

**Министерство образования и науки Республики Татарстан
Государственное автономное профессиональное образовательное
учреждение
«Камский строительный колледж имени Е.Н. Батенчука»**

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ
по дисциплине СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ
Основы PostgreSQL**

Специальность подготовки: 09.02.05 Прикладная информатика (по
отраслям)

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Автор: Еремина И.И.

2020 г.

Содержание

Лабораторная работа №1. Ознакомление с основами PostgreSQL.....	3
Цель работы	3
Задание лабораторной работы.....	3
Коды команд	3
Лабораторная работа №2. Ознакомление с числовыми функциями PostgreSQL	7
Цель работы	7
Задание лабораторной работы.....	7
Числовые функции	7
Тригонометрические функции.....	11
Задание лабораторной работы.....	11
Лабораторная работа №3. Работа со строковыми и символьными функциями PostgreSQL	13
Цель работы	13
Задание лабораторной работы.....	13
Строковые и символьные функции	13
Лабораторная работа №4. Функции работы с датой и временем PostgreSQL	18
Цель работы	18
Задание лабораторной работы.....	18
Функции работы с датой и временем	18
Контрольная работа. Ознакомление с основами PostgreSQL	25
Цель работы	25
Задание контрольной работы	25

Лабораторная работа №1. Ознакомление с основами PostgreSQL

по дисциплине

Базы данных

Цель работы

Изучить базовые операции по работе с базой данных. Изучить синтаксис команд. Приобрести навыки создания баз данных, создания, заполнения и модификации таблиц в PostgreSQL.

Задание лабораторной работы

Общая часть: Создать и заполнить базу данных в PostgreSQL. Таблицы (минимум по 5 записей в каждой) связать между собой полями идентификаторов. С помощью команд интерактивного терминала *psql* просмотреть структуру базы данных, структуру таблиц, просмотреть данные в них, изменить структуру таблиц, добавить столбцы, добавить данные, создать столбцы с пользовательскими типами данных.

Вариант 0: Создать и заполнить базу данных для учета работы продуктового магазина, состоящую из трех таблиц. Первая таблица должна содержать поля: наименование товара, количество товара на складе, стоимость покупки (за единицу измерения). Вторая: наименование производителя, адрес. Третья: цена продажи, проданное количество товара. На основании созданных таблиц создать таблицу, содержащую поля: Наименование товара, наименование производителя, стоимость покупки, цена продажи.

Коды команд

1. Создание базы данных:
`create database db2;`

2. Вход в базу данных:

```
\c db2;
```

3. Создание нового типа данных — адрес.

```
create type adress as (ylica char(15), nomer int);
```

```
create type tip_oplati as enum ('nal', 'besnal');
```

4. Создание первой таблицы (наименование товара, количество товара на складе, стоимость покупки (за единицу измерения)):

```
create table t2(id int, naz_tovar char(15), count_sklad int, cost int);
```

5. Заполнение первой таблицы:

```
insert into t2 values (1, 'hleb', 300, 14, 'nal');
```

и т.д.

Просмотр результатов заполнения:

```
postgres=# select * from t2;
 id | naz_tovar | count_sklad | tip_oplati | cost
----+-----+-----+-----+-----
  1 | hleb     |      300   | nal       |   230
  2 | atata    |      233   | nal       |   630
  3 | saf      |      150   | nal       |   870
  4 | kalach   |      480   | besnal    |   900
  5 | koleso   |      560   | besnal    |   100
(5 rows)
```

6. Создание второй таблицы (наименование производителя, адрес):

```
create table t3(id int, naz_creator char(15), adres adress);
```

7. Заполнение второй таблицы:

```
insert into t3 values (1, 'OOO', ('Petuhova', 132));
```

и т.д.

Просмотр результатов заполнения:

```
postgres=# select * from t3;
 id | naz_creator | adres
----+-----+-----
  1 | OOO         | ('Petuhova', 132)
  2 | OAO         | ('Zorge', 122)
  3 | ZAO         | ('Marksa', 12)
  4 | IP          | ('Gromova', 15)
  5 | IP          | ('Hilok', 18)
(5 rows)
```

8. Создание третьей таблицы (цена продажи, проданное количество товара):

```
create table t4(id int, cena_prod int, count_prod int);
```

9. Заполнение третьей таблицы:

```
insert into t4 values (1, 120, 500);
```

и т.д.

Просмотр результатов заполнения:

```
postgres=# select * from t4;
 id | cena_prod | count_prod
-----+-----+-----
  1 |      120 |         500
  2 |      220 |         500
  3 |       10 |         350
  4 |      180 |          67
  5 |       90 |         190
(5 rows)
```

10. Создание новой таблицы на основании созданных ранее таблиц.

Новая таблица содержит поля: Наименование товара, наименование производителя, стоимость покупки, цена продажи.

```
create table t_out as select
t2.naz_tovar,
t3.naz_creator,
t2.cost,
t4.cena_prod
from t2
inner join t3 on t2.id=t3.id
inner join t4 on t3.id=t4.id
```

```
postgres=# create table t_out as select t2.naz_tovar, t3.naz_creator, t2.cost, t
4.cena_prod from t2 inner join t3 on t2.id=t3.id inner join t4 on t3.id=t4.id;
SELECT
postgres=# select * from t_out;
ERROR:  syntax error at or near "selcet"
СИРОКА 1:selcet * from t_out;
      ^
postgres=# select * from t_out;
 naz_tovar | naz_creator | cost | cena_prod
-----+-----+-----+-----
 hleb     | 000        | 14  | 120
 bulka    | 0A0        | 24  | 220
 pechenka | ZA0        | 15  | 10
 kalach   | IP         | 65  | 180
 rulka    | IP         | 95  | 90
(5 rows)
```

11. Изменение структуры таблицы (добавление нового столбца):

```
postgres=# alter table t2 add new_col int;
ALTER TABLE
postgres=# select * from t2;
 id | naz_tovar | count_sklad | cost | new_col
-----+-----+-----+-----+-----
  1 | hleb     | 300         | 14  |
  2 | bulka    | 500         | 24  |
  3 | pechenka | 150         | 15  |
  5 | rulka    | 560         | 95  |
  4 | kalach   | 480         | 65  |
(5 rows)
```

12. Добавление, изменение и удаление записи в таблице:

```

postgres=# update t2 set naz_tovar='koleso' where id=5;
UPDATE 1
postgres=# select * from t2;
 id | naz_tovar | count_sklad | cost | new_col
----+-----+-----+-----+-----
  1 | hleb     |          300 |   14 |
  2 | bulka    |          500 |   24 |
  3 | pechenka|          150 |   15 |
  4 | kalach   |          480 |   65 |
  5 | koleso   |          560 |   95 |
(5 rows)

postgres=# delete from t2 where cost=24;
DELETE 1
postgres=# select * from t2;
 id | naz_tovar | count_sklad | cost | new_col
----+-----+-----+-----+-----
  1 | hleb     |          300 |   14 |
  3 | pechenka|          150 |   15 |
  4 | kalach   |          480 |   65 |
  5 | koleso   |          560 |   95 |
(4 rows)

postgres=# insert into t2 values (2, 'chto-to', 400, 15, 99);
INSERT 0 1
postgres=# select * from t2;
 id | naz_tovar | count_sklad | cost | new_col
----+-----+-----+-----+-----
  1 | hleb     |          300 |   14 |
  3 | pechenka|          150 |   15 |
  4 | kalach   |          480 |   65 |
  5 | koleso   |          560 |   95 |
  2 | chto-to  |          400 |   15 |        99
(5 rows)

```

Лабораторная работа №2. Ознакомление с числовыми функциями PostgreSQL

по дисциплине
Базы данных

Цель работы

Изучить базовые операции по числовыми функциями. Изучить синтаксис команд. Приобрести навыки работы с числовыми выражениями в PostgreSQL.

