.

38. Вывести список всех сотрудников, их табельные номера, даты и суммы получения зарплаты на руки и зарплаты, если бы у них не брали ‘подоходный налог’, результат поместить в столбец Sum_With_Nalog:

– в VFP, MS SQL Server, Access:

SELECT Staff.T_number, Name, Surname, Pay_day, Sum_pay, (Sum_pay-Item_sum) AS Sum_With_Nalog FROM Staff INNER JOIN Pay INNER JOIN Items_pay ON Pay.Code_pay = Items_pay.Code_pay ON Staff.T_number = Pay.T_number WHERE Item_pay = 'подоходный налог'

– в Oracle:

SELECT Staff.T_number, Name, Surname, Pay_day, Sum_pay, (Sum_pay-Item_sum) AS Sum_With_Nalog FROM ADMIN_PAY.Staff INNER JOIN ADMIN_PAY.Pay INNER JOIN ADMIN_PAY.Items_pay ON Pay.Code_pay = Items_pay.Code_pay ON Staff.T_number = Pay.T_number WHERE Item_pay = 'подоходный налог';

В формуле запроса стоит минус, так как в таблице значения налогов хранятся как отрицательные числа.

39. Объединить данные фамилии, имена, отчества в одном столбце с названием FIO (рис. 14):

– в VFP, MS SQL Server, Access:

SELECT (RTRIM(Surname) + ' ' + RTRIM(Name) + ' '+ Lastname) AS FIO FROM Staff

– в Oracle:

SELECT (RTRIM(Surname) + ' ' + RTRIM(Name) + ' '+ Lastname) AS FIO FROM ADMIN_PAY.Staff;

FIO
Иванов Иван Петрович
Сидоров Василий Михайлович
Васильков Петр Аркадьевич
Артемьев Иван Васильевич
Соянов Савел Игнатьевич
Ушаков Виктор Семенович
Иванова Анна Михайловна

Рис. 14. Объединение данных

40. Объединить данные фамилии, имена, отчества и названия должности в одном столбце с названием FIO_Post:

– в VFP, MS SQL Server, Access:

SELECT (RTRIM(Surname) + ' ' + RTRIM(Name) + ' ' + RTRIM(Lastname) + ' в должности ' + Post) AS FIO_Post FROM Staff

– в Oracle:

SELECT (RTRIM(Surname) + ' ' + RTRIM(Name) + ' ' + RTRIM(Lastname) + ' в должности ' + Post) AS FIO_Post FROM ADMIN_PAY.Staff;

Использование переменных в условии.

41. Вывести список сотрудников, принятых на работу за последний месяц:

– в VFP:

```
Local Perem_B, Perem_E              && объявление местной переменной
Perem_B=GOMONTH(Date(),-1)          && дата начала интересующего периода
Perem_E = Date()                    && дата конца интересующего периода
```

SELECT Name, Lastname, Surname FROM Staff WHERE Date_Input BETWEEN Perem_B AND Perem_E

– в MS SQL Server:

```
-- объявление местной переменной
Declare @Perem_B DateTime, @Perem_E  DateTime
-- дата начала интересующего периода
SET @Perem_B=DATEADD ( month , -1, getdate())
-- дата конца интересующего периода
SET @Perem_E = GetDate( )
```

SELECT Name, Lastname, Surname FROM Staff WHERE Date_Input BETWEEN @Perem_B AND @Perem_E

– в Oracle:

```
SET SERVEROUTPUT ON;
Declare
Perem_B Date;
Perem_E Date;
Name_ ADMIN_PAY.Staff.Name%TYPE;
Lastname_ ADMIN_PAY.Staff.Lastname%TYPE;
Surname_ ADMIN_PAY.Staff.Surname%TYPE;

BEGIN
 Perem_B:= ADD_MONTHS(Sysdate,-1) ;
 Perem_E:= Sysdate;
 DBMS_OUTPUT.PUT_LINE(Perem_B||' '||Perem_E);
```

SELECT Name, Lastname, Surname INTO Name_, Lastname_, Surname_ FROM ADMIN_PAY.Staff WHERE Date_Input BETWEEN Perem_B AND Perem_E;

END;

42. Вывести список сотрудников, возраст которых меньше заданного (рис. 15):

– в VFP:

```
Local Perem                    && объявление местной переменной
Perem = 45
SELECT Name, Lastname, Surname FROM Staff WHERE
((Day(Birthday)+Month(Birthday)*30.5)/365.25-Year(Birthday)+Year(Date()))
< Perem
```

– в MS SQL Server:

```
Declare @Perem Int
-- назначение возраста
SET @Perem= 45
SELECT Name, Lastname, Surname FROM Staff WHERE CAST(
(getdate( )-Birthday) AS INT) < @Perem
```

– в Oracle:

```
-- объявление местной переменной
Declare
 Perem number(2);
 Name_ ADMIN_PAY.Staff.Name%TYPE;
 Lastname_ ADMIN_PAY.Staff.Lastname%TYPE;
 Surname_ ADMIN_PAY.Staff.Surname%TYPE;
BEGIN
 -- назначение возраста
 Perem:= 45;
 SELECT Name, Lastname, Surname INTO Name_, Lastname_, Surname_
FROM ADMIN_PAY.Staff WHERE trunc((Sysdate-Birthday)/365.25) <
Perem;
END;
```

Name	Lastname	Surname
Иван	Петрович	Иванов
Петр	Аркадьевич	Васильков
Иван	Васильевич	Артемьев
Савел	Игнатьевич	Соянов
Виктор	Семенович	Ушаков

Рис. 15. Результат запроса
с использованием переменных

43. Вывести список сотрудников с фамилиями, начинающимися на ‘Ив’:

– в VFP:

```
Local Perem              && объявление местной переменной
Perem = ‘Ив’
SET ANSI OFF             && настройка правила сравнения
SELECT Name, Lastname, Surname FROM Staff WHERE  Surname = Perem
```

– в MS SQL Server:

```
Declare @Perem VarChar(10)
-- назначение переменной
SET @Perem= 'Ив'
SELECT Name, Lastname, Surname FROM Staff WHERE  Surname LIKE RTRIM(@Perem)+'%'
```

– в Oracle:

```
Declare
Perem VarChar2(10);
Surname_ ADMIN_PAY.Staff.Surname%TYPE;
-- назначение переменной
BEGIN
Perem:= 'Ив';
SELECT Surname INTO Surname_ FROM ADMIN_PAY.Staff WHERE Surname LIKE RTRIM(Perem)+'%';
END;
```

Использование переменных вместо названий таблиц.

44. Вывести список всех сотрудников, их табельные номера, даты и суммы получения зарплаты на руки и зарплаты, если бы у них не брали ‘подоходный налог’:

– в VFP, MS SQL Server, Access:

```
SELECT a.T_number, Name, Surname, Pay_day, Sum_pay, (Sum_pay-Item_sum) FROM Staff a, Pay b, Items_pay c WHERE b.Code_pay = c.Code_pay AND a.T_number = b.T_number AND Item_pay = 'подоходный налог'
```

– в Oracle:

```
SELECT a.T_number, Name, Surname, Pay_day, Sum_pay, (Sum_pay-Item_sum) FROM ADMIN_PAY.Staff a, ADMIN_PAY.Pay b, ADMIN_PAY.Items_pay c WHERE b.Code_pay = c.Code_pay AND a.T_number = b.T_number AND Item_pay = 'подоходный налог';
```

Использование переменных вместо названий таблиц позволяет сократить размер кода создаваемого запроса и сделать его более читаемым.

