**Лабораторная работа №1**

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

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

**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**

1. Напишите запрос к таблице Продавцов, чей вывод может включить всех продавцов, причем с комиссионными  не меньше 0.13, если они не находятся в Лондоне

2. Запросите двумя способами все заказы, обслуживаемые продавцами с номерами 1002 и 1007.

3. Напишите запрос, который сосчитал бы сумму всех заказов на 3 и 5 Октября 1990 г.

4. Напишите запрос, который выбрал бы среднюю сумму заказа для каждого продавца с номерами 1001, 1002 и 1003.

5. Напишите запрос, который выбрал бы самый поздний заказ каждого продавца с сортировкой по убыванию дат заказов.

**Лабораторная работа №2**

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

**Тема 2:** Подзапросы

**Тема 2:** Представления

**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**

1. Напишите запрос, который вывел бы для каждого заказа его номер, стоимость и имя заказчика. Данные вывести для заказчиков, размещенных в Лондоне и Нью-Йорке.  
2. Напишите запрос, который выводит все заказы, сумма которых больше средней по все заказам, используя подзапрос.  
3. Создайте представление на основе запроса из задания 1 лабораторной работы № 1.  
4. Выведите покупателей из последних трех по алфавиту городов и все их заказы.   
5. Используя представление, созданное в задании 3, выведите данные о продавца с комиссионными больше 0,11.

**Лабораторная работа №3**

**Тема 1:** Создание таблиц. Последовательности

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

**Тема 3:** Транзакции

**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:**

1. Создать таблицу для хранения данных о спортивных соревнованиях. Таблица должна содержать поле для уникального номера, названия соревнования, количества команд участников.

2. Напишите команды для вставки в таблицу 5-7 записей о соревнованиях. Создайте последовательность и используйте ее в командах вставки для заполнения поля уникального номера.

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

4. Напишите команду удаления строк с данными о соревнованиях 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

**Тема 2:** Вывод данных

**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:**

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, причем переменной типа даты присвойте значение системной даты. Напишите функцию вывода значений всех переменных.  
2. Составить и выполнить программу PL/SQL, которая вычисляет квадраты первых пяти положительных чисел, кратных 2, используя цикл While, и выводит результаты.  
3. Составить и выполнить программу PL/SQL, которая определяет, на какой квартал (кварталы) года попадает Ваш знак Зодиака. Для определения используйте оператор CASE.

**Лабораторная работа 5**

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

**Тема 2:** Обработка исключительных ситуаций

**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:**

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

2. Составить и выполнить программу PL/SQL, которая считывает из базы данных номера заказов для продавцов, работающих в Лондоне, 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 имена продавцов, работающих не в Лондоне, и выводит результат.

**Лабораторная работа 6**

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

**Тема 2:** Пакеты PL/SQL

**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:**

1. Создать хранимую процедуру PL/SQL, которая выводит разницу длин двух строк, 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 трех самых поздних заказах, выполненных до даты, переданной в параметре. Вызовите процедуру пакета из безымянного блока.

**Лабораторная работа № 7**

**Тема 1:** Триггеры PL/SQL

**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:**

1. Создать триггер PL/SQL, который запрещает удаление продавцов, если количество выполненных им заказов больше трех.

2. Создать триггер PL/SQL, который регистрирует в журнальной таблице операции вставки, изменения и удаление заказов. Регистрация выполняется только в нерабочее время и включает имя пользователя, вид операции и стоимость заказа.

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

**Курсовая работа**

Работу следует выполнять в среде программирования Oracle 10g XE.

**Требования к оформлению работы**

Необходимо представить следующее:

1. Текст задания ( с подставленными данными своего варианта);
2. Текст программы (текстовый файл или файл с расширением .sql) ;
3. Результат работы программы (т.е. все данные, которые выводятся на экран, – текстовый файл или файл, содержащий скрин-шоты панели вывода среды разработки Oracle XE).
4. **3. Общая часть задания**
5. Создать две таблицы, имеющие уникальные поля и первичные ключи. Таблицы должны быть связаны с помощью внешнего ключа.
6. Для одной из таблиц разработать триггер для обеспечения дополнительных действий при изменение данных таблицы (см. свой вариант задания).
7. Создать представление для связанных таблиц, которое позволяет вывести только часть полей и часть строк таблиц.
8. Создать пакет PL/SQL, содержащий процедуру начального заполнения таблиц данными (по 5-7 записей в таблице) и процедуру очистки таблиц (удаления записей).
9. Добавить в пакет процедуру (см. свой вариант задания) изменения данных в таблицах. Значения изменяемых данных должны передаваться в процедуру как параметры.
10. В процедурах предусмотреть обработку исключений.

**Задания на выполнение контрольной работы**

Внести в таблицу данные об Авиарейсах и Аэропортах. Каждый рейс может проходить по маршруту, включающему несколько аэропортов. Рейс имеет названия конечных пунктов, авиакомпанию и время отправления. Аэропорт имеет название и класс обслуживания. Процедура должна удалять N самых поздних рейсов; число N удаляемых рейсов должно передаваться в процедуру как параметр. Триггер должен запрещать удаление в последний день месяца. Включить в пакет еще одну процедуру, которая выводит рейсы, обслуживаемые аэропортами заданного в параметре класса.