МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Дальневосточный федеральный университет»**

|  |
| --- |
| **ШКОЛА ЕСТЕСТВЕННЫХ НАУК** |

**Кафедра информационных систем управления**

Котик Софья Валентиновна

|  |
| --- |
| РАЗРАБОТКА БАЗЫ ДАННЫХ ДЛЯ АДМИНИСТРИРОВАНИЯ ЗАЛА РЕСТОРАНА |

**КУРСОВОЙ ПРОЕКТ**

|  |  |
| --- | --- |
| Студентка гр. Б8319 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
| Руководитель ст. преподаватель | |
|  | Л.В.Красюк |
|  | |
|  | |
| Регистрационный № \_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись И.О.Фамилия  « \_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018г. | Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись И.О.Фамилия  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018г. | |

г. Владивосток

2018

**Оглавление**

[Введение 3](#_Toc517832474)

[1 Предметная область 4](#_Toc517832475)

[2 Анализ предметной области 8](#_Toc517832476)

[3 Обоснование решаемой задачи 16](#_Toc517832477)

[3.1 Постановка задачи 16](#_Toc517832478)

[3.2 Описание автоматизированной работы зала ресторана 18](#_Toc517832479)

[4 Инфологическая модель проектируемой базы данных 21](#_Toc517832480)

[5 Логическая модель данных 24](#_Toc517832481)

[6 Физическая модель базы данных и ее реализация 24](#_Toc517832482)

[6.1 Роли и подсистемы 25](#_Toc517832483)

[6.2 Справочники системы 29](#_Toc517832484)

[6.3 Документы и регистры системы 30](#_Toc517832485)

[6.4 Перечисления, используемые в системе 38](#_Toc517832486)

[6.5 Отчеты, создаваемые системой 39](#_Toc517832487)

[6.6 Формы 41](#_Toc517832488)

[Заключение 45](#_Toc517832489)

[Список литературы 46](#_Toc517832490)

[Приложение А 47](#_Toc517832491)

[Руководство пользователя для роли «Администратор» 47](#_Toc517832492)

[Руководство пользователя для роли «Бухгалтер» 70](#_Toc517832493)

[Приложение Б 74](#_Toc517832494)

[Справочник «Официанты». Модуль формы элемента 74](#_Toc517832495)

[Справочник «Общее меню». Модуль формы элемента 74](#_Toc517832496)

[Документ «Приходная накладная». Модуль формы элемента 75](#_Toc517832497)

[Документ «Приходная накладная». Модуль объекта 75](#_Toc517832498)

[Документ «Приходная накладная». Модуль менеджера 76](#_Toc517832499)

[Документ «Расходная накладная». Модуль объекта 77](#_Toc517832500)

[Документ «Расходная накладная». Модуль формы элемента 78](#_Toc517832501)

[Документ «Расходная накладная». Модуль менеджера 78](#_Toc517832502)

[Документ «Бронирование». Модуль формы элемента 79](#_Toc517832503)

[Документ «Бронирование». Модуль объекта 80](#_Toc517832504)

[Документ «Бронирование». Модуль формы элемента 80](#_Toc517832505)

[Документ «Бронирование». Модуль объекта 82](#_Toc517832506)

[Документ «Бронирование». Модуль менеджера 82](#_Toc517832507)

[Документ «Чек». Модуль объекта 83](#_Toc517832508)

[Документ «Чек». Модуль менеджера 83](#_Toc517832509)

[Документ «График». Модуль формы элемента 84](#_Toc517832510)

[Документ «График». Модуль менеджера 87](#_Toc517832511)

[Приложение В 90](#_Toc517832512)

**Введение**

Цель курсового проекта– создание автоматизированной системы для улучшения работы ресторана.

Объектом анализа данной курсового проекта является деятельность ресторана.

Предметом исследования является деятельность администратора ресторана.

В указанной области ресторана фигурирует большое количество данных, например, данные об официантах, о клиентах, о загруженности зала, бронировании и так далее. Вся эта информация хранится в бумажном виде, что очень замедляет работу администраторов зала, которым необходимо затрачивать огромное количество времени и ресурсов для организации основной работы зала.

