

Лабораторная работа №7

Хранимый код, триггеры, события

Теоретический раздел	2
Хранимый код	2
Триггеры	3
События	4
Задания к лабораторной работе	6
<i>Общие правила выполнения заданий</i>	6
Вариант 1. Проект ПОСТАВКА ТОВАРОВ	7
Вариант 2. Проект РОЗНИЧНАЯ ТОРГОВЛЯ	8
Вариант 3. Проект БАНКОВСКИЕ ВКЛАДЫ	9
Вариант 4. Проект АУДИТ	10
Вариант 5. Проект ФАБРИКА	11
Вариант 6. Проект ТУРАГЕНСТВО	12
Вариант 7. Проект ТОРГОВЛЯ	13
Вариант 8. Проект КОСМЕТИЧЕСКАЯ ПРОДУКЦИЯ	14
Вариант 9. Проект ПОДПИСНЫЕ ИЗДАНИЯ	15
Вариант 10. Проект ПОЛИКЛИНИКА	15
Вариант 11. Проект СПЕЦОДЕЖДА	17
Вариант 12. Проект ГАИ	18
Вариант 13. Проект ЖЭС	19
Вариант 14. Проект ОБЩЕПИТ	20
Вариант 15. Проект ИНТЕРНЕТ-ПРОДАЖИ	21
Вариант 16. Проект БАНКОМАТЫ	22
Вариант 17. Проект РЕМОНТ БЫТОВОЙ ТЕХНИКИ	23
Вариант 18. Проект МЕБЕЛЬ	24
Вариант 19. Проект ТИПОГРАФИЯ	25
Вариант 20. Проект АПТЕЧНЫЙ СКЛАД	26
Вариант 21. Проект ДИПЛОМНЫЙ ПРОЕКТ	27
Вариант 22. Проект НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ	27
Вариант 23. Проект КОММЕРЧЕСКАЯ ФИРМА	29
Задание 24. Проект ИЗДАТЕЛЬСКИЙ ЦЕНТР	30
Вариант 25. Проект АВТОЗАПРАВКИ	31
Вариант 26. Проект БИБЛИОТЕКА	32
Вариант 27. Проект РОСТОВЗЕЛЕНСТРОЙ	33
Вариант 28. Проект АВИАЛИНИИ	34
Вариант 29. Проект ПРОКАТ ВЕЛОСИПЕДОВ ФИТНЕС-КЛУБА	35
Вариант 30. ПРОЕКТ КАНЦЕЛЯРСКАЯ ПРОДУКЦИЯ	36

Теоретический раздел

Хранимый код

Хранимая процедура (функция) – это последовательность операций, хранящаяся как единое целое в каталоге базы данных на сервере. Приложения могут вызывать и запускать хранимые процедуры. Таким образом, вместо того, чтобы хранить часто используемый запрос, клиенты могут ссылаться на соответствующую хранимую хранимый код. Это обеспечивает лучшую производительность, поскольку данный запрос должен анализироваться только однажды и уменьшается трафик между сервером и клиентом. Концептуальный уровень можно также повысить за счет создания на сервере библиотеки функций.

```
CREATE PROCEDURE имя_процедуры ([параметр_процедуры[, ...]])  
[характеристика ...] тело_подпрограммы  
  
CREATE FUNCTION имя_функции ([параметр_функции[, ...]])  
RETURNS тип  
[характеристика ...] тело_подпрограммы  
  
параметр_процедуры:  
[ IN | OUT | INOUT ] имя_параметра тип  
параметр_функции:  
имя_параметра тип  
  
тип:  
Любой тип данных MySQL  
  
характеристика:  
LANGUAGE SQL  
| [NOT] DETERMINISTIC  
| { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }  
| SQL SECURITY { DEFINER | INVOKER }  
| COMMENT 'string'  
  
тело_подпрограммы:  
Правильное SQL выражение.
```

Хранимые процедуры используются, когда приложению необходимо выполнить сложную задачу с использованием информации реляционной базы данных. Примером может служить заявка на получение ссуды для определения платежеспособности клиента и его кредитоспособности. Чтобы проверить платежеспособность клиента, кредитный специалист сравнивает среднемесячный доход клиента с его ежемесячной суммой снятия средств со счета за 24-месячный период. Чтобы проверить кредитоспособность, кредитный специалист отправляет идентификатор клиента или номер социального страхования на веб-сайт кредитной отчетности.

