

Курс подготовки программиста для разработки конфигураций на платформе Клиентской базы

Оглавление

Урок 1. Общие сведения о платформе КБ	3
Урок 2. Интерфейс настроек КБ	5
Урок 3. Конструктор вычислений и допдействий	7
Настройка “Вычислять при условии”	9
Урок 4. Синтаксис кода вычислений	11
Формат типов полей	11
Глобальные переменные	12
Использование переменной \$user	13
Примеры вычислений	13
Урок 5. Синтаксис кода допдействий	17
Типы допдействий	17
Урок 6. Создаем простые отчеты	20
Урок 7. Внутренние функции КБ	23
Частые внутренние функции КБ	23
Урок 8. Структура базы данных	25
Основные таблицы	25
Урок 9. Рекомендации по написанию кода	27
Стандарт кода	27
Безопасность кода	28
Ввод данных	28
Изменение данных	28
Заключение	30

Вступление

Добрый день, уважаемый разработчик!

Изучив материал и выполнив уроки опубликованные в данном документе вы получите общие сведения по разработке собственных решений на платформе Клиентской базы (КБ) www.clientbase.ru.

Предполагается, что вы уже имеете достаточные навыки программирования на PHP/MySQL в целом, поэтому данный документ не является курсом по программированию, а рассматривает лишь программирование в применении к нашей платформе. Если вы не уверены в своих знаниях PHP/MySQL, то вы можете [пройти тест](#).

Также в данном курсе не рассматривается или рассматривается минимально обучение конфигурированию программы Клиентская база без программирования. В конфигурирование входит: создание таблиц и полей, создание подтаблиц и связанных таблиц, настройка всех типов полей под свои задачи, настройка групп доступа, создание шаблонов печати и рассылки и так далее. Конфигурированию у нас посвящен [отдельный курс Конфигуратора](#), который также можно пройти предварительно для более глубокого погружения, однако столь подробные знания не всегда обязательны для написания кода вычислений и отчетов.

В чем преимущества использования платформы КБ для Вас как для разработчика?

- **существенная экономия по времени разработки.** Вы не изобретаете велосипед, типа таблицы Пользователи или конструктора таблиц, а используете уже существующий функционал
- **широчайший функционал платформы** позволит создать большую часть разработки в режиме конструктора, программирование потребуется лишь точечное.
- **платформа развернута в облаке**, и вы всегда можете создать бесплатный аккаунт для тестов и разработки на 14 дней и продемонстрировать результат клиенту
- сопровождение и техническая поддержка самой платформы и серверной части лежит на нас
- **платформа имеет коробочную версию**
- **конфигурации, можно сохранять в виде бэкапа**, переносить на другие аккаунты, выкладывать в Магазин конфигураций, либо просто сохранять у себя на диске.

Итак, в случае, если мы вас убедили, приступим к изучению платформы.

Внимание! При прохождении курса, записывайте все вопросы, которые у Вас возникали (что непонятно или не подробно расписано). Список этих вопросов необходимо будет приложить к [анкете заполняемой в конце курса](#).

Урок 1. Общие сведения о платформе КБ

Клиентская база это онлайн-платформа с помощью которой можно создавать персональные решения по автоматизации любого бизнеса.

Сайт: www.clientbase.ru

С точки зрения разработчика в Клиентской базе присутствуют следующие возможности:

- создавать любые таблицы с любым набором полей и связывать их между собой, как в связанные таблицы, так и в подтаблицы.
- при создании таблиц использовать 9 типов полей: текст, число, дата и время, список, связь (с другой таблицей), файл, изображение, пользователь, группа. Каждое из которых имеет множество дополнительных настроек.
- есть возможность создавать различные группы доступа и устанавливать для каждой группы права на просмотр и редактирование любого поля в созданных вами таблицах. Если же групп доступа не хватает, то существуют также правила доступа, которые определяют доступ в зависимости от условий (например Статус строки равен Разрешен, или Пользователь равен Иванов Иван).
- создавать шаблоны документов, которые заполняются автоматически, по тем данным, которые есть в записях таблицы (форматы документов HTML, PDF, DOC, RTF и т.д.)
- в зависимости от условий, необходимых разработчику создавать: цветовое форматирование строк, фильтры, напоминания
- проводить e-mail и смс-рассылки по созданным разработчиком шаблонам
- создавать внешние формы для интеграции платформы с другими сайтами и системами
- использовать API
- и довольно обширный дополнительный функционал, который вы при желании можете изучить используя нашу документацию.

Но главное для вас, то что на нашей платформе есть возможность дорабатывать программу используя вычисления, которые создаются через

Конструктор вычислений и допдействий или пишутся с нуля на PHP/MySQL, либо Javascript.

Разберем, что можно реализовать с помощью программирования.

Во первых это ВЫЧИСЛЕНИЯ. Это скрипты которые отработывают при заданных условиях, самое частое это Сохранение записи.

Например у нас есть поле А и поле В. Нам требуется, чтобы в поле С записалось их произведение. Соответственно нам требуется создать ВЫЧИСЛЕНИЕ, которое будет записывать в поле $C = A * B$

Это можно сделать как с помощью конструктора вычислений, так и просто написанием программного кода. Вы можете также комбинировать эти методы, например, создать базовое вычисление с помощью конструктора вычислений, а затем подправить код, уже непосредственно в редакторе.

Во-вторых вы можете создать ДОПДЕЙСТВИЯ. Это кнопки, которые вы можете добавить в любую таблицу, внутрь каждой записи. По нажатию кнопки будет выполняться код, описанный в ДОПДЕЙСТВИИ.

Например, Допдействие "Списать товар", оно должно по щелчку изменять Статус товара на "Списан", заполнять ФИО пользователя который списал товар, а также фиксировать Дату списания.

На платформе КБ, Вы также, как и в случае с вычислением, можете создать это допдействие с помощью конструктора, а можете написать код вручную. Также можно комбинировать эти два метода.

И третий случай применения программирования это ПРЕДСТАВЛЕНИЯ.

Представления это страницы на которых вы можете выводить в произвольной форме совокупную информацию по одной или нескольким таблицам.

