**Лабораторная работа 5**

**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).*  
**Тема 3:** Триггеры PL/SQL   
                *(в лекциях см. п. 9.6).*

**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

3. 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.

4. 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 пакет, в который поместить процедуру, которая выводит произведение двух чисел, заданных в качестве аргументов. Вызвать процедуру пакета из безымянного блока.  
2. 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.  
3. 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.  
4. 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.

**Вариант 1.**  
1. Создать пакет, в который поместить функцию, которая возвращает минимальное из двух чисел, заданных в качестве аргументов. Вызвать процедуру пакета из безымянного блока.  
2. 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.  
3. Создать триггер PL/SQL, который запрещает изменение рейтинга для покупателей, если изменение происходит после 25-го числа.  
4. Создать триггер PL/SQL, который регистрирует в журнальной таблице операции вставки, изменения и удаления данных в таблице продавцов. При регистрации необходимо сохранять вид операции и время ее выполнения.

**Вариант 2.**  
1. Создать пакет, в который поместить процедуру, которая принимает в параметрах Ваши фамилию, имя и отчество, а выводит инициалы и фамилию. Вызвать процедуру пакета из безымянного блока.  
2. Включите в пакет процедуру, которая считывает из базы данных информацию о первых N (передать в параметре) по возрастанию стоимостей заказах и именах их покупателей. Вызовите процедуру пакета из безымянного блока.  
3. Создать триггер PL/SQL, который разрешает изменение суммы заказа только в рабочее время.  
4. Создать триггер PL/SQL, который регистрирует в журнальной таблице операции вставки и изменения заказов. Регистрация выполняется, если стоимость заказа превышает $100, и включает имя пользователя, вид операции и стоимость заказа.

**Вариант 3.**  
1. Создать пакет, в который поместить функцию, которая возвращает куб числа, заданного в качестве аргумента. Вызвать процедуру пакета из безымянного блока.  
2. Включите в пакет процедуру, которая считывает из базы данных информацию о последних N (передать в параметре) заказах, отсортированных по возрастанию стоимостей, и именах их продавцов. Вызовите процедуру пакета из безымянного блока.  
3. Создать триггер PL/SQL, который регистрирует изменение комиссионных продавцов с указанием только имени таблицы и времени изменения.  
4. Создать триггер PL/SQL, который запускается при удалении покупателей. Триггер должен запрещать удаление, если покупатель имеет более одного заказа.

**Вариант 4.**  
1. Создать пакет, в который поместить процедуру, которая выводит заглавными буквами строку, заданную в качестве аргумента. Вызвать процедуру пакета из безымянного блока.  
2. Включите в пакет процедуру, которая считывает из базы данных информацию о первых N (передать в параметре) по алфавиту продавцах и их заказах. Вызовите процедуру пакета из безымянного блока.  
3. Создать триггер PL/SQL, который запрещает удаление заказов продавца в последний день месяца.  
4. Создать триггер PL/SQL, который регистрирует в журнальной таблице операции вставки, изменения и удаления данных в таблице заказов. При регистрации необходимо сохранять вид операции, номер и сумму заказа.

**Вариант 5.**  
1. Создать пакет, в который поместить функцию, которая возвращает самую длинную из двух строк, заданных в качестве аргументов. Вызвать процедуру пакета из безымянного блока.  
2. Включите в пакет процедуру, которая считывает из базы данных информацию о продавцах из первых N (передать в параметре) по алфавиту городов и их заказах. Вызовите процедуру пакета из безымянного блока.  
3. Создать триггер PL/SQL, который разрешает удаление покупателей с количеством заказов меньше трех.  
4. Создать триггер PL/SQL, который регистрирует в журнальной таблице операции вставки, изменения и удаления данных в таблице заказов, а также округляет стоимость заказа до целого числа при вставке и изменении.

**Вариант 6.**  
1. Создать пакет, в который поместить процедуру, которая выводит самую короткую из двух строк, заданных в качестве аргументов. Вызвать процедуру пакета из безымянного блока.  
2. Включите в пакет процедуру, которая считывает из базы данных информацию о  последних N (передать в параметре) по алфавиту покупателях и их заказах. Вызовите процедуру пакета из безымянного блока.  
3. Создать триггер PL/SQL, который регистрирует изменение комиссионных, только если новое значение больше прежнего.  
4. Создать триггер PL/SQL, который запускается при вставке, изменении и удалении заказов. Триггер должен регистрировать тип и дату операции и запрещать удаление, если стоимость заказа превышает $5000.

**Вариант 7.**  
1. Создать пакет, в который поместить функцию, которая возвращает суммарную длину двух строк, заданных в качестве аргументов. Вызвать процедуру пакета из безымянного блока.  
2. Включите в пакет процедуру, которая считывает из базы данных информацию о покупателях из последних N (передать в параметре) по алфавиту городов и их заказах. Вызовите процедуру пакета из безымянного блока.  
3. Создать триггер PL/SQL, который регистрирует изменение городов для покупателей с сохранением в журнале имени покупателя, прежнего и нового названия города.  
4. Создать триггер PL/SQL, который запускается при вставке, изменении и удалении заказов. Триггер должен регистрировать тип и время операции и запрещать удаление, если количество заказов в таблице не превышает 10.

