***Лабораторная работа 2***

***Создание и использование представлений***

***Цель работы: освоить технику создания и использования представлений, а также закрепить навыки разработки запросов на выборку и работы с консольной утилитой SQL\*Plus***

1. **Учебные примеры - представления для кассиров и диспетчера Автовокзала (внимательно прочитать перед выполнением заданий – сами задания расположены ниже)**

Кассиры и диспетчер – это основные пользователи базы данных Автовокзал, поэтому рассмотрим примеры представлений, которые могут использоваться в приложениях для кассиров и диспетчера. Будем преследовать две важных цели создания представлений:

1. ограничить права доступа к данным для кассиров и диспетчеров на уровне базы данных, разместив в представлении только те элементы данных, которые действительно нужны этим сотрудникам;
2. упростить запросы, которые часто исполняются на сервере при работе приложений кассира и диспетчера (представления не могут заметно ускорить эти запросы – только значительно сократить их текст и упростить процесс разработки).

Основные обязанности кассиров – поиск рейсов, которые могут довести пассажиров до тех пунктов, в которые им нужно добраться, и продажа билетов на выбранный рейс, если есть свободные места на этот рейс.

Следовательно, представление кассира должно содержать всю информацию о рейсах и пунктах, через которые они проходят, в том числе, количество свободных мест (напомню, что в БД хранится только количество проданных билетов, значит, количество свободных мест надо считать).

Представление кассира **points\_trips** (пункты-рейсы):

**create view points\_trips**

**as**

**select points\_routes.cod\_point cod\_point,**

**trips.cod\_trip cod\_trip,**

**routes.name\_route name\_route,**

**trips.week\_day week\_day,**

**trips.hour||':'|| trips.minute time,**

**models.class class,**

**models.places - trips.tickets free\_places**

**from points\_routes, routes, trips, buses, models**

**where points\_routes.cod\_route=routes.cod\_route and**

**routes.cod\_route= trips.cod\_route and**

**trips.cod\_bus=buses.cod\_bus and**

**buses.cod\_model=models.cod\_model**

Обратите внимание на то, как в Oracle обозначается операция конкатенации  — ||. Преобразование из числового типа в строковый для столбцов **hour** и **minute** при этом будет выполнено автоматически.

Если представление создано без ошибок, можно получить создаваемую им виртуальную таблицу при помощи команды

select \* from points\_trips

Основная обязанность диспетчера - отправка рейсов в соответствии с расписанием. Поэтому в представлении диспетчера должны отображаться только те рейсы, которые отправляются ***сегодня,*** причем ***в порядке возрастания времени*** (т.е. в том порядке, в котором нужно их отправлять).

Представление диспетчера имеет следующий вид:

**create view trips\_now**

**as**

**select trips.cod\_trip cod\_trip,**

**routes.name\_route name\_route,**

**trips.week\_day week\_day,**

**trips.hour hour,**

**trips.minute minute,**

**trips.tickets tickets**

**from routes, trips**

**where routes.cod\_route= trips.cod\_route and**

**week\_day=to\_number(to\_char(sysdate,'d'))**

**order by hour, minute**

Здесь мы используем функцию **sysdate**, которая возвращает текущую дату, параметр **'d'** функции **to\_char** преобразует дату в номер дня недели.

***2. Порядок выполнения и задания к лабораторной работе***

База данных Автовокзал вам доступна через утилиту SQL\*Plus (и только в лабораториях кафедры АВТ), поэтому задания по этой БД будем делать в SQL\*Plus. Логины у вас должны совпадать с логинами в Apex, пароль тоже, возможно, 11111 (возможно и другое, поскольку учётные записи вы сами себе создавали в прошлом семестре). Имя ORCL в верхней строке не меняйте – просто нажмите Enter. Чтобы не вводить пароль вслепую, можно ввести имя с паролем – имя/пароль.

Если кто-то хочет использовать Oracle SQL Developer вместо SQL\*Plus – это не запрещается, но указания ориентированы на SQL\*Plus.

**Задания**

1. Убедитесь, что у вас имеются все таблицы базы данных и хранилища (хранилище в этой работе не понадобится, но в следующих работах будет использоваться). Проверьте, что таблицы оперативной базы данных заполнены.
2. Создайте в SQL\*Plus представления, приведённые в качестве примеров выше. Убедитесь, что они созданы без ошибок.

3. На их основе напишите запросы:

А. Названия всех остановок рейса с заданным кодом (для этого надо соединить представление с таблицей points). Расположить остановки в порядке следования

Б. Время отправления, название маршрута, количество остановок в этом маршруте и количество свободных мест для рейса с заданным кодом (возьмите любой, какой у вас есть)

В. Все рейсы в ближайшую субботу, проходящие через Череповец (или любой другой пункт), на которые сейчас есть свободные места. Указать название маршрута и время отправления

Г. Все рейсы по маршруту Вологда-Череповец (или любому другому), которые отправляются СЕГОДНЯ – указать только время отправления

Д. Список автобусов с указанием кода и номера автобуса, его модели и количества мест, которые СЕГОДНЯ СВОБОДНЫ (не задействованы ни в одном из рейсов, которые сегодня отправляются).

Убедитесь в работоспособности запросов в SQL\*Plus - к сожалению, наша проверяющая система представления не поддерживает, поэтому запросы сдаются преподавателю и помещаются в Отчёт по лабораторной работе (со скриншотами).

Представления не удаляйте, могут пригодиться при выполнении следующих лабораторных работ.

2. Создайте несколько представлений (минимум 2) для своей предметной области, ориентируясь на пользователей вашей базы данных и наиболее востребованные запросы. Предложите и реализуйте несколько актуальных запросов на выборку, которые могут быть существенно упрощены, если в них использовать созданные вами представления (минимум четыре), обязательно отладьте запросы. Поместите представления и запросы в отчёт вместе со скриншотами.

В отчёт поместите все представления и запросы (со скриншотами), написанные в ходе выполнения работы, включая БД Автовокзал и вашу индивидуальную БД.