Оба вышеуказанных действия сложны и трудны для выполнения с помощью основных команд SQL. Кроме того, процесс утверждения ссуды клиента может выполняться в разное время для разных клиентов (т. е. одно и то же действие повторяется несколько раз), но с каждым действием связана разная информация о клиенте.

Вышеуказанная ситуация является идеальным примером того, когда должны быть сохранены хранимые процессы, т. е. При выполнении сложного действия с использованием комбинации SQL, процедурного языка / языка структурированных запросов (PL / SQL) или внешнего языка программирования. Во-вторых, одно и то же действие выполняется многократно, и единственными изменениями являются параметры или данные, подлежащие обработке.

Хранимые процедуры обычно обеспечивают выигрыш в производительности по сравнению с написанием кода приложения по следующим двум причинам:

- Они не требуют дополнительного межпрограммного взаимодействия между базой данных и внешним приложением.
- Не нужно компилировать и выполнять для каждого экземпляра, так как сохраненные процессы компилируются только один раз.

Хранимые процедуры хранятся как часть словаря данных базы данных, а не как приложение, которое ссылается на базу данных.

Триггеры

Триггер — это хранимая процедура особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено наступлением определенного события (действием) — по сути добавлением INSERT или удалением DELETE строки в заданной таблице, или модификации UPDATE данных в определенном столбце заданной таблицы реляционной базы данных. Триггеры применяются для обеспечения целостности данных и реализации сложной бизнес-логики. Триггер запускается сервером автоматически при попытке изменения данных в таблице, с которой он связан. Все производимые им модификации данных рассматриваются как выполняемые в транзакции, в которой выполнено действие, *вызвавшее срабатывание триггера*.

CREATE

```
[DEFINER = {user | CURRENT_USER}]  
TRIGGER trigger_name trigger_time trigger_event  
ON tbl_name FOR EACH ROW trigger_stmt
```

В параметре *trigger_name* задается имя триггера. В параметре *tbl_name* задается имя таблицы, для которой создается триггер (также поддерживаются триггеры для представлений, на что указывает наличие параметра *view_name*.)

Также можно задать тип триггера (*trigger_time*) с помощью двух дополнительных параметров: **AFTER** и **BEFORE**. Триггеры типа **AFTER** вызываются после выполнения действия, запускающего триггер, а триггеры типа **BEFORE** выполняются до действия, запускающего триггер.

Параметры **INSERT**, **UPDATE** и **DELETE** задают действие триггера (*trigger_event*). Под действием триггера имеется в виду инструкция SQL, которая запускает триггер.

При запуске триггера исполняющее ядро SQL создает и заполняет две структуры данных, которые работают подобно записям. Это псевдозаписи **NEW** и **OLD** (приставка «псевдо» объясняется тем, что они обладают не всеми свойствами настоящих записей PL/SQL). **OLD** хранит начальные значения записи, обрабатываемой триггером, а **NEW** - новые значения.

При работе с псевдозаписями NEW и OLD необходимо учитывать несколько правил:

- Для триггеров, относящихся к командам INSERT, структура OLD не содержит никаких данных, «старого» набора значений не существует.
- Для триггеров, относящихся к командам UPDATE, заполняются обе структуры: OLD и NEW. OLD содержит исходные значения (до обновления), а NEW - те значения, которые получит строка после выполнения обновления.
- Для триггеров, относящихся к командам DELETE, структура NEW не содержит никаких данных, ведь речь идет об удалении записи.
- Изменение значений полей структуры OLD запрещено, попытка такого изменения приведет к возникновению ошибки. Изменение значений полей структуры NEW допустимо.
- Структура NEW или OLD не может передаваться как параметр типа запись в хранимый код или функцию, вызываемую внутри триггера. Можно передавать только отдельные поля псевдозаписей.