45. Вывести список сотрудников и суммарную зарплату каждого (рис. 16):

– в VFP, MS SQL Server, Access:

SELECT Name, Lastname, Surname, d.T_number, SUM(Sum_pay) FROM Staff d, Pay f WHERE (d.T_number = f.T_number) GROUP BY d.T_number

– в Oracle:

SELECT Surname, d.T_number, SUM(Sum_pay) FROM ADMIN_PAY.Staff d, ADMIN_PAY.Pay f WHERE (d.T_number = f.T_number) GROUP BY d.T_number, Surname;

Name	Lastname	Surname	T_number	Sum_sum_pay
Иван	Петрович	Иванов	1	19607.00
Василий	Михайлович	Сидоров	2	5732.00
Петр	Аркадьевич	Васильков	3	7595.00
Савел	Игнатьевич	Соянов	4	2456.00

Рис. 16. Результат запроса

46. Вывести список сотрудников, получающих одну из следующих надбавок к зарплате: ‘премию’, ‘оплату учебы’, ‘поощрение’, и коды их зарплат:

– в VFP, MS SQL Server, Access:

SELECT Name, Lastname, Surname, b.Code_pay FROM Staff a, Pay b, Items_pay c WHERE b.Code_pay = c.Code_pay AND a.T_number = b.T_number AND Item_pay IN('премия', 'оплата учебы', 'поощрение')

– в Oracle:

SELECT Name, Lastname, Surname, b.Code_pay FROM ADMIN_PAY.Staff a, ADMIN_PAY.Pay b, ADMIN_PAY.Items_pay c WHERE b.Code_pay = c.Code_pay AND a.T_number = b.T_number AND Item_pay IN('премия', 'оплата учебы', 'поощрение');

Выбор результата в курсор.

47. Вывести все сведения о зарплатах сотрудника с фамилией ‘Алеев’ и именем ‘Павел’ и поместить результат во временную таблицу с названием Temp1:

– в VFP:

SELECT Name, Lastname, Surname, Sum_pay, Pay_Day FROM Staff, Pay INTO CURSOR Temp1 WHERE (Staff.T_number = Pay.T_number) AND Surname = ‘Алеев’ AND Name = ‘Павел’

– в MS SQL Server:

```
Declare TEMP1 CURSOR FOR SELECT Name, Lastname, Surname, Sum_pay, Pay_Day FROM Staff, Pay WHERE (Staff.T_number = Pay.T_number) AND Surname = 'Алеев' AND Name = 'Павел'
```

– в Oracle (с примером построчного вывода данных из курсора):

```
SET SERVEROUTPUT ON
DECLARE
Name1 ADMIN_PAY.Staff.Name%TYPE;
Lastname1 ADMIN_PAY.Staff.Lastname%TYPE;
Surname1 ADMIN_PAY.Staff.Surname%TYPE;
Sum_pay1 ADMIN_PAY.Pay.Sum_pay%TYPE;
Pay_Day1 ADMIN_PAY.Pay.Pay_Day%TYPE;

CURSOR TEMP1 IS SELECT Name, Lastname, Surname, Sum_pay, Pay_Day FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE (Staff.T_number = Pay.T_number) AND Surname = 'Алеев' AND Name = 'Павел';
BEGIN
OPEN TEMP1;
WHILE TEMP1%found LOOP
FETCH TEMP1 INTO Name1, Lastname1, Surname1, Sum_pay1, Pay_Day1;
DBMS_OUTPUT.PUT_LINE(Name1||' '|| Lastname1||' '||Surname1||' '|| Sum_pay1||' '|| Pay_Day1);
END LOOP;
CLOSE TEMP1;
END;
```

Для того чтобы использовать результаты запроса в дальнейшем коде программы, необходимо запрос сохранить либо на диске в таблице (ключевая фраза INTO DBF или INTO TABLE) с заданным названием, либо во временной таблице, которая сохраняется только на период работы программы или в рамках сессии.

INTO CURSOR – поместить результат запроса во временную таблицу с указанным названием (в примере Temp1), которая будет удалена из памяти по окончании работы программы.

48. Вывести все сведения о сотрудниках с табельными номерами 12–54 и поместить результат во временную таблицу с названием Temp2 (рис. 17):

– в VFP:

```
SELECT * FROM Staff INTO CURSOR Temp2 WHERE T_number BETWEEN 12 AND 54
```

– в MS SQL Server:

Declare Temp2 CURSOR FOR SELECT * FROM Staff WHERE T_number BETWEEN 12 AND 54

– в Oracle:

```
DECLARE
CURSOR TEMP1 IS SELECT * FROM ADMIN_PAY.Staff WHERE T_number BETWEEN 12 AND 54;
BEGIN
  Null;
END;
```

T_number	Surname	Name	Lastname	Birthday	...	Date_input
15	Иванова	Анна	Михайловна	12.03.1960	...	12.11.1979

Рис. 17. Фрагмент выбора результата в курсор

Использование совместно с подзапросом квантора существования.

49. Вывести неповторяющийся список сотрудников, которые получали премию:

– в VFP, MS SQL Server, Access:

SELECT DISTINCT Name, Lastname, Surname FROM Staff, Pay WHERE Staff.T_number = Pay.T_number AND EXISTS(SELECT * FROM Items_pay WHERE Items_pay.Code_pay = Pay.Code_pay AND Item_pay='премия')

– в Oracle:

SELECT DISTINCT Name, Lastname, Surname FROM ADMIN_PAY.Staff, ADMIN_PAY.Pay WHERE Staff.T_number = Pay.T_number AND EXISTS(SELECT * FROM ADMIN_PAY.Items_pay WHERE Items_pay.Code_pay = Pay.Code_pay AND Item_pay='премия');

EXISTS( ) – квантор существования, понятие, заимствованное из формальной логики. Возвращает два значения: либо ИСТИНА, либо ЛОЖЬ. ИСТИНА – если условие, указанное в скобках, выполнилось и имеет ненулевой результат, ЛОЖЬ – если условие вернуло пустое множество.

50. Вывести список сотрудников, которые ни разу не получали зарплаты:

– в VFP, MS SQL Server, Access:

SELECT Surname, Name, Lastname FROM Staff WHERE NOT EXISTS(SELECT * FROM Pay WHERE Staff.T_number = Pay.T_number)

– в Oracle:

SELECT Surname, Name, Lastname FROM ADMIN_PAY.Staff WHERE NOT EXISTS(SELECT * FROM ADMIN_PAY.Pay WHERE Staff.T_number = Pay.T_number);

51. Вывести список сотрудников, у которых размер зарплаты не меньше 3000 руб. (рис. 18):

– в VFP, MS SQL Server, Access:

SELECT Surname, Name, Lastname FROM Staff WHERE EXISTS(SELECT * FROM Pay WHERE Staff.T_number = Pay.T_number AND Sum_pay >=3000)

– в Oracle:

SELECT Surname, Name, Lastname FROM ADMIN_PAY.Staff WHERE EXISTS(SELECT * FROM ADMIN_PAY.Pay WHERE Staff.T_number = Pay.T_number AND Sum_pay >=3000);

Surname	Name	Lastname
Иванов	Иван	Петрович
Васильков	Петр	Аркадьевич

Рис. 18. Результат запроса
с использованием квантора
существования

Использование функций совместно с подзапросом.

52. Вывести список сотрудников и даты с размерами полученных зарплат, которые превысили средний размер их же зарплат (рис. 19):

– в VFP, MS SQL Server, Access:

SELECT Surname, Name, Lastname, Sum_pay, Pay_Day FROM Staff INNER JOIN PAY ON Staff.T_number = Pay.T_number WHERE Pay.Sum_pay>(SELECT AVG(Sum_pay) FROM Pay)

– в Oracle:

SELECT Surname, Name, Lastname, Sum_pay, Pay_Day FROM ADMIN_PAY.Staff INNER JOIN ADMIN_PAY.PAY ON Staff.T_number = Pay.T_number WHERE Pay.Sum_pay>(SELECT AVG(Sum_pay) FROM ADMIN_PAY.Pay);

Surname	Name	Lastname	Sum_pay	Pay_day
Иванов	Иван	Петрович	12542.00	01.03.2003
Иванов	Иван	Петрович	4521.00	01.02.2003
Васильков	Петр	Аркадьевич	4511.00	01.01.2003

Рис. 19. Результат запроса с подзапросом

Оператор обработки данных Update.

