

Лабораторный практикум 7
Тема: Проектирование бизнес-логики на языке PL/SQL
Вариант 2

1 Описание тестовой базы данных

Для выполнения заданий используется стандартная демо-схема БД *Oracle - HUMAN RESOURCES (HR)* из лабораторной работы 1.

Задания

Подготовка базы данных

Выполнить скрипт `hr_create.sql` с командами создания таблиц БД.

Выполнить скрипт `hr_popul.sql` с командами заполнения таблиц БД.

Задание 1 Массовое внесение изменений в БД

Часто для проведения тестирования производительности, иногда, функционального тестирования, необходимо использовать таблицы с количеством строк, соизмеримым с реальным состоянием на производстве (сотни, тысячи, миллионы). Для этого используются генераторы искусственных (суррогатных) строк.

1.1 В предыдущей лабораторной работе при рассмотрении иерархических запросов был указан способ их использования для массовой (пакетной) генерации значений атрибутов таблиц.

Создать запрос типа *INSERT ALL* по автоматической регистрации в БД 10000 сотрудников, учитывая следующее:

- для идентификаторов сотрудника использовать значение, сгенерированное иерархическим запросом (значение генератора);
- имя, фамилия сотрудника определяется как Ваше имя, фамилия + значение генератора;
- E-mail сотрудника определяется как Ваше имя + значение генератора;
- дата зачисления определяется как '01.01.2000' + значение генератора;
- должность = «Finance Manager»;
- остальные значения колонок оставить NULL.

1.2 Предыдущее решение позволяет создавать простые генераторы.

Отменить операцию внесения данных предыдущего задания.

Создать анонимный PL/SQL-блок, автоматически регистрирующий в БД 10000 сотрудников, учитывая условия из задания 1.1.

Задание 2 Обработка исключений

В решение задания 1.2 добавить контроль ограничений целостности:

- внесения дубликатов по E-mail сотрудников, нарушающих ограничение целостности *UNIQUE*, с выводом ошибки, например, “E-mail Ivanov already exists!”;
- внесения отрицательной зарплаты с выводом сообщения об ошибке, например, “Salary = -100 is incorrect salary!”

Для проверки срабатывания созданных исключений последовательно выполнить PL/SQL-блок, вносящий:

- 1) те же E-mail, что и при выполнении задания 1.2;
- 2) любую отрицательную зарплату.

Задание 3 Работа с курсорами

Описать операции транзакции в виде PL/SQL-кода:

- 1) получить список идентификаторов подразделений, в которых есть сотрудники;
- 2) получить список сотрудников 2-го по списку подразделения;
- 3) перевести сотрудников в 3-е по списку подразделение;
- 4) сохранить данные о сотрудниках в таблице `job_history`

Задание 4 Автоматическая инициализация генераторов уникальных значений

В предыдущих лабораторных работах необходимо было создавать генераторы последовательностей для PK-атрибутов таблиц с уже заполненными данными.

В Oracle 11XE отсутствует возможность создавать генераторы (sequence), автоматически проставляя начальные значения с учетом содержимого таблиц БД, что требует от администратора вручную выполнять запросы на получение максимальных значений PK-атрибутов.

4.1 Создать анонимный PL/SQL-блок, автоматизирующий этот процесс на основе шагов:

- определение максимального значения идентификатора подразделения в таблице Departments и идентификатора сотрудника в таблице Employees;
- проверка наличия генератора в БД с учетом заранее известных названий для таблиц Departments, Employees, используя запрос по шаблону `select sequence_name from user_sequences where sequence_name = 'название_в_верхнем_регистре'`;
- если генераторы уже существуют, выполнение команды удаления генераторов;
- создание генераторов с учетом смещений начального значения, превышающего на 1 полученные максимальные значения.

4.2 В решение 1-го задания изменить PL/SQL-код так, чтобы не было необходимости проверять наличие генераторов в БД через создание заглушки

Задание 5 Динамические запросы

Отменить ранее выполненные операции внесения записей по сотрудникам.

Создать анонимный PL/SQL-блок, который автоматически регистрирует сотрудников с фамилией (*last_name*), начинающейся на букву *C* или *D*, как пользователей Oracle с учетом условий:

- имена пользователей (логины) совпадают с E-mail сотрудников из таблицы *employees*;
- пароль генерируется как любая константа;
- пользователю-сотруднику после регистрации предоставляется право входа в систему, т.е. автоматически выполняется команда *GRANT CONNECT TO пользователь*;
- пользователю-сотруднику, работающему на должности, связанной с управлением (в названии должности есть слово *manager*), предоставить право управлять ресурсами, т.е. автоматически выполняется команда *GRANT RESOURCE TO пользователь*;

Необходимо учесть, что пользователь, запускающий созданный скрипт, должен иметь полномочия на выполнение команд: *CREATE USER*, *GRANT CONNECT*, *GRANT RESOURCE*.

Задание 6 Пакетная работа с данными

Повторить задание 1.2, используя пакетную операцию внесения *FORALL*.

Предварительно отменить все ранее внесенные операции.

Сравнить времена выполнения PL/SQL-блоков этого задания и задания 1.2.

Требования к оформлению отчета решений по лабораторной работе

Все команды оформить в виде файла-скрипта *Фамилия_7.sql*

По каждому заданию включите в файл:

- 1) условие задания (включите в виде комментариев)
- 2) PL/SQL-код