Самый простой пример Представления это Отчет по продажам, который выводит информацию по продажам за определенный период, основываясь на таблице Счета и и разбивкой данных продаж по менеджерам.

На данный момент в КБ нет конструктора отчетов (он в разработке), но есть предустановленные отчеты, код которых вы можете изучить и доработать под свои задачи в случае необходимости.

ЗАДАНИЕ 1:

Создайте аккаунт <https://clientbase.ru/create/> со Стандартной конфигурацией.

Урок 2. Интерфейс настроек КБ

Программа КБ имеет два интерфейса: для пользователей и для администратора.

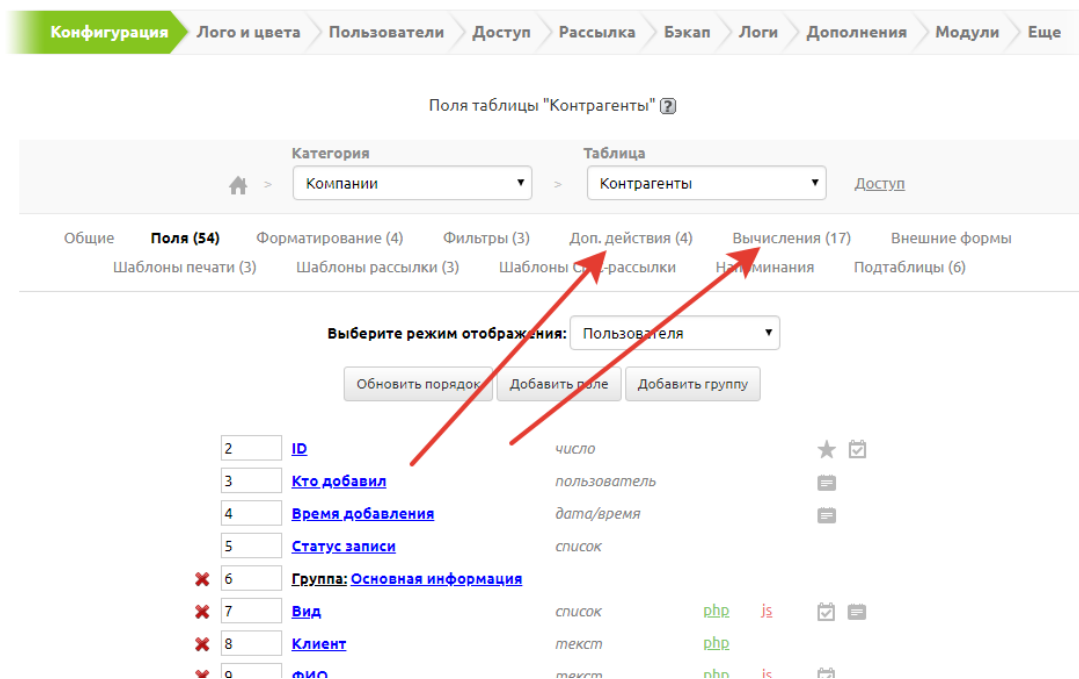
Для перехода в интерфейс администратора кликните на значок шестеренки:

The screenshot shows the K6 ClientBase interface. The top header is dark red with the K6 logo and 'Клиентская база'. Below the header, there's a navigation bar with 'Конструктор' and 'Рассылки'. The main content area is titled 'Автомобили' and contains a table with columns: 'МАРКА АВТОМОБИЛЯ', 'МОДЕЛЬ АВТОМОБИЛЯ', 'ГОД ВЫПУСКА', 'СТОИМОСТЬ', 'ДАТА ПОКУПКИ', 'ИЗОБРАЖЕНИЕ', 'СТАТУС', 'СОТРУДНИК', 'СОСТОЯНИЕ', 'С НАЗНАЧКОМ', 'ПОЛЬЗОВАТЕЛЬ', 'ДАТА СЛЕДУЮЩЕГО ТЕХОСМОТРА', and 'КОЛИЧЕСТВО ДНЕЙ ОЖИДАНИЯ'. The table lists several cars, including Toyota Land Cruiser and VW. A red arrow points to the settings gear icon in the top right corner of the header.

Вы попадаете в Настройки, где у вас есть возможность создавать или изменять существующие Таблицы, а также создавать для них вычисления или допдействия.

The screenshot shows the K6 ClientBase interface in the 'Настройка' (Settings) section. The top header is dark red with the K6 logo and 'Клиентская база'. Below the header, there's a navigation bar with 'Конструктор' and 'Рассылки'. The main content area is titled 'Настройка' and contains a list of categories and tables. The 'Категории' (Categories) section is visible, showing a list of categories and tables. The categories are: 'Начало работы', 'Компания', 'Работа', 'Заявки', 'Документы', 'Расходы', 'Сотрудники', 'Справочники', 'Отчеты', 'Конструктор', and 'Рассылки'. The 'Таблица' (Table) section is also visible, showing a list of tables. The 'Категория' (Category) dropdown is set to 'Все категории' and the 'Таблица' (Table) dropdown is set to 'Все'. The 'Обновить порядок' (Update order) and 'Добавить категорию' (Add category) buttons are at the bottom.

Для того, чтобы создать вычисление выберите необходимую вам Категорию, а затем Таблицу. Вы попадете в настройки таблицы, где появятся вкладки “Вычисления” и “Допдействия”, которые относятся к этой таблице.



Войдя в любой из этих пунктов вы сможете создавать новые, либо просматривать и изменять существующие вычисления и допдействия.

Задание 2. Создайте собственную категорию, например "Тест", в ней таблицу "Тест".

В таблице "Тест" создайте поля Наименование товара (текст), Цена (число), Количество (число) и Сумма (число). Можете создать какие-нибудь дополнительные поля других типов по своему усмотрению. В этой таблице в следующих уроках мы будем создавать вычисления и допдействия.

Если вы не разобрались, как создать собственную таблицу самостоятельно, то можете посмотреть видеоурок

<https://www.youtube.com/watch?v=zAbPo7vdNnl&t>

Урок 3. Конструктор вычислений и допдействий

Самым простым способом создать вычисление или допдействие в программе Клиентская база является использование конструктора.