Указанные проблемы являются актуальными и требуют соответствующего решения. Следовательно, задачи, которые необходимо решить:

1) автоматизировать организацию бронирования;

2) автоматизировать проведение мероприятия;

3) автоматизировать создание отчетности;

4) автоматизировать расчеты;

5) автоматизировать составление графика работы официантов;

6) создать единую базу данных.

После определения задач, которые будут решаться системой, ее необходимо спроектировать. В данной работе использовались методология SADT (семейство стандартов IDEF) и нотация Питера Чена.

Завершающий этап разработки программного продукта – это его реализация при помощи программного средства 1С: Предприятие 8.3.

После создания и внедрения автоматизированной системы администратор сможет избавиться от ручного заполнения данных, ускорить организацию мероприятий и бронирования, а также составление графика работы официантов.

**1 Предметная область**

Ресторан — предприятие общественного питания с широким ассортиментом блюд сложного приготовления, и различным персоналом.

Общая организационная структура ресторана представлена на рисунке 1.



Рисунок 1 – Оргструктура ресторана

Время работы ресторана с 10:00 – 00:00, семь дней в неделю.

Далее подробно рассмотрена часть ресторана, а именно реализация работа в зале.

Зал ресторана имеет общую зону, которая состоит из 20 столиков по 4 – 6 посадочных мест.

Максимальная вместимость зала – примерно 120 человек.

Количество персонала:

1) администраторы зала – 2 человека;

2) официанты – 20 человек;

3) бухгалтер – 1 человек.

Администратор зала является неотъемлемой составляющей в хорошей работе любого ресторана. Это многофункциональный человек, который должен с большой ответственностью подходить к своей работе, и особенно к обслуживанию клиентов. Помимо всего он ведет работу с официантами – работниками предприятия общественного питания, обслуживающих посетителей.

В основные функции администратора входят:

1) организация бронирования столиков для клиентов. Администратор запрашивает личные данные клиента (ФИО, номер телефона), оговаривает с ним необходимое количество мест, дату, время, длительность;

2) организация мероприятий. Администратор с клиентом оговаривают вид мероприятия, размещение (выбор столов), количество гостей и меню (необходимо присутствие шеф-повара), далее составляется план мероприятия, где подсчитывается итоговая сумма;

4) график работы для официантов. Администратор на основе графика за прошлую неделю и загруженности ресторана (назначенные на предстоящую неделю мероприятия) разрабатывает график на следующую неделю;

5) подсчет выручки после обслуживания клиентов, в конце рабочего дня;

6) составление списка официантов для получения премиальных на основе показателей труда за неделю (общая выручка за неделю по одному официанту должна быть больше или равна 14000) и участия в мероприятиях (график работы официантов).

В обязанности официантов входят:

- выполнение заказов посетителей;

- получение от них расчёта за услуги;

- уборка столов после ухода посетителей;

- сервировка столов к приходу новых посетителей.

Ниже представлены алгоритмы по каждой функции администратора зала.

Клиент может позвонить в ресторан или обратиться напрямую к администратору, чтобы забронировать столик. Администратор оповестит его о наличии мест в ресторане. Информация по загруженности ресторана хранится у администратора в журнале «Загруженность ресторана». Если мест нет, клиента оповещают о невозможности бронирования столиков, если места есть, то администратор попросит сообщить необходимое количество мест, дату, время и длительность трапезы. Далее администратор предложит возможный вариант размещения.

Если клиенту не подходит данный вариантов, то заказ на бронирование аннулируется или администратор предлагает другой вариант размещения. Если же клиента все устраивает, то ему сообщается номер столика и запрашиваются его контактные данные (ФИО и номер телефона). Всю полученную информацию администратор заносит в журнал загруженности.