53. Перевести всех сотрудников в статус ‘ИТР’, у которых название должности начинается с ‘главный’:

– в VFP:

```
SET ANSI OFF
UPDATE Staff SET Type_post = ‘ИТР’ WHERE Post = ‘главный’
```

– в MS SQL Server:

```
UPDATE Staff SET Type_post = 'ИТР' WHERE Post LIKE 'главный%'
```

– в Oracle:

```
UPDATE ADMIN_PAY.Staff SET Type_post = 'ИТР' WHERE Post LIKE 'главный%';
```

Оператор UPDATE обновляет значения одного или нескольких столбцов в выбранных строках одной таблицы. В операторе указывается целевая таблица, которая должна быть модифицирована, при этом пользователь должен иметь право на обновление. Предложение WHERE отбирает строки таблицы, подлежащие обновлению. В предложении SET указывается, какие столбцы должны быть обновлены, и для них закладываются новые значения.

54. Перевести всех сотрудников в статус ‘пенс.’, а значение должности удалить, если стаж их работы больше 20 лет и возраст больше 60 лет:

– в VFP:

```
UPDATE Staff SET Type_post = ‘пенс.’, Post = ‘’ WHERE (Date()-Date_Input)/365.25>20 AND (Date()-Birthday)/365.25>60
```

– в MS SQL Server:

```
UPDATE Staff SET Type_post = 'пенс.', Post = ' ' WHERE CAST(DATEDIFF(Year,Date_Input,GetDate()) AS INT)>20 AND CAST(DATEDIFF(Year,Birthday,GetDate()) AS INT)>60
```

– в Oracle:

```
UPDATE ADMIN_PAY.Staff SET Type_post = 'пенс.', Post = ' ' WHERE (Sysdate-Date_Input)/365.25>20 AND (Sysdate- Birthday)/365.25>60;
```

55. Изменить значение Post на ‘нет сведений’, если значение поля является пустым:

– в VFP:

```
UPDATE Staff SET Post = ‘нет сведений’ WHERE ALLTRIM(Post) = ‘’
```

– в MS SQL Server:

```
UPDATE Staff SET Post = 'нет сведений' WHERE RTRIM(Post) = '' OR Post IS Null
```

– в Oracle:

UPDATE ADMIN_PAY.Staff SET Post = 'нет сведений' WHERE RTRIM(Post) = '' OR Post IS Null;

Оператор обработки данных Insert.

56. Добавить в таблицу сотрудников новую запись, причем так, чтобы табельный номер был автоматически увеличен на 1, а в должности стояло значение ‘нет сведений’:

– в VFP (если табельный номер без автонаращивания):

SELECT MAX(T_number) AS Max_ FROM Staff INTO CURSOR Temp

INSERT INTO Staff(T_number, Post)VALUES(Temp.Max_+1, ‘нет сведений’)

– в MS SQL Server (если табельный номер с автонаращиванием, объявленным через IDENTITY(1,1)):

INSERT INTO Staff(Post)VALUES('нет сведений')

– в MS SQL Server(если табельный номер без автонаращивания):

Declare @max bigint

SET @max = (SELECT MAX(T_number) FROM Staff)

SET @max = ISNULL(@max, 0)

INSERT INTO Staff(T_number, Post)VALUES(@max+1, 'нет сведений')

– в Oracle (если для табельного номера объявлена последовательность SEQUENCE для генерации уникальных ключей):

INSERT INTO ADMIN_PAY.Staff(T_number, Post)VALUES (ADMIN_PAY.ID_STAFF_SEQ.NextVal, 'нет сведений');

– в Oracle (если для табельного номера нет объявленной последовательности):

```
DECLARE
Max_ int;
BEGIN
SELECT MAX(T_number) INTO Max_ FROM ADMIN_PAY.Staff;
IF Max_ IS NULL THEN
  Max_:=0;
END IF;
INSERT INTO ADMIN_PAY.Staff(T_number, Post)VALUES (Max_+1, 'нет сведений');
END;
```

Перед выполнением оператора добавления новой строки использован SELECT для того, чтобы определить максимальное значение ключевого индексного поля таблицы и при добавлении новой строки автоматически нарастить его, тем самым не нарушая

целостности таблицы (исключая появление в ключевом поле двух одинаковых значений).

Оператор INSERT добавляет в таблицу новую строку. В предложении INTO указывается таблица, в которую добавляется новая строка (целевая таблица), а в предложении VALUES содержатся значения данных для новой строки. Список столбцов определяет, какие значения в какой столбец заносятся. В столбцы, не перечисленные в первых скобках, будут автоматически записаны значения типа NULL (пустые или неопределенные).

Список столбцов в первых скобках должен строго соответствовать списку записываемых значений во вторых скобках.

57. Добавить в таблицу Pay новую запись, причем так, чтобы код зарплаты был автоматически увеличен на 1, табельный номер =23, дата зарплаты = текущей дате, а размер зарплаты = 5000:

– в VFP (если код зарплаты без автонаращивания):

```
SELECT MAX(Code_pay) AS Max_ FROM Pay INTO CURSOR Temp
INSERT INTO Pay(T_number, Code_pay, Pay_day, Sum_pay) VALUES(23, Temp.Max_+1, Date(), 5000)
```

– в MS SQL Server (если код зарплаты с автонаращиванием, объявленным через IDENTITY(1,1)):

```
INSERT INTO Pay(T_number, Pay_day, Sum_pay) VALUES(23, GetDate(), 5000)
```

– в MS SQL Server(если код зарплаты без автонаращивания):

```
Declare @max bigint
SET @max = (SELECT MAX(Code_pay) FROM Pay)
SET @max = ISNULL(@max, 0)
INSERT INTO Pay(T_number, Code_pay, Pay_day, Sum_pay) VALUES(23, @max+1, GetDate(), 5000)
```

– в Oracle (если для кода зарплаты объявлена последовательность SEQUENCE для генерации уникальных ключей):

```
INSERT INTO ADMIN_PAY.Pay(T_number, Code_pay, Pay_day, Sum_pay) VALUES(23, ADMIN_PAY.ID_PAY_SEQ.NextVal, SysDate, 5000);
```

– в Oracle (если для кода зарплаты нет объявленной последовательности):

```
DECLARE
Max_ int;
BEGIN
SELECT MAX(Code_pay) INTO Max_ FROM ADMIN_PAY.Pay;
```

```
if Max_ IS NULL then
  Max_:=0;
END IF;
INSERT INTO ADMIN_PAY.Pay(T_number, Code_pay, Pay_day, Sum_pay) VALUES(23, Max_+1, SysDate, 5000);
END;
```

Данная команда INSERT сработает, если в главной таблице Staff есть запись с T_number=23 или при отсутствии поддержки целостности БД.

58. Добавить в таблицу Items_pay новую запись, причем так, чтобы код зарплаты был 45, название статьи зарплаты = ‘премия’, а размер премии = 1500 руб. (рис. 20):

– в VFP, MS SQL Server, Access (если ключевое поле Code_Items с автонаращиванием):

```
INSERT INTO Items_pay(Code_pay, Item_pay, Item_sum) VALUES(45, 'премия', 1500)
```

– в Oracle (если ключевое поле Code_Items наращивается через SEQUENCE):

```
INSERT INTO ADMIN_PAY.Items_pay(Code_pay, Item_pay, Item_sum, Code_Items) VALUES(45, 'премия', 1500, ADMIN_PAY.ID_ITEM_SEQ.NextVal);
```

Данная команда INSERT сработает, если в главной таблице Pay есть запись с Code_pay=45 или при отсутствии поддержки целостности БД.

Оператор обработки данных Delete.

59. Удалить из таблицы всех сотрудников, у которых возраст больше 80 лет (рис. 20):

Code_pay	Item_pay	Item_sum
1	Оклад	1457.00
1	Поощрение	4512.00
1	Оплата учебы	145.00
2	Оклад	4656.00
2	Налог	-415.00
2	Поощрение	326.00
3	Оклад	1654.00
3	Премия квартальная	1213.00
10	За бездетность	-154.00
10	Оклад	1456.00
10	Премия разовая	1245.00
10	Налог подоходный	-452.00
45	Премия	1500.00

Рис. 20. Результат запроса
с использованием оператора Insert

– в VFP:

DELETE FROM Staff WHERE (Date()-Birthday)>80

– в MS SQL Server:

DELETE FROM Staff WHERE CAST(DATEDIFF(Year, Birthday, GetDate()) AS INT) >80

– в Oracle:

DELETE FROM ADMIN_PAY.Staff WHERE (Sysdate-Birthday)/365.25>80;

Оператор DELETE удаляет выбранные строки данных из одной таблицы. В предложении FROM указывается таблица, содержащая строки, которые требуется удалить. В предложении WHERE указываются строки, которые должны быть удалены.