Задание лабораторной работы

Общая часть: Выполнить вычисления для выражений в PostgreSQL по примерам. Выполнение каждого примера показать в скриншоте экрана и ниже записать вводимое выражение. С помощью команд записи числовых функций *psql* получить значение выражения F(X), результат получить в виде таблицы

X	F(X)

Вычислить выражение Y, разработать структуру таблицы по порядку выполнения действий, добавить столбцы для заданных параметров с пользовательскими данными.

Например:

$$Y = \frac{\operatorname{tg}\left(\frac{1}{x}\right) - \lg(1 - z^2)}{\sqrt{(a + b)^3}}$$

x	1	2	3	4	5	6	7	$8Y = \frac{\operatorname{tg}\left(\frac{1}{x}\right) - \lg(1 - z^2)}{\sqrt{(a + b)^3}}$

Числовые функции

В PostgreSQL предусмотрен ряд встроенных функций для работы с числами.

Функция ABS

Функция ABS(n) возвращает абсолютное значение числа n. Например:
SELECT ABS(100) x1, ABS(-100) x2, ABS(-100.2) x3;

X1	X2	X3
----	----	----

100	100	100.2
-----	-----	-------

Функция CEIL

Функция CEIL(n) возвращает наименьшее целое, большее или равное переданному в качестве параметра числу n.

Например:

```
SELECT CEIL(100) x1, CEIL(-100) x2, CEIL(100.2) x3, CEIL(-100.2) x4;
```

X1	X2	X3	X4
100	-100	101	100

Функция FLOOR

Функция FLOOR(n) возвращает наибольшее целое, меньшее или равное переданному в качестве параметра числу n.

Например:

```
SELECT FLOOR(100.22) x1, FLOOR(-100.22) x2,
       FLOOR(100.99) x3, FLOOR(100.01) x4;
```

X1	X2	X3	X4
100	-101	100	100

Функция TRUNC

Функция TRUNC(n[, m]) возвращает число n, усеченное до m знаков после десятичной точки. Параметр m может не указываться – в этом случае n усекается до целого.

```
SELECT TRUNC(100.25678) x1, TRUNC(-100.25678) x2,
       TRUNC(100.99) x3, TRUNC(100.25678, 2) x4;
```

X1	X2	X3	X4
100	-100	100	100.25

Функция *ROUND*

Функция *ROUND*(n[, m]) возвращает число n, округленное до m знаков после десятичной точки по правилам математического округления. Параметр m может не указываться – в этом случае округляется до целого.

```
SELECT ROUND(100.25678) x1, ROUND(100.5) x2, ROUND(100.99)
x3, ROUND(100.25678, 2) x4;
```

X1	X2	X3	X4
100	101	101	100.26

Функция *SIGN*

Функция *SIGN*(n) определяет знак числа. Если n положительное, то функция возвращает 1. Если отрицательное – возвращается -1. Если равно нулю, то возвращается 0.

Например:

```
SELECT SIGN(100.22) x1, SIGN(-100.22) x2, SIGN(0) x3;
```

X1	X2	X3
1	-1	0

Функция *MOD*

Функция *MOD*(n,m) возвращает остаток от деления n на m.

Например:

```
SELECT MOD(10, 3) x1, MOD(10, 2) x2, MOD(100, 98) x3;
```

X1	X2	X3
1	0	2

Функция *POWER*

Функция *POWER*(n,m) возводит число n в степень m. Степень может быть дробной и отрицательной, что существенно расширяет возможности данной функции.

```
SELECT POWER(10, 2) x1, POWER(100, 0.5) x2,
POWER(1000, 0.33333333) x3, POWER(1000, -0.33333333) x4;
```

X1	X2	X3	X4
100	10	9.999...934	0.100...851

В некоторых случаях при вызове данной функции может возникнуть исключительная ситуация.

Например:

```
SELECT POWER(-100, 0.5) x2;
```

В данном случае производится попытка вычисления квадратного корня от отрицательного числа, что приведет к возникновению ошибки «Отрицательное число в дробной степени дает комплексный результат».

Функция SQRT

Функция SQRT(n) возвращает квадратный корень от числа n.

Например:

```
SELECT SQRT(100) x;
```

X

10

Функции EXP и LN

Функция EXP(n) возводит e в степень n, а функция LN(n) вычисляет натуральный логарифм от n (при этом значение n должно быть больше нуля).

Пример:

```
SELECT EXP(1) x1, LN(1) x2, LN(EXP(2)) x3;
```

X1	X2	X3
2,71828182845905	0	2

Попытка передать функции LN отрицательное значение приводит к возникновению ошибки «Вычислить логарифм отрицательного числа нельзя».

Функция LOG

Функция LOG(n, m) производит вычисление логарифма m по основанию n.

Пример:

```
SELECT LOG(2, 8) x1, LOG(10, 100) x2;
```

X1	X2
3	2

Тригонометрические функции

PostgreSQL поддерживает вычисление основных тригонометрических функций:

SIN(n) – синус n (где n – угол в радианах);

COS(n) – косинус n (где n – угол в радианах);

TAN(n) – тангенс n (где n – угол в радианах);

COT(n) – котангенс n (где n – угол в радианах). Пример:

```
SELECT SIN(0) x1, COS(0) x2, TAN(0) x3, COT(0);
```