Если клиент хочет организовать мероприятие, то ему необходимо напрямую обратиться в ресторан. Возможность организовать мероприятие зависит от загруженности ресторана (журнал «Загруженность ресторана»), его вместимости и видов банкетных мероприятий (банкет с полным или частичным обслуживанием, фуршет, бизнес-ланч, кофе-брейк). Все планируемые мероприятия заносятся в отдельный журнал «Мероприятия», но предварительно вся информация фиксируется на черновике. Администратор приглашает шеф-повара, а затем администратор попросит клиента сообщить его личные данные (ФИО и номер телефона), дату, время, длительность и количество гостей, чтобы выбрать подходящие столики. Если клиента все устраивает, то переходят к меню, если нет – выбирается другой вариант размещения или заказ на мероприятие отменяется. Далее администратор должен узнать, какие блюда предпочтительны для клиента (рыбную, мясные, овощные и так далее), нет ли аллергии на что-то, после чего совместно с шеф-поваром предлагают блюда из общего меню, с учетом предпочтений клиента. Поставка продуктов осуществляется кухней отдельно от поставки для общего меню. Клиент, администратор и шеф-повар составляют список продуктов (наименование, вес (кг), цена за кг продукта). Составляется план мероприятия, и подсчитывается общая стоимость. При удачном раскладе, копия плана отдается клиенту и производится оплата, если клиента что-то не устраивает, то в план вносятся изменения или он может полностью отказаться от организации мероприятия.

У администратора имеются данные по каждому из официантов (ФИО, номер телефона, привязанные столы). При составлении нового графика, администратор анализирует график за прошлую неделю, если имеются официанты, которые участвовали в мероприятиях или общая выручка одного официанта по чекам за неделю была больше 14000 рулей, то они вносятся в список для получения премии (список «Сведения о премиальных» передается бухгалтеру), если таких работников нет, то администратор переходит к составлению графика работы для официантов. График делится следующим образом, имеются четные (вторник, четверг, суббота) и нечетные (понедельник, среда, пятница, воскресенье) дни недели, а также разделение на дневную (10:00-17:00) и вечернюю (17:00-00:00) смены. Администратор опросит официантов, чтобы узнать предпочтительное для них время работы на следующую неделю. Далее распределит работников с учетом предпочтений, при этом те официанты, которые работали по нечетным дням недели не могут быть включены на эти же дни, и наоборот. После распределения официантов составляется график работы (указываются дни недели, даты, смены, ФИО официантов, номера столиков, закрепленные за каждым из них). Если на следующую неделю запланированы мероприятия (журнал «Мероприятия»), то имеющиеся официанты распределяются между ними. Работники, выделенные для мероприятий, выносятся в примечание графика работы. График вывешивается на общем стенде, любой из официантов может попросить у администратора зала сделать его копию.

С учетом загруженности столиков и графика работы официантов, официанты сервируют столы, обслуживают клиентов, получают от них расчеты и убирают столы. После обслуживания чек сразу отдается администратору. В конце рабочего дня по всем чекам за день подсчитывается выручку и составляется отчет о выручке для бухгалтера.

**2 Анализ предметной области**

В первую очередь необходимо проанализировать предметную область и найти проблемы, которые будут решаться за счет автоматизации работы зала ресторана.

Для начала надлежит представить описанную ранее предметную область при помощи диаграмм AS-IS, то есть «такими, какие они есть».

Модель IDEF0 предназначена для формализации и описания бизнес-процессов.

Самый верхний уровень модели IDEF0 – контекстная диаграмма, представленная на рисунке 2, которая является общим описанием бизнес-процессов с логическим отношениями между функциями администратора. Интерфейсные дуги данной диаграммы отображают различные объекты, которые необходимы для того или иного процесса, а также частично описывают его.

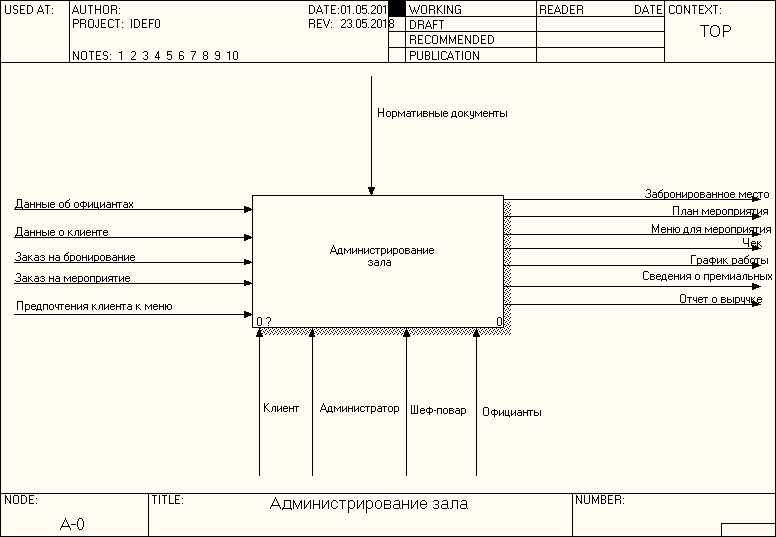


Рисунок – Контекстная диаграмма предметной области, модель IDEF0

Исходя из построенной контекстной диаграммы можно увидеть является входными, выходными или управляющими данными, а также механизмы.