Для понимания работы конструктора требуется посмотреть два небольших видеоурока:

Конструктору вычислений

https://www.youtube.com/watch?v=j3a2XR4_MKU&t=43s

Конструктору допдействий

<https://www.youtube.com/watch?v=N0n7ZJCzNb4&t=10s>

Также есть общая справочная информация по конструктору

https://clientbase.ru/help/for_admin_16/calculations/#construct

Конструктор состоит из шаблонов - типовых вычислений, например вычисление по копированию поля из одного столбца в другой, либо вычисление по созданию математической формулы и т.д.

Около 80% всех вычислений и допдействий на данный момент можно создать с помощью встроенных шаблонов, которые в том числе можно объединять в одном вычислении/допдействии. Однако, всегда есть вероятность того, что возможностей конструктора будет недостаточно и потребуются внести коррективы непосредственно в код.

Даже если, Вам предстоит написать сложное вычисление, базовую структуру мы все равно рекомендуем создать с помощью конструктора (это быстрее, удобнее и задаст правильный стандарт синтаксиса), а уже далее исправить сформированный код под свои задачи.

Для исправления кода вычисления или допдействия, вам необходимо переключить его в режим эксперта:

Описание вычисления : Покупка автомобиля

Условия :

Режим : ☒ Режим конструктора ☐ Режим эксперта

Шаблон : Изменить поле  

Дата: Изменить дату  

Далее следует нажать “Редактировать”

Последнее изменение: 15.08.2018 16:08 , Гарифуллин Марат, IP 91.245.36.4.

Описание вычисления : Покупка автомобиля

Режим : ☐ Режим конструктора ☒ Режим эксперта

Вычисления : [Редактировать](#)

Действия для запуска вычисления

или	Сохранение в таблице 	
или	Изменение поля 	Дата покупки
или	Изменение поля 	Количество дней
или		

и вы можете вносить изменения в уже сформированный код.

```

1 $line['Статус'] = 'Продан';
2
3 $line['Дата следующего техосмотра'] = date('Y-m-d H:i:s', str

```

Позиция: Стр 1, Стлб 1 Всего: Стр 3, Стлб 167

Важно! После внесения изменений в код в режиме эксперта вы уже не сможете вернуть его в режим конструктора.

Для получения **примеров кода по типовым действиям** таким как: копирование поля, копирования поля из связанной таблицы, добавление/копировании записи в таблицу, удаление записи, операции с датами и так далее, мы рекомендуем создать данной вычисление с помощью

одного или нескольких шаблонов конструктора и ознакомиться с полученным кодом.

Также стоит отметить, что всегда есть возможность сразу написать код самостоятельно, перейдя в режим эксперта при создании вычисления или допдействия, однако мы рекомендуем это делать только опытным разработчикам уже знакомым с синтаксисом вычислений, структурой базы данных и внутренними функциями Клиентской базы.

Выбор типа вычисления

Выберите тип вычисления:

Связанная таблица

Вычисления из этого списка создаются в текущей и изменяют данные в связанных таблицах

Родительская таблица

Вычисления из этого списка создаются в текущей и изменяют данные в дочерних таблицах

Режим эксперта

Перейти в режим эксперта

Назад



Настройка “Вычислять при условии”

Данная настройка является важной частью вычисления написанного как в режиме конструктора, так и в режиме эксперта. Она позволяет задавать **события** при которых требуется срабатывание вычисления.

В режиме конструктора данные события формируются самостоятельно, но их можно изменять, в зависимости от необходимых задач

- *Сохранение в таблице* - вычисление срабатывает при любом редактировании/добавлении записи в таблице.
- *Отображение в таблице* - вычисление будет срабатывать для всех строк, каждый раз при просмотре таблицы (данное событие крайне не экономичное в плане использования ресурсов и не рекомендуется к

использованию, однако в некоторых исключительных случаях оно необходимо).

- *Изменение поля* - вычисление сработает только при изменении определенного поля (данное событие самое эффективное в плане экономии ресурсов)
- *Удаление в таблице* - вычисление сработает при удалении записи в таблице.
- *Импорт в таблице* - вычисление сработает для импортируемых строк.

Следует пояснить, что вычисления сработают именно для тех строк, с которыми были проведены данные действия, остальные строки таблицы останутся без изменений.

ЗАДАНИЕ 3.

А. Создайте собственное вычисление с помощью конструктора в таблице “Тест”, (которую вы создавали в Уроке 2).

Необходимо, чтобы при создании новой записи, или изменении текущей Поле Сумма автоматически заполнялось значению равному полю Количество умноженному на поле Цена. Это необходимо сделать с помощью конструктора вычислений, с помощью шаблона “Математическая формула”.

Б. Добавьте в таблицу поля “Ставка НДС” и “Сумма НДС”.

Создайте допдействие “Рассчитать НДС”, которое будет:

1. выводить всплывающее окно, в котором можно будет ввести Ставку НДС для выбранной записи и сохранять это значение в поле Ставка НДС
2. Рассчитывать Сумму НДС, по формуле $\text{Сумма НДС} = \text{Сумма} * \text{Ставка НДС} / 100$ и записывать полученное значение в поле Сумма НДС

В. Зайдите в режим эксперта и ознакомьтесь с кодом созданных вами Вычисления и допдействия. Это необходимо для дальнейшего понимания принципа работы вычислений и допдействий.

Внимание! И вычисление и допдействие необходимо реализовать только с помощью конструктора без изменения кода в режиме эксперта.

Урок 4. Синтаксис кода вычислений

Вычисления на платформе Клиентской базы имеют определенный синтаксис.

Внимание! Для более удобной работы рекомендуется включить “Режим отображения разработчика” в списке полей таблицы в Настройках. В этом случае начнут отображаться все служебные названия полей, которые вы можете использовать в вычислениях.

Конфигурация > Лого и цвета > Пользователи > Доступ > Рассылка > Бэкап > Логи > Дополнения > Модули > Еще

Поля таблицы "Автомобили" (?)