**Вариант 8.**  
1. Создать пакет, в который поместить функцию, которая принимает в параметрах Ваши фамилию, имя и отчество, а возвращает фамилию и инициалы. Вызвать процедуру пакета из безымянного блока.  
2. 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.  
3. Создать триггер PL/SQL, который запрещает удаление продавцов, если количество выполненных им заказов больше трех.  
4. 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.

**Вариант 9.**  
1. Создать пакет, в который поместить процедуру, которая выводит строчными (маленькими) буквами строку, заданную в качестве аргумента. Вызвать процедуру пакета из безымянного блока.  
2. Включите в пакет процедуру, которая считывает из базы данных информацию о первых двух продавцах, имеющих заказы после даты, переданной в параметре. Вызовите процедуру пакета из безымянного блока.  
3. Создать триггер PL/SQL, который разрешает удаление покупателей только из города Лондон.  
4. Создать триггер PL/SQL, который регистрирует в журнальных таблицах операции вставки, изменения и удаления данных в таблице продавцов. Каждый вид операции необходимо регистрировать в своей таблице. При регистрации необходимо сохранять пользователя и время ее выполнения.

**ЗАДАНИЕ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ**

**1. Рекомендации по выполнению курсовой работы**

Прежде, чем приступать к выполнению курсовой работы, необходимо выполнить все лабораторные работы данного курса.

Курсовая работа состоит из задания, которое выбирается согласно своему варианту. Если работа выполнена в соответствии с заданием, в ней нет грубых ошибок, то вы получаете положительную оценку по курсовой работе. В противном случае работа отправляется вам на доработку.

**!!! Номер варианта курсовой работы определяется по последней цифре пароля**

Работу следует выполнять в среде программирования Oracle 10g XE или более высокой версии XE.

**2. 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).

**3. Критерии оценки курсовой работы**

Оценка "отлично" выставляется, если: созданы все объекты, указанные в задании; созданные объекты отвечают указанным требованиям полноты и функциональности; при обработке данных максимально используются возможности, предоставляемые языками программирования SQL и PL/SQL.

Оценка «хорошо» выставляется, если: созданы все объекты, указанные в задании; не более, чем два объекта имеют несущественные отклонения от указанных требований, не влияющие на результирующую функциональность; программные процедуры дают правильный результат, но обработка данных выполняется нерационально, не используя возможности языков SQL и PL/SQL.

Оценка «удовлетворительно» выставляется, если: созданы не все объекты, указанные в задании, но остается возможность хранить и обрабатывать данные в соответствии с правилами предметной области; не более, чем две программные процедуры не отвечают требованиям функциональности.

Оценка «неудовлетворительно» выставляется, если имеет место одно из следующих: созданные объекты не обеспечивают хранение и обработку данных в соответствии с правилами предметной области; более, чем две программные процедуры не отвечают требованиям функциональности.

**4. Общая часть задания**

Создать две таблицы, имеющие уникальные поля и первичные ключи. Таблицы должны быть связаны с помощью внешнего ключа.

Для одной из таблиц разработать триггер для обеспечения дополнительных действий при изменение данных таблицы (см. свой вариант задания).

Создать представление для связанных таблиц, которое позволяет вывести только часть полей и часть строк таблиц.

Создать пакет PL/SQL, содержащий процедуру начального заполнения таблиц данными (по 5-7 записей в таблице) и процедуру очистки таблиц (удаления записей).

Добавить в пакет процедуру (см. свой вариант задания) изменения данных в таблицах. Значения изменяемых данных должны передаваться в процедуру как параметры.

В процедурах предусмотреть обработку исключений.

Обеспечить подтверждение транзакций при их успешном выполнении и откат - в случае возникновения исключительной ситуации.

**5. Варианты задания на выполнение контрольной работы**

**0.** В таблицах должны содержаться данные о Строительных компаниях и Газопроводах. Одна компания может строить несколько газопроводов. Компании должны иметь номер, наименование и страну размещения. Газопроводы должны иметь номер, наименование и стоимость строительства. Процедура должна увеличить стоимость строительства газопроводов для двух компании с наименьшими стоимостями своих строительных работ; процент увеличения задать в параметре процедуры. Триггер должен запрещать изменения 13-го числа. Включить в пакет еще одну процедуру, которая, используя представление, выводит компании и количества и суммарные стоимости их газопроводов.   
  
**1.** В таблицах должны находиться данные о Компьютерных фирмах и Типах компьютеров. Компьютеры каждого типа можно приобрести в нескольких фирмах. Фирмы должны иметь номер, название и город размещения. Компьютеры должны иметь номер, тип и рабочую частоту. Процедура должна удалить компьютер с самой низкой частотой и все фирмы, в которых продавался этот компьютер. Триггер должен разрешать удаление только собственнику таблицы. Включить в пакет еще одну процедуру, которая выводит компьютеры и количества фирм, продающих компьютеры этого типа; выводиться должны компьютеры, для которых количество фирм более заданного в параметре.   
  