X1	X2	X3	X4
0	1	0	Infinity

Задание лабораторной работы

№ варианта	X	F(X)	Y
1.	-5	ABS(X-1)+0,5	$\tau = \lambda_0 \frac{\cos(\pi \lambda_0^3)}{3,2 \cdot 10^3 - \lambda_1^3 \ln \lambda_1 - \lambda_2 } - \sin(\lambda_0^2)$
2.	2	X^2-3*X+5,3	$\gamma = \alpha_1 + \frac{b \cdot x}{6,9 - 2,3b \cdot x^3} + \lg^3(3k_0) - x^4 \cos(k_0^2)$
3.	3	X^3-2*X+0,85	$y = \frac{1}{x^2 + \alpha_1^2} \sqrt{x^3(\beta - \alpha_2) - \cos^3(l + 2)}$
4.	-4	ABS(X-1)+0,5	$\gamma = \sqrt{1,7 \cdot 10^{-2} + 2 \ln[\sin^2 k + (x^2 - 1,8)^3]} + \beta_0$
5.	10	10/(X-5)+1,5	$\psi = k_1 \left(\frac{k_2}{E} + \frac{a}{P_{\max}} \right) (P_{\max} - k_2) \cdot e^{-k_2 P}$
6.	5	1/ABS(12-X)	$e = \frac{\ln^3 \alpha_2 + 1,07 \cdot 10^{-7} + \beta x}{\sqrt{\alpha_1} \cos(4n)}$
7.	0	-5*X+10,25	$t = \frac{v \cdot (\beta - 1)}{k \cdot \mu \cdot f \cdot v_{kp} \cdot h} \left(\ln \frac{P_n}{P_0} + (\beta - 1) \cdot \ln \frac{P_n - P_1}{P_0 - P_1} \right)$
8.	2	(X^2+1)/X+3	$\omega_{11} = \lambda \cdot \omega \cdot \frac{\cos \varphi}{\sqrt{1 - \lambda^2 (\sin \varphi - k)^2}} + \frac{2,7 \cdot 10^3}{\lambda^2 \cdot \omega}$
9.	-4	ABS(X-1)+0,5	$\varphi = \left[x \left(\frac{x + 2,7 \cdot 10^{-3} k}{x} - 4,5 \right)^3 + \cos(k^3) \right] \cdot \sin^4 x$
10.	-10	5*X^2-3*X+1	$t = 2\pi \cdot \sqrt{\frac{m}{E \cdot f \cdot g} \cdot \frac{ab}{a+b} \left(1 + \frac{\gamma F}{3} \cdot \frac{a+b}{m} \right)}$

11.	6	$-2 \cdot X^2 - X + 1$	$\omega_2 = \frac{J_m \omega_1 \cdot (\beta_1 - 1)}{J_m (\beta - 1) + \beta \frac{J_a}{i_k^2}} + \sqrt[3]{J_a \omega_1^2}$	
12.	-3	$(X+1)^2 + 0,5$	$\omega_{\text{кон}} = r \omega (\sin \varphi_1 + \frac{\lambda}{2} \sin(2\varphi + 4,9 \cdot 10^{-7}) - \lambda^k \cos \varphi_1)$	
13.	1	$\text{ABS}(1-X) + 1/X$	$\omega_{\text{У.}} = \left[\omega_k - \frac{k_3 M_{-p} t (1 - \frac{k_2}{\beta})}{2 J_m} \right] \cdot \frac{J_m (\beta - k_3)}{J_m (\beta - k_3) + J_m (\beta - k_2)}$	
14.	5	$2 \cdot (X-5)^{0,5} - 3$	$\theta = \lambda_0 \frac{l_1}{\lambda_1^2 + l_2^3 + \varphi \cos(\pi \lambda_0^2)} - 6,5 \cdot 10^2 l_2$	
15.	15	$3 \cdot \text{COS}(X/2)$	$y = 8,36 \cdot 10^2 - (\frac{k_1}{k_0 - m} - 1) \cdot \cos^3 k_1 + (\frac{\alpha}{\alpha + \beta} + 2,306 \cdot \alpha^2)^3$	
16.	0	$-0,5 \cdot \text{SIN}(2 \cdot X) + 1$	$\omega_1 = \sqrt{\frac{m \omega^2 \rho}{F} + \frac{m \rho \cdot (g + 2 \omega^2 \rho)}{\omega}} \cdot \alpha_\beta + \lg(m \omega^2 \rho)$	
17.	-3,2	$3 \cdot \text{COS}(X/2) - 1$	$\beta = \frac{(\text{tg}(x^3) + \cos \sqrt{x})^{2x}}{e^x + \lg 2,4x^3 } + 2,385 \cdot 10^5 \cos \sqrt{x}$	
18.	-6,5	$0,5 \cdot \text{COS}(X/2)$	$d_{\text{кон.}} = 2 \sqrt{\left(\frac{335}{\sigma_n} \right)^2 \frac{M_2 k_2 U_5}{(1 - 0,5 \psi_{\text{вкл}})^2 \psi_{\text{вкл}}} + \sqrt[3]{(\psi_{\text{вкл}} + \sigma_n)^2}}$	
19.	-3,1	$-2 \cdot \text{COS}(X/3)$	$\varphi = k_1^5 \sqrt[5]{\left(\frac{k_2}{e^{k_3 - 6,4} + k_1} \right)^3} - \cos^2(2k_1)$	
20.	3,1	$1 - \text{COS}(X/2)$	$Y = \frac{x \cdot \sin\left(\frac{\pi}{4}\right)}{1 - 2x \cdot \cos\left(\frac{\pi}{4}\right) + x^2} - \ln \left 2 \sin \frac{x}{2} \right $	

Лабораторная работа №3. Работа со строковыми и символьными функциями PostgreSQL

по дисциплине
Базы данных

Цель работы

Изучить базовые операции со строковыми и символьными функциями. Изучить синтаксис команд. Приобрести навыки работы с числовыми выражениями в PostgreSQL.

Задание лабораторной работы

Общая часть: Выполнить вычисления для выражений в PostgreSQL по примерам. Выполнение каждого примера показать в скриншоте экрана и ниже записать вводимое выражение. С помощью команд записи строковых и символьных функций *psql* получить значение выражения.

Строковые и символьные функции

В этой части речь пойдет о функциях работы с текстовой информацией, которые могут применяться в запросах и программном коде на языке PL/pgSQL.

Функция *CONCAT*

Функция *CONCAT(str1, str2)* выполняет конкатенацию строк *str1* и *str2*. Если один из аргументов равен *NULL*, то он воспринимается как пустая строка. Если оба аргумента равны *NULL*, то функция возвращает *NULL*.

Пример:

```
SELECT CONCATY('попа ', 'была' 'собака') x1,  
CONCAT('Test', NULL) x2,  
CONCAT(NULL, 'Test') x3,  
CONCAT(NULL, NULL) x4;
```

X1	X2	X3	X4
У попа была собака	Test	Test	NULL

Для конкатенации строк PostgreSQL поддерживает специальный оператор конкатенации `||`, который работает аналогично функции *CONCAT*, например:

```
SELECT CONCATY(' попа ', была' собака') X1,У 'попа' || была'  
собака' X2;
```

X1	X2
У попа была собака	У попа была собака

Функция LOWER

Функция LOWER(str) преобразует все символы строки str в строчные.