Далее необходимо составить декомпозиция контекстной диаграммы для представления работы зала, чтобы избежать затруднения в анализе предметной области. При этом один единственный процесс разбивается на подпроцессы, которые описывают отдельные функции администратора зала.

Декомпозированная контекстная диаграмма представлена на рисунке 3.

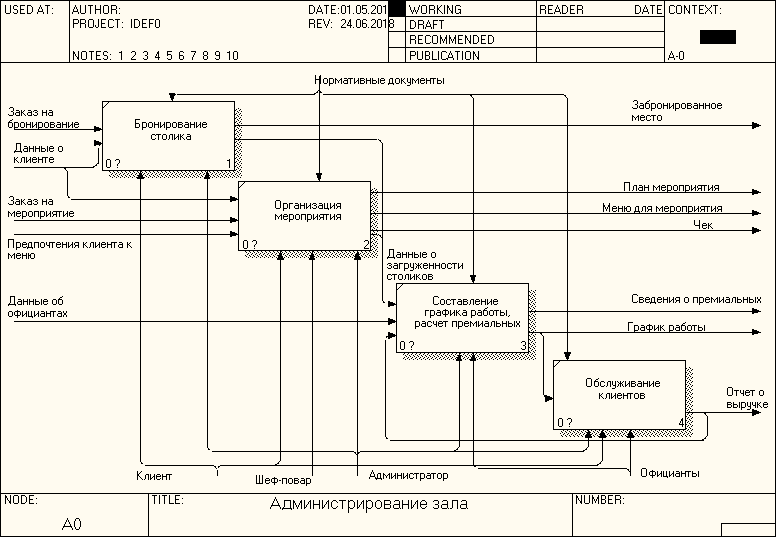


Рисунок – Декомпозиция контекстной диаграммы, модель IDEF0

В ходе анализа предметной области были выделены следующие ключевые бизнес-процессы:

1) бронирование столика;

2) организация мероприятия;

3) составление графика работы, расчет премиальных;

4) обслуживание клиентов.

Для того, чтобы более детально углубиться в проблемы анализируемой предметной области была произведена повторная декомпозиция, но уже по методологии IDEF3, чтобы отобразить технологию выполнения каждого бизнес-процесса.

Диаграмма декомпозиции процесса «Бронирование столика» изображена на рисунке 4.

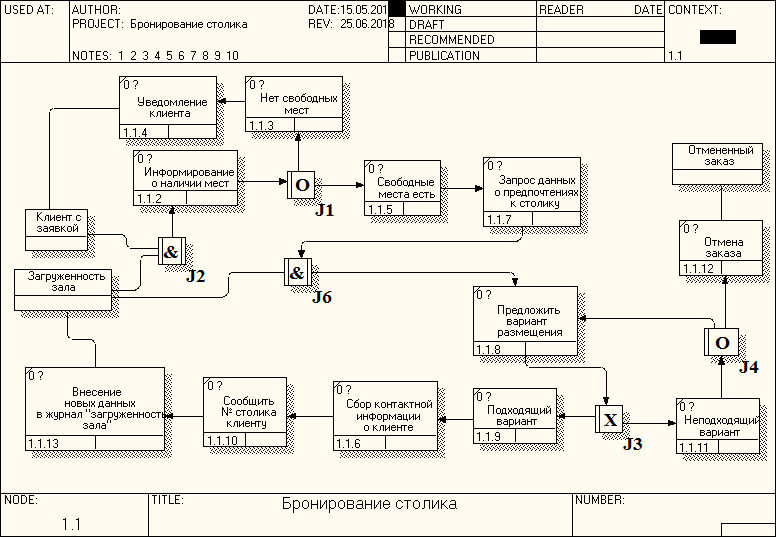


Рисунок – Декомпозиция «Бронирование столика», модель IDEF3

В данной диаграмме содержится множество проблем, во-первых, при информировании клиента о наличии мест, администратор вручную просматривает данные по загруженности зала, во-вторых, при сборе контактной информации администратор заносит данные вручную, что занимает не мало времени. Не исключается возможность допущения ошибок, например, дважды указывается один и тот же стол для бронирования.

