# Автоматизация работы магазина.

В рамках данного цикла лабораторных работ необходимо автоматизировать работу магазина. 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.

Магазин занимается продажей со склада товаров оптом и в розницу. При этом менеджеры следят за тем, чтобы на складе постоянно были товары в наличии. Товары закупаются у посредника по установленной цене. В дальнейшем магазин продает их по другой цене. Продажи записываются в журнал продаж, где отмечается дата продажи, количество проданного товара и цена за единицу. Полученные от разницы в покупке и продаже товара деньги магазин тратит на различные статьи расхода. Данные о расходах отмечаются в журнале расходов и характеризуются датой, суммой и самой статьей расхода. В частности, каждый месяц магазин выплачивает работникам зарплату.

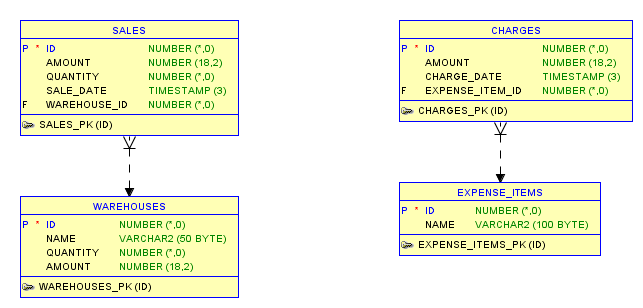
**База данных должна удовлетворять следующим требованиям:**

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, должны быть реализованы следующие ограничения:
   * Не позволяет добавить в таблицу товаров товар, у которой не задана сумма
   * Не позволяет добавлять расход, с суммой большей заданной
   * Не позволяет изменять данные в таблице продаж задним числом от сегодняшней даты
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:
   * Товаров;
   * Статей расходов.
2. В главном окне приложения должны быть реализованы интерфейсы:
   * Журнала продаж с возможностью задавать товар, дату продажи, количество и цену за единицу.
   * Журнал расходов с возможностью задавать дату, сумму и статью расхода.
3. Необходимо реализовать возможность просмотра оператором следующих показателей:
   * Прибыль магазина за заданный месяц.
   * Пять самых доходных товаров за заданный интервал дат.

## 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*;

1. Создать необходимые таблицы.

|  |  |  |
| --- | --- | --- |
| **Имя таблицы** | **Имя колонки** | **Расшифровка** |
| warehouses |  | Таблица склада товаров |
|  | id | Идентификатор записи |
|  | name | Наименование |
|  | quantity | Количество товара |
|  | amount | Стоимость одной единицы товара |
| sales |  | Таблица продаж |
|  | id | Идентификатор записи |
|  | amount | Стоимость одной единицы товара |
|  | quantity | Количество товара |
|  | sale\_date | Дата продажи |
|  | warehouse\_id | Товар |
| charges |  | Таблица расходов |
|  | Id | Идентификатор записи |
|  | amount | Сумма |
|  | charge\_date | Дата |
|  | expense\_item\_id | Статья расхода |
| expense\_items |  | Таблица статей расхода |
|  | id | Идентификатор записи |
|  | name | Наименование статьи |

1. Создать связи между таблицами:

|  |  |  |
| --- | --- | --- |
| **Название** | **Primary Key** | **Foreign Key** |
| fk\_sales\_warehouses | warehouses.id | sales.warehouse\_id |
| fk\_charges\_expense\_items | expense\_items.id | charges.expense\_item\_id |

1. Создать Backup базы и запомнить место его расположения.
2. Удалить базу с сервера
3. Восстановить базу из Backup базы
4. 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 слов.

## Работа №2. Язык SQL-DML

### Выборка данных

* однотабличная выборка
  1. Вычислить общее количество проданных товаров и сумму за все время работы магазина
  2. Вывести все расходы за последний месяц
* соединение таблиц (join)
  1. Вывести все товары, которые в данный момент есть на складе и по которым за последний месяц были продажи
  2. Вывести стоимость каждой статьи расхода за последний год, упорядочив по убыванию стоимости
* для реализации проекта
  1. Вычислить прибыль магазина за последний месяц
  2. Пять самых доходных товаров за все время работы магазина.

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

### Вставка данных

* однотабличная вставка
  1. Добавить новую статью расхода
  2. Добавить в таблицу расходов расход по статье из п1
* многотабличная вставка в рамках транзакции
  1. Добавить в рамках транзакции новый товар на склад и добавить в журнал запись о его продаже. Если количество товара на складе меньше, чем пытаются продать, транзакцию откатить.

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

### Удаление данных

* удаление по фильтру и удаление из связанных таблиц
  1. Удалить статью расхода и все расходы по ней
* удаление в рамках транзакции
  1. Удалить в рамках транзакции продажу определенного товара, заданного по наименованию, с наименьшим количеством этого товара
  2. то же, что и п1, но транзакцию откатить

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

### Модификация данных

* модификация по фильтру
  1. Увеличить цену всех товаров на складе на 10%
* модификация в рамках транзакции
  1. В рамках транзакции в таблице продаж увеличить цену последней продажи определенного товара на 5.00
  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 статьи расхода, по которым за все время сумма превысила некоторую границу
2. Создать представление, отображающее общее количество расходов за последний месяц в разрезе статей расходов

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

### Хранимые процедуры

* без параметров

1. Создать хранимую процедуру, выводящую все товары и среднюю стоимость их продаж за все время

* с входными параметрами

1. Создать хранимую процедуру, имеющую два параметра «товар1» и «товар2». Она должна возвращать даты продаж, в которых эти два товара продавались одновременно.

* с выходными параметрами

1. Создать хранимую процедуру с двумя входными параметрами «дата1» и «дата2» и двумя выходными параметрами, возвращающими общую сумму дохода и расхода за данный период с «дата1» по «дата2».

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

### Триггера

* Триггера на вставку

1. Создать триггер, который не позволяет добавлять расход, с суммой большей заданной

* Триггера на модификацию

1. Создать триггер, который не позволяет изменять данные в таблице продаж задним числом от сегодняшней даты

* Триггера на удаление

1. Создать триггер, который при удалении расхода в случае, если расход был более чем месяц назад, откатывает транзакцию

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

### Курсоры

* Хранимая процедура для расчета суммы предполагаемой прибыли на ближайший месяц

Необходимо реализовать хранимую процедуру, рассчитывающую предполагаемую прибыль на ближайший месяц. Хранимая процедура должна иметь один выходной параметр, в котором возвращать размер прибыли.

Предлагаемый алгоритм: заводим две переменные ср\_расх, ср\_дох типа decimal; создаем курсор, который пробегает по статьям расхода за последние 4 месяца. Для каждой строки проверяем условие, если дата расхода попадает в последний месяц, то суммируем в ср\_расх с коэффициентов 1, если в предыдущий месяц, то с коэффициентом ½, в противном случае, ¼. Аналогично, создаем курсор по продажам за последние 4 месяца и так же с коэффициентами 1, ½, ¼ наращиваем сумму в переменную ср\_дох.

Некоторой переменной, значение которой по окончании работы курсора будет выдано в качестве выходного параметра, присваиваем значение ср\_дох - ср\_расх.

Hints: CURSOR, %NOTFOUND, FETCH