Пример:

```
SELECT LOWER('text DATA') X;
```

```
X  
text data
```

Функция UPPER

Функция UPPER(str) преобразует все символы строки str в прописные.

Пример:

```
SELECT UPPER('text DATA') X;
```

```
X  
TEXT DATA
```

Функция INITCAP

Функция INITCAP(str) возвращает строку str, в которой первые буквы всех слов преобразованы в прописные. Функция удобна для форматирования полного имени при построении отчетов.

Пример:

```
SELECT INITCARIВАНОВ(' петр сидорович') X;
```

```
X  
Иванов Петр Сидорович
```

Функции LTRIM и RTRIM

Функция LTRIM(str [,set]) удаляет все символы с начала строки до первого символа, которого нет в наборе символов set. По умолчанию set состоит из одного пробела и может не указываться. Функция RTRIM(str [,set]) аналогична

LTRIM, но удаляет символы, начиная от конца строки. Рассмотрим несколько примеров:

```
SELECT LTRIM('Text DATA') X1,
LTRIM(' _ # Text DATA', ' #_') X2,
LTRIM(' 1234567890 Text DATA', ' 1234567890') X3
UNION ALL
SELECT RTRIM('Text DATA') X1,
RTRIM('Text DATA _ # ', ' #_') X2,
RTRIM('Text DATA 1234567890 ', ' 1234567890') X3;
```

```
X1    X2    X3
Text DATA Text DATA Text DATA
Text DATA Text DATA Text DATA
```

Функция REPLACE

Функция REPLACE(str, search_str, replace_str) осуществляет поиск образца search_str в строке str и каждое найденное вхождение заменяет на replace_str. Поиск подстроки ведется с учетом регистра.

Пример:

```
SELECT REPLACE(' попа была собака', 'собака', 'кошка') X1,
REPLACE(' попа была злая собака', 'злая', '') X2,
REPLACE(' попа была собака', 'Собака', 'Кошка') X3;
```

X1	X2	X3
У попа была кошка	У попа была собака	У попа была собака

Функция TRANSLATE

Функция TRANSLATE(str, from_mask, to_mask) анализирует строку str и заменяет в ней все символы, встречающиеся в строке from_mask, на соответствующие символы из to_mask. Для корректной работы функции строки from_mask и to_mask должны иметь одинаковую длину или строка from_mask должна быть длиннее, чем to_mask. Если from_mask длиннее, чем to_mask, и в процессе обработки строки str обнаружатся символы, соответствующие одному из символов from_mask, и при этом им не найдется соответствия в to_mask, то такие символы будут удалены из строки str. Если передать from_mask или to_mask значение, равное NULL, то функция возвратит значение NULL. Сравнение производится с учетом регистра.

Примеры:

```
SELECT TRANSLATE('Test 12345', 'e2', 'E!') X1,
TRANSLATE('Test 12345', 'e234', 'E') X2;
```

X1	X2
TEst 1!345	TEst 15

Данная функция удобна для решения ряда практических задач, связанных с перекодировкой символов или с поиском запрещенных символов. Например, необходимо проанализировать пароль и выяснить, содержит ли он хотя бы одну цифру. Реализация данной проверки при помощи TRANSLATE имеет вид:

```
CREATE OR REPLACE FUNCTION is_password_correct(
    password IN char)
    RETURNS int AS $is_password_correct$
BEGIN
    IF TRANSLATE(password, '0123456789', '*') = password THEN
        RAISE WARNING
            'Пароль' должен содержать хотя бы одну цифру!';
        RETURN 0;
    END IF;

    RAISE INFO 'Корректный' пароль!';
    RETURN 1;
END;
$is_password_correct$ LANGUAGE plpgsql;
```

Другой пример: идет подготовка числа к его преобразованию в NUMBER. Необходимо заменить разделители десятичных знаков ',' на '.' и удалить пробелы. Реализация данной операции при помощи TRANSLATE имеет вид:

```
SELECT TRANSLATE('123 455,23', ',', '.', '') x1,
TRANSLATE('-123 455.23', '..', ', ', '..') x2;
```

X1	X2
123455.23	-123455.23

Функция SUBSTR

Функция SUBSTR(str, m [,n]) возвращает фрагмент строки str, начиная с символа m длиной n символов. Длину можно не указывать – в этом случае возвращается строка от символа m и до конца строки str. Нумерация символов идет с 1. Если указать m равное 0, то копирование все равно начнется с первого символа. Задание отрицательного значения m приводит к тому, что символы отсчитываются от конца строки, а не от начала. Задание значений m, превышающих по абсолютному значению длину строки, приводит к тому, что функция возвращает NULL.

Пример:

```
SELECT SUBSTR('попа была собака', 13) x1,
SUBSTR('попа была собака', -6) x2,
SUBSTR('тестовый текст', 5, 8) x3,
SUBSTR('попа была собака', 150) x4;
```

X1	X2	X3	X4
----	----	----	----

собака	собака	ТЕКСТОВЫЙ	NULL
--------	--------	-----------	------

Функция *LENGTH*

Функция *LENGTH*(str) возвращает длину строки str в символах. Для пустой строки функция вернет 0, а для значения NULL – NULL. Пример:
 SELECT LENGTH(' по па была собака') x1,
 LENGTH('') x2,
 LENGTH(NULL) x3;

X1	X2	X3
18	0	NULL

Функция *ASCII*

Функция *ASCII*(str) возвращает ASCII-код первого символа строки str в случае применения кодировок ASCII и UTF-8.

Пример:
 SELECT ASCII('test') x1,
 ASCIIТест('') x2;

X1	X2
84	1058

Функция *CHR*

Функция *CHR*(n) возвращает символ по его коду. Пример:
 SELECT
 CHR(84) x1,
 CHR(1058) x2,
 CHR(80) x3,
 CHR(1060) x4;

X1	X2	X3	X4
T	T	P	Ф

Лабораторная работа №4. Функции работы с датой и временем PostgreSQL

по дисциплине
Базы данных

Цель работы

Изучить базовые операции работы с датой и временем. Изучить синтаксис команд. Приобрести навыки работы с с датой и временем в PostgreSQL.

Задание лабораторной работы

На компьютере проделать примеры, получить результат. Вставить результат в качестве скриншота в отчет по лабораторной работе.

Функции работы с датой и временем

В данном подразделе речь пойдет о функциях работы с датой и временем, а также функциями преобразования типов для даты. Для хранения даты и времени в PostgreSQL предусмотрен специальный тип `timestamp`. С физической точки зрения это дробное число, целая часть которого хранит количество дней с некоторой базовой даты, а дробная – время. Это позволяет совершать над датами арифметические операции – сложение и вычитание.

Функция *NOW*

Это одна из самых часто употребляемых функций, она возвращает текущую дату и время по часам сервера.