Триггеры могут использоваться для следующих операций: создания журнала логов действий в таблицах базы данных; реализации бизнес-правил; принудительного обеспечения ссылочной целостности.

События

События MySQL - задачи, которые выполняются согласно плану. Следовательно, мы иногда обращаемся к ним как к планируемым событиям. Когда Вы создаете событие, Вы создаете именованный объект базы данных, содержащий одну или большее количество инструкций SQL, которые будут выполнены в одном или более регулярных интервалах, начиная и заканчивая в специфическую дату и время.

```
CREATE EVENT [IF NOT EXISTS] event_name
ON SCHEDULE schedule
[ON COMPLETION [NOT] PRESERVE] [ENABLE | DISABLE]
[COMMENT 'comment']
DO sql_statement;
```

schedule:

```
AT timestamp [+ INTERVAL interval]
| EVERY interval [STARTS timestamp]
[ENDS timestamp]
```

interval:

```
quantity {YEAR | QUARTER | MONTH | DAY | HOUR |
          MINUTE | WEEK | SECOND | YEAR_MONTH | DAY_HOUR |
          DAY_MINUTE | DAY_SECOND | HOUR_MINUTE | HOUR_SECOND |
          MINUTE_SECOND}
```

- После **CREATE EVENT** необходимо указать имя события. Оно должно быть уникальным в структуре базы данных;
- После оператора **ON SCHEDULE** вы задаете график. Если событие является единичным, используется синтаксис: **AT timestamp [+ INTERVAL]**. Если событие периодическое, используется условие **EVERY: EVERY interval STARTS timestamp [+INTERVAL] ENDS timestamp [+INTERVAL]**;
- Вы размещаете оператор SQL после ключевого слова **DO**. Стоит отметить, что вы можете вызвать хранимую хранимый код внутри тела события. В случае если у вас есть составные операторы SQL, вы можете заключить их в блок **BEGIN END**.

Для выполнения всех запланированных событий MySQL использует специальный поток, который называется «поток планировщика событий». По умолчанию поток планировщика событий отключен. Чтобы включить его и запустить поток планировщика событий, необходимо выполнить следующую команду:

```
SET GLOBAL event_scheduler = ON
```

Создание события производится аналогично созданию других объектов базы данных: таких как хранимые процедуры и триггеры.

Событие представляет собой проименованный объект, который содержит состояние SQL. Хранимая процедура выполняется только при прямом вызове; триггер выполняется, когда происходит событие, связанное с таблицей: например, добавление, обновление, или удаление. В то время как само событие может выполняться один или несколько раз с регулярным интервалом.

Задания к лабораторной работе

Общие правила выполнения заданий

Требования к ПО:

- Рекомендуется установка в качестве среды разработки openServer.
- В качестве IDE могут быть использованы как проприетарные (phpStorm и др.), так и свободно распространяемые инструменты (Sublime Text, NetBeans, Visual Studio Code и др.).
- Хранимый код (триггеры и события) могут быть реализованы как во вкладках phpMyAdmin так и в Workbench.

Требования и рекомендации по выполнению:

- Весь код лабораторной работы выполняется на сервере.
- Необходимо протестировать работу кода в случае, если база данных находится в несогласованном состоянии или наступает одно из событий.
- При вызове хранимого кода на php входные параметры задаются в соответствующих полях формы.
- В лабораторной работе может быть изменена структура базы данных путем добавления в отношения новых атрибутов для отражения «флагов состояния».

Правила оценивания:

- Лабораторная работа оценивается из 100%. В случае пересдачи максимально возможный балл – 60%.
- В случае нарушения дедлайнов лабораторная работа оценивается в 1%.

