**Лабораторная работа**

**Разработка хранимого кода. Триггеры и пакеты**

***Цель работы: освоить технику создания и использования триггеров в базе данных. Познакомиться с техникой создания пакетов.***

***Порядок выполнения работы***

Большинство заданий этой работы выполняется без автоматической проверки. Поэтому их помещаем в отчёт и сдаём преподавателю.

1. В предыдущей лабораторной работе была создана процедура отправки («закрытия») рейса. Очевидно, она у Вас работает.

Для освоения таких интересных объектов базы данных как **триггеры** реализуем **тот же самый функционал «Отправка рейса» несколько другим способом – урежем функционал процедуры (она будет только обнулять количество проданных билетов), зато дополним её триггером (он будет отправлять данные об отправленном рейсе в архив).**

!!! Этот способ нисколько не лучше предыдущего, на практике я бы не стала его применять, но как учебная задача на разработку триггеров он вполне годится.

2. Выполним по порядку такие задания.

***Задание 2.1.*** Напишите альтернативную хранимую процедуру «Отправка рейса» (параметр «код рейса»), которая проверяет, продан ли на рейс хотя бы один билет, и в этом случае обнуляет количество проданных билетов в таблице **trips,** но в архив она данные не заносит.

***Задание 2.2.*** Напишите триггер на обновление таблицы **trips,** который перед обнулением количества проданных билетов (при работе процедуры «отправка рейса») заполнит строку таблицы **sent\_trips** из хранилища данными об отправленном рейсе. Не забудьте при реализации, что этот триггер должен срабатывать только на обнуление количества билетов при отправке рейса, и не реагировать на действия кассиров, продающих билеты (проверьте условие :new.tickets=0).

***Задание 2.3.*** Дополните триггер обработкой исключительной ситуации, которая при обнаружении нулевого количества проданных билетов отменяет запрос на обнуление и возвращает клиенту сообщение «на этот рейс не продано ни одного билета». А из хранимой процедуры эту проверку уберите. Этот альтернативный вариант мы выполняем исключительно с целью опробовать стандартную процедуру ***raise\_application\_error (***или можно сгенерировать исключительную ситуацию при помощи ***raise)***. Похожий пример размещён в файле Дополнение к лекции по триггерам – он взят из книги Тома Кайта.

Теперь мы убедились на опыте, что триггер может отменить ту команду SQL, на которую он назначен.

3. В качестве ещё одного примера использования триггеров создайте триггеры для поддержки в актуальном состоянии справочников хранилища данных – это последняя часть файла Примеры хранимого кода (Лабораторная работа 3). В этом файле приведены примеры триггеров для синхронизации таблицы a\_trips с таблицей trips, а вы создайте аналогичные триггеры для таблицы buses, которые будут выполнять нужные изменения в a\_buses.

4. Для окончательного закрепления темы «Триггеры» решите задачу 592 «Автоматизация мест» в проверяющей системе.

5. Продолжайте работу над реализацией 2-3 хранимых процедур/функций для вашей индивидуальной базы данных.

**Создайте 2 триггера для своей индивидуальной базы данных (обсудите с преподавателем).**

**Отладьте хранимый код в Apex.**

**6. Оформите реализованный вами хранимый код для индивидуальной базы данных (конечно, без триггеров) в виде пакета, ориентируясь на пример, который приведён в лекции по Пакетам Oracle. Он выложен в файле Пример создания пакета на портале.**

**Оформите единый отчёт по работам 3-5 – Содержание отчёта выложено на портале.**