

Лабораторная работа №6

ORM Readbean

Теоретический раздел	3
Подключение библиотеки.....	3
CRUD: Create (Создание записи)	3
CRUD: Read (Чтение)	4
CRUD: Update (Обновление записи).....	4
CRUD: Delete (Удаление).....	4
Поиск данных.....	5
Построение запросов (Querying)	6
Работа с Базами Данных и их таблицами	8
Связи (отношения) в RedBeanPHP	8
Методы подсчёта (Counting).....	9
Задания к лабораторной работе	10
<i>Общие правила выполнения заданий.....</i>	<i>10</i>
Вариант 1. Проект ПОСТАВКА ТОВАРОВ	11
Вариант 2. Проект РОЗНИЧНАЯ ТОРГОВЛЯ	12
Вариант 3. Проект БАНКОВСКИЕ ВКЛАДЫ	13
Вариант 4. Проект АУДИТ	14
Вариант 5. Проект ФАБРИКА.....	15
Вариант 6. Проект ТУРАГЕНСТВО	16
Вариант 7. Проект ТОРГОВЛЯ.....	17
Вариант 8. Проект КОСМЕТИЧЕСКАЯ ПРОДУКЦИЯ	18
Вариант 9. Проект ПОДПИСНЫЕ ИЗДАНИЯ	19
Вариант 10. Проект ПОЛИКЛИНИКА.....	20
Вариант 11. Проект СПЕЦОДЕЖДА	21
Вариант 12. Проект ГАИ.....	22
Вариант 13. Проект ЖЭС.....	23
Вариант 14. Проект ОБЩЕПИТ	24
Вариант 15. Проект ИНТЕРНЕТ-ПРОДАЖИ	25
Вариант 16. Проект БАНКОМАТЫ.....	26
Вариант 17. Проект РЕМОНТ БЫТОВОЙ ТЕХНИКИ	27
Вариант 18. Проект МЕБЕЛЬ.....	28
Вариант 19. Проект ТИПОГРАФИЯ	29
Вариант 20. Проект АПТЕЧНЫЙ СКЛАД	30
Вариант 21. Проект ДИПЛОМНЫЙ ПРОЕКТ	31
Вариант 22. Проект НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ	32
Вариант 23. Проект КОММЕРЧЕСКАЯ ФИРМА.....	33

Задание 24. Проект ИЗДАТЕЛЬСКИЙ ЦЕНТР	34
Вариант 25. Проект АВТОЗАПРАВКИ.....	35
Вариант 26. Проект БИБЛИОТЕКА	36
Вариант 27. Проект РОСТОВЗЕЛЕНСТРОЙ	37
Вариант 28. Проект АВИАЛИНИИ	38
Вариант 29. Проект ПРОКАТ ВЕЛОСИПЕДОВ ФИТНЕС-КЛУБА	39
Вариант 30. ПРОЕКТ КАНЦЕЛЯРСКАЯ ПРОДУКЦИЯ	40

Теоретический раздел

ORM или **Object-relational mapping** (рус. Объектно-реляционное отображение) — это технология программирования, которая позволяет преобразовывать несовместимые типы моделей в ООП, в частности, между хранилищем данных и объектами программирования. ORM используется для упрощения процесса сохранения объектов в реляционную базу данных и их извлечения, при этом ORM сама заботится о преобразовании данных между двумя несовместимыми состояниями.

RedBeanPHP был создан для значительного облегчения жизни программистов в процессе работы с базами данных. RedBeanPHP требует версию PHP $\geq 5.3.4$.

В RedBeanPHP каждый объект записи называется бином. Эти бины можно воспринимать как самые обычные объекты, свойства которых представляют собой записи в Ваших таблицах. То есть одна запись - это один бин, а его свойства это поля Вашей записи. Работать с бином можно точно также как с обычным массивом.

Скачать библиотеку RedBeanPHP можно на [официальном сайте](#).

Подключение библиотеки

```
1. require 'libs/rb.php';
```

Для подключения к базе данных в RedBeanPHP есть статичный метод **setup()**, который принимает 4 опциональных аргумента. Опциональными они являются, потому что Вы можете никакой аргумент не задать и тогда RedBeanPHP создаст временную базу данных в формате SQLite в Вашей временной директории. Вызывается метод **setup()** для MySQL следующим образом:

```
1. R::setup('mysql:host=localhost; dbname=dbname','login','password');
2. if (!R::testConnection() )
3. {
4.     exit ('Нет соединения с базой данных');
5. }
```

Метод **testConnection** проверяет есть ли у нас фактическое подключение к базе.

