

Практические задания по курсу PL/SQL

1. Модульное программирование

- 1.1. Создайте процедуру для включения нового отдела в таблицу S_DEPT.
- 1.2. Создайте процедуру, обновляющую номер региона для существующего отдела.
- 1.3. Создайте процедуру для удаления отдела, созданного в упражнении 1.
- 1.4. Создайте процедуру NEW_EMP для включения записи о новом служащем в таблицу S_EMP.

2. Простой блок PL/SQL

- 2.1. Создайте процедуру, принимающую два числа через переменные. Первое нужно разделить на второе и к результату прибавить второе число. Результат должен быть записан в переменную PL/SQL и выведен на экран.
- 2.2. Создайте функцию, вычисляющую общее вознаграждение за год. Функции должны передаваться годовая зарплата и процент премиальных. Премиальные необходимо преобразовать из целого числа в десятичное. Если зарплата не определена, то функция должна выдавать значение «ноль». Если премия не определена, функция должна выдавать только зарплату.

3. Управление потоками в блоках PL/SQL

- 3.1. Скопируйте таблицу s_emp в s_emp_copy. Для s_emp_copy напишите набор функций:
 - Вставка строки в таблицу. Если отдел не введен то сотрудник заносится в самый малочисленный отдел.
 - Выбор строки. Если указан id, то выводится строка с указанным ключом. Если нет, то происходит поиск по фамилии, имени, дате начала работы, году начала работы, отделу и заработной плате.
 - Удаление. Если указана id, то удаляется строка с указанным ключом. Если нет, то происходит поиск по фамилии, имени, дате начала работы, году начала работы, отделу и заработной плате.
- 3.2. Создайте процедуру SET_COMM, которая устанавливает процент комиссионных для служащего в зависимости от общего объема продаж
 - Создайте параметр, чтобы можно было ввести номер служащего.
 - Подсчитайте сумму всех заказов, размещенным этим служащим.
 - Если эта сумма меньше 10000 установите процент комиссионных «10».
 - Если сумма находится в диапазоне от 10000 до 1000000, установите процент комиссионных «15».
 - Если в таблице S_ORD нет заказов, размещенных данным служащим, установите процент комиссионных «0».
- 3.3. Создайте процедуру CUST_UPDATE с циклом для обработки кредитного рейтинга всех заказчиков по регионам. Не фиксируйте изменения
 - Если номер региона четный, установите кредитный рейтинг “Excellent”, даже если он уже был таким; в противном случае установите кредитный рейтинг “Good”.
 - Когда строки будут обновлены, определите количество обновленных строк. В зависимости от того, каким будет это количество, выдайте на экран следующую информацию:
 - если обновлено менее трех строк, сообщение : “Fewer then 3 customer records updated for region number X”, где X – номер региона;
 - в противном случае – “Y rows updated for region number X”, где Y – количество обновленных строк, X – номер региона.

- Отмените изменения. Задайте точку останова на проверке количества обновленных строк.
- 3.4. Создайте процедуру EMP_MESSAGE, которая выбирает фамилию служащего, дату начала работы и заработную плату по номеру служащего, задаваемому во времени выполнения. Выдайте на экран сообщение по любой комбинации нижеуказанных критериев (используйте вложенные операторы IF)

Критерий	Сообщение
Зарплата более 1200	Salary >1200
Фамилия содержит "R"	Name contains "R"
Дата начала работы приходится на март	March start date
Ничего из вышеуказанного	**None**

4. Циклы

- 4.1. Создайте таблицу ZAKAZ со следующими полями (id number, order_name varchar2, product_id number, quantity number, comment varchar2).
- Для всех заказчиков, общая сумма заказов которых превышает 50000, внесите данные в таблицу. Для задания первичного ключа создайте последовательность.
 - Для всех заказчиков не принадлежащих региону «Европе» добавить комментарий «out region»
 - Проверить количество обновленных строк. Если были обновлены все строки, то вывести сообщение «нет заказчиков из европейского региона».
- 4.2. Создайте функцию вычисления либо общего количества лет, которые сотрудник проработал на фирме либо даты десятилетия его рабочей деятельности в зависимости от флага.
- Создайте таблицу STAG со следующими полями (id number, emp_id, FIO varchar2, ord_id number, ord_name varchar2, commentar varchar2). Заполните таблицу данными, имеющимися в базе (используйте данные из таблиц). Для задания первичного ключа создайте последовательность.
 - Для всех сотрудников, внесенных в таблицу STAG, с четными номерами добавьте в столбец комментария сообщение «стаж XXX лет», где XXX-вычисленное количество отработанных сотрудником лет.
 - Для всех сотрудников, внесенных в таблицу STAG, с нечетными номерами добавьте в столбец комментария сообщение «До 10-тилетия раб. деят. Осталось XXX лет», где XXX – вычисленное количество лет, которые осталось проработать сотруднику до 10-тилетнего юбилея.
 - Определить количество сотрудников, проработавших на фирме более 10 лет. Вывести сообщение «XXX сотрудников проработало на фирме более 10 лет», где XXX – вычисленное количество сотрудников.
 - Если у сотрудника в текущую дату (sysdate) – юбилей, то вывести на экран следующее сообщение «Поздравляем СОТРУДНИК_ИМЯ с 10-тилетием его рабочей деятельности!»