Вариант 1. Проект ПОСТАВКА ТОВАРОВ

1. Разработайте хранимый код для фиксации несоответствия объема отгруженного товара и плановой поставки. Выполните согласование данных, отразите разницу (отклонение).
2. Разработайте механизмы логов для фиксации всех изменений в базе данных.
3. Разработайте хранимый код для запроса: Вывести список товаров, заказанных компанией X. Запустите ее на php. Название компании должно быть выбрано в форме.
4. Запланируйте событие, которое бы еженедельно рассчитывало величину отклонения.

Вариант 2. Проект РОЗНИЧНАЯ ТОРГОВЛЯ

1. Изменить структуру таблицы базы данных для учета изменения количества товара на складе. Разработайте хранимый код для согласования данных и отображения текущего количества товара с учетом продаж.
2. Подготовьте механизмы автоматического изменения количества товара на складе в случае новой продажи или возврата товара.
3. Разработайте хранимый код для запроса: Вывести список товаров, поставляемых поставщиком X. Запустите ее на php. Название поставщика должно быть получено из формы.
4. Запланируйте событие, которое бы раз в неделю помечало товар, поступивший более года назад как Уцененный, а поступивший за последний месяц как Новинка.

Вариант 3. Проект БАНКОВСКИЕ ВКЛАДЫ

1. Изменить структуру таблицы базы данных для учета суммы счетов на вкладе. Разработайте хранимый код для согласования данных и отображения текущей суммы.
2. Подготовьте механизмы изменения Итоговой суммы по счетам в случае открытия нового счета, его закрытия или изменения суммы на счете.
3. Разработайте хранимый код, для отражения перечня вкладов, открытых клиентом, имя которого вводится в форму. Вывести результат на веб-странице.
4. Запланируйте событие, которые бы каждый месяц начисляло проценты по вкладу пока вклад не закрыт.

Вариант 4. Проект АУДИТ

1. Изменить структуру таблицы базы данных для учета количества выполненных работ каждым сотрудником. Разработайте хранимый код для согласования данных и отображения текущего количества выполненных работ каждым сотрудником.
2. Подготовьте механизмы автоматического изменения количества выполненных работ в зависимости от запросов пользователей.
3. Разработайте хранимый код для запроса: Вывести все работы, которые выполнял сотрудник Х. Запустите ее на php. Имя сотрудника должно быть получено из формы.
4. Запланируйте событие, которое бы раз в месяц перемещало всех сотрудников с флагом «Удалено» в таблицу Архив.

Вариант 5. Проект ФАБРИКА

1. Изменить структуру таблицы базы данных для учета максимальной стоимости товара. Разработайте хранимый код для согласования данных и отображения максимальной стоимости модели для каждого вида товара.
2. Подготовьте механизмы изменения значения максимальной стоимости товара при изменениях в таблице модели.
3. Разработайте хранимый код для запроса: Вывести модели товара категории X. Запустите ее на php. Вид товара должен быть получен из формы.
4. Запланируйте событие, которые бы раз в квартал увеличивало бы стоимость моделей на 3%. Обеспечьте соответствующие изменения в таблице Товары.

Вариант 6. Проект ТУРАГЕНСТВО

1. Изменить структуру таблицы базы данных для учета количества маршрутов, реализуемых в каждой стране. Разработайте хранимый код для согласования данных и отображения текущего количества маршрутов для каждой страны.
2. Подготовьте механизмы автоматического изменения количества маршрутов по стране в случае удаления или добавления нового маршрута.
3. Разработайте хранимый код для запроса: Вывести список стран, в которые были проданы туры за определенный период. Запустите ее на php. Период указывается в полях формы.
4. Запланируйте событие, которое бы каждый квартал помечало бы самые популярные маршруты как «Популярные», а маршруты с нулевыми продажами как «Низкий спрос»

Вариант 7. Проект ТОРГОВЛЯ

1. Разработайте хранимый код для расчета объема реализации товара в день по отделам. Выполните согласование данных.
2. Подготовьте механизмы изменения объема реализации товара при осуществлении новой продажи.
3. Разработайте хранимый код для запроса: Вывести товары, проданные в указанный период. Запустите ее на php. Период задается в виде интервала дат и вводится в соответствующие поля формы.
4. Запланируйте событие, которые бы раз в квартал не востребованные товары списывала с продажи.

