Автоматизация работы автосервиса.

В рамках данного цикла лабораторных работ необходимо автоматизировать работу автосервиса. Для этого в рамках базы данных PostgreSQL необходимо создать объекты в схеме вашего пользователя (логин и пароль пользователя для доступа к базе данных студент должен получить у преподавателя) и написать клиентское приложение на базе компонентов ADO.NET или JDBC.

Автосервис специализируется на ремонте и техническом обслуживании автомобилей. При этом цена услуг для отечественных и зарубежных отличается. В сервисе есть ограниченный круг клиентов для которых производятся работы. Каждый автомобиль клиента характеризуется государственным номером, цветом и маркой. Периодически для автомобиля мастер выполняет работу, данные о которой заносятся в журнал работ. Одну услугу оказывает один мастер.

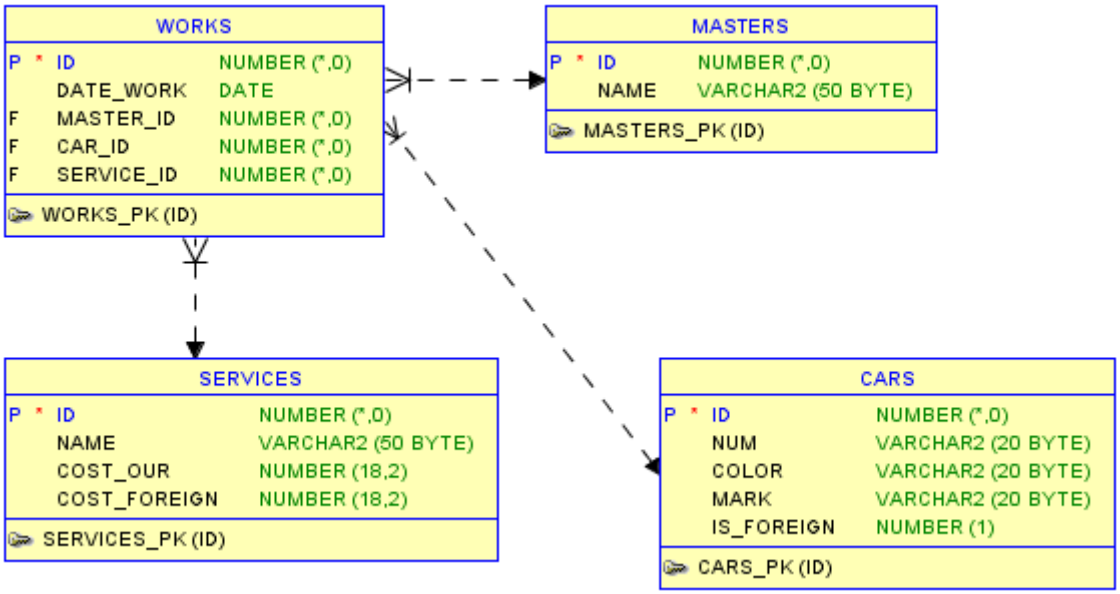
База данных должна удовлетворять следующим требованиям:

1. Контроль целостности данных, используя механизм связей
2. Операции модификации групп данных и данных в связанных таблицах должны быть выполнены в рамках транзакций.
3. Логика работы приложения должна контролироваться триггерами. В частности:
    - Не позволяет добавить автомобиль с уже существующим номером
    - Не позволяет добавить мастера, если их уже больше 10
    - Не позволяет дать работу мастеру, если он в этот день уже выполнил больше одной работы
4. Все операции вычисления различных показателей (из требований к клиентскому приложению) должны реализовываться хранимыми процедурами.

Требования к клиентскому приложению:

1. Необходимо реализовать интерфейсы для ввода, модификации и удаления справочников:
    - Мастеров;
    - Автомобилей;
    - Услуг.
2. В главном окне приложения должен быть реализован интерфейс, позволяющий оператору назначать работы из перечня услуг мастерам автосервиса для обслуживания имеющейся базы клиентов с возможностью указания даты проведения работы.
3. Необходимо реализовать возможность просмотра оператором следующих показателей:
    - Общая стоимость обслуживания отечественных и импортных автомобилей (с возможностью фильтрации по датам оказания услуги).
    - Пять мастеров, которые в заданном месяце выполнили наибольшее число работ для разных автомобилей.

Работа №1. Создание базы данных.



1. Запустить pgAdmin или psql.
2. Создать соединение, используя логин и пароль.
3. Изменить пароль вашего пользователя на свой, используя команду

```
ALTER USER your_user_name IDENTIFIED BY new_password;
```

4. Создать необходимые таблицы.

Имя таблицы	**Имя колонки**	**Расшифровка**
cars		Таблица автомобилей
	id	Идентификатор записи
	num	Номер
	color	Цвет
	mark	Марка
	is_foreign	Иностранная (1) или нет (0)
masters		Таблица мастеров
	id	Идентификатор записи
	name	Имя
services		Таблица услуг
	iid	Идентификатор записи
	name	Наименование
	cost_our	Стоимость для отечественной машины
	cost_foreign	Стоимость для иномарки
works		Таблица работ
	id	Идентификатор записи
	date_work	Дата работы
	master_id	Мастер
	car_id	Машина
	service_id	Услуга

5. Создать связи между таблицами:

Название	**Primary Key**	**Foreign Key**
fk_works_cars	cars.id	works.car_id
fk_works_masters	masters.id	works.master_id
fk_works_services	services.id	works.service_id

6. Создать Backup базы и запомнить место его расположения.
7. Удалить базу с сервера
8. Восстановить базу из Backup базы
9. Для всех таблиц реализовать автоматическое заполнение первичного ключа при вставке данных (автоинкремент).