5. Обработка исключений

- 5.1. Создайте процедуру, которая бы обновляла номера регионов для служащих отделов. (Входные данные – название региона и новый номер отдела для данного служащего).
- Напишите обработчик исключений, который будет выдавать сообщение о том, что указанный регион не существует.

- Напишите обработчик исключений, который бы выдавал пользователю сообщение о том, что для указанного региона уже есть отдел с таким названием.
- Напишите обработчик исключений, который выдавал бы пользователю сообщение о том, что указанный номер отдела не существует (используйте атрибут SQL%NOTFOUND и возбуждите исключение вручную). Напишите процедуру для вывода на фамилии и названия отдела для служащих, чья заработная плата лежит в диапазоне плюс-минус 100\$ от введенного значения.
- Напишите обработчик исключений, который будет выдавать сообщение о том, что служащих с такой зарплатой нет.
- Напишите обработчик исключений, который будет выдавать сообщение о том, что служащих с такой зарплатой несколько. Сообщение должно указывать, сколько сотрудников попадает в этот диапазон зарплат.

6. Курсоры

- 6.1. Создайте процедуру TOP_DOGS1 для определения самых высокооплачиваемых служащих
- Для этого упражнения создайте новую таблицу с данными о служащих и их заработной плате.
 - Включите параметр, чтобы пользователь мог ввести нужное количество самых высокооплачиваемых служащих (n).
 - Создайте цикл FOR с курсором для выборки из таблицы S_EMP фамилий и заработной платы n самых высокооплачиваемых служащих.
 - Сохраните фамилию и заработную плату в таблице TOP_DOGS.
 - Предполагается что двух служащих с одинаковой заработной платой не существует.
 - Проверьте особые случаи – например, с n=0 и с n, превышающим количество служащих в таблице S_EMP.
 - После каждого теста удаляйте данные из таблицы TOP_DOGS. Создайте хранимую процедуру ADD_STARS, которая в новом столбце STARS проставляет для каждого служащего по призовой звёздочке за каждый процент заработанных комиссионных. Используйте курсор и цикл WHILE.
 - Для этого упражнения создайте в таблице S_EMP новый столбец для хранения звёздочек (*)
 - Определите процент комиссионных для каждого служащего, округлив его до ближайшего целого числа. Рассмотрите случай, когда служащий не получает комиссионных.
 - Добавляйте звёздочку в строку звёздочек за каждый процент комиссионных. Если, например, служащий получает 10 процентов комиссионных, символьная строка в столбце STARS должна содержать десять звёздочек.
 - Проставьте соответствующее количество звёздочек для каждого служащего в столбце STARS.
- 6.3. Скопируйте процедуру TOP_DOGS1 из упражнения 6.1 и назовите новый вариант TOP_DOGS2. Измените процедуру TOP_DOGS2 с учётом случая, когда несколько служащих из упражнения 1 имеют одинаковую заработную плату. Для каждой фамилии в списке должны быть перечислены все служащие с такой же заработной платой. Выполните процедуру TOP_DOGS2. В качестве n введите число 6, 7 или 8. В этом случае в выходных данных должны появиться фамилии Ngao, Dumas и Quick-To_See. Если же n равно 9, 10 или 11, должны появиться фамилии Nagayama, Magee и Maduro. Не забывайте полностью удалять данные из таблицы TOP_DOGS после каждого теста.