Вариант 8. Проект КОСМЕТИЧЕСКАЯ ПРОДУКЦИЯ

1. Изменить структуру таблицы базы данных для учета средней стоимости товаров каждого бренда. Разработайте хранимый код для согласования данных и отображения средней стоимости товаров по каждому бренду.
2. Подготовьте механизмы изменения значения средней стоимости товаров бренда при изменениях в таблице бренды.
3. Разработайте хранимый код для запроса: Вывести товары бренда X. Запустите ее на php. Бренд выбирается из формы.
4. Запланируйте событие, которые бы ежегодно в конце декабря все заказа, с момента которых прошел год, отправляло в таблицу Архив.

Вариант 9. Проект ПОДПИСНЫЕ ИЗДАНИЯ

1. Изменить структуру таблицы базы данных для учета количества подписок, осуществляемых получателями. Разработайте хранимый код для согласования данных и отображения текущего количества подписок каждого получателя.
2. Подготовьте механизмы автоматического изменения количества подписок в случае удаления или добавления новой подписки.
3. Разработайте хранимый код для запроса: Найти перечень изданий, на которые подписан Пользователь. Запустите ее на php. Пользователь выбирается в форме.
4. Запланируйте событие, которые бы ежемесячно для пользователей, у которых более 5 подписок формировало скидку в размере 5%, а для пользователей с 10 и более подписками – 10%. Сохраните величину скидку в соответствующем столбце базы данных.

Вариант 10. Проект ПОЛИКЛИНИКА

1. Изменить структуру таблицы базы данных для учета суммарной стоимости посещения каждым клиентом поликлиники. Разработайте хранимый код для согласования данных и отображения суммы, потраченной каждым клиентом при посещении поликлиники.
2. Подготовьте механизмы автоматического изменения суммарной стоимости посещения каждым клиентом поликлиники (при новом приеме, отмене приема, изменении стоимости приема)
3. Разработайте хранимый код для запроса: Узнать, к каким врачам на прием приходил клиент Х. Запустите ее на php. Клиент выбирается в соответствующем поле формы.
4. Запланируйте событие, которые бы ежедневно проверяло сумму, накопленную при посещении поликлиники. При достижении 10000 – выставить скидку 3%, при 15000 – 5%, при 20000 – 7%.

Вариант 11. Проект СПЕЦОДЕЖДА

1. Разработайте хранимый код для фиксации необходимости возврата спецодежды. Если прошел месяц с даты выдачи – пометить запись соответствующим флагом. В случае если месяц еще не прошел – отразить количество дней до возврата.
2. Подготовьте механизмы фиксации всех изменений в таблице Получение.
3. Разработайте хранимый код для запроса: Узнать начальника цеха, в котором работает сотрудник X. Сотрудник выбирается в соответствующем поле формы. Протестируйте на php.
4. Запланируйте событие, которые бы ежедневно обновляло количество дней до возврата спецодежды.

Вариант 12. Проект ГАИ

1. Изменить структуру таблицы базы данных для учета количества штрафов для каждого водителя. Разработайте хранимый код для согласования данных и отображения текущего количества штрафов.
2. Подготовьте механизмы увеличения количества штрафов в случае нового получения взыскания.
3. Разработайте хранимый код для поиска информации о водителе по номеру автомобиля, введенного в форму. Протестируйте на php.
4. Запланируйте событие, которое бы ежедневно проверяла своевременность оплаты штрафа (срок – 3 месяца) и неоплаченные вовремя помечало бы флагом «Просрочен».