Пример:

```
SELECT NOW();
```

NOW

2016-09-05 21:31:10.095806+04

Функция *JUSTIFY_INTERVAL*

Функция `JUSTIFY_INTERVAL(interval)` преобразует интервал (тип `interval`), указанный в виде строки в соответствующее значение типа `timestamp`.

Пример:

```
SELECT NOW() D1,  
       NOW() + JUSTIFY_INTERVAL('30 DAYS 1 HOUR 2 MINUTE')  
       D2, NOW() - JUSTIFY_INTERVAL('30 DAYS 1 HOUR 2  
       MINUTE') D3;
```

D1	D2	D3
2016-09-05 21:39:08...+04	2016-10-05 22:41:08...+04	2016-08-05 20:37:08...+04

Функция *DATE_TRUNC*

Функция *DATE_TRUNC(timestamp)* используется для обрезки даты или интервала (*DATE_TRUNC(interval)*) до определенной точности.

Пример:

```
SELECT  
DATE_TRUNC('HOUR', NOW()) D1,  
DATE_TRUNC('DAY', NOW()) D2,  
DATE_TRUNC('MONTH', NOW()) D3;
```

D1	D2	D3
2016-09-05 22:00:00+04	2016-09-05 00:00:00+04	2016-09-01 00:00:00+04

Получение начала и конца месяца

Для получения дат соответствующих началу и концу месяца необходимо использовать функции *DATE_TRUNC* и *JUSTIFY_INTERVAL*.

Пример:

```
SELECT  
DATE_TRUNC('MONTH', NOW()) D1,  
DATE_TRUNC('MONTH', NOW())  
+ JUSTIFY_INTERVAL('1 MONTH - 1 DAY') D2;
```

D1	D2
2016-09-01 00:00:00+04	2016-09-30 00:00:00+04

Данные функции также могут быть использованы для определения количества дней в заданном месяце.

Например:

```
SELECT NOW() D1,  
TO_CHAR(DATE_TRUNC('MONTH', NOW())  
+ JUSTIFY_INTERVAL('1 MONTH - 1 DAY'), 'DD') D2;
```

D1	D2
2016-09-05 21:51:19.727702+04	30

Функция *AGE*

Функция *AGE([end_date,]start_date)* возвращает разницу между датами, обозначенными как *end_date* и *start_date*. Если параметр *end_date* опущен, то используется значение глобальной переменной *CURRENT_DATE*, которая содержит текущую дату (тип *date*, дата без времени).

Пример:

```
SELECT CURRENT_DATE D1,
```

```
AGE(MAKE_TIMESTAMP(2013, 7, 15, 8, 15, 23.5))
D2, AGE(MAKE_DATE(2016, 3, 3),
MAKE_TIMESTAMP(2013, 7, 15, 8, 15, 23.5)) D3;
```

D1	D2	D3
2016-09-05	3 years 1 mon 20 days 15:44:36.5	2 years 7 mons 18 days 15:44:36.5

В данном примере также используются функции MAKE_TIMESTAMP и MAKE_DATE, которые возвращают значения типов timestamp и date соответственно.

Функция EXTRACT

Функция EXTRACT(field FROM timestamp) извлекает элемент даты field из значения типа timestamp. Также существует функция EXTRACT(field FROM interval) для работы со значениями типа interval.

Пример:

```
SELECT
NOW()
D1,
EXTRACT(MONTH FROM NOW()) D2,
EXTRACT(YEAR FROM NOW()) D3,
EXTRACT(MINUTE FROM NOW()) D4;
```

D1	D2	D3	D4
2016-09-05 22:37:54.6046+04	9	2016	37

Количество месяцев между двумя датами

Функции EXTRACT, AGE и DATE_TRUNC могут быть использованы для нахождения количества месяцев между двумя датами.

Пример:

```
SELECT
NOW()
D1,
EXTRACT(MONTH FROM AGE(DATE_TRUNC('MONTH', NOW()),
DATE_TRUNC('MONTH', MAKE_DATE(2016, 3, 3)))) D2;
```

D1	D2
2016-09-05 22:41:08.68334+04	6

Функция TO_DATE

Функция TO_DATE(str, mask) преобразует строку str в дату. Преобразование ведется по маске mask.

Пример:
 SELECT
 TO_DATE('05 Dec 2000', 'DD Mon YYYY') D1,
 TO_DATE('15.12.2000', 'dd.mm.yy') D2;

D1	D2
2000-12-05	2000-12-05

Функция TO_CHAR

Функция TO_CHAR(date, mask) преобразует дату date в символную строку в соответствии с заданной маской.

Пример:
 SELECT NOW() D1,
 TO_CHAR(NOW(), 'DD.MM.YY HH24:MI') D2;

Например:

D1	D2
2016-09-05 22:49:37.044199+04	05.09.16 22:49

Некоторые форматные маски представлены в таблице 5.1.

Маски для форматирования даты и времени

Маска	Назначение
CC	Номер века
YYYY, или YY	Четыре или последние две цифры номера года
Q	Номер квартала
MONTH, или MON, или MM	Строковое или числовое представление номера месяца
WW	Номер недели в году
W	Номер недели в месяце
DDD	Номер дня в году
DD	Номер дня в месяце
DAY, или DY, или ID	Номер дня недели
HH, или HH12, или HH24	Час
MI	Минута

Как получить текущую дату в SQL

1 вариант:

```
1 WHERE date = CURDATE()
```

2 вариант:

```
1 WHERE date = STR_TO_DATE(now(), '%Y-%m-%d')
```

Прибавить к дате один час в SQL

```
1 DATE_ADD('2013-03-30', INTERVAL 1 HOUR)
```

Прибавить к дате один день в SQL

```
1 DATE_ADD('2013-03-30', INTERVAL 1 DAY)
```

Аналогично можно прибавлять любое количество дней к текущей дате.

Прибавить к дате один месяц в SQL

```
1 DATE_ADD('2013-03-30', INTERVAL 1 MONTH)
```

Аналогично можно прибавлять любое количество месяцев к текущей дате.

Получить вчерашний день в SQL

Первый вариант:

```
1 DATE_ADD(CURDATE(), INTERVAL -1 DAY)
```

Второй вариант:

```
1 DATE_SUB(CURDATE(), INTERVAL 1 DAY)
```

Получить дату начала текущей недели в SQL

Вот эта одна из самых сложных на первый взгляд задач, но решается очень просто:

```
1 CURDATE() - WEEKDAY(CURDATE());
```

Получить выборку с этого понедельника по текущий день недели в SQL

```
1 WHERE (  
2 date BETWEEN  
3 (CURDATE() - WEEKDAY(CURDATE()))  
4 AND  
5 CURDATE())
```

6)

Получить выборку с первого числа текущего месяца по текущий день недели в SQL

```
1 WHERE (  
2 date BETWEEN  
3 (CURDATE()-WEEKDAY(CURDATE()))  
4 AND  
5 CURDATE())
```

Как получить дату рождения пользователя в SQL

```
1 SELECT name, birth, CURRENT_DATE,  
2 (YEAR(CURRENT_DATE)-YEAR(birth))  
3 - (RIGHT(CURRENT_DATE,5)<RIGHT(birth,5))  
4 AS age  
5 FROM user;
```

Найти всех пользователей у которых день рождение в следующем месяце в SQL

```
1 SELECT name, birth FROM user  
2 WHERE MONTH(birth) = MONTH(DATE_ADD(NOW(), INTERVAL 1 MONTH));
```

Или второй вариант:

```
1 SELECT name, birth FROM pet  
2 WHERE MONTH(birth) = MOD(MONTH(NOW()), 12) + 1;
```

Кроме вышеприведенных кейсов по работе с датами в SQL, рекомендую ознакомиться с документацией по следующим операторам :

NOW() — Возвращает текущую дату и время.