Правила выполнения работ

1. Засчитываются только целиком сделанные работы
2. Преподаватель оставляет за собой право на выдачу дополнительных заданий, в случае возникновения подозрения на стороннюю помощь при выполнении работы.
3. Посещения занятий ОБЯЗАТЕЛЬНО, кроме случаев оговоренных лично с преподавателем.
4. Если студент сдает работу, то он должен знать, как он ее сделал.
5. Все работы можно делать дома, но это не является основанием для пропусков занятий.
6. Перед уходом студент должен удалить с сервера свою базу. Рекомендую не забывать делать Backup в осознанное место, что бы на следующем занятии не пришлось делать работу с начала.
7. К каждому заданию прилагается список операторов, ключевых.

Работа №2. Язык SQL-DML

Выборка данных

- однотабличная выборка
    1. Вычислить общее услуг и общую сумму стоимости для отечественных и импортных автомобилей
    2. Вывести все работы за последний месяц
- соединение таблиц (join)
    1. Вывести стоимость обслуживания каждого автомобиля за последний год, включая автомобили, которые не обслуживались, упорядочив по убыванию стоимости
- для реализации проекта
    1. Вычислить общую стоимость обслуживания отечественных и импортных автомобилей за все время существования сервиса

Hints: select, count, join, where, in, exists, order by, group by, having

Вставка данных

- однотабличная вставка
    1. Добавить новую услугу
    2. Добавить работу по новой услуге из п1
- многотабличная вставка в рамках транзакции
    1. Добавить в рамках транзакции новый автомобиль и услугу и провести работу для этого автомобиля по новой услуге.

Hints: insert, where, in, exists, commit, rollback

Удаление данных

- удаление по фильтру и удаление из связанных таблиц
    1. Удалить статью автомобиль и все работы по нему
- удаление в рамках транзакции
    1. Удалить в рамках транзакции услуги, которые оказывались только заданным мастером. Удалить работы по таким услугам этого мастера.
    2. то же, что и п1, но еще удалить мастера и транзакцию откатить

Hints: delete, where, in, exists, commit, rollback

Модификация данных

- модификация по фильтру
    1. Увеличить стоимость всех услуг на 15%
- модификация в рамках транзакции
    1. В рамках транзакции в таблице услуг увеличить цену услуги, оказанной последней, на 10.00
    2. то же, что и п1, но транзакцию откатить

Hints: update, where, in, exists, commit, rollback

Дополнительные задания

Работа №3. Представления, хр. процедуры, триггера и курсоры.

Представления

1. Создать представление, отображающее все услуги, по которым за все время сумма превысила некоторую границу
2. Создать представление, отображающее общий доход мастеров за последний год, включая мастеров, которые ничего не получили

Hints: select, where, count, max, group by, having, like, create view, drop view

Хранимые процедуры

- без параметров
    1. Создать хранимую процедуру, выводящую все автомобили и среднюю стоимость услуги для них, включая автомобили, по которым не производились работы
- с входными параметрами
    1. Создать хранимую процедуру, имеющую два параметра «услуга» и «машина». Она должна возвращать общую стоимость этой услуги для машины за все время существования автосервиса и количество проводимых по этой услуге работ.
- с выходными параметрами
    1. Создать хранимую процедуру с входными параметрами «мастер1» и «мастер2» и выводящую количество услуг, которые оказывали оба мастера.

Hints: select, where, count, max, group by, having, create procedure, drop procedure

Триггера

- Триггера на вставку
    1. Создать триггер, который не позволяет добавить автомобиль с уже существующим номером
- Триггера на модификацию
    1. Создать триггер, который не позволяет изменить дату работы более чем на один день
- Триггера на удаление
    1. Создать триггер, который при удалении автомобиля, в случае если по нему были какие-то работы, откатывает транзакцию

Hints: select, where, in, exists, join, commit, rollback, create trigger, drop trigger

Курсоры

- Хранимая процедура для распределения премии мастерам за период

Необходимо реализовать хранимую процедуру, рассчитывающую зарплату+премию, полученную мастером за текущий месяц. Хранимая процедура должна иметь два входных параметра: мастера, для которого производим расчет и его зарплата и один выходной, в котором возвращать рассчитанную сумму.

Предлагаемый алгоритм: создаем курсор, который пробегает по работам для данного мастера.

Для каждой строки проверяем сколько раз для данной машины в заданном периоде оказывались услуги. Если меньше 3-х, то премия составляет 5% от з/п, если больше, то 7% (по каждой машине). Суммируем полученный результат в некоторой переменной, значение которой по окончании работы курсора будет выдано в качестве выходного параметра.

Hints: CURSOR, %NOTFOUND, FETCH