Вариант 13. Проект ЖЭС

1. Разработайте хранимый код для таблицы Оплата за услугу, которая бы отмечала своевременность оплаты услуги и корректировала дату оплаты при необходимости.
2. Подготовьте механизмы фиксации в таблице Услуги даты изменения тарифа и процент его увеличения.
3. Разработайте хранимый код, для отражения перечня услуг, которые заключены по определенном лицевому счету, указанном в форме php. Вывести результат на веб-странице.
4. Запланируйте событие, которые бы каждый месяц контролировала бы своевременность оплаты услуг.

Вариант 14. Проект ОБЩЕПИТ

1. Разработайте хранимый код, позволяющий подсчитать калорийность блюда и отметить блюда с калорийностью менее 300 ккал как «Низкокалорийные»
2. Подготовьте механизмы логов для фиксации изменений в базе данных.
3. Разработайте хранимый код для поиска продуктов в определенном диапазоне цен, заданным в полях формы. Протестировать выполнение на php.
4. Запланируйте событие, которое бы каждый месяц отмечало одно случайное блюдо как Блюдо месяца.

Вариант 15. Проект ИНТЕРНЕТ-ПРОДАЖИ

1. Разработайте хранимый код для фиксации в таблице Интернет-магазин суммарной выручки за квартал.
2. Обеспечьте механизмы изменения выручки интернет-магазина после доставки заказа.
3. Разработайте хранимый код, для вывода списка товаров интернет-магазина, выбранного в форме html.
4. Запланируйте событие, которое бы каждый квартал данные о выручке интернет-магазина фиксировала в таблице Архив и очищало столбец «Выручка за текущий квартал».

Вариант 16. Проект БАНКОМАТЫ

1. Разработайте хранимый код для согласования количества банкоматов определенного банка. Добавьте новый атрибут в таблицу «Банки» с названием «Банкоматы». Обновите количество банкоматов актуально текущему состоянию базы данных.
2. Подготовьте механизмы изменения количества банкоматов в таблице «Банки» в случае их открытия/закрытия.
3. Разработайте хранимый код для поиска банкоматов, расположенных на определенной улице, выбранной в html-форме.
4. Запланируйте событие, которое бы в конце каждого месяца проверяло бы сумму снятых наличных. В случае, если эта сумма больше 100000 – ставило флаг – «Превышен процент снятия наличных»

Вариант 17. Проект РЕМОНТ БЫТОВОЙ ТЕХНИКИ

1. Изменить структуру таблицы базы данных для учета количества заказов, исполненных каждым сотрудником. Разработайте хранимый код для согласования данных адекватно текущему состоянию базы данных.
2. Обеспечьте механизмы изменения количества исполненных заказов каждым сотрудником при поступлении новых заказов, удалении или назначении других сотрудников.
3. Разработайте хранимый код для запроса: Выведите перечень товаров, поступивших за определенный период. Запустите код на php. Выбор периода осуществляется в форме.
4. Запланируйте событие, которое бы раз в месяц перемещало бы в Архив заказы, исполненные более года назад

Вариант 18. Проект МЕБЕЛЬ

1. Разработайте хранимый код для подсчета количества дней, которое прошло между оформлением и исполнением договора. Если договор еще не исполнен – выведите информацию о договоре и статус «В исполнении», в противном случае – «Исполнен».
2. Разработайте механизмы логов для фиксации всех изменений в базе данных.
3. Разработайте хранимый код для запроса: Узнать, какую мебель купил клиент X. Запустите его на php. Имя клиента выбирается в форме
4. Запланируйте событие, которое бы ежеквартально мебель, которая не имеет спроса отмечало бы флагом «Под списание»

Вариант 19. Проект ТИПОГРАФИЯ

1. Изменить структуру таблицы базы данных для учета количества продукции, изготавливаемой каждым цехом. Разработайте хранимый код для согласования данных адекватно текущему состоянию базы данных.
2. Обеспечьте механизмы изменения количества продукции, изготавливаемой каждым цехом при добавлении и удалении продукции.
3. Разработайте хранимый код для запроса: Определить, какая продукция была заказана в определенном месяце. Запустите ее на php. Выбор месяца реализовать в виде выпадающего списка формы.
4. Запланируйте событие, которое бы раз в месяц увеличивало бы стоимость самой дешевой продукции на 10% и фиксировало в Архиве динамику изменения цены.

