**Лабораторная работа 1**

**Тема 1:** Типы данных SQL Oracle. Стандартные функции. Арифметические и логические выражения.

*(в лекциях см. п. 2, 1.4, 3 1.1)*

**Тема 2:** Агрегатные функции. Группировка строк. Сортировка строк

*(в лекциях см. п.3.1.1)*

**Пример задания:**

1. Вывести имена всех продавцов и города, в которых они работают, кроме продавцов, работающих в Лондоне.

**Решение:**

select sname,city from sal where city<>'London'

**Результат:**

|  |  |
| --- | --- |
| **SNAME** | **CITY** |
| Serres | San Jose |
| Rifkin | Barcelona |
| Axelrod | New York |

**Задание:**

**Вариант 0.**

1. Напишите запрос к таблице Покупателей, чей    вывод может включить всех покупателей, причем с оценкой выше 200, если они находятся не в Риме

2. Запросите двумя способами все заказы на 3 и 4 января 2010 г.

3. Напишите запрос, который сосчитал бы сумму всех заказов на 3 января 2010 г.

4. Напишите запрос, который выбрал бы наименьший заказ для каждого заказчика.

5. Напишите запрос, который выбрал бы наивысший рейтинг в каждом городе с сортировкой по возрастанию рейтингов.

**Лабораторная работа 2**

**Тема 1:** Выборка данных из объединенных таблиц

*(в лекциях см. п.3 1.1).*

**Тема 2:** Подзапросы

*(в лекциях см. п. 3.1.1).*

**Тема 2:** Представления

*(в лекциях см. п.3.3.3).*

 **Пример задания:**

1. Вывести для каждого продавца номера его заказов.

**Решение:**

select s.sname, o.onum

from sal s,ord o

where s.snum=o.snum

order by s.sname,o.onum;

**Результат:**

|  |  |
| --- | --- |
| **SNAME** | **ONUM** |
| Axelrod | 3009 |
| Motica | 3002 |
| Peel | 3003 |
| Peel | 3008 |
| Peel | 3011 |
| Rifkin | 3001 |
| Rifkin | 3006 |
| Serres | 3005 |
| Serres | 3007 |
| Serres | 3010 |

**Задание:**

**Вариант 0.**
1. Напишите запрос, который вывел бы для каждого  заказа на 3 января его номер, стоимость заказа, имя продавца и  размер комиссионных, полученных продавцом за этот заказ.
2. Выведите имена и города всех заказчиков с тем же рейтингом, что и у Hoffmana, используя подзапрос.
3. Создайте представление на основе запроса из задания 1 лабораторной работы № 1.
4. Выведите информацию о первых двух продавцах, имеющих заказы после 3-го числа.
5. Используя представление, созданное в задании 3, выведите данные о продавца из Берлина.

**Лабораторная работа 3**

**Тема 1:** Создание таблиц. Последовательности

*(в лекциях см. п.3.3.1, 3.3.2).*

**Тема 2:** Вставка, изменение и удаление данных из таблиц

*(в лекциях см. п.3.1.2).*

**Тема 3:** Транзакции

*(в лекциях см. п.3.2).*

**Пример задания:**

1. Создать таблицу с именем tab1 в базе данных одним числовым полем для целых двузначных чисел и двумя символьными полями размером по 50 байтов.

**Решение:**

create table tab1 (

col1 number(2),

col2 varchar2(50),

col3 varchar2(50)

)

**Результат:**

Table created.

**Задание:**

**Вариант 0.**

1. Создать таблицу для хранения данных о высших учебных заведениях. Таблица должна содержать поле для уникального номера ВУЗа, названия, количества факультетов. Создать первичный ключ для уникального номера.

2. Напишите команды для вставки в таблицу 5-7 записей о ВУЗах. Создайте последовательность и используйте ее в командах вставки для заполнения поля первичного ключа.

3. Напишите две команды изменения данных: одну - для изменения названия ВУЗа с самым большим номером, вторую - для увеличения вдвое количества факультетов для ВУЗа с самым маленьким номером. После этих двух команд напишите команду подтверждения транзакции.

4. Напишите команду удаления строк с данными о ВУЗах с нечетными номерами.

5. Напишите команду отмены транзакции, а затем повторите команду удаления (см. п.4), но для четных номеров. Подтвердите транзакцию.

6. Используя команду CREATE TABLE … AS SELECT, создайте таблицу для хранения данных, формируемых представлением из задания 3 лабораторной работы № 2.

**Лабораторная работа 4**

**Тема 1:** Блоки PL/SQL. Типы данных и операторы языка PL/SQL

*(в лекциях см. п.5, 6, 7).*

**Тема 2:** Вывод данных

*(в лекциях см. п. 10.1).*

**Пример задания:**

1. Написать и выполнить безымянный блок PL/SQL, который выводит значение символьной переменной.

**Решение:**

declare

vv varchar2(10);

begin

vv := ‘ABCD’;

DBMS\_OUTPUT.PUT\_LINE(‘значение vv=’||vv);

end;

/

Примечание: при наборе текста в окне SQL Commands апострофы (’) должны быть прямыми, а не наклонными.

**Результат:**

значение vv=ABCD

Statement processed.

0,00 seconds

**Задание:**

**Вариант 0.**
1. Написать и выполнить безымянный блок PL/SQL, в котором объявите одну переменную числового типа, одну переменную символьного типа и одну переменную типа даты. В основной части блока присвойте этим переменным конкретные значения и напишите функцию вывода значений этих переменных.
2. Составить и выполнить программу PL/SQL, которая вычисляет квадраты первых пяти положительных чисел, кратных 2, используя простой цикл Loop, и выводит результаты.
3. Составить и выполнить программу PL/SQL, которая определяет, под каким знаком Зодиака Вы родились. Для определения используйте оператор CASE.