Следующий процесс, который был декомпозирован – «Организация мероприятий», рисунок 5.

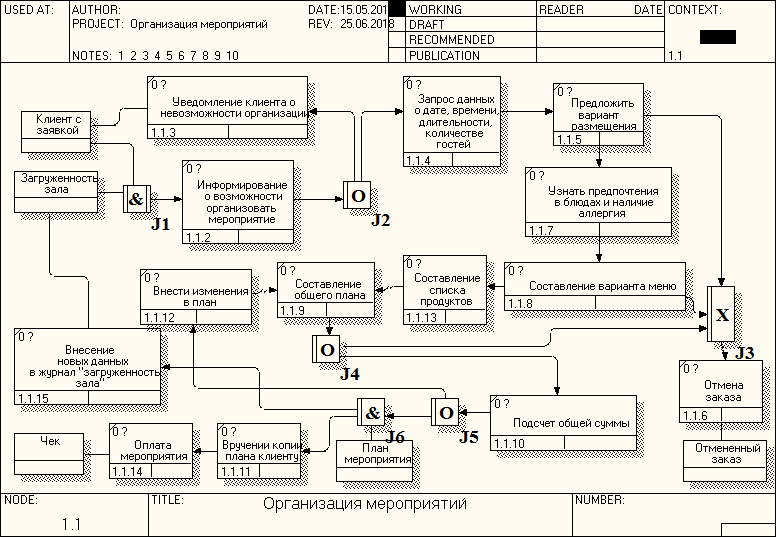


Рисунок – Декомпозиция «Организация мероприятий», модель IDEF3

При декомпозиции данного процесса видно, что затрачивается очень много усилий и времени для заказа мероприятия клиентом. Во-первых, поиск данных о загруженности зала, во-вторых, ручное составление меню и списка продуктов, в-третьих, внесение изменений в план, подсчет общей суммы мероприятия и, в-четвертых, повторное составление плана, но уже на чистовике.

Следующий процесс для декомпозиции – «Составление графика работы, расчет премиальных», представлен на рисунке 6.

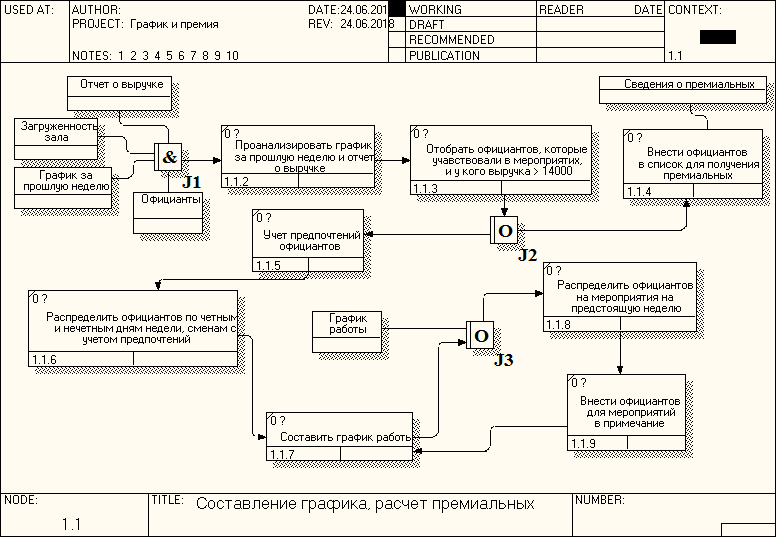


Рисунок – Декомпозиция «Составление графика, расчет премиальных», модель IDEF3

Составление графика вызывает множество проблем, которые связаны с возможным дублированием информации, неправильным распределением официантов, затратой большого количества времени.

И последний процесс для декомпозиции – это «Обслуживание клиентов», представленная на рисунке 7.