При выполнении команды обратите внимание на выполнение правил целостности баз данных.

60. Удалить из таблицы Статьи зарплат (таблица Items_pay) все записи, у которых в поле названия статьи зарплаты = ‘не известно’:

– в VFP, MS SQL Server, Access:

DELETE FROM Items_pay WHERE Item_pay='не известно'

– в Oracle:

DELETE FROM ADMIN_PAY.Items_pay WHERE Item_pay='не известно';

61. Удалить из таблицы зарплат все записи, в которых значения полей сумма зарплаты и табельный номер сотрудника равны 0:

– в VFP, MS SQL Server, Access:

DELETE FROM Pay WHERE T_number = 0 AND Sum_pay = 0

– в Oracle:

DELETE FROM ADMIN_PAY.Pay WHERE T_number = 0 AND Sum_pay = 0;

2. УПРАЖНЕНИЯ НА SQL

2.1. База данных «Книжное дело»

На рисунке 21 представлен фрагмент упрощенной схемы данных для выполнения индивидуальных заданий. Прежде чем приступить к выполнению заданий необходимо в новой базе данных с названием DB_BOOKS создать таблицы со структурой, показанной в таблицах 7–11, установить связи между таблицами, затем заполнить тестовыми данными.

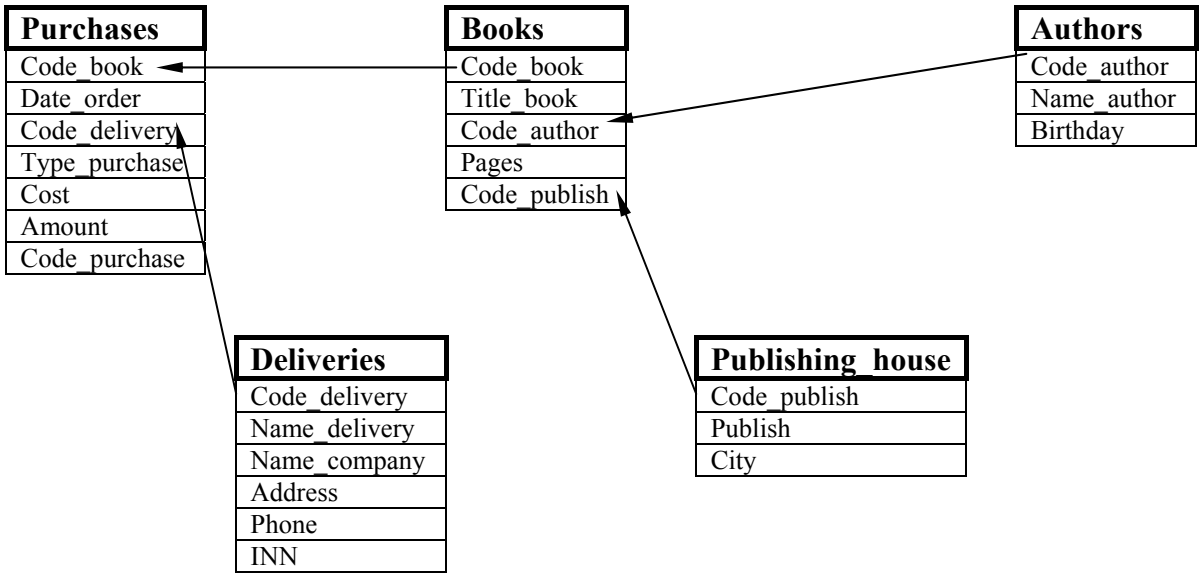

Рис. 21. Фрагмент базы данных «Книжное дело»

Связь между таблицами осуществляется с помощью следующих пар полей с типом связи «один-ко-многим» соответственно:

1. Books.Code_book - Purchases.Code_book;
2. Deliveries.Code_delivery - Purchases.Code_delivery;
3. Authors.Code_author - Books.Code_author;
4. Publising_house.Code_publish - Books.Code_publish.

Таблица 7

Покупки (название таблицы Purchases)

Название поля	Тип поля	Описание поля
Code_book	Integer	Код закупаемой книги
Date_order	Datetime	Дата заказа книги
Code_delivery	Integer	Код поставщика
Type_purchase	Bit	Тип закупки (опт/ розница)
Cost	Money	Стоимость единицы товара
Amount	Integer	Количество экземпляров
Code_purchase	Integer	Код покупки

Таблица 8

Справочник книг (название таблицы Books)

Название поля	Тип поля	Описание поля
Code_book	Integer	Код книги
Title_book	Char	Название книги
Code_author	Integer	Код автора
Pages	Integer	Количество страниц
Code_publish	Integer	Код издательства

Таблица 9

Справочник авторов (название таблицы Authors)

Название поля	Тип поля	Описание поля
Code_author	Integer	Код автора
Name_ author	Char	Фамилия, имя, отчество автора
Birthday	Datetime	Дата рождения

Таблица 10

Справочник поставщиков (название таблицы Deliveries)

Название поля	Тип поля	Описание поля
Code_delivery	Integer	Код поставщика
Name_delivery	Char	Фамилия, и., о. ответственного лица
Name_company	Char	Название компании-поставщика
Address	Char	Юридический адрес
Phone	Numeric	Телефон контактный
INN	Char	ИНН

Таблица 11

Справочник издательств (название таблицы Publishing_house)

Название поля	Тип поля	Описание поля
Code_publish	Integer	Код издательства
Publish	Char	Издательство
City	Char	Город

2.2. Упражнения с использованием операторов обработки данных для БД «Книжное дело»

Сортировка.

1.Выбрать все сведения о книгах из таблицы Books и отсортировать результат по коду книги (поле Code_book).

2.Выбрать из таблицы Books коды книг, названия и количество страниц (поля Code_book, Title_book и Pages), отсортировать результат по

названиям книг (поле Title_book по возрастанию) и по полю Pages (по убыванию).

3.Выбрать из таблицы Deliveries список поставщиков (поля Name_delivery, Phone и INN), отсортировать результат по полю INN (по убыванию).

Изменение порядка следования полей.

4.Выбрать все поля из таблицы Deliveries таким образом, чтобы в результате порядок столбцов был следующим: Name_delivery, INN, Phone, Address, Code_delivery.

5.Выбрать все поля из таблицы Publishing_house таким образом, чтобы в результате порядок столбцов был следующим: Publish, City, Code_publish.

Выбор некоторых полей из двух таблиц.

6.Выбрать из таблицы Books названия книг и количество страниц (поля Title_book и Pages), а из таблицы Authors выбрать имя соответствующего автора книги (поле Name_ author).

7.Выбрать из таблицы Books названия книг и количество страниц (поля Title_book и Pages), а из таблицы Deliveries выбрать имя соответствующего поставщика книги (поле Name_delivery).

8.Выбрать из таблицы Books названия книг и количество страниц (поля Title_book и Pages), а из таблицы Publishing_house выбрать название соответствующего издательства и места издания (поля Publish и City).

Условие неточного совпадения.

9.Выбрать из справочника поставщиков (таблица Deliveries) названия компаний, телефоны и ИНН (поля Name_company, Phone и INN), у которых название компании (поле Name_company) начинается с ‘ОАО’.

10. Выбрать из таблицы Books названия книг и количество страниц (поля Title_book и Pages), а из таблицы Authors выбрать имя соответствующего автора книг (поле Name_ author), у которых название книги начинается со слова ‘Мемуары’.

11. Выбрать из таблицы Authors фамилии, имена, отчества авторов (поле Name_ author), значения которых начинаются с ‘Иванов’.

Точное несовпадение значений одного из полей.

12. Вывести список названий издательств (поле Publish) из таблицы Publishing_house, которые не находятся в городе ‘Москва’ (условие по полю City).

13. Вывести список названий книг (поле Title_book) из таблицы Books, которые выпущены любыми издательствами, кроме издательства ‘Питер-Софт’ (поле Publish из таблицы Publishing_house).

Выбор записей по диапазону значений (Between).