CURDATE() — Возвращает текущую дату.

CURTIME() — Возвращаем текущее время.

DATE() — Состоит из двух частей даты и времени.

EXTRACT() — Возвращает одно значения даты/времени.

DATE_ADD() — Добавляет до выборки указанное число дней/мину/часов и т.д.

DATE_SUB() — Вычитываем указанный интервал от даты.

DATEDIFF() — Возвращает значение времени между двумя датами.

DATE_FORMAT() — Функция для различного вывода информации о времени.

Работа с датами в SQL, как оказывается не такая сложная, и теперь вместо того чтобы вычислять периоды средствами PHP можно делать это еще на этапе выполнения SQL запроса и получать необходимую выборку данных.

Контрольные вопросы

Перечислите основные функции PostgreSQL для работы с числами.

Перечислите основные тригонометрические функции PostgreSQL.

Перечислите основные функции PostgreSQL для работы со строками.

Перечислите основные функции PostgreSQL для работы с датой и временем.

1. Дана строка 'minsk'. Сделайте из нее строку 'MINSK'
2. Дана строка 'MINSK'. Сделайте из нее строку 'Minsk'.
3. В переменной \$date лежит дата в формате '31-12-2030'. Преобразуйте эту дату в формат '2030.12.31'.
4. Дана строка 'php'. Сделайте из нее строку 'PHP'.
5. Дана строка 'PHP'. Сделайте из нее строку 'php'.
6. Дана строка 'london'. Сделайте из нее строку 'London'.
7. Дана строка 'London'. Сделайте из нее строку 'london'.
8. Дана строка 'london is the capital of great britain'. Сделайте из нее строку 'London Is The Capital Of Great Britain'.
9. Дана строка 'LONDON'. Сделайте из нее строку 'London'.
10. Дана строка 'html css php'. Вырежьте из нее и выведите на экран слово 'html', слово 'css' и слово 'php'.
11. Дана строка. Вырежьте и выведите на экран последние 3 символа этой строки.
12. Дана строка. Проверьте, что она начинается на 'http://'. Если это так, выведите 'да', если не так - 'нет'.
13. Дана строка. Проверьте, что она начинается на 'http://' или на 'https://'. Если это так, выведите 'да', если не так - 'нет'.
14. Дана строка. Проверьте, что она заканчивается на '.png'. Если это так, выведите 'да', если не так - 'нет'.
15. Дана строка. Проверьте, что она заканчивается на '.png' или на '.jpg'. Если это так, выведите 'да', если не так - 'нет'.

Контрольная работа. Ознакомление с основами PostgreSQL

по дисциплине

Базы данных

Цель работы

Проанализировать исходную информацию, сформировать по исходным данным Варианта работы таблицы для работы с базой данных. Создать соответствующие запросы, используя синтаксис команд. Закрепить навыки создания баз данных, создания, заполнения и модификации таблиц в PostgreSQL.

Задание контрольной работы

Общая часть: Создать и заполнить базу данных в PostgreSQL. Таблицы (минимум по 10 записей в каждой) связать между собой полями идентификаторов. С помощью команд интерактивного терминала *psql* просмотреть структуру базы данных, структуру таблиц, просмотреть данные в них, изменить структуру таблиц, добавить столбцы, добавить данные, создать столбцы с пользовательскими типами данных.

Вариант 1

Исходная информация содержится в двух таблицах:

1. «Багаж» (№ п/п; номер рейса (ключевое поле); допустимый вес багажа; стоимость 1 кг перевозки багажа) - 7 строк.

2. «Регистрация» (№ п/п; ФИО пассажира (7 разных значений); номер рейса; стоимость 1 кг перевозки багажа; общий вес багажа (10 разных значений); сумма оплаты за багаж - 15 строк.

Получить список пассажиров указанного рейса с указанием веса багажа и размера оплаты за багаж.

Вариант 2

Исходная информация содержится в двух таблицах:

1. «Прейскурант цен телевизионных передач» (№ п/п; код передачи (ключевое поле); название передачи; цена минуты эфира) - 7 строк.

2. «Регистрация телепередач» (№ п/п; дата выхода в эфир (7 разных значений); код передачи; название передачи; регулярность выхода в эфир (10 разных значений); время в эфире (5 разных значений); стоимость передачи) - 15 строк.

Выдать передачи, наиболее выгодные (по стоимости) для размещения рекламы с заданной цикличностью выхода в эфир.

Вариант 3

Исходная информация содержится в двух таблицах:

1. «Прейскурант цен» (№ п/п; код товара (ключевое поле); наименование товара; цена единицы товара) - 7 строк.

2. «Учет реализации товаров» (№ п/п; дата продажи (7 разных значений); код товара; наименование товара; количество проданного товара (10 разных значений); стоимость товара) - 15 строк.

Составить в алфавитном порядке список товаров реализованных после указанной даты с указанием количества, цены и объема выручки.

Вариант 4

Исходная информация содержится в двух таблицах:

1. «Прейскурант цен» (№ п/п; код товара (ключевое поле); наименование товара; цена единицы товара) - 7 строк.

2. «Учет реализации товаров» (№ п/п; ФИО продавца (7 разных фамилий), код товара; наименование товара; количество проданных единиц товара (10 разных значений); стоимость проданного товара) - 15 строк.

Получить отчет о выручке каждого продавца, с указанием количества и стоимости проданного товара.

Вариант 5

Исходная информация содержится в двух таблицах:

1. «Ассортимент» (№ п/п; код изделия (ключевое поле); наименование изделия; цена за единицу веса) - 7 строк.

2. «Регистрация заказов» (№ п/п; фирма-заказчик (7 разных значений); код изделия; наименование; вес изделия (10 разных значений); дата исполнения заказа (7 разных значений); стоимость заказа) - 15 строк.

Составить ведомость заказов на определенный день с указанием наименования изделия, веса, стоимости заказа.

Вариант 6

Исходная информация содержится в двух таблицах:

1. «Регистрация передач» (№ п/п; код передачи (ключевое поле); название рекламы; цена рекламы за единицу времени) - 7 строк.