- 6.4. Напишите процедуру для печати фамилии служащих, чья заработная плата лежит в диапазоне плюс минус \$100 от введённого значения.
- Если служащего с такой заработной платой нет, пользователь должен получить соответствующее сообщение. Используйте исключение.
 - Если служащих с такой зарплатой более 3, сообщение должно указывать, сколько сотрудников попадёт в этот диапазон зарплат.
- 6.5. Создайте процедуру для определения:
- Сотрудников, работающих больше заданного числа лет.
 - Для этих сотрудников определите менеджеров. Выведите без повторов полученный список менеджеров с указанием рядом с каждым именем через запятую имён сотрудников из начального списка.
 - Определите и выведите на экран менеджеров, принявших суммарные заказы более заданной суммы.
- 6.6. Напишите процедуру, определяющую рейтинг лучших товаров по:
- суммарной цене приобретённого за указанный срок товара;
 - количеству единиц приобретённого за указанный срок товара;
 - количеству заказов, в которые входил приобретённого за указанный срок товар. Т.е. первым выдаётся на печать товар, у которого максимальный показатель по выбранному критерию, затем товар с более низким показателем и т.д.
 - Какой из критериев использовать определяется по входным параметрам. Если указано несколько критериев, то их приоритет соответствует списку, указанному выше.
- 6.7. Скопируйте таблицу `s_emp` в `s_emp_soru`. Для `s_emp_soru` напишите пакет функций:
- Вставка строки в таблицу. Если отдел не введён то сотрудник заносится в самый малочисленный отдел.
 - Выбор строки. Если указан `id`, то выводится строка с указанным ключом. Если нет, то происходит поиск по фамилии, имени, дате начала работы, году начала работы, отделу и заработной плате.
 - Удаление. Если указана `id`, то удаляется строка с указанным ключом. Если нет, то происходит поиск по фамилии, имени, дате начала работы, году начала работы, отделу и заработной плате.
 - Разработать структуру логирования информации для таблицы `s_emp_soru` с возможностью восстановления данных за нужный период.
- 6.8. Напишите процедуру, осуществляющую поиск в зависимости от входных параметров:
- 1. Указан интервал времени. Поиск заказчиков у которых время, прошедшее от заказа до получения товара больше чем указанный интервал.
 - 2. Наименование товара. Список заказчиков, закупувших товар с таким наименованием (при этом не поиск должен работать независимо от указанного регистра или числа пробелов).
 - 3. Сумма. Заказчиков, приобретших товар на сумму более указанной.
 - Добавить в качестве входного параметра флаг. В первом случае он позволяет выбирать заказчиков по условиям независимо. Выбранных заказчиков без повторения поместить в специальную таблицу и в отдельном столбце в ней ставить столько плюсикув, сколько пунктов для заказчика истинно.
 - Второе значение флага позволяет выбирать заказчиков удовлетворяющих всем указанным пунктам.

- 6.9. Создайте процедуру EMP_MESSAGE, которая выбирает фамилию служащего, дату начала работы и зарплату по номеру служащего заданному при выполнении. Выдайте на экран сообщение с любым сочетанием нижеуказанных критериев. Проверьте номера сотрудников 2,5,16,17 и 18.

7. Динамический SQL

- 7.1. Напишите процедуру, возвращающую список товаров, количество каждого товара и его цену. Процедура должна иметь два входных параметра - название фирмы и название товара. Если указано название фирмы, выдаётся список всех купленных ею товаров. Если указано название фирмы и дополнительно указано наименование товара, то выдаётся товар, купленный фирмой, наименование которого совпадает с заданным наименованием. Если указано только наименование товара, то выдаётся указанный товар. Процедуру написать с использованием позиций предназначенных для размещения переменных PL/SQL, указываемых в предложении USING оператора EXECUTE IMMEDIATE.

8. Триггеры DML

- 8.1. Для отслеживания изменений информации о сотрудниках, создайте таблицу s_emp_log и напишите триггер, заносащий в s_emp_log информацию о времени изменения и информации о сотруднике до изменения.
- 8.2. Для отслеживания удалений информации о сотрудниках, напишите триггер, заносащий в s_emp_log удалённую информацию, а также время удаления.
- 8.3. Напишите триггер для таблице s_item, позволяющий контролировать изменение цены товара. Если новая цена отличается от старой более чем на 30%, выдаётся соответствующее сообщение и запрещается изменения данных. Реализовать используя исключения.
- 8.4. Требуется отслеживать статистические показатели, касающиеся продуктов. Т.е. для каждого существующего товара указывается количество заказчиков этого товара, количество заказанных единиц, сумма заказа. Результаты будут храниться в таблице major_stats.
- 8.5. Напишите триггер который разрешает работу команды DML с таблицами в нормальное рабочее время с 8:45 до 5:30, с понедельника по пятницу. Создайте процедуру SECURE_DML, которая при попадании в эти часы выводит сообщении: "Data changes restricted to office hours".

9. Триггеры замещения

- 9.1. Создайте представление, содержащее имя отдела, номер региона, Фамилию, имя, должность и номер сотрудника. Операции DML для данного представления определим по следующим правилам:
- INSERT – Назначить отдел, введённому сотруднику. В результате s_dept (при необходимости s_region) обновляются.
 - UPDATE – Изменить отдел, назначенный сотруднику. Это может привести к обновлению или s_emp или s_dept, в зависимости от того, какой столбец представления обновляется.
 - DELETE – Очистить идентификатор отдела для сотрудника (сотрудник не зачислен ни в один отдел). В результате s_emp обновляется: ID устанавливается в значении NULL.
 - Создайте триггер, реализующий сформулированные выше правила и позволяющий правильно выполнять операции DML над созданным представлением.

- Входные данные процедуры – ФИО, должность сотрудника, название отдела. Зарботную плату сотруднику определить как среднюю з/п всех сотрудников фирмы.