14. Вывести фамилии, имена, отчества авторов (поле Name_author) из таблицы Authors, у которых дата рождения (поле Birthday) находится в диапазоне 01.01.1840 – 01.06.1860.

15. Вывести список названий книг (поле Title_book из таблицы Books) и количество экземпляров (поле Amount из таблицы Purchases), которые были закуплены в период с 12.03.2003 по 15.06.2003 (условие по полю Date_order из таблицы Purchases).

16. Вывести список названий книг (поле Title_book) и количество страниц (поле Pages) из таблицы Books, у которых объем в страницах укладывается в диапазон 200 – 300 (условие по полю Pages).

17. Вывести список фамилий, имен, отчеств авторов (поле Name_author) из таблицы Authors, у которых фамилия начинается на одну из букв диапазона ‘В’ – ‘Г’ (условие по полю Name_author).

Выбор записей по диапазону значений (In).

18. Вывести список названий книг (поле Title_book из таблицы Books) и количество (поле Amount из таблицы Purchases), которые были поставлены поставщиками с кодами 3, 7, 9, 11 (условие по полю Code_delivery из таблицы Purchases).

19. Вывести список названий книг (поле Title_book) из таблицы Books, которые выпущены следующими издательствами: ‘Питер-Софт’, ‘Альфа’, ‘Наука’ (условие по полю Publish из таблицы Publishing_house).

20. Вывести список названий книг (поле Title_book) из таблицы Books, которые написаны следующими авторами: ‘Толстой Л.Н.’, ‘Достоевский Ф.М.’, ‘Пушкин А.С.’ (условие по полю Name_author из таблицы Authors).

Выбор записей с использованием Like.

21. Вывести список авторов (поле Name_author) из таблицы Authors, которые начинаются на букву ‘К’.

22. Вывести названия издательств (поле Publish) из таблицы Publishing_house, которые содержат в названии сочетание ‘софт’.

23. Выбрать названия компаний (поле Name_company) из таблицы Deliveries, у которых значение оканчивается на ‘ский’.

Выбор записей по нескольким условиям.

24. Выбрать коды поставщиков (поле Code_delivery), даты заказов (поле Date_order) и названия книг (поле Title_book), если количество книг

(поле Amount) в заказе больше 100 или цена (поле Cost) за книгу находится в диапазоне от 200 до 500 руб.

25. Выбрать коды авторов (поле Code_author), имена авторов (поле Name_author), названия соответствующих книг (поле Title_book), если код издательства (поле Code_Publish) находится в диапазоне от 10 до 25 и количество страниц (поле Pages) в книге больше 120.

26. Вывести список издательств (поле Publish) из таблицы Publishing_house, в которых выпущены книги, названия которых (поле Title_book) начинаются со слова ‘Труды’ и город издания (поле City) – ‘Новосибирск’.

Многотабличные запросы (выборка из двух таблиц, выборка из трех таблиц с использованием JOIN).

27. Вывести список названий компаний-поставщиков (поле Name_company) и названия книг (поле Title_book), которые они поставили в период с 01.01.2002 по 31.12.2003 (условие по полю Date_order).

28. Вывести список авторов (поле Name_author), книги которых были выпущены в издательстве ‘Мир’ (условие по полю Publish).

29. Вывести список поставщиков (поле Name_company), которые поставляют книги издательства ‘Питер’ (условие по полю Publish).

30. Вывести список авторов (поле Name_author) и названия книг (поле Title_book), которые были поставлены поставщиком ‘ОАО Книготорг’ (условие по полю Name_company).

Вычисления.

31. Вывести суммарную стоимость партии одноименных книг (использовать поля Amount и Cost) и название книги (поле Title_book) в каждой поставке.

32. Вывести стоимость одной печатной страницы каждой книги (использовать поля Cost и Pages) и названия соответствующих книг (поле Title_book).

33. Вывести количество лет с момента рождения авторов (использовать поле Birthday) и имена соответствующих авторов (поле Name_author).

Вычисление итоговых значений с использованием агрегатных функций.

34. Вывести общую сумму поставок книг (использовать поле Cost), выполненных ‘ЗАО Оптторг’ (условие по полю Name_company).

35. Вывести общее количество всех поставок (использовать любое поле из таблицы Purchases), выполненных в период с 01.01.2003 по 01.02.2003 (условие по полю Date_order).

36. Вывести среднюю стоимость (использовать поле Cost) и среднее количество экземпляров книг (использовать поле Amount) в одной поставке, где автором книги является ‘Акунин’ (условие по полю Name_author).

37. Вывести все сведения о поставке (все поля таблицы Purchases), а также название книги (поле Title_book) с минимальной общей стоимостью (использовать поля Cost и Amount).

38. Вывести все сведения о поставке (все поля таблицы Purchases), а также название книги (поле Title_book) с максимальной общей стоимостью (использовать поля Cost и Amount).

Изменение наименований полей.

39. Вывести название книги (поле Title_book), суммарную стоимость партии одноименных книг (использовать поля Amount и Cost), поместив результат в поле с названием Itogo, в поставках за период с 01.01.2002 по 01.06.2002 (условие по полю Date_order).

40. Вывести стоимость одной печатной страницы каждой книги (использовать поля Cost и Pages), поместив результат в поле с названием One_page, и названия соответствующих книг (поле Title_book).

41. Вывести общую сумму поставок книг (использовать поле Cost) и поместить результат в поле с названием Sum_cost, выполненных ‘ОАО Луч’ (условие по полю Name_company).

Использование переменных в условии.

42. Вывести список сделок (все поля из таблицы Purchases) за последний месяц (условие с использованием поля Date_order).

43. Вывести список авторов (поле Name_author), возраст которых меньше заданного пользователем (условие с использованием поля Birthday).

44. Вывести список книг (поле Title_book), которых закуплено меньше, чем указано в запросе пользователя (условие с использованием поля Amount).

Использование переменных вместо названий таблиц.

45. Вывести список названий компаний-поставщиков (поле Name_company) и названия книг (поле Title_book), которые они поставили.

46. Вывести список авторов (поле Name_author), книги которых были выпущены в издательствах ‘Мир’, ‘Питер Софт’, ‘Наука’ (условие по полю Publish).

47. Вывести список издательств (поле Name_company), книги которых были поставлены по цене 150 руб. (поле Cost).

Выбор результата в курсор.

48. Вывести список названий книг (поле Title_book) и количества страниц (поле Pages) в каждой книге и поместить результат в курсор с названием Temp1.

49. Вывести список названий компаний-поставщиков (поле Name_company) и поместить результат в курсор с названием Temp2.

50. Вывести список авторов (поле Name_author) и поместить результат в курсор с названием Temp3.

Использование функций совместно с подзапросом.

51. Вывести список книг (поле Title_book), у которых количество страниц (поле Pages) больше среднего количества страниц всех книг в таблице.

52. Вывести список авторов (поле Name_author), возраст которых меньше среднего возраста всех авторов в таблице (условие по полю Birthday).

53. Вывести список книг (поле Title_book), у которых количество страниц (поле Pages) равно минимальному количеству страниц книг, представленных в таблице.

Использование квантора существования в запросах.

54. Вывести список издательств (поле Publish), книги которых были приобретены оптом (‘опт’ из поля Type_Purchase).

55. Вывести список авторов (поле Name_author), книг которых нет в таблице Books.

56. Вывести список книг (поле Title_book), которые были поставлены поставщиком ‘ЗАО Квантор’ (условие по полю Name_company).

Оператор обработки данных Update.

57. Изменить в таблице Books содержимое поля Pages на 300, если код автора (поле Code_author) =56 и название книги (поле Title_book) =’Мемуары’.

58. Изменить в таблице Deliveries содержимое поля Address на ‘нет сведений’, если значение поля является пустым.

59. Увеличить в таблице Purchases цену (поле Cost) на 20 процентов, если заказы были оформлены в течение последнего месяца (условие по полю Date_order).

Оператор обработки данных Insert.

60. Добавить в таблицу Purchases новую запись, причем так, чтобы код покупки (поле Code_purchase) был автоматически увеличен на единицу, а в тип закупки (поле Type_purchase) внести значение ‘опт’.