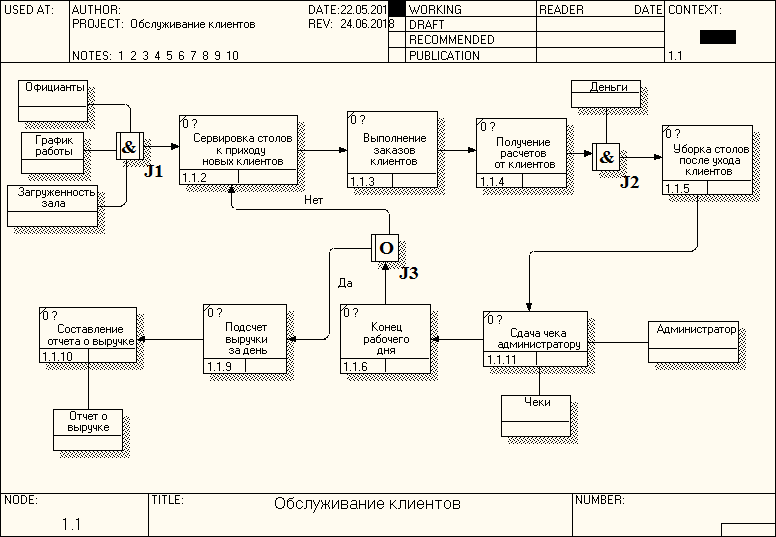


Рисунок – Декомпозиция процесса «Обслуживание клиентов», модель IDEF3

Проблемой данного процесса является большая вероятность ошибки при вычислении общей выручки за день.

Приведенные выше диаграммы отображают только технологию каждого бизнес-процесса, а для того чтобы проследить движение потоков информации была использовала диаграмма методологии DFD, по которой можно создать базу данных. Контекстная диаграмма с указанными источниками и получателями данных приведена на рисунке 8.

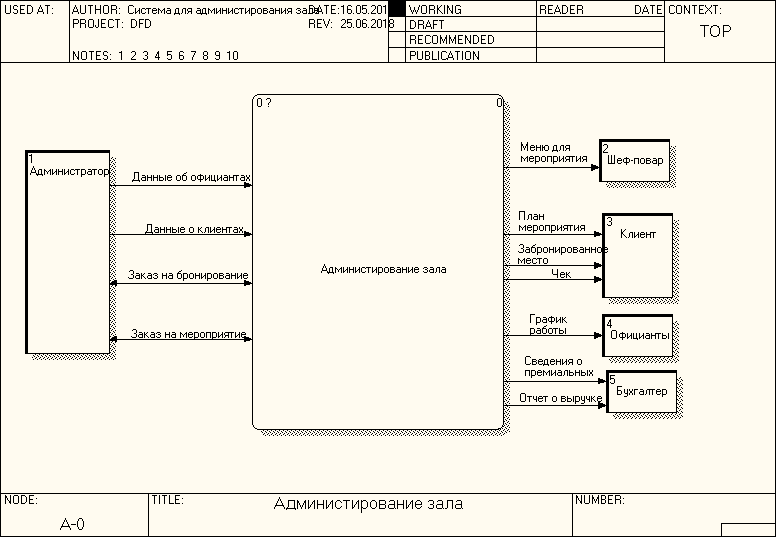


Рисунок – Контекстная диаграмма «Администрирование зала», модель DFD

В качестве источников и получателей данных, то есть внешних объектов, выступают администратор, шеф-повар, клиенты, официанты, бухгалтер.

Для определения хранилищ данных, то есть внутренних объектов, используется декомпозиция контекстной диаграммы, рисунок 9.