2. «Регистрация заказов на рекламу» (№ п/п; фирма-заказчик (7 разных значений); код передачи; название рекламы; длительность рекламы (10 разных значений); стоимость рекламы) - 15 строк.

Найти среднюю стоимость всех реклам и список реклам, превышающих указанную величину стоимости. Составить ведомость передач, пользующихся наибольшей популярностью у рекламодателей (по максимальному времени).

Вариант 7

Исходная информация содержится в двух таблицах:

1. «Прейскурант цен» (№ п/п; код посуды (ключевое поле); наименование посуды; цена посуды) - 7 строк.

2. «Регистрация продаж» (№ п/п; дата продажи (7 разных значений); код посуды; название посуды; количество проданных единиц (10 разных значений); стоимость) - 15 строк.

Составить список наиболее популярной у покупателей посуды, а также список посуды, приносящей магазину наибольшую выручку. Составить ведомость работы магазина на конкретную дату с указанием количества и стоимости проданной посуды.

Вариант 8

Исходная информация содержится в двух таблицах:

1. «Регистрация поступлений» (№ п/п; код игрушки (ключевое поле); название игрушки; цена за единицу) - 7 строк.

2. «Ведомость покупок» (№ п/п; дата продажи (7 разных значений); код игрушки; название игрушки; возраст детей (10 разных значений); количество проданных игрушек (10 разных значений); стоимость) - 15 строк.

Составить ведомость игрушек наиболее популярных за последний месяц (с указанием количества и суммы продаж), а также получить ведомость игрушек, которые подходят для детей заданного возраста. Составить отчет о работе магазина на указанную дату.

Вариант 9

Исходная информация содержится в двух таблицах:

1. «Прейскурант цен» (№ п/п; код товара (ключевое поле); наименование товара; цена за единицу товара) - 7 строк

2. «Регистрация поставок» (№ п/п; код товара; наименование товара; страна, импортирующая товар (7 разных значений); объём поставляемой партии (10 разных значений); стоимость партии) - 15 строк.

Составить список товаров, пользующихся наибольшим спросом за рубежом, найти страну, экспортирующую товары на наибольшую сумму. Составить ведомость импортируемых товаров по каждой стране.

Вариант 10

Исходная информация содержится в двух таблицах:

1. «Тарифы» (№ п/п; код специальности (ключевое поле); название специальности; цена рабочего дня) - 7 строк.

2. «Табель учета рабочего времени» (№ п/п; ФИО рабочего (7 разных значений); код специальности; наименование цеха (5 разных значений); специальность; количество отработанных дней (10 разных значений); зарплата; удержания (13% от зарплаты); сумма к выдаче (зарплата - удержания)) - 15 строк.

Составить ведомость начисления зарплаты по определенному цеху с указанием суммы удержания и выдачи, а также вычислить среднемесячный заработок рабочего заданного цеха и среднемесячный заработок всех рабочих завода.

Вариант 11

Исходная информация содержится в двух таблицах:

1. «Расценки» (№ п/п; код изделия (ключевое поле); наименование изделия; стоимость сборки изделия) - 7 строк.

2. «Регистрация сборки изделий» (№ п/п; дата сборки (7 разных значений); фамилия сборщика (5 разных значений); код изделия; наименование изделия; количество собранных изделий (10 разных значений); стоимость работы) - 15 строк.

Найти сборщика, собравшего наибольшее число изделий, и день, когда он достиг наивысшей производительности труда, подсчитать месячный заработок каждого сборщика и указать фамилию самого отстающего сборщика. Составить ведомость рабочих дней каждого сборщика с указанием собранных изделий и их суммарной стоимости.

Вариант 12

Исходная информация содержится в двух таблицах:

1. «Регистрация наличия лекарств» (№ п/п; код лекарства (ключевое поле); название лекарства; цена за единицу) - 7 строк.

2. «Ведомость продаж» (№ п/п; дата продажи (7 разных значений); код лекарства; название лекарства; количество проданных упаковок (10 разных значений); стоимость лекарств) - 15 строк.

Составить ведомость лекарств, пользующихся наибольшим спросом за месяц с указанием объема выручки, получить ведомость популярных показаний за месяц. Составить отчет о проданных лекарствах на указанную дату.

Вариант 13

Исходная информация содержится в двух таблицах:

1. "Абонент" (№ п/п; фамилия абонента (ключевое поле); номер телефона; размер месячной оплаты) - 7 строк

2. "Регистрация оплаты" (№ п/п; фамилия абонента; номер телефона; дата оплаты (7 разных значений); размер месячной оплаты; сумма оплаты (10

разных значений); размер задолженности или переплаты (сумма оплаты - размер месячной оплаты)) - 15 строк.

Составить ведомость задолжников с указанием суммы, организовать справочную поисковую систему по абоненту, по телефону.

Вариант 14

Исходная информация содержится в двух таблицах:

1. «Аптеки» (№ п/п; код лекарства (ключевое поле); название лекарства; цена за единицу) - 7 строк.

2. «Лекарства» (№ п/п; код лекарства; название лекарства; дата изготовления (7 разных значений); срок годности (10 разных значений); количество упаковок (10 разных значений); номер аптеки (7 разных значений); стоимость лекарств) - 15 строк.

Получить сведения об аптеках, в которых есть указанное лекарство (с указанием цены лекарства), удалить сведения о лекарствах, срок годности которых истек.

Вариант 15

Исходная информация содержится в двух таблицах:

1. «Тарифы» (№ п/п; код товара (ключевое поле); название товара; цена единицы товара) - 7 строк.

2. «Поступления» (№ п/п; код товара; название товара; размер партии (10 разных значений); дата поступления (7 разных значений); стоимость партии) - 15 строк.

Составить ведомость наличия товаров с указанием общей стоимости, по запросу на получение товара выдать информацию о наличии товара в нужном количестве.

Вариант 16

Исходная информация содержится в двух таблицах:

1. «Прейскурант» (№ п/п; тип квартиры (ключевое поле); стоимость метра жилой площади; стоимость коммунальных услуг на человека) - 7 строк.

2. «Квартиросъемщик» (№ п/п; ФИО владельца (7 разных значений); количество проживающих (5 разных значений); жилая площадь (10 разных значений); тип квартиры; стоимость квартиры; сумма квартплаты) - 15 строк.

Составить ведомость задолжников по квартплате с указанием суммы задолженности.

Вариант 17

Исходная информация содержится в двух таблицах:

1. «Процент» (№ п/п; номер вклада (ключевое поле); название вклада; процент начисления) - 7 строк.

2. «Вкладчик» (№ п/п; номер вклада; название вклада; ФИО вкладчика (7 разных значений); сумма вклада (5 разных значений); дата вложения (7

разных значений); процент начисления; общая сумма с начислениями (сумма вклада + проценты)) - 15 строк.