61. Добавить в таблицу Books новую запись, причем вместо ключевого поля поставить код (поле Code_book), автоматически увеличенный на единицу от максимального кода в таблице, вместо названия книги (поле Title_book) написать ‘Наука. Техника. Инновации’.

62. Добавить в таблицу Publish_house новую запись, причем вместо ключевого поля поставить код (поле Code_publish), автоматически увеличенный на единицу от максимального кода в таблице, вместо названия города – ‘Москва’ (поле City), вместо издательства – ‘Наука’ (поле Publish).

Оператор обработки данных Delete.

63. Удалить из таблицы Purchases все записи, у которых количество книг в заказе (поле Amount) = 0.

64. Удалить из таблицы Authors все записи, у которых нет имени автора в поле Name_Author.

65. Удалить из таблицы Deliveries все записи, у которых не указан ИНН (поле INN пустое).

2.3. База данных «Успеваемость студентов»

На рисунке 22 представлен фрагмент упрощенной схемы данных для выполнения индивидуальных заданий. Прежде чем приступить к выполнению заданий необходимо в новой базе данных с названием DB_STUDY создать таблицы со структурой, показанной в таблицах 12–16, установить связи между таблицами, затем заполнить тестовыми данными.

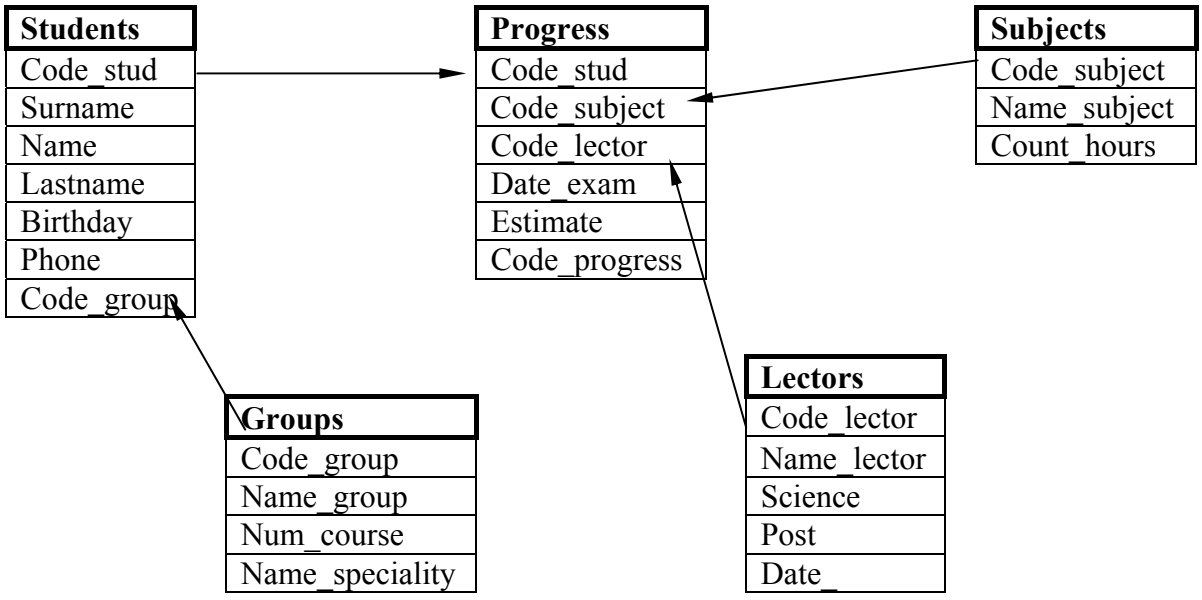

Рис. 22. Фрагмент базы данных «Успеваемость студентов»

Связь между таблицами осуществляется с помощью следующих пар полей с типом связи «один-ко-многим» соответственно:

1. Groups.Code_group - Students.Code_group;
2. Students.Code_stud - Progress.Code_stud;
3. Subjects.Code_subject - Progress.Code_subject;
4. Lectors.Code_lector - Progress.Code_lector.

Таблица 12

Список групп (название таблицы Groups)

Название поля	Тип поля	Описание поля
Code_group	Integer	Код группы
Name_group	Character	Название группы
Num_course	Integer	Номер курса
Name_speciality	Character	Название специальности

Таблица 13

Справочник студентов (название таблицы Students)

Название поля	Тип поля	Описание поля
Code_stud	Character	Номер зачетной книжки
Surname	Character	Фамилия студента
Name	Character	Имя студента
Lastname	Character	Отчество студента
Code_group	Integer	Код группы
Birthday	Date	Дата рождения студента
Phone	Numeric	Контактный телефон студента

Таблица 14

Справочник изучаемых дисциплин (название таблицы Subjects)

Название поля	Тип поля	Описание поля
Code_subject	Integer	Код дисциплины
Name_subject	Character	Название дисциплины
Count_hours	Integer	Количество часов в курсе

Таблица 15

Таблица успеваемости (название таблицы Progress)

Название поля	Тип поля	Описание поля
Code_stud	Character	Номер зачетной книжки
Code_subject	Integer	Код дисциплины
Code_lector	Integer	Код преподавателя
Date_exam	Date	Дата сдачи экзамена
Estimate	Integer	Оценка
Code_progress	Integer	Ключевое поле

Таблица 16

Справочник преподавателей (название таблицы Lectors)

Название поля	Тип поля	Описание поля
Code_lector	Integer	Код преподавателя
Name_lector	Character	Фамилия, имя, отчество преподавателя
Science	Character	Ученая степень
Post	Character	Должность
Date_	Date	Дата приема на работу

2.4. Упражнения с использованием операторов обработки данных для БД «Успеваемость студентов»

Сортировка.

66. Вывести все сведения о студентах из таблицы Students и отсортировать результат по коду студента (поле Code_stud).

67. Вывести список фамилий, имен, отчеств преподавателей (поле Name_lector), их должности (поле Post) и ученые степени (поле Science) из таблицы Lectors, отсортировать результат по названиям должностей по возрастанию (использовать поле Post) и по ученым степеням по убыванию (использовать поле Science).

68. Выбрать из таблицы Groups названия групп и номера курсов (поля Name_group, Name_course) и отсортировать результат по полю Name_course по убыванию.

Изменение порядка следования полей.

69. Вывести все сведения о студентах из таблицы Students таким образом, чтобы в результате порядок столбцов был следующим: Code_group, Name, Surname, Lastname, Phone, Birthday.

70. Выбрать все поля из таблицы Subjects таким образом, чтобы в результате порядок столбцов был следующим: Name_subject, Code_subject.

Выбор некоторых полей из двух таблиц.

71. Вывести список фамилий (поле Surname), имен (поле Name), отчеств (поле Lastname) студентов из таблицы Students и названий групп (поле Name_group) из таблицы Groups, в которых они обучаются.

72. Вывести даты экзаменов (поле Date_exam) из таблицы Progress и фамилии, имена, отчества преподавателей (поле Name_lector) из таблицы Lectors, принимавших в эти даты экзамены.

73. Вывести даты экзаменов (поле Date_exam) из таблицы Progress и названия дисциплин (поле Name_subject) из таблицы Subjects, по которым сдавали экзамены в указанные даты.

Условие неточного совпадения.

74. Вывести список преподавателей (поле Name_lector) из таблицы Lectors с ученой степенью кандидат каких-либо наук, т.е. у которых значение поля Science начинается с ‘к’.

75. Вывести список студентов (поля Surname, Name, Lastname) из таблицы Students и названия групп (поле Name_group) из таблицы Groups, значения которых начинаются с сочетания ‘АС’.

76. Вывести список дисциплин (поле Name_subject) из таблицы Subjects, значение которых начинается с ‘математ’.

Точное несовпадение значений одного из полей.

77. Вывести список преподавателей (поле Name_lector) из таблицы Lectors и их должности (поле Post), которые не являются докторами технических наук, т.е. значение поля Science не равно ‘д.т.н.’.

78. Вывести список групп (поле Name_group) из таблицы Groups, которые не относятся к специальности ‘электротехника’ (условие по полю Name_speciality).

79. Вывести все сведения о всех предметах из таблицы Subjects, кроме предмета ‘высшая математика’ (условие по полю Name_subject).