Категория: Конструктор > Таблица: Автомобили > Доступ

Общие Поля (19) Форматирование (2) Фильтры Доп. действия (3) Вычисления (1) Внешние формы
Шаблоны печати Шаблоны рассылки Шаблоны СМС-рассылки Напоминания Подтаблицы (1)

Выберите режим отображения: Разработчика

Обновить порядок Добавить поле Добавить группу

1	id	\$line['id']	ID	число	★
2	user_id	\$line['user_id']	Кто добавил	пользователь	☰
3	add_time	\$line['add_time']	Время добавления	дата/время	☰
4	status	\$line['status']	Статус записи	список	☰
✗ 5	f5391	\$line['f5391']	Марка автомобиля	текст	
✗ 6	f5401	\$line['f5401']	Модель автомобиля	текст	
✗ 7	f5411	\$line['f5411']	Год выпуска	текст	

Итак, разберем основные моменты написания кода вычислений.

Формат типов полей

Как было описано ранее, в программе Клиентская база можно создавать 9 типов полей, также есть 4 служебных типа. Для каждого из этих типов используется свой формат хранения значений (не всегда он совпадает с отображением значения поля в режиме пользователя). При работе с вычислениями необходимо оперировать внутренним форматом хранения поля. Ниже кратко описан формат всех типов полей.

- “Текст” - формат хранения не отличается от представления в таблице
- “Число” - формат хранения дробного числа представлен в виде числа с точкой в качестве десятичного разделителя (в отличие от его представления в таблице, которое зависит от выбранного языка, например с запятой для русского)
- “Дата/время” - дата/время хранится в формате “Y-m-d H:i:s” (см. описание форматов для php-функции date), например “2018-12-31 23:55:00” для времени 23:55 31 декабря 2018 года.

- “Список” - формат хранения аналогичен текстовому полю, за исключением того, что записываемые в поле значения должны соответствовать набору значений поля. В случае множественного списка значения разделяются переводом строки, например “Первый\nВторой”.
- “Связь” - в поле связи хранится id связанной записи. При этом при обращении к полю связи через переменную \$line (см. описание данной переменной в следующем разделе) она выдает массив всех полей связанной записи, например \$line['Компания']['Телефон'], где Компания - собственно поле связи, а Телефон - поле связанной записи. В случае, если связанное поле представляет собой тоже поле связи, оно также раскрывается в массив и так далее.
- “Файл” и “Изображение” - формат работы с этими полями идентичен, за исключением того, что в поле с типом “изображение” можно сохранять только файлы изображений. В самом поле хранится только имя файла, сам файл хранится в папке files директории приложения. Для сохранения содержимого файла необходимо использовать специальные функции КБ для работы с файлами, их описание можно найти на code-stock.clientbase.ru
- “Пользователь” - в поле хранится id выбранного пользователя
- “Группа” - в поле хранится id выбранной группы доступа

Также в каждой таблице есть 4 служебных поля: ID, Кто добавил, Время добавления и Статус записи. Данные поля не рекомендуется менять (кроме Статус записи), только читать в случае необходимости. Формат полей Кто добавил и Время добавления аналогичен полям “пользователь” и “дата/время” соответственно. Формат поля Статус записи несколько отличается от обычного списка - там хранятся числовые коды соответствующих статусов. Это “0” для активных записей, “1” - для записей в архиве и “2” - для удаленных.

Глобальные переменные

Часто используемые в вычислениях:

- **\$line** - массив данных текущей строки, например, \$line['Сумма'], \$line['Цена'], \$line['Количество'], где Сумма, Цена и Количество это названия полей в текущей таблице;
- **\$user** - массив данных текущего пользователя (он расписан далее)
- **\$ID** - то же, что и \$line['ID'], т.е. ID текущей строки

Редко используемые, либо не используемые в вычислениях:

- **\$table** - массив настроек текущей таблицы
- **\$config** - массив глобальных настроек программы
- **\$lang** - массив локализации для текущего языка
- **\$smarty** - системный объект Smarty
- **\$ses_id** - ключ текущей сессии

В вычислениях нельзя переопределять глобальные переменные, такие как \$table. Например, код:

```
$table = 'Моя таблица';
```

приведет к ошибке программы.

При этом данные переменные могут быть прочитаны и использованы в вычислениях. Переопределение допускается только значений полей в переменной \$line.

Использование переменной \$user

Иногда появляется необходимость использования данных текущего пользователя в вычислениях. Для этого необходимо использовать переменную \$user, являющуюся массивом. Ниже приведено соответствие переменных и значений, которым они соответствуют:

- \$user['login'] - Логин пользователя;
- \$user['password']- Пароль пользователя. Стоит учесть, что пароль находится в хешированном виде;
- \$user['fio'] - ФИО пользователя;
- \$user['e_mail'] - E-mail пользователя;
- \$user['phone'] - Телефон пользователя (указывается в поле СМС настроек пользователя);
- \$user['group_id'] - Группа доступа пользователя;
- \$user['lang'] - Установленный язык пользователя;
- \$user['time_zone'] - Часовой пояс пользователя;
- \$user['last_time'] - Время последнего логина в программу;
- \$user['last_ip'] - IP с которого в последний раз подключался пользователь.

Примеры вычислений

Пример 1. Вычисление без использования данных из таблиц

\$line - это текущая строка, точнее переменная-массив с данными текущей строки.

Мы можем присвоить полям текущей строки любые данные:

```
$line['Дата'] = date("Y-m-d 00:00:00"); // записываем текущую дату  
$line['ФИО'] = 'Иванов Иван Иванович'; // записываем ФИО
```

Следует отметить, что для корректной работы вычисления необходимо совпадение типа поля, которое было создано в таблице и тех данных которые мы в него записываем. Например, данные по дате и времени следует

присваивать только соответствующим полям. Можно присвоить их и текстовому полю, но тогда они запишутся в виде текста.