Вариант 20. Проект АПТЕЧНЫЙ СКЛАД

1. Разработайте хранимый код, который бы к лекарству добавлял флаг «Популярно» и «Низкие продажи» в зависимости от того, в каком объеме лекарство участвовало в закупках. Актуализируйте столбец в соответствии с текущим состоянием базы данных.
2. Разработайте механизмы логов для фиксации всех изменений в базе данных.
3. Разработайте хранимый код для запроса: Узнать, какие лекарства были в заявке аптеки «ФармаПлюс» от 11.02.2022. Запустите ее на php. Выбор аптеки и даты осуществляется с использованием формы. Список аптек подгружается из базы данных.
4. Запланируйте событие, которое бы раз в месяц актуализировало флаг «Популярно» и «Низкие продажи» для лекарства.

Вариант 21. Проект ДИПЛОМНЫЙ ПРОЕКТ

1. Измените структуру базы данных для учета среднего балла студентов, полученного на госэкзамене и при защите дипломной работы.
2. Реализуйте механизмы фиксации прошлой темы дипломной работы в случае ее изменения. Также отслеживайте и фиксируйте, если были выполнены изменения оценок после их выставления.
3. Разработайте хранимый код для запроса: Найти темы, которые предложил определенный преподаватель. Запустите на php. Преподаватель выбирается из списка в html-форме.
4. Запланируйте событие, которое бы ежегодно в июле месяца архивировало всю информацию о студентах, их темах, оценках и руководителях.

Вариант 22. Проект НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ

1. Изменить структуру таблицы базы данных для учета премии сотрудников. Разработайте хранимый код для вычисления премии сотрудника в зависимости от количества продаж. При 3 продажах начисляется премия в размере 10% от оклада. При 5 продажах начисляется премия в размере 20% от оклада. Выполните согласование данных.
2. Обеспечьте механизмы обновления величины премии в зависимости от числа продаж.
3. Разработайте хранимый код для запроса: Найти организации, с которыми был заключен договор в определенном году (год вводится в форме). Запустите на php.
4. Запланируйте событие, которое бы ежедневно проверяло договора, и договора, заключенные за последний месяц отмечало флагом «Актуально», а договора, заключенные более 3 месяцев назад отмечало как «Прошедшие».

Вариант 23. Проект КОММЕРЧЕСКАЯ ФИРМА

1. Разработайте хранимый код для подсчета количества автомобилей, купленных каждым клиентом. Выполните согласование данных.
2. Обеспечьте механизмы изменения количества проданных автомобилей клиентов в случае новой покупки или возврата.
3. Разработайте хранимый код для запроса: Вывести технические характеристики автомобилей, которые можно купить в указанном диапазоне. Запустите ее на php. Выбор диапазона выполнять в форме.
4. Запланируйте событие, которое бы каждый квартал добавляло удаленных поставщиков в архив.

Задание 24. Проект ИЗДАТЕЛЬСКИЙ ЦЕНТР

1. Разработайте хранимый код, который бы автоматически выставлял дату расторжения контракта после истечения срока с момента его заключения.
2. Подготовьте механизмы реагирования на факт расторжения контракта. Расторгнутые контракты добавляйте в таблицу «Архив».
3. Разработайте хранимый код для запроса: Найти автора книги X. Запустите ее на php. Название книги выбирается в форме.
4. Разработайте событие, которое бы каждый месяц самые популярные книги помечало товар флагом «Лидер продаж». Данные предыдущего месяца необходимо удалять.