Выбор записей по диапазону значений (Between).

80. Вывести даты экзаменов (поле Date_exam) из таблицы Progress и список дисциплин (поле Name_subject) из таблицы Subjects, по которым сдавали экзамены в период с 01.01.2003 по 01.02.2003 (условие по полю Date_exam).

81. Вывести список преподавателей (поле Name_lector) из таблицы Lectors и их должности (поле Post), которые были устроены на работу в период с 12.03.2000 по 15.06.2000 (условие по полю Date_).

82. Вывести список студентов (поля Surname, Name, Lastname) и их телефоны (поле Phone) из таблицы Students, если значения телефонов находятся в диапазоне от 220000 до 226666 (условие по полю Phone).

83. Вывести список дисциплин (поле Name_subject) из таблицы Subjects, у которых название начинается на одну из букв диапазона ‘В’–‘Г’ (условие по полю Name_subject).

Выбор записей по диапазону значений (In).

84. Вывести список групп и названия специальностей (поля Name_group и Name_speciality из таблицы Groups), в которых учатся студенты со следующими номерами зачетной книжки ‘АС-12-02’, ‘ПИ-14-03’, ‘АС-21-03’, ‘БИ-12-02’ (условие по полю Code_stud из таблицы Students).

85. Вывести список преподавателей (поле Name_lector) из таблицы Lectors и их должности (поле Post), у которых есть одна из следующих ученых степеней: ‘к.т.н.’, ‘к.э.н.’, ‘д.т.н.’ (условие по полю Science).

86. Вывести список студентов (поля Surname, Name, Lastname) из таблицы Students, которые сдавали экзамены по дисциплинам со следующими кодами: 5, 8, 12, 25 (условие по полю Code_subject).

Выбор записей с использованием Like.

87. Вывести список дисциплин (поле Name_subject) из таблицы Subjects, которые начинаются на букву ‘М’.

88. Вывести список студентов (поля Surname, Name, Lastname) и даты рождения (поле Birthday) из таблицы Students, которые содержат в фамилии сочетание букв ‘нова’ (условие по полю Surname).

89. Выбрать список групп (поле Name_group) из таблицы Groups, у которых значение оканчивается на ‘0’ (ноль).

Выбор записей по нескольким условиям.

90. Вывести всех студентов (поля Surname, Name, Code_group) из таблицы Students, которые сдавали экзамен по дисциплине (поле Name_subject из таблицы Subjects) ‘математический анализ’.

91. Вывести список преподавателей (поле Name_lector) из таблицы Lectors, которые принимали экзамены по дисциплинам с кодами (условие по полю Code_subject из таблицы Progress) 5 – 12 и в период с 01.01.2003 по 01.02.2003 (условие по полю Date_exam из таблицы Progress).

92. Вывести список групп (поле Name_group) и номера курсов (поле Num_course) из таблицы Groups, в которых учатся студенты с датами рождения с 01.01.1976 по 01.01.1978 (условие по полю Birthday из таблицы Students) и кодами из диапазона 10 – 150 (условие по полю Code_stud из таблицы Students).

Многотабличные запросы (выборка из двух таблиц, выборка из трех таблиц с использованием JOIN).

93. Вывести список названий дисциплин (поле Name_subject из таблицы Subjects) и имен преподавателей (поле Name_lector из таблицы Lectors), которые принимали по этим дисциплинам экзамены.

94. Вывести список студентов (поля Surname, Name из таблицы Students) и номер курса (поле Num_course из таблицы Groups), учащихся в группе ‘Ас-31’ (условие по полю Name_group).

95. Вывести список имен преподавателей (поле Name_lector из таблицы Lectors), которые принимали экзамены у студентов с кодами групп 10, 12, 15 (условие по полю Code_group из таблицы Students).

96. Вывести список названий дисциплин (поле Name_subject из таблицы Subjects) и имен преподавателей (поле Name_lector из таблицы Lectors), которые принимали по этим дисциплинам экзамены в период с 15.01.2003 по 16.02.2003 (условие по полю Date_exam из таблицы Progress).

Вычисления.

97. Вывести список всех преподавателей (Name_lector), их ученые степени (поле Science) и срок их работы в годах (использовать поле Date_ из таблицы Lectors).

98. Вывести список всех студентов (поля Surname, Name, Lastname) и их возраст в годах (использовать поле Birthday из таблицы Students).

99. Вывести список всех студентов (поля Surname, Name, Lastname из таблицы Students) и номер курса, на котором они занимаются, а также количество лет оставшейся учебы (использовать поле Num_course из таблицы Groups).

Вычисление итоговых значений с использованием агрегатных функций.

100. Вывести список всех групп (поле Name_group из таблицы Groups) и количество студентов в каждой группе (по любому полю из таблицы Students).

101. Вывести средний балл (использовать поле Estimate из таблицы Progress) по результатам экзаменов каждого студента и имена студентов (поля Surname, Name из таблицы Students) за период сдачи экзаменов с 05.01.2003 по 25.01.2003 (условие по полю Date_exam из таблицы Progress).

102. Вывести фамилии и имена студентов (поля Surname, Name из таблицы Students) с максимальным средним баллом за весь период обучения (условие по полю Estimate из таблицы Progress).

103. Вывести все сведения о преподавателях (все поля таблицы Lectors) с максимальным общим стажем работы (использовать поле Date_).

Изменение наименований полей.

104. Вывести список групп (поле Name_group из таблицы Groups) и количество студентов в каждой группе (по любому полю из таблицы Students), поместив результат в новое поле Count_Students.

105. Вывести средний балл (использовать поле Estimate из таблицы Progress) по результатам экзаменов каждого студента, поместив результат в поле Avg_estimate, и имена студентов (поля Surname, Name из таблицы Students) за период сдачи экзаменов 05.01.2003 по 25.01.2003 (условие по полю Date_exam из таблицы Progress).

106. Вывести список преподавателей (Name_lector), их ученые степени (поле Science) и срок их работы в годах (использовать поле Date_ из таблицы Lectors), поместив резльтат в поле Old_years.

Использование переменных в условии.

107. Вывести список студентов (поля Surname, Name, Lastname) и их телефоны (поле Phone) из таблицы Students, если значения телефонов находятся в диапазоне, хранящемся в переменных Phone_begin и Phone_end.

Например, пусть Phone_begin = 125478, а Phone_end = 352456.

108. Вывести все сведения о студентах и их даты рождения (поле Birthday) из таблицы Students, значения которых находятся в диапазоне, хранящемся в переменных Birthday_begin и Birthday_end.

Например, пусть Birthday_begin = 12.03.1978, а Birthday_end = 12.03.1980.

109. Вывести список студентов (поля Surname, Name, Lastname) и названия их групп (поле Name_group) для значений кодов групп (поле Code_group), находящихся в диапазоне, хранящемся в переменных Group_begin и Group_end.

Например, пусть Group_begin = 12, а Group_end = 35.

Использование переменных вместо названий таблиц.

110. Вывести коды студентов (поле Code_stud) и имена (поля Surname, Name), названия и коды групп (поля Name_group, Code_group из таблицы Groups), причем таблица Students будет использоваться с именем ‘A’, а таблица Groups будет использоваться с именем ‘B’.

111. Вывести имена студентов (поля Surname, Name), названия и коды предметов (поля Name_subject, Code_subject из таблицы Subjects), которые сдавали студенты, а также оценки за предметы (поле Estimate), причем таблица Students будет использоваться с именем ‘A’, таблица Progress будет использоваться с именем ‘B’, а таблица Subjects будет использоваться с именем ‘C’.

112. Вывести имена студентов (поля Surname, Name), названия и коды преподавателей (поля Name_lector, Code_lector из таблицы Lectors), которым сдавали студенты экзамены, а также оценки за предметы (поле Estimate), причем таблица Students будет использоваться с именем ‘A’, таблица Progress будет использоваться с именем ‘B’, а таблица Lectors будет использоваться с именем ‘C’.

Выбор результата в курсор.

113. Вывести все сведения о сданных экзаменах (все поля из таблицы Progress) для студента с фамилией ‘Васькov’ и именем ‘Павел’ (условия по

полям Surname, Name из таблицы Students) и поместить результат в курсор с названием Temp1.