Пример 2. Использование данных из текущей таблицы

Наиболее частым случаем является работа вычисляемых полей с данными из текущей таблицы. Наглядный пример представлен в таблице "Позиции счета". В ней в поле "Сумма" вносится число равное "Цене" умноженной на "Количество". Код этого вычисления выглядит так:

```
$line['Сумма'] = $line['Цена'] * $line['Количество'];
```

Пример 3. Использование данных из связанной таблицы

Допустим есть две таблицы: Компании и Заявки. И в той и в другой таблицы есть поле "Телефон", но заполнено оно только в таблице "Компании". Таблица "Заявки" имеет поле связи с таблицей "Компании". Нам требуется чтобы телефон копировался из таблицы "Компании" в таблицу "Заявки", если он заполнен в таблице "Компании".

Вычисление, размещенное в таблице "Заявки":

```
$line['Телефон'] = $line['Компания']['Телефон']; // где Телефон, Компания это поля  
таблицы Заявки
```

- `$line['Телефон']` это поле в таблице "Заявки", в которое требуется записать значение;
- `$line['Компания']['Телефон']` **это поле связи** в таблице "Заявки", где "Компания" это название поля, а "Телефон" это название поля в связанной таблице "Компании".

Подобное вычисление может работать и в обратную сторону, когда согласно записи в подтаблице определенное значение будет переноситься в основную таблицу (вычисление также размещено в таблице "Заявки"). Например, так:

```
$line['Компания']['Телефон'] = $line['Телефон'];
```

В вычислении используются те же поля, что и в предыдущем примере, но присвоение производится полю связи. Согласно этому вычислению, в родительскую таблицу "Компании" в поле "Телефон" будет внесено значение из поля "Телефон" подчиненной таблицы.

Пример 4. Добавление записи в таблицу

Для автоматического добавления записи в произвольную таблицу можно использовать функцию `insert_query`. В отличие от прямых запросов в БД, функция `insert_query`, во-первых, позволяет работать с внешними именами полей, а не с внутренними, типа "f235", во-вторых, автоматически заполняются служебные поля и значения по умолчанию, и в-третьих, автоматом запускаются имеющиеся вычисления в таблице (на события изменения полей).

Алгоритм использования функции такой:

1. Заполняем массив полей нужными нам данными
2. Добавляем новую запись через функцию `insert_query`

```
$data['Имя'] = "Петр";  
$data['Фамилия'] = "Иванов";  
insert_query($data, "Сотрудники");
```

Примечание: функция `insert_query` возвращает id добавленной записи, который можно записать в переменную и затем использовать в последующей обработке.

Аналогично, для обновления произвольной записи можно использовать функцию `update_query`, а для удаления - `delete_query`:

```
$data['Должность'] = "Старший менеджер";  
update_query($data, "Сотрудники", "`Фамилия`='Иванов'");
```

```
delete_query("Сотрудники", "`Фамилия`='Иванов'");
```

В примерах выше, параметр `"`Фамилия`='Иванов'"` - условие отбора записей

Дополнительные примеры

С более подробной информацией по примерам вычислений можно ознакомиться в документации

https://clientbase.ru/help/for_admin_16/calculations/#examples, однако повторимся, что самый большой набор примеров на данный момент находится в шаблонах конструктора вычислений.

ЗАДАНИЕ 4:

Создайте в таблице Тест текстовое поле ФИО (в котором будет заполняться Фамилия, Имя и Отчество) и Сокращенное ФИО.

Напишите в режиме эксперта вычисление, которое будет сокращать значения поля ФИО до Фамилии и первых букв Имени и Отчества (Например до Иванова И.И.) и записывать результат в поле “Сокращенное ФИО”.

Через конструктор данное задание на текущий момент реализовать невозможно.

Урок 5. Синтаксис кода допдействий

В уроке 2 мы рассмотрели как создавать допдействия (сокращенно ДД) с помощью конструктора. В данном уроке мы рассмотрим создание допдействий более подробно.

Типы допдействий

Еще раз напомним, что Допдействие это кнопка, которая отображается в таблице и в записи и которая может запускать заданное вычисление для одной записи, для нескольких выделенных записей, либо для всех записей.

Чаще всего допдействия изменяют данные в записях, либо добавляют новые записи в зависимости от условий.

Допдействия делятся на два типа:

- обычные допдействия
- допдействия со вводом параметров (и всплывающим окном)

В случае **обычного допдействия**, при нажатии на кнопку, отрабатывается код и выдается сообщение об успешном, либо неуспешном выполнении действия.

Пример:

```
$line['Принято'] = 'Да';  
$line['Дата выполнения'] = date("Y-m-d H:i:s");
```

Данное ДД устанавливает значение “Да” в поле “Принято” и текущую дату и время в поле “Дата выполнения”.

В случае **допдействия со вводом параметров**, при нажатии на кнопку у пользователя сначала появляется всплывающее окно, с возможностью ввести определенные параметры, а затем, после нажатия на кнопку “Отправить” выполняется сам код ДД, с учетом тех параметров, которые были введены. Также допдействия со вводом параметров могут служить только для ввода и сохранения параметров, без отработки дополнительного кода.

В конструкторе ввод параметров устанавливается через “Ввод параметров” - “Да” - > “Добавить новый параметр”.

Создание доп. действия таблицы "Автомобили"

Название доп. действия ? :

Режим цвета ? : ☒ Стандартный ☐ Палитра

Цвет кнопки ? : 

Режим ? : ☒ Режим конструктора ☐ Режим эксперта

Ввод параметров ? : ☐ Нет ☒ Да

Параметры ? :

Условия ? :

Шаблон ? :  

Пример кода ДД со вводом параметра:

```
if (isset($_REQUEST['rating'])) {  
    // производим нужные нам действия с необходимыми полями  
    $line['Оценка'] = $_REQUEST['rating'];  
    $line['Принято'] = 'Да';  
    $line['Дата выполнения'] = date('Y-m-d H:i:s');  
    // закрываем всплывающее окно  
    echo '<script>>window.opener.location.reload(); window.close();</script>';  
    exit;  
}  
// выводим в всплывающем окне форму для ввода параметров  
echo '<form method="get">';  
echo 'Введите оценку :';  
echo '<input name="id" type="hidden" value="" . $button_id . "" />'; // id кнопки  
echo '<input name="line_id" type="hidden" value="" . $ID . "" />'; // id текущей строки  
echo '<input name="rating" type="text" value="5" />'; // поле для оценки  
echo '<input type="submit" value="Оценить" />';  
echo '</form>';
```

Данное ДД устанавливает значение "Да" в поле "Принято" и текущую дату и время в поле "Дата выполнения".