Закрыть соединение с базой данных Вы можете при помощи метода **close()**.

```
1. R::close();
```

CRUD: Create (Создание записи)

```
1. // Указываем, что будем работать с таблицей book
2. $book = R::dispense('book');
3. // Заполняем объект свойствами
4. $book->title = 'Призрак победы';
5. $book->price = 199;
6. // Можно обращаться как к массиву
```

```
7. $book['author'] = 'Макс Глебов';
8. // Сохраняем объект
9. R::store($book);
```

CRUD: Read (Чтение)

Если нужно получить данные без каких-либо условий, то легче это сделать методами **load()** и **loadAll()**

```
2. // Получаем все записи, ID которых указаны в массиве ids
3. $ids = [1,2,3];
4. $books = R::loadAll('book', $ids);
5. foreach ($books as $book){
6.     echo $book->title.'<br>';
7. }
8. // Получаем одну запись по её ID
9. $id = 1;
10. $book = R::load('book', $id);
11. echo $book->title;
```

Если по каким-то причинам вам понадобится именно массив данных, то на этот случай есть метод **export()**:

```
1. $id = 1;
2. $book = R::load('book', $id);
3. $book = $book->export();
4. echo $book['title'];
```

CRUD: Update (Обновление записи)

```
1. $id = 1;
2. // Загружаем объект с ID = 1
3. $book = R::load('book', $id);
4. // Обращаемся к свойству объекта и назначаем ему новое значение
5. $book->price = 210;
6. // Сохраняем объект
7. R::store($book);
```

CRUD: Delete (Удаление)

Удалить запись с **ID = 5**

```
1. $id = 5;
2. $book = R::load('book', $id);
3. R::trash($book);
```

Удалить записи с **ID = 6, 7**

```
1. $ids = [6, 7];
2. $book = R::loadAll('book', $ids);
3. R::trashAll($book);
4. // Начиная с версии 5.1 данную задачу лучше выполнить методом R::trashBatch(). В
   таком случае нет необходимости создавать (получать) бин - объект RedBeanPHP
5. $ids = [6, 7];
6. R::trashBatch('book', $ids);
7. // Удаление записи с ID = 3
8. $id = 3;
9. R::hunt('book', 'id = ?', [$id]);
```

Метод **wipe()** полностью очищает указанную таблицу:

```
1. R::wipe('book');
```

Метод **nuke()** полностью очищает всю базу данных. Режим заморозки должен быть выключен:

```
1. R::freeze(false);
2. R::nuke();
```

Поиск данных

find(), findOne(), findAll(), findLike()

Если вы не знаете идентификатор бина, вы можете искать бины, используя метод **find()**:

```
1. $min_price = 250;
2. $books = R::find('book', 'price > ?', [$min_price]);
3. $search = 'строка';
4. $books = R::find('book', 'author LIKE ?', ["%$search%"]);
5. $id = 1;
6. $min_price = 300;
7. $books = R::find('book', 'id > :id AND price < :price', [':price' => $min_price, ':id'
   => $id]);
8. $ids = [1, 3, 5];
9. $books = R::find('book', 'id IN (' . R::genSlots($ids) . ')', $ids);
```

Если необходимо получить только одну запись, используем метод **findOne()**:

```
1. $id = 1;
2. $book = R::findOne('book', 'id = ?', [$id]);
3.
4. $title = 'гостя из будущего';
5. $book = R::findOne('book', 'title = ?', [$title]);
```

Если необходимо получить все данные без особых условий, используем метод **findAll()**:

```
1. $books = R::findAll('book');
2. $limit = 5;
3. $books = R::findAll('book', 'ORDER BY id ASC LIMIT ?', [$limit]);
```

Метод **findLike()** предназначен для поиска по записям.

```
1. $search_1 = 'Джон Пристли';
2. $search_2 = 'Сергей Тармашев';
3. $books = R::findLike('book',
4.   ['author' => [$search_1, $search_2]],
5.   'ORDER BY title ASC'
6. );
```

Построение запросов (Querying)

При использовании RedBeanPHP (как и любой другой ORM) не всегда можно ограничиться простыми методами поиска (Finding). Часто существует необходимость сделать более сложный запрос, который сделать простыми методами крайне проблематично.