**2.** Внести в таблицы данные о Морских круизах и Страховых компаниях. Страховая компания может обслуживать несколько круизов. Круизы должны иметь номер, порт отправления и сумму страховки. Компании должны иметь номер, наименование и страну размещения. Процедура должна удалить круизы для компании с максимальным четным номером. Триггер должен регистрировать удаление с указанием пользователя, даты и времени операции. Включить в пакет еще одну процедуру, которая выводит компании, количества обслуживаемых круизов и средние суммы их страховки, кроме компании, указанной в параметре.   
  
**3.** Занести в таблицы данные о Научных конференциях и Организаторах. Каждая конференция может иметь несколько организаторов. Конференции должны иметь номер, наименование и число участников. Организаторы должны иметь номер, наименование и бюджеты финансирования. Процедура должна изменять первичный ключ конференции, при этом ее организаторы не должны потеряться. Триггер должен регистрировать изменение с указанием пользователя, даты и времени операции, если значение ключа увеличивается. Включить в пакет еще одну процедуру, которая, используя представление, выводит конференции и количества их организаторов; в вывод должны попадать конференции с количеством организаторов, не менее, чем задано в параметре.   
  
**4.** Требуется обрабатывать данные о Телефонных компаниях и Услугах связи. Компания предоставляет несколько видов услуг. Компании должны иметь номер, наименование и количество абонентов. Услуги должны иметь номер, наименование и тариф. Процедура должна добавлять новую компанию и ее услугу; данные о компании и услуге должны быть переданы в параметрах процедуры. Триггер должен разрешать добавление, если число компаний на момент добавления не превышает семи. Включить в пакет еще одну процедуру, которая выводит компании, количества и средние тарифы предоставляемых услуг; выводиться должны компании, для которых средние тарифы попадают в диапазон, заданный в параметрах.   
  
**5.** Поместить в таблицы информацию о Теннисных турнирах и Теннисистах. В каждом турнире принимает участие не менее четырех теннисистов. Турниры должны иметь номер, наименование и страну проведения. Теннисисты должны иметь номер, имя и страну, за которую он выступает. Процедура должна удалить турниры с количеством участников, менее заданного в параметре. Триггер должен запрещать удаление после 25-го числа. Включить в пакет еще одну процедуру, которая, используя представление, выводит турниры и имена их участников; в вывод должны попадать турниры с количеством участников, заданном в параметре.   
  
**6.** Таблицы должны содержать данные об Издательствах и их Спонсорах, причем каждый спонсор может финансировать несколько издательств. Издательства должны иметь номер, наименование и сумму спонсорской помощи. Спонсоры должны иметь номер, наименование и страну пребывания. Процедура должна изменять спонсорскую помощь тем издательствам, чей спонсор размещается в указанной стране; страну и сумму изменения задать в параметре процедуры. Триггер должен регистрировать изменение с указанием пользователя, даты и времени операции, если сумма помощи увеличивается. Включить в пакет еще одну процедуру, которая выводит N самых щедрых спонсоров; N должно передаваться как параметр процедуры. Выводиться должны спонсоры и их суммарная спонсорская помощь.   
  
**7.** Занести в таблицы данные о Кинофильмах и Киностудиях. Каждая киностудия может снимать несколько фильмов. Кинофильмы должны иметь номер, наименование и бюджет. Киностудии должны иметь номер, наименование и страну пребывания. Процедура должна добавлять новую киностудию и ее фильм без указания бюджета; данные о киностудии и фильме должны быть переданы в параметрах процедуры. Триггер должен регистрировать добавление с указанием пользователя, даты и времени операции. Включить в пакет еще одну процедуру, которая, используя представление, выводит киностудии и снимающиеся на них фильмы. Для каждой студии должны выводиться только фильмы с минимальным и максимальным бюджетами.   
  
**8.** Поместить в таблицы информацию о Клубах кинологов и Породах собак. Собаку любой породы можно приобрести в нескольких клубах. Клубы должны иметь номер, наименование и диапазон цен продажи щенков. Породы должны иметь номер, наименование и тип. Процедура должна удалять породы, которые продаются в наименьшем числе клубов. Триггер должен разрешать удаление, если количество пород в таблице не менее пяти. Включить в пакет еще одну процедуру, которая выводит количество клубов, где можно приобрести собаку указанной породы, и название клуба, в котором нижняя граница цены минимальна.   
  
**9.** Поместить в таблицы информацию об Автомобильных фирмах и Странах. Каждая фирма может иметь офисы в нескольких странах. Фирмы должны иметь номер, наименование и страну принадлежности. Страны должны иметь номер, наименование и число выпускаемых автомобилей. Процедура должна изменять число выпускаемых в каждой стране автомобилей для фирмы с максимальным номером, кратным трем; величину изменения передавать в параметре процедуры. Триггер должен разрешать изменение только собственнику таблицы. Включить в пакет еще одну процедуру, которая выводит суммарное количество выпускаемых автомобилей заданная в параметре фирма, а также страны, где она имеет свои офисы.