Во всем остальном Синтаксис допдействий аналогичен синтаксису вычислений, рассмотренном в предыдущем уроке.

Более подробно о допдействиях можно прочитать в документации:

https://clientbase.ru/help/for_admin_16/dop/

ЗАДАНИЕ 5:

Создайте в таблице Тест поле Статус с типом “список”, с набором значений “Новый”, “В работе”, “Закрыт”.

Напишите в режиме эксперта доп.действие, которое будет устанавливать поле Статус в значение “Закрыт” и отправлять запись в архив (для отправки записи в архив необходимо установить значение поля “Статус записи” равным 1).

Урок 6. Создаем простые отчеты

В Клиентской базе есть возможность создавать страницы следующих типов:

- таблицы и записи
- календари
- представления (отчеты)

В данном уроке мы поговорим о представлениях. Представления служат для отображения данных в требуемом разработчику виде, которым чаще всего являются отчеты, например отчет по продажам.

Выглядит отчет следующим образом:

По продажам 

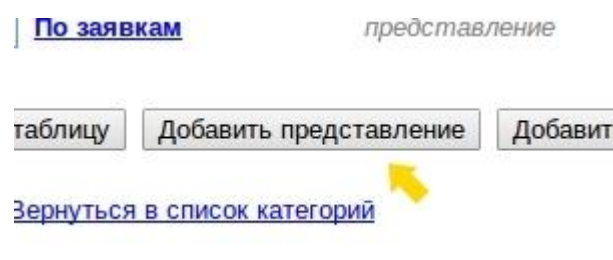
Экспорт в Excel  Печать отчёта

	Период с	по	Обновить
	27.08.2018	03.09.2018	

	Бонус, руб.	Выставлено счетов, руб.	Оплачено счетов, руб.
admin	0,00	0,00	0,00
Иванов Иван	0,00	0,00	0,00
Кротов Семен	0,00	0,00	0,00
Васильев Антон	0,00	0,00	0,00
Тихонова Мария	0,00	0,00	0,00
Головин Виктор	0,00	0,00	0,00
Петренко Федор Иванович	0,00	0,00	0,00
codeg	0,00	0,00	0,00
Итого	0,00	0,00	0,00

Код представления состоит из двух частей — подготовки данных для отображения и собственно отображения полученных данных. Первая часть пишется на языке PHP с извлечением при необходимости данных из БД с помощью MySQL запросов, и их обработки. Вторая часть пишется в формате шаблона Smarty, основой которого является HTML.

Для создания представления необходимо зайти в меню Настройки => вкладка Конфигурация, после этого выберите нужную вам категорию, в которой Вы хотите создать представление/отчет и нажмите на кнопку "Добавить представление"



После нажатия кнопки "Добавить представление", Вы перейдете в окно редактирования представлений. Первым, что необходимо сделать, это задать имя создаваемого представления.

Для того чтобы извлечь данные из таблиц программы Клиентская база, необходимо знать структуру базы данных, в частности как обращаться к конкретным таблицам и полям. Физически в базе они хранятся под номерами, а не под "русскими" именами. Например, таблица Контрагенты в стандартной конфигурации имеет номер 42, поле Название (название компании) в той же таблице — номер 435. Их истинные имена выглядят как <prefix>data42 и f435 соответственно, где <prefix> — текущий префикс таблиц. Таким образом запрос на выборку данных о контрагенте с названием «Аргумент» будет выглядеть так:

```
$sqlQuery = "SELECT * FROM ".DATA_TABLE."42 WHERE f435='Аргумент'";  
$result = sql_query($sqlQuery); // аналог функции mysql_query  
$row = sql_fetch_array($result); // аналог функции mysql_fetch_array  
$company['name'] = $row['f435']; // поле Название  
$company['email'] = $row['f442']; // поле E-mail
```

Здесь DATA_TABLE — константа, равная <prefix>data (текущий префикс берется из файла config.php, он задается при установке программы). Номера полей и таблиц можно посмотреть в Настройках соответствующих таблиц и полей, в режиме отображения разработчика.

Далее нужно передать их в шаблон отображения через метод \$smarty->assign. Делается это так:

```
$smarty->assign("data", $data);
```

здесь \$data — передаваемая в шаблон Smarty переменная, "data" — имя переменной внутри шаблона

Полученный код необходимо разместить в поле "Подготовка данных".

Данные мы подготовили, теперь необходимо их отобразить. Для этого нужно написать код шаблона отображения на Smarty с подстановкой переданных данных.

Подробную инструкцию по созданию шаблонов Smarty вы можете найти здесь:

<http://www.smarty.net/manual/ru/>

Опять же самым простым способом изучить принцип работы отчетов в КБ и создать собственные является просмотр кода существующих отчетов и их доработка под собственные задачи. Посмотреть код отчета можно в Настройках программы, выбрав нужный отчет и нажав на ссылку “Редактировать” подготовку данных или отображение.

Конфигурация > Лого и цвета > Пользователи > Доступ > Рассылка > Бэкап > Логи > Дополнения > Модули > Еще

Редактирование представления "По продажам"

Категория ? : Отчеты ▼

Имя ? : По продажам

Подготовка данных ? : [Редактировать](#)

Отображение ? : [Редактировать](#)

Справка ? :
Отчет по продажам показывает сколько и на какую сумму за выбранный период было выставлено счетов и сколько оплачено. В отчете используются данные из таблицы Счета.

Отправлять по e-mail ? : ☐

[Перейти в доступ](#)

Сохранить Отмена

Подробнее об отчетах https://clientbase.ru/help/for_admin_16/presentation/

Задание 6.