Важно! Рассмотренные выше методы **Finding** необходимо применять, если требуется сделать простой запрос, без каких-либо сложных условий. В рассмотренных ниже примерах всегда возвращается массив данных (а не объекты-бины).

Метод exec()

Метод для произвольного SQL запроса (чаще всего применяется для добавления, изменения и удаления):

```
1. $id = 3;
2. $title = 'New title';
3.
4. R::exec('UPDATE `book` SET `title` = :title WHERE id = :id', [
5.   'id' => $id,
6.   'title' => $title
7. ]);
```

Метод getAll()

Вернёт массив данных (все записи/несколько по условию) из указанной таблицы:

```
1. //$books = R::getAll('SELECT `title` FROM `book`');
2. $id = 1;
3. $books = R::getAll('SELECT `title` FROM `book` WHERE `id` > ?', [$id]);
4. foreach ($books as $book){
5.   echo $book['title'].'<br>';
6. }
```

Метод `getRow()`

Вернёт все записи, но выводит только одну. Рекомендуется добавлять **LIMIT 1**, чтобы и запрашивалась тоже только одна запись:

```
1. $search = 'поворот';
2. $book = R::getRow('SELECT * FROM `book` WHERE `author` LIKE :search LIMIT 1', [
3.     'search' => "%$search%"
4. ]);
```

Метод `getCol()`

Вернёт колонку:

```
1. // Выбрать все названия всех книг
2. $books = R::getCol('SELECT `title` FROM book');
```

Метод `getCell()`

Вернёт ячейку одной записи:

```
1. $id = 5;
2. $title = R::getCell('SELECT `title` FROM book WHERE `id` = ? LIMIT 1', [$id]);
```

Метод `getAssoc()`

Чтобы получить ассоциативный массив с указанным столбцом ключа и значения, используйте:

```
1. R::getAssoc('SELECT id, title FROM book');
```

Метод `getInsertID()`

Вернёт ID последней вставленной записи:

```
1. $res = R::exec("INSERT INTO book (title, author, price) VALUES (?, ?, ?)", ['New Book',
    'New Author', 10]);
2. $id = R::getInsertID();
```

Методы `convertToBean()` и `convertToBeans()`

Конвертация массива записей в бины или один бин (`convertToBean()`)

```
1. $books = R::getAll("SELECT * FROM book");
2. $books = R::convertToBeans('book', $books);
3.
4. $book = R::getRow("SELECT * FROM book WHERE `id` = ?", [1]);
5. $book = R::convertToBean('book', $book);
```

Работа с Базами Данных и их таблицами

Метод **inspect()** возвращает названия таблиц в БД. Если параметром передать название таблицы, то он вернёт все поля этой таблицы:

```
1. // Какие таблицы есть в БД
2. $tables = R::inspect();
3.
4. // Какие поля есть в указанной таблице
5. $fields = R::inspect('book');
```

Связи (отношения) в RedBeanPHP

One-to-many (связь один ко многим)

Работа с реляционной базой данных предполагает наличие связанности между таблицами. RedBeanPHP поддерживает все необходимые типы связей. Для создания связи один-ко-многим мы используем свойство с префиксом **own**.

Own поле является ассоциативным массивом

RedBeanPHP поддерживает концепцию “ленивой загрузки”. То есть загрузка связанных данных произойдет в момент обращения к полю объекта

Достанем из БД все книги, у которых **category_id = 1**

```
1. $category_id = 1;
2. $category = R::load('category', $category_id);
3. $books = $category->ownBookList;
4. // Сортировка и лимит
5. $books = $category->with('ORDER BY `title` ASC LIMIT 3')->ownBookList;
6. // Но более предпочтительным способом является метод withCondition()
7. $status = 1;
8. $limit = 3;
9. $books = $category
10.  ->withCondition('status = ? ORDER BY title ASC LIMIT ?', [$status, $limit])
11.  ->ownBookList;
12. foreach ($books as $book){
13.     echo $book->title.'<br>';
14. }
```

Many-to-one (связь Многие к одному)

Для создания связи многие-ко-многим используем свойство с префиксом **shared**.

