

## Лабораторный практикум 7

### Тема: Проектирование бизнес-логики на языке *PL/SQL*

#### 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 Автоматическая инициализация генераторов уникальных значений

В предыдущих лабораторных работах необходимо было создавать генераторы последовательностей для РК-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 РК-атрибутов.

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

- 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;
- определение максимального значения идентификатора подразделения в таблице *Departments* и идентификатора сотрудника в таблице *Employees*;
- создание генераторов с учетом смещений начального значения, превышающего на 1 полученные максимальные значения.

##### Задание 2 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.

2.1 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 генератор;
- имя, фамилия сотрудника определяется как Ваше имя, фамилия + значение генератора;
- логин сотрудника определяется как Ваше имя + значение генератора;
- дата зачисления определяется как '01.01.2000' + значение генератора;
- остальные значения колонок - произвольные, но не противоречащие ограничениям целостности.

2.2 Предыдущее решение позволяет создавать простые генераторы.

Создать анонимный PL/SQL-блок, автоматически регистрирующий в БД 10000 сотрудников, учитывая условия из задания 2.1.

##### Задание 3 Обработка исключений

3.1 В решение 1-го задания изменить PL/SQL-код так, чтобы не было необходимости проверять наличие генераторов в БД.

3.2 В решение задания 2.2 добавить контроль ограничений целостности:

- внесения дубликатов по логинам сотрудников, нарушающих ограничение целостности *UNIQUE*, с выводом ошибки типа "Login Ivanov already exists";
- внесения отрицательной зарплаты с выводом ошибки типа "Salary = -10. But salary must be >= 0"

Внести изменения в PL/SQL-код, приходящий к срабатыванию указанных исключений.

#### **Задание 4 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 5 Динамические запросы**

Создать анонимный *PL/SQL*-блок, который автоматически регистрирует пользователей Oracle с учетом условий:

- имена пользователей совпадают с логинам сотрудников из таблицы employees;
- пароль генерируется как любая константа;
- пользователю 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*;

#### **Задание 6 Пакетная работа с данными**

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

Предварительно отменить все ранее внесенные операции.

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

#### **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