А. Доработайте “Отчет по продажам” в Стандартной конфигурации программы таким образом, чтобы в отчете появились два дополнительных столбца “Количество выставленных счетов” и “Количество оплаченных счетов”, а также столбец “Средний чек”, который показывал бы среднюю сумму по оплаченным счетам.

Б. По аналогии с “Отчетом по продажам” создайте собственный отчет “Тестовый отчет” на основе таблицы “Тест”, которую вы создали. (Отчет по продажам берет данные из таблицы Счета, а ваш отчет должен брать данные из таблицы Тест).

Урок 7. Внутренние функции КБ

Для корректной работы кода любого типа (вычислений, допдействий, отчетов), требуется использовать внутренние функции Клиентской базы, а не стандартные функции PHP/MySQL.

Используя внутренние функции вы:

- избегаете ошибок с уязвимостями
- избегаете ошибок связанными с конфликтами с другими инструментами
- поддерживаете правильную работу логов, синхронизации и т.д.

Частые внутренние функции КБ

Название	Описание	Параметры	Примеры
data_select	Выборка данных из таблицы dataN (запрос "SELECT")	<code>\$table_id</code> - id таблицы 2й и последующие параметры - условия отбора записей	<code>\$result = data_select(111, "id=", 2);</code>
data_update	Обновление строк в таблице dataN (запрос "UPDATE").	<code>\$table_id</code> - id таблицы <code>\$data</code> - массив данных (ключи - имена полей) 3й и последующие параметры - условия отбора записей Вторым аргументом может быть специальное значение <code>EVENTS_ENABLE</code> , в этом случае при обновлении будут запущены вычисления при изменении полей.	<code>// без вычислений</code> <code>data_update(111, array('f22'=>'Измененный клиент'), "id=", 2);</code> <code>// с запуском вычислений</code> <code>data_update(111, EVENTS_ENABLE, array('f22'=>'Измененный клиент'), "id=", 2);</code>
data_insert	Вставка строки в таблицу dataN (запрос "INSERT"). Возвращает id добавленной записи.	<code>\$table_id</code> - id таблицы <code>\$data</code> - массив данных (ключи - имена полей)	<code>// без вычислений</code> <code>\$new_id = data_insert(111, array('f22'=>'Новый клиент'));</code>

		Вторым аргументом может быть специальное значение EVENTS_ENABLE, в этом случае при добавлении будут запущены вычисления и заполнены значения по умолчанию для полей.	// с запуском вычислений \$new_id = data_insert(111, EVENTS_ENABLE, array('f22'=>'Новый клиент'));
--	--	--	--

С полным списком внутренних функций, на текущий момент их 76, вы можете ознакомиться по ссылке <https://code-stock.clientbase.ru/fields.php?table=50>
Чтобы получить доступ к программе, пройдите регистрацию по ссылке: <https://code-stock.clientbase.ru/questionare.php?id=10> и дождитесь письма-приглашения.

Задание 7.

Доработайте вычисление в режиме эксперта используя внутренние функции КБ.

Создайте таблицу “Тест 2” с числовыми полями Курс доллара и Курс евро и Дата (поле типа дата). Добавьте 3 записи с тремя разными датами и разными курсами.

В таблице Тест добавьте поле типа список “Валюта” со значениями Доллар и Евро. Также добавьте поле “Текущий Курс”.

Напишите вычисление которое заполняет поле “Текущий курс” беря значения из таблицы “Тест 2”.

Урок 8. Структура базы данных

Знание структуры базы данных не столь необходимо, в случае использования внутренних функций, однако в некоторых случаях, ее использование напрямую возможно, поэтому в данном уроке мы дадим информацию по ней.

Начнем с того, что в каждом аккаунте в корневом каталоге есть скрипт `edit_sql.php`, который предоставляет прямой доступ к базе данных. Данный файл доступен только после авторизации под администратором.

Внимание! Стоит сразу отметить, что просмотр, а тем более изменение данных напрямую через базу данных является скорее форс-мажорной ситуацией. В подавляющем большинстве основных случаев следует пользоваться стандартным функционалом настроек.

Язык: Русский язык

MySQL » Сервер » База данных: c_const

Выйти

ClientBase 4.2.5

База данных: c_const

[Изменить базу данных](#) [Схема базы данных](#) [Полномочия](#)

Таблицы и представления

Поиск в таблицах (191)

выбрать f_acc_buttons

выбрать f_acc_buttons_par

выбрать f_acc_calendars

выбрать f_acc_calendars_par

выбрать f_acc_cats

выбрать f_acc_cats_par

выбрать f_acc_fields

выбрать f_acc_fields_par

выбрать f_acc_filters

выбрать f_acc_filters_par

выбрать f_acc_forms

выбрать f_acc_forms_par

выбрать f_acc_messages

выбрать f_acc_reports

выбрать f_acc_reports_par

выбрать f_acc_reports_users

выбрать f_acc_rules

выбрать f_acc_subadmin

выбрать f_acc_subadmin_rules

выбрать f_acc_sync

выбрать f_acc_tables

выбрать f_acc_tables_par

выбрать f_acc_tips

выбрать f_acc_tips_par

выбрать f_acc_tips_buttons

Таблица	Тип	Режим сопоставления	Объем данных	Объем индексов	Свободное место	Автоматическое приращение	Строк	Комментарий
☐ f_acc_buttons	InnoDB	utf8_general_ci	16 384	32 768	0	702	~ 58	
☐ f_acc_buttons_par	InnoDB	utf8_general_ci	16 384	32 768	0	1	0	
☐ f_acc_calendars	InnoDB	utf8_general_ci	16 384	32 768	0	31	~ 3	
☐ f_acc_calendars_par	InnoDB	utf8_general_ci	16 384	32 768	0	1	0	
☐ f_acc_cats	InnoDB	utf8_general_ci	16 384	32 768	0		~ 32	
☐ f_acc_cats_par	InnoDB	utf8_general_ci	16 384	32 768	0		0	
☐ f_acc_fields	InnoDB	utf8_general_ci	98 304	147 456	0	10 702	~ 1 544	
☐ f_acc_fields_par	InnoDB	utf8_general_ci	16 384	49 152	0	1	0	
☐ f_acc_filters	InnoDB	utf8_general_ci	16 384	49 152	0	2 422	~ 91	
☐ f_acc_filters_par	InnoDB	utf8_general_ci	16 384	49 152	0	1	0	
☐ f_acc_forms	InnoDB	utf8_general_ci	16 384	49 152	0	1 132	~ 85	
☐ f_acc_forms_par	InnoDB	utf8_general_ci	16 384	49 152	0	1	0	
☐ f_acc_messages	InnoDB	utf8_general_ci	16 384	16 384	0	3	~ 2	
☐ f_acc_reports	InnoDB	utf8_general_ci	16 384	32 768	0	182	~ 23	

Назначение таблиц можно понять из их названия. Опишем основные таблицы, которые вам могут понадобиться.