Составить ведомость вкладов по названию с указанием суммы с начислениями; определить общую сумму начислений по всем видам вкладов.

Вариант 18

Исходная информация содержится в двух таблицах:

1. «Поступление» (№ п/п; артикул товара (ключевое поле); наименование товара; стоимость единицы товара) - 7 строк.

2. «Продажа» (№ п/п; дата продажи (7 разных значений); артикул товара; наименование товара; количество проданного товара (10 разных значений); стоимость проданного товара) - 15 строк.

Составить ведомость продаж за определенный день и ведомость наличия обуви заданного артикула.

Вариант 19

Исходная информация содержится в двух таблицах:

1. «Тариф» (№ п/п; код диагноза (ключевое поле); наименование диагноза; стоимость лечения за 1 день) - 7 строк.

2. «Пациент» (№ п/п; ФИО пациента (7 разных значений); пол, возраст (10 разных значений); место проживания (5 разных значений); код диагноза; наименование диагноза; количество дней в клинике (10 разных значений); стоимость лечения) - 15 строк.

Составить ведомость иногородних пациентов с указанием стоимости лечения и ведомость выручек за лечение (по каждому диагнозу).

Вариант 20

Исходная информация содержится в двух таблицах:

1. «Прейскурант» (№ п/п; код специальности (ключевое поле); название специальности; стоимость обучения за год) - 7 строк.

2. «Студент» (№ п/п; ФИО студента (7 разных значений); год поступления (5 разных значений); код специальности; название специальности; стоимость обучения; внесенная сумма оплаты (10 разных значений) - 15 строк.

Составить список студентов, переплативших на текущую дату и список должников по каждой специальности.

Вариант 21

Исходная информация содержится в двух таблицах:

1. «Тарифы» (№ п/п; код города (ключевое поле); название города; цена минуты разговора) - 7 строк.

2. «Переговоры» (№ п/п; ФИО владельца (7 разных значений); код города; название города; количество минут разговора (10 разных значений); стоимость разговора) - 15 строк.

Выписать счет на оплату телефонного разговора; составить ведомость по переговорам с данным городом на определенную дату.

Вариант 22

Исходная информация содержится в двух таблицах:

1. «Тариф» (№ п/п; тип кассеты (SONY, JVC и др) (ключевое поле); жанр фильма; цена проката за сутки) - 7 строк.

2. «Кассеты» (№ п/п; тип кассеты; жанр фильма; наименование фильма (10 разных значений); дата выдачи (7 разных значений); ФИО клиента (5 разных значений); срок проката (7 разных значений); сумма оплаты за прокат) - 15 строк.

Составить ведомость выручек по категориям кассет за определенный срок и хит-парад наиболее популярных за последний месяц фильмов.

Вариант 23

Исходная информация содержится в двух таблицах:

1. «Цены» (№ п/п; код товара (ключевое поле); наименование товара; цена единицы товара) - 7 строк.

2. «Товары» (№ п/п; код товара; наименование товара; номер магазина (5 разных значений); размер партии (10 разных значений); дата отправления (7 разных значений); стоимость партии) - 15 строк.

Составить ведомость товара определенного наименования, отправленного в магазины, с указанием общего веса и суммарной стоимости, а также подсчитать общую стоимость товаров, отправленных в магазины за определенный день.

Вариант 24

Исходная информация содержится в двух таблицах:

1. «Тариф» (№ п/п; код отправления (ключевое поле); название отправления (посылка, бандероль и др); цена за единицу веса) - 7 строк.

2. «Посылка» (№ п/п; ФИО отправителя (7 разных значений); код отправления; название отправления; вес отправления (10 разных значений); пункт назначения (5 разных значений); стоимость посылки) - 15 строк.

Выписать квитанцию на указанного отправителя; составить ведомость посылок в указанные пункты назначения с указанием общего веса и общей стоимости.

Вариант 25

Исходная информация содержится в двух таблицах:

1. «Номенклатура» (№ п/п; код изделия (ключевое поле); наименование изделия; цена за единицу изделия) - 7 строк.

2. «Заказы» (№ п/п; фирма-заказчик (5 разных значений); код изделия; наименование изделия; количество изделий (10 разных значений); дата заказа (7 разных значений); стоимость заказа) - 15 строк.

Составить ведомость исполненных за указанный месяц заказов с указанием количества и общей суммы выручки, а также ведомость невыполненных заказов.

Вариант 26

Исходная информация содержится в двух таблицах:

1. «Акции» (№ п/п; наименование акций (ключевое поле); стоимость 1 акции; % дивидендов на акцию в год) - 7 строк.

2. «Акционер» (№ п/п; ФИО акционера (5 разных значений); наименование акций; стоимость акции; количество акций (10 разных значений); дата приобретения акций (7 разных значений); начисления по акциям) - 15 строк.

Вычислить дивиденды указанного акционера, подсчитать общую сумму, которую акционерное общество обязалось выплатить акционерам по дивидендам.

Вариант 27

Исходная информация содержится в двух таблицах:

1. «Ассортимент» (№ п/п; код изделия (ключевое поле); наименование изделия; цена за единицу изделия) - 7 строк.

2. «Ведомость продаж» (№ п/п; дата продажи (7 разных значений); код изделия; наименование изделия; количество изделий (10 разных значений); стоимость изделий) - 15 строк.

Составить ведомость продаж на конкретную дату с указанием общего количества и стоимости проданных товаров.

Вариант 28

Исходная информация содержится в двух таблицах.

1. «Ассортимент» (№ п/п; код товара (ключевое поле); наименование товара; цена проката за день) - 7 товаров.

2. «Салон проката» (№ п/п; дата выдачи (7 разных значений); ФИО клиента (5 разных значений); код товара; наименование товара; количество дней проката (10 разных значений); стоимость проката) - 15 строк.

Составить ведомость работы салона проката: на конкретную дату с указанием наименования товара и стоимости.

Вариант 29

Исходная информация содержится в двух таблицах.

«Ассортимент» (№ п/п; код изделия (ключевое поле); наименование изделия; стоимость изделия) - 7 строк.

«Ведомость продаж» (№ п/п; дата продажи (7 разных значений); код изделия; наименование изделия; количество проданных изделий (10 разных значений); стоимость проданной партии) - 15 строк.

Составить ведомость продаж на указанную дату с указанием общей выручки.

Вариант 30

Исходная информация содержится в двух таблицах:

1. «Прейскурант цен» (№ п/п; марка автомобиля (ключевое поле); тип комплектации; стоимость базового автомобиля) - 7 строк.

2. «Ведомость продаж» (№ п/п; дата продажи (7 разных значений); ФИО менеджера (5 разных значений); марка автомобиля; тип комплектации; % скидки (10 разных значений); окончательная стоимость автомобиля) - 15 строк.

Составить ведомость продаж автомобилей заданной марки, а также ведомость работы салона на указанную дату.