114. Вывести список групп и специальности (поля Name_group, Name_speciality из таблицы Groups), у которых номер курса = 2 (условие по полю Num_course), поместить результат в курсор с названием Temp2.

115. Вывести список всех изучаемых дисциплин (поле Name_subject из таблицы Subjects) и поместить результат в курсор с названием Temp3.

Использование функций совместно с подзапросом.

116. Вывести список преподавателей (все поля из таблицы Lectors), дата устройства которых меньше средней даты устройства всех преподавателей в таблице (условие по полю Date_).

117. Вывести все сведения о сдачах экзаменов (все поля из таблицы Progress) и список студентов (поля Surname, Name из таблицы Students), которые по таблице Progress сдавали экзамены самыми последними (дата сдачи экзаменов Date_exam максимальна).

118. Вывести список групп (поле Name_group), в которых студентов больше 25.

Использование квантора существования в запросах.

119. Вывести список студентов (поля Surname, Name, Lastname из таблицы Students), которые сдали все экзамены без двоек (подзапрос по таблице Progress).

120. Вывести список студентов (поля Surname, Name, Lastname из таблицы Students), которые не сдавали ни одного экзамена (подзапрос по таблице Progress).

121. Вывести список студентов (поля Surname, Name, Lastname из таблицы Students), которые сдали все экзамены на "отлично" (подзапрос по таблице Progress).

Оператор обработки данных Update.

122. Увеличить номер курса (поле Num_course) у всех групп на единицу в таблице Groups, если системная дата равна значению «первое сентября нового учебного года».

123. Изменить в таблице Students содержимое поля Lastname на ‘нет сведений’, если значение поля является пустым.

124. Изменить в таблице Subjects значение поля Name_subject на ‘математический анализ’, если название дисциплины ‘высшая математика’.

Оператор обработки данных Insert.

125. Добавить в таблицу Student новую запись, причем так, чтобы код студента (по полю Code_stud) был автоматически увеличен на единицу, а ФИО = ‘Иванов’ ‘Павел’ ‘Сергеевич’ (поля Surname, Name, LastName).

126. Добавить в таблицу успеваемости Progress новую запись, вместо кода студента (поле Code_stud) поставить 45, вместо кода предмета (поле Code_subject) – 12, вместо кода лектора (поле Code_lector) – 11, вместо даты экзамена (поле Date_exam) –’12.03.2003’.

127. Добавить в таблицу преподавателей Lectors новую запись, причем вместо ключевого поля поставить код (по полю Code_lector), автоматически увеличенный на единицу от максимального кода в таблице, вместо имени (поле Name_lector) – ‘Петров Савелий Яковлевич’, вместо ученой степени (поле Science) – ‘к.т.н.’.

Оператор обработки данных Delete.

128. Удалить из таблицы Students все записи, код группы которых равен 35, или 15, или 19 (условие по полю Code_group).

129. Удалить из таблицы Subjects все записи, в которых в поле Name_subject нет данных или в поле содержится пустое значение.

130. Удалить из таблицы Progress все записи, в которых не указана дата экзамена (поле Date_exam пустое).

3. ВАРИАНТЫ ЗАДАНИЙ

Используя схему и структуру данных, соответствующих вашему варианту, создайте в выбранном СУБД новый проект и реализуйте базу данных.

Для проверки правильности работы команд SQL наберите в таблицах по несколько (5 – 10) записей, соответствующих предметной области и с соблюдением правил целостности базы данных.

Далее по вашему варианту выполните упражнения в созданном проекте.

Вариант	Список упражнений												
1	1	6	11	16	21	26	31	36	41	46	51	56	61
2	2	7	12	17	22	27	32	37	42	47	52	57	62
3	3	8	13	18	23	28	33	38	43	48	53	58	63
4	4	9	14	19	24	29	34	39	44	49	54	59	64
5	5	10	15	20	25	30	35	40	45	50	55	60	65
6	66	71	76	81	86	91	96	101	106	111	116	121	126
7	67	72	77	82	87	92	97	102	107	112	117	122	127
8	68	73	78	83	88	93	98	103	108	113	118	123	128
9	69	74	79	84	89	94	99	104	109	114	119	124	129
10	70	75	80	85	90	95	100	105	110	115	120	125	130

Библиографический список

1. *Астахова И. Ф.* SQL в примерах и задачах/ И. Ф. Астахова, А. П. Толстобров, В.М. Мельников.– М.: Новое знание, 2002.– 176 с.
2. *Грабер М.* SQL. Описание SQL92, SQL99 и SQLJ.–М.: Лори, 2001.– 644 с.
3. *Грабер М.* SQL. Справочное руководство.– М.: Лори, 2001.–354 с.
4. *Грабер М.* Понимание SQL.– М.: Лори, 1993.–420 с.
5. *Грабер М.* Справочное руководство по SQL.– М.: Лори, 1997.– 291 с.
6. *Гроff Дж.* Энциклопедия SQL.– 3-е изд.– СПб: Питер, 2003.–896 с.
7. *Гроff Дж., Вайнберг Пол Н.* SQL: Полное руководство.–Киев: Издательская группа BHV, McGraw–Hill Companies, 2001.–816 с.
8. *Гроff Дж., Вайнберг Пол Н.* SQL: Полное руководство.– Киев: Издательская группа BHV, 1998.– 608 с.
9. *Дворжецкий А.* SQL: Structured Query Language. Руководство пользователя.–М.: Познавательная Книга Плюс, 2001.–416 с.
10. Документация Oracle 10g: PL/SQL Packages and Types Reference.– Электрон. дан.– 2007.–Режим доступа: http://www.oracle.com/pls/db102/to_pdf?pathname=appdev.102%2Fb14258.pdf&remark=portal+%28Books%29.– Загл. с экрана. – Яз. англ.– ©Oracle.
11. Документация Oracle 10g: PL/SQL User's Guide and Reference.– Электрон. дан.– 2007.–Режим доступа: http://www.oracle.com/pls/db102/to_pdf?pathname=appdev.102%2Fb14261.pdf&remark=portal+%28Books%29.– Загл. с экрана. – Яз. англ.– ©Oracle.
12. *Кауффман Дж.* SQL. Программирование/ Джон Кауффман, Брайан Матсик, Кевин Спенсер.– М.: Бином. Лаборатория знаний, 2002.–746 с.
13. *Кириллов В.В.* Структурированный язык запросов (SQL).– СПб.: ИТМО, 1994.– 80 с.
14. *Кузнецов С.* SQL. Язык реляционных баз данных.– М.: Майор, 2001.– 192 с.
15. *Летучий С.* Первые шаги: PL/SQL в Oracle.– Электрон. дан.– 2007.–Режим доступа: http://www.firststeps.ru/sql/oracle/oracle1.html, http://www.firststeps.ru/sql/oracle/oracle2.html, http://www.firststeps.ru/sql/oracle/oracle3.html.– Загл. с экрана. – Яз. рус.– © Летучий С.
16. *Сичкаренко В. А.* SQL–99. Руководство разработчика баз данных.– М.: ДиаСофтЮП, 2002.–816 с.
17. *Стивенс Р.К. , Плю Р.Р.* SQL.– М.: БИНОМ, 1998.– 400 с.
18. *Тейлор Аллен Дж.* SQL для «чайников». – М., 2001.– 368 с.

Учебное издание

Ирина Ивановна Семенова

SQL СТАНДАРТ В СУБД MS SQL SERVER, ORACLE, VFP И ACCESS: МАНИПУЛИРОВАНИЕ ДАННЫМИ

Учебное пособие

* * *
Редактор И.Г. Кузнецова
Компьютерную верстку выполнила автор
* * *

Подписано в печать _____
Формат 60x90 $^{1}/_{16}$. Бумага писчая
Оперативный способ печати
Гарнитура Таймс
Усл.п.л. ___, уч.-изд.л. ___
Тираж 100 экз. Заказ ___
Цена договорная

Издательство СибАДИ
644099, Омск, ул. П.Некрасова, 10

Отпечатано в ПЦ издательства СибАДИ
644099, Омск, ул. П.Некрасова, 10