Вариант 25. Проект АВТОЗАПРАВКИ

1. Изменить структуру таблицы базы данных для учета количества автозаправок каждой фирмы. Разработайте хранимый код для согласования данных адекватно текущему состоянию базы данных.
2. Обеспечьте механизмы изменения количества автозаправок каждой фирмы при их добавлении и удалении.
3. Разработайте хранимый код для запроса: Найти заправки, расположенные по адресу X. Запустите ее на php. Часть адреса вводится в поле формы.
4. Запланируйте событие, которое бы раз в месяц увеличивало бы стоимость самого популярного топлива на 3% и фиксировало в Архиве динамику изменения цены.

Вариант 26. Проект БИБЛИОТЕКА

1. Изменить структуру таблицы базы данных для учета количества книг каждого издательства. Разработайте хранимый код для согласования данных и отображения текущего количества книг по каждому издательству.
2. Подготовьте механизмы автоматического изменения количества книг каждого издательства в случае удаления или добавления книги.
3. Разработайте хранимый код для запроса: Найти книги, которые взял в библиотеке определенный читатель. Запустите код на php. Пользователь выбирается из формы.
4. Запланируйте событие, которые бы каждый квартал помечало бы самые часто выдаваемые книги как «Популярные», а книги, которые не выдавались как «Низкий спрос»

Вариант 27. Проект РОСТОВЗЕЛЕНСТРОЙ

1. Разработайте хранимый код для подсчета срока выполнения договора. Если договор был исполнен менее чем за 3 дня пометьте его флагом срочный.
2. При изменении цены рассады фиксируйте процент ее увеличения и дату изменения в таблице «Динамика изменения цены».
3. Разработайте хранимый код для запроса: Найти заказчиков цветов по цене рассады X-Y. Числовые величины вводятся в поля формы. Протестируйте на php.
4. Запланируйте событие, которое бы ежедневно договора, до даты исполнения которых осталось меньше 3 дней помечало флагом «Срочный».

Вариант 28. Проект АВИАЛИНИИ

1. Изменить структуру таблицы базы данных для подсчета количества рейсов для каждого маршрута. Разработайте хранимый код для согласования данных в базе данных.
2. Подготовьте механизмы автоматического изменения количества рейсов по каждому маршруту в случае удаления или добавления рейса.
3. Разработайте хранимый код для запроса: Определить, куда летал клиент в n квартале этого года. Клиент и квартал выбираются в выпадающем списке. Протестируйте на php.
4. Запланируйте событие, которые бы ежедневно проверяло срок эксплуатации самолета. Если срок эксплуатации закончился, такой самолет помечается специальным флагом.

Вариант 29. Проект ПРОКАТ ВЕЛОСИПЕДОВ ФИТНЕС-КЛУБА

1. Разработайте хранимый код, позволяющий подсчитать количество велосипедов на пункте проката. Актуализируйте состояние базы данных для поддержки ее согласованности.
2. Подготовьте механизмы логов для фиксации изменений в базе данных.
3. Разработайте хранимый код Вывести статистику по выдаче велосипедов марки Cube за апрель месяц. Марка велосипеда и дата задаются в форме. Протестировать выполнение на php.
4. Запланируйте событие, которые бы каждый месяц клиентам, которые брали велосипед больше 5 раз назначала скидку 3%.

Вариант 30. ПРОЕКТ КАНЦЕЛЯРСКАЯ ПРОДУКЦИЯ

1. Разработайте хранимый код отражения количества договоров, которые заключила каждая организация. Выполните согласование данных в базе данных.
2. Подготовьте механизмы увеличения количества договоров в таблице организации при заключении нового договора.
3. Разработайте хранимый код для запроса: Узнать перечень продукции, заказанной по договорам на определенную дату. Дата выбирается в форме. Протестируйте на php.
4. Запланируйте событие, которые бы каждый квартал увеличивало бы стоимость продукции, цена которой выше средней, на 1%.