Основные таблицы

cb_acc_* - таблицы доступов к разным элементам программы (фильтры, отчеты, таблицы и все остальное)

cb_tables - таблица с данными всех Таблиц в КБ, которые отображаются пользователям

cb_fields - таблица с данными всех Полей в таблицах cb_data*

cb_data* - таблицы с данными (для вывода данных, расчетов и прочего)

cb_users - таблица пользователей (для вывода пользователей, добавление пользователей)

cb_calc - таблица вычислений (для быстрого поиска вычислений)

cb_cron - таблица cron, там cron задачи (для быстрого поиска cron задач или создание cron задачи из кода)

cb_reports - таблица отчетов

cb_config - таблица для хранения каких либо конфигурационных данных (вкл/выкл каких нибудь параметров)

Задание 8.

Зайдите в `edit_sql` и найдите таблицы "Тест" и "Тест 2". Ознакомьтесь со структурой полей. Сопоставьте поля в базе данных Mysql с полями в таблицах КБ.

Урок 9. Рекомендации по написанию кода

Стандарт кода

Настоятельно рекомендуется, при написании кода следовать общепринятому стандарту PSR, в частности по следующим пунктам:

- Все блоки кода в фигурных скобках необходимо писать с отступами. Для отступов должны использоваться четыре пробела (но не знак табуляции).
- Открывающая фигурная скобка в определении функции должна располагаться на новой строке, а закрывающая фигурная скобка должна располагаться на следующей строке после тела функции.
- Открывающая фигурная скобка в управляющих конструкциях должна располагаться в той же строке, что и сама конструкция, а закрывающая фигурная скобка должна располагаться на следующей строке после тела конструкции.
- После ключевых слов в управляющих конструкциях должен располагаться один пробел, а после вызовов функций – не должен.
- После открывающей круглой скобки и перед закрывающей круглой скобкой в управляющих конструкциях не должно быть пробела.
- В коде можно использовать только теги
- Кодировка php файлов должна быть UTF-8 без BOM-байта.

Пример:

```
function my_function()
{
    foreach ($arr as $elem) {
        $a = 2;
        if ($elem > 2) {
            $b = 3;
            $c = 4;
        }
    }
}
```

Безопасность кода

Ввод данных

Все входящие данные `$_REQUEST`, `$_GET`, `$_POST`, `$_FILES`, `$_COOKIE` и т. п. должны фильтроваться. Если в вычислении ожидается ввод определенного типа данных, то должны быть введены данные именно этого типа.

Числовые параметры

```
$table_id = intval($_REQUEST['table']); // число  
$table_id = floatval($_REQUEST['table']); // число с точкой
```

Чекбоксы

```
$is_summa = $_REQUEST['summa']?1:0;
```

Даты

```
$end_date = form_eng_time($_REQUEST['end_date']);
```

Строки

Строки это самый сложный вид фильтрации. При вводе в скрипт, строка должна проходить очистку от принудительно расставленных кавычек (в случае использования `magic_quotes`), с помощью функции `form_input`.

```
$fio = form_input($_REQUEST['fio']);
```

В случае использования строкового значения в функции `sql_query`, обязательно экранировать значения с помощью `addslashes`.

Изменение данных

Все изменение данных, а также ввод новых, должно быть защищено от атак типа CSRF. Ввод данных, изменение, создание, удаление чего-либо, любое действие, приводящие к изменению состояния, должно отправляться с помощью метода POST. В случае использования форм ввода, следует использовать в шаблоне Smarty: `<input type=hidden name=csrf value='{$_csrf}'>`. В случае если действие производится по ссылке, в ссылку необходимо добавить класс: `class='href_post'`. В случае если в ссылке есть свой обработчик `onclick`, следует использовать переменную `href_post` для предотвращения отправки формы.

Примеры

// Форма ввода логина и пароля с защитой от CSRF атак

```
<form action="login.php" method="post">
<input type="text" name="password">
<input type="text" name="login" >
<input type="submit" value="Отправить">
<input type="hidden" name="csrf" value="{$_csrf}">
</form>
```

// Ссылка удаляющая поле таблицы

```
<a class='href_post'
href='edit_field.php?table=42&field=442&delete'>Удалить</a>
```

Более подробная информация

https://clientbase.ru/help/for_admin_16/oformlenie_php_koda/#security

Задание 9.

Создайте доп.действие в всплывающем окне с формой для ввода и сохранением полученных данных в полях таблицы, с учетом вышеперечисленных мер безопасности (метод POST, защита от CSRF, фильтрация полученных данных перед их сохранением).

Заключение

Благодарим Вас за прохождение курса разработчика КБ. Мы очень надеемся, что данный курс был вам полезен и поможет в дальнейшем в использовании платформы КБ для разработки клиентских решений.

В завершении мы просим Вас заполнить анкету о прохождении курса
<https://shkb.clientbase.ru/questionare.php?id=21>

После проверки ваших заданий Вам на почту будет выслан сертификат от компании Клиентская база о прохождении курса разработчика.

Вы можете смело писать что вам понравилось, и главное что не понравилось, либо не хватило в данном курсе. Мы постараемся учитывать ваши замечания и улучшать данный курс в соответствии с вашими запросами.

В случае, если вы хотите связаться с нами, пишите нам на info@clientbase.ru и мы постараемся Вам помочь.

Желаем успехов!