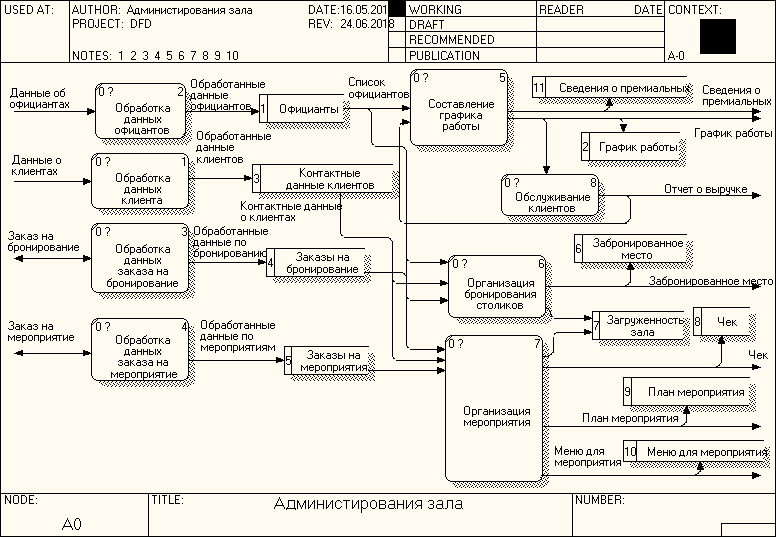


Рисунок – Декомпозиция контекстной диаграммы «Администрирование зала», модель DFD

По последней декомпозиции можно заметить множество проблем, связанных с ведением и хранением данных, в виду того, что имеется около 11 хранилищ данных, то есть своеобразных журналов, бланков, списков.

По итогу анализа можно выделить ряд следующих проблем:

1) использование бумажных носителей, которые занимают много времени для заполнения, редактирования, копирования и последующей работы с данными;

2) дублирование данных;

3) ручные расчеты;

4) человеческий фактор, который не исключает ошибки;

5) избыток данных.

3 Обоснование решаемой задачи

По результатам анализа предметной области и в виду того, что огромное количество информации хранится на бумажных носителях, целесообразно будет создать автоматизированную систему.

3.1 Постановка задачи

Целями создания автоматизированной системы администрирования зала являются упрощение создания, редактирования и удаления данных, поиска нужной информации, а также хранения данных.

Будущая система позволит автоматизировать следующие процессы:

1) работа с данными официантов;

2) работа с данными клиентов;

3) учет загруженности зала;

4) организация бронирования столиков;

5) организация мероприятий для клиентов с составлением меню;

6) учет продуктов (хранение приходной и расходной накладных, остатки);

7) подсчет общей суммы мероприятия;

8) создание графика работы для официантов;

9) подсчет выручки за день, создание отчета для бухгалтера;

10) создание списка официантов для бухгалтера, сведения о премиальных.

Пользователи системы:

- администраторы зала, которые работают посменно (две смены по 7 часов) и являются также и системными администраторами;

- бухгалтер, который работает с 10:00-18:00.

Создаваемая система должна состоять из следующих функциональных подсистем:

1) бронирование;

2) мероприятие;

3) график работы;

4) продукты;

5) отчетность;

6) справочники.

Входной информацией для системы являются:

1) данные о количестве столиков в зале (наименование столов) и его вместимости (количество мест за каждым столом);

2) данные по каждому официанту, которые вносятся по требованию администратора (ФИО, номер телефона, привязанные столы);

3) контактные данные клиента, которые вносятся по требованию администратора (ФИО, номер телефона);

4) чеки после обслуживания клиентов;

5) данные о поступающих на склад продуктах;

6) данные о продуктах для составления меню мероприятий;

7) данные для бронирования;

8) данные для мероприятия.

К выходной информации относится:

1) документ график работы официантов, который создаются раз в неделю (по воскресеньям);

2) документ с планом мероприятия, которое создается по требованию администратора;

3) меню для мероприятий, которое создается по требованию администратора, входит в план мероприятия;

4) документ на печать о выручке, который создается администратором ежедневно для бухгалтера;

5) отчет со списком официантов для получения премии за участие в мероприятиях, или общей выручки за неделю больше 14000 рублей, который создается администратором еженедельно, если имеются такие работники, для бухгалтера;

6) отчет о продуктах на складе, создает по требованию администратора;

7) подробный отчет по загруженности зала, создает по требованию администратора;

8) документ на бронирование столика, создает по требованию администратора;

9) документ «Расходная накладная», создает по требованию администратора.

3.2 Описание автоматизированной работы зала ресторана

Далее автоматизированная система будет описана при помощи диаграмм TO-BE, то есть «такими, какие они будут».

Контекстная диаграмма модели TO-BE методологии IDEF0 представлена на рисунке 10.