**Лабораторная работа 5**

**Тема 1:** Работа с базами данных в программах PL/SQL. Курсоры

*(в лекциях см. п. 8).*

**Тема 2:** Обработка исключительных ситуаций

*(в лекциях см. п. 5.4).*

**Пример задания:**

1. Составить и выполнить программу PL/SQL, которая считывает из базы данных количество продавцов в Лондоне и выводит результат.

**Решение:**

declare

n number(2);

begin

select count(\*) into n from sal where city=’London’;

DBMS\_OUTPUT.PUT\_LINE(‘продавцов в Лондоне: ’||n);

end;

/

Примечание: при наборе текста в окне SQL Commands апострофы (’) должны быть прямыми, а не наклонными.

**Результат:**

продавцов в Лондоне: 2

Statement processed.

0,00 seconds

**Задание:**

**Вариант 0.**

1. Составить и выполнить программу PL/SQL, которая считывает из базы данных имя продавца, работающего в Барселоне, количество его заказов и выводит результат.

2. Составить и выполнить программу PL/SQL, которая считывает из базы данных имя продавца, работающего в Лондоне, количество его заказов и выводит результат. Добавить в программу раздел Exception с обработчиком OTHERS и выводом сообщения об ошибке из этого раздела.

3. Составить и выполнить программу PL/SQL, которая, используя курсор, считывает из базы данных номера заказов, сделанных до 04.01.2009, и выводит результат.

**Лабораторная работа 6**

**Тема 1:** Хранимые процедуры и функции PL/SQL

*(в лекциях см. п. 9.1, 9.2).*

**Тема 2:** Пакеты PL/SQL

*(в лекциях см. п. 9.3).*

**Пример задания:**

1. Создать хранимую процедуру PL/SQL, которая выводит наибольшее из двух чисел, заданных в качестве аргументов.

**Решение:**

create or replace procedure PrMax (

p\_N1 in number,

p\_N2 in number ) is

res number;

begin

res := case when(p\_N1 >= p\_N2) then p\_N1 else p\_N2 end;

DBMS\_OUTPUT.PUT\_LINE('вывод: '||res );

end;

/

**Результат:**

Procedure created.

2. Вызвать процедуру PrMax из безымянного блока, задав числовые аргументы.

**Решение:**

begin

PrMax(2, 8.9);

end;

/

**Результат:**

вывод: 8,9

**Задание:**

**Вариант 0.**
1. Создать хранимую процедуру PL/SQL, которая выводит произведение двух чисел, заданных в качестве аргументов. Вызвать процедуру из безымянного блока.
2. Создать пакет, в который поместить процедуру из п.1. Вызвать процедуру пакета из безымянного блока.
3. Включите в пакет процедуру, которая считывает из базы данных информацию о первых трех заказах, чьи суммы выше значения, переданного в параметре. Вызовите процедуру пакета из безымянного блока.

**Вариант 1.**
1. Создать хранимую процедуру PL/SQL, которая выводит минимальное из двух чисел, заданных в качестве аргументов. Вызвать процедуру из безымянного блока.
2. Создать пакет, в который поместить процедуру из п.1. Вызвать процедуру пакета из безымянного блока.
3. Включите в пакет процедуру, которая считывает из базы данных информацию о первых двух покупателях, имеющих заказы до даты, переданной в параметре. Вызовите процедуру пакета из безымянного блока.

**Лабораторная работа № 7**

**Тема 1:** Триггеры PL/SQL

                *(в лекциях см. п. 9.6).*

**Пример задания:**

1. Создать триггер PL/SQL, который регистрирует в журнальной таблице вставку строки в таблицу Ord. В журнал необходимо заносить вид операции, пользователя и время выполнения.

**Решение:**

CREATE OR REPLACE TRIGGER Tr\_Ord\_Log

AFTER INSERT ON Ord

BEGIN

INSERT INTO Ord\_Log (logid, lsql, luser, ltime)

VALUES(sq\_log.nextval, ‘INSERT’, USER, SYSDATE);

END Tr\_Ord\_Log;

/

**Результат:**

Trigger created.

2. Создать триггер PL/SQL, который регистрирует в журнальной таблице изменение комиссионных для продавцов только в том случае, если комиссионные увеличиваются.

**Решение:**

CREATE OR REPLACE TRIGGER Tr\_Sal\_Log

AFTER UPDATE ON Sal

FOR EACH ROW

WHEN (new.comm. > old.comm)

BEGIN

INSERT INTO Sal\_Log (logid, lsql, luser, ltime)

VALUES(sq\_log.nextval, ‘UPDATE’, USER, SYSDATE);

END Tr\_Sal\_Log;

/

**Результат:**

Trigger created.

**Задание:**

**Вариант 0.**

1. Создать триггер PL/SQL, который регистрирует в журнальной таблице изменение рейтинга для покупателей. При регистрации необходимо сохранять в журнале старое и новое значения рейтинга и время изменения.

2. Создать триггер PL/SQL, который регистрирует в журнальных таблицах операции изменения и удаления данных в таблице заказов. Каждый вид операции должен регистрироваться в своей журнальной таблице.

3. Создайте триггер, предназначенный для формирования первичного ключа при вставке строки в таблицу, созданную в задании 1 лабораторной работы 3.