Достанет из базы название категории, с которой связана книга

```
1. $book = R::load('book', 1);
2. $category = $book->category->title;
```


Many-to-many (связь Многие к одному)

Достанет из базы (из связующей таблицы) все книги этой категории:

```
1. $category = R::load('category', 1);  
2. $books = $category->sharedBookList;  
3. print_r($books);
```

Методы подсчёта (Counting)

Простой подсчёт элементов:

```
1. // Сколько записей (элементов) в таблице book  
2. $books = R::count( 'book' );  
3. // Сколько записей (элементов) в таблице book, у которых поле status = 1  
4. $books = R::count( 'book', 'status = ?', [1] );
```

Подсчёт элементов связанных таблиц:

```
1. // Сколько записей (элементов) в таблице book, связанных с категорией с ID = 1  
2. $category = R::load('category', 1);  
3. $numBook = $category->countOwn('book');
```

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.).
- Библиотека RedBeanPHP используется в качестве ORM

Задание:

Добавить в веб-приложение вкладку для работы с ORM (возможна многоуровневая структура для каждого запроса согласно индивидуальному варианту задания). Доступ ограничить для всех, кроме администратора. Реализовать CRUD-интерфейс для указанной таблицы и перечень запросов на выборку согласно индивидуальному варианту задания.

Требования и рекомендации по выполнению:

- Запросы выполняются с использованием ORM
- Не использовать метод `exec()` в запросах на выборку
- Использовать связи типа `own`.
- При выводе информации идентификаторы (внешние ключи) должны быть заменены на значения из связанных таблиц адекватно содержанию запроса.
- Формулировка «выбранный» («введенный») в запросах означает подгрузку соответствующих значений из базы данных (при их наличии) или ввод значений (при указании диапазона или необходимости использования маски ввода), если это целесообразно.
- Формат выбора не ограничен (возможно отображение таблицы с кнопками, списка с выбором и т.д.)
- Формат вывода результата не ограничен (табличный вывод, список и т.д.)
- После выполнения работы оформить ее по своему усмотрению (настроить шрифты, отступы и т.д.), используя CSS, Bootstrap и др.

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

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Отгрузки»
2. Вывести список товаров, отгруженных за *выбранный* квартал *выбранного* года
3. Определить количество отгруженного товара (шт) по цене, лежащей в *выбранном* диапазоне
4. Вывести список товаров, заказанный компанией, в названии которой содержится «ООО», «АОО», «ИП» и т.п. в зависимости от *введенного* значения

Вариант 2. Проект РОЗНИЧНАЯ ТОРГОВЛЯ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Товары»
2. Вывести список товаров, проданных за *выбранный* период
3. Вывести список товаров, поставляемых *выбранным* поставщиком, отсортированный по цене закупке (*по убыванию или возрастанию*)
4. Определить выручку проданного товара для *выбранного* поставщика

Вариант 3. Проект БАНКОВСКИЕ ВКЛАДЫ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Счет»
2. Определить клиентов, сумма на счете у которых лежит в *выбранном* диапазоне
3. Вывести наименования вкладов, открытых *выбранным* клиентом
4. Найти клиентов, у которых счет будет закрыт в *выбранном* месяце *выбранного* года

Вариант 4. Проект АУДИТ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Сотрудники»
2. Вывести всех сотрудников *выбранной* категории
3. Подсчитать количество часов, которые отработал каждый из сотрудников, отсортировав по выбранному столбцу по *выбранному* условию (по возрастанию/по убыванию)
4. Определить виды работ, которые *выбранный* сотрудник осуществлял в *выбранный период*

Вариант 5. Проект ФАБРИКА

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Модели»
2. Вывести список моделей *выбранного вида*, цена которых находится в *выбранном диапазоне цен*
3. Подсчитать количество моделей каждого вида. Виды, в которых более 2 моделей выделить
4. Вывести модели с указанием их типа, поступившие в *выбранном месяце*

Вариант 6. Проект ТУРАГЕНСТВО

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Маршруты»
2. Определить наименования маршрутов, реализуемых турагенством в *выбранной* стране
3. Узнать, сколько маршрутов организуется в *выбранную* страну, отсортировав по стоимости визы (по возрастанию или убыванию в зависимости от *выбора*)
4. Вывести список стран, в которые были проданы маршруты в *выбранном* квартале *выбранного* года

Вариант 7. Проект ТОРГОВЛЯ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Продажи»
2. Вывести товары, проданные в *выбранном* месяце
3. Найти товары, которые были проданы в отделе с *выбранным* заведующим отдела
4. Вывести товары с указанием количества их продаж, отсортированных *выбранным типом* (по возрастанию/убыванию количества продаж)

Вариант 8. Проект КОСМЕТИЧЕСКАЯ ПРОДУКЦИЯ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Товары»
2. Найти все товары *выбранного* бренда
3. Подсчитать количество товаров каждого бренда. Выделить бренды, в которых более 1 товара
4. Вывести перечень товаров с указанием их бренда, заказанных в *выбранном* месяце *выбранного* года

Вариант 9. Проект ПОДПИСНЫЕ ИЗДАНИЯ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Подписка»
2. Найти перечень изданий, на которые подписан *выбранный* получатель
3. Узнать, на какое количество изданий подписан каждый получатель. Получателя с максимальным количеством подписок выделить.
4. Найти получателей, подписка на издания в месяц которых составляет сумму в *выбранном* диапазоне

Вариант 10. Проект ПОЛИКЛИНИКА

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Прием»
2. Узнать, к каким врачам на прием приходил *выбранный* пациент
3. Узнать, сколько приемов осуществил *выбранный* врач за *выбранный* период
4. Подсчитать, какую сумму за посещение поликлиники потратил *выбранный* пациент

Вариант 11. Проект СПЕЦОДЕЖДА

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Работники»
2. Вывести работников цеха под руководством *выбранного* начальника
3. Определить, какое количество видов спецодежды получает каждый сотрудник
4. Вывести перечень сотрудников, которые получили спецодежду в *выбранном* квартале *выбранного* года

Вариант 12. Проект ГАИ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Автомобиль»
2. Найти автомобиль, который принадлежит водителю с *выбранным* номером удостоверения
3. Подсчитайте количество автомобилей каждого водителя, выполнив сортировку по их количеству в зависимости от *выбора* (по возрастанию или убыванию)
4. Определите, кем и какие были совершены нарушения в *выбранном* месяце *выбранного* года

Вариант 13. Проект ЖЭС

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Квартира»
2. Найти квартиросъемщиков, снимающих квартиру на *выбранной* улице
3. Вывести список квартиросъемщиков-задолжников по оплате *выбранной* услуги
4. Вывести перечень услуг, заключенных с *выбранным* квартиросъемщиком

Вариант 14. Проект ОБЩЕПИТ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Приготовление»
2. Определить перечень продуктов, необходимых для приготовления *выбранного* блюда
3. Определить, для приготовления каких блюд используется *выбранный* продукт
4. Вывести стоимость всех блюд предприятия общественного питания

Вариант 15. Проект ИНТЕРНЕТ-ПРОДАЖИ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Заказы»
2. Вывести список товаров, реализуемый через *выбранный* интернет-магазин
3. Определить перечень товаров, заказанных *в выбранный* период *выбранным* клиентом
4. Создать фильтр для вывода товаров (доставленных или нет)

Вариант 16. Проект БАНКОМАТЫ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Банкомат»
2. Определить, какое количество банкоматов обслуживает каждый банк
3. Узнать, на какую сумму и в каких банкоматах были сняты наличные *выбранным* клиентом
4. Выведите информацию о совершенных операциях по выдаче денег в выбранную дату с комиссией или без в зависимости от *выбора*

Вариант 17. Проект РЕМОНТ БЫТОВОЙ ТЕХНИКИ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Заказы»
2. Узнайте, ремонт какого товара по гарантии был заказан в *выбранный* период
3. Узнайте, сколько ремонтов произвел *выбранный* сотрудник
4. Узнайте, на какую сумму произвела фирма ремонт в *выбранном* месяце

Вариант 18. Проект МЕБЕЛЬ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Договоры»
2. Узнать перечень покупателей, заключивших договор за *выбранный* период
3. Узнать, какую мебель купил *выбранный* клиент
4. Узнать, сколько договоров заключил на покупку каждый покупатель

Вариант 19. Проект ТИПОГРАФИЯ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Продукция»
2. Определить продукцию, за изготовление которой отвечает цех под руководством *выбранного* начальника
3. Определить, какое количество видов продукции изготавливается каждым цехом.
Цех-лидер по количеству изготавливаемым видам продукции выделить
4. Определить, какая продукция была заказана в *выбранный период*

Вариант 20. Проект АПТЕЧНЫЙ СКЛАД

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Заявка»
2. Найдите лекарства *введенного* производителя. Добавьте сортировку по цене по возрастанию/убыванию
3. Узнать, сколько заявок составила каждая аптека в *выбранном* месяце *выбранного* года
4. Узнать, какие лекарства были в заявке аптеки *выбранной* аптеки на *выбранную* дату

Вариант 21. Проект ДИПЛОМНЫЙ ПРОЕКТ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Темы»
2. Найти темы, которые предложил *выбранный* преподаватель и подсчитать их количество
3. Найти студентов, у которых руководителем был *выбранный* преподаватель
4. Найти студентов, которые получили на защите дипломной работы *выбранные* оценки (возможен множественный выбор)

Вариант 22. Проект НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Договор»
2. Узнать, какой тип оборудования поставляется *выбранной* организации
3. Найти организации, с которыми был заключен договор в *выбранном* месяце
4. Узнать, сколько программного обеспечения заказала каждая организация, отсортировав по *выбору* (от большего к меньшему или от меньшего к большему)

Вариант 23. Проект КОММЕРЧЕСКАЯ ФИРМА

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Модель»
2. Узнать цены на автомобиль *выбранную* модель автомобиля с учетом *выбора* коробки передач
3. Узнать, сколько автомобилей было продано за *выбранный* период
4. Узнать, какую модель автомобиля и по какой цене купил *выбранный* клиент

Задание 24. Проект ИЗДАТЕЛЬСКИЙ ЦЕНТР

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Заказы»
2. Узнать, сколько книг издал каждый писатель в издательском центре. Отсортировать по количеству (по возрастанию/убыванию в зависимости от *выбора*)
3. Найти книги, которые были заказаны за *выбранный* период
4. Найти авторов *выбранной* книги

Вариант 25. Проект АВТОЗАПРАВКИ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Автозаправки»
2. Найти фирмы, у которых количество заправок лежит в *выбранном* диапазоне
3. Подсчитать, сколько топлива продал Лукойл за *выбранный* квартал *выбранного* года
4. Узнать цену *выбранного* вида топлива у различных фирм. Отсортировать по цене (по возрастанию/убыванию в зависимости от выбора)

Вариант 26. Проект БИБЛИОТЕКА

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Книги»
2. Найти книги, которые взял в библиотеке *выбранный* читатель за *выбранный* период
3. Найти читателей, которые брали *выбранную* книгу
4. Определить какие книги и какого издательства были выданы в *выбранном* году

Вариант 27. Проект РОСТОВЗЕЛЕНСТРОЙ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Цветы»
2. Узнать, кто из поставщиков поставляет цветов в количестве *выбранного* диапазона
3. Узнать, цветы какого поставщика были заказаны в *выбранном* месяце *выбранного* года. Отсортировать по цене рассады (по возрастанию/убыванию в зависимости от *выбора*)
4. Найти заказчиков цветов по цене рассады в *выбранном* диапазоне

Вариант 28. Проект АВИАЛИНИИ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Рейсы»
2. Найти сколько рейсов осуществляется по каждому маршруту
3. Найти командиров корабля, которые осуществляли полет в *выбранный период*
4. Определить, куда летал *выбранный* клиент в *выбранном* году

Вариант 29. Проект ПРОКАТ ВЕЛОСИПЕДОВ ФИТНЕС-КЛУБА

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Экземпляр»
2. Узнать, на каком пункте проката хранятся велосипеды *выбранной* марки
3. Узнать, сколько раз каждый клиент брал велосипед
4. Найти велосипеды, которые находятся в пункте проката в фитнес-клубе по *выбранному* адресу (указывается название улицы)

Вариант 30. ПРОЕКТ КАНЦЕЛЯРСКАЯ ПРОДУКЦИЯ

Перечень запросов для реализации (общий вид):

1. Реализовать CRUD-интерфейс для таблицы «Договоры»
2. Узнать перечень продукции, заказанной по договорам за *выбранный* период
3. Подсчитать, сколько раз совершали заказ все организации-заказчики. Выделить те, которые заключили более 1 договора
4. Узнать какую продукцию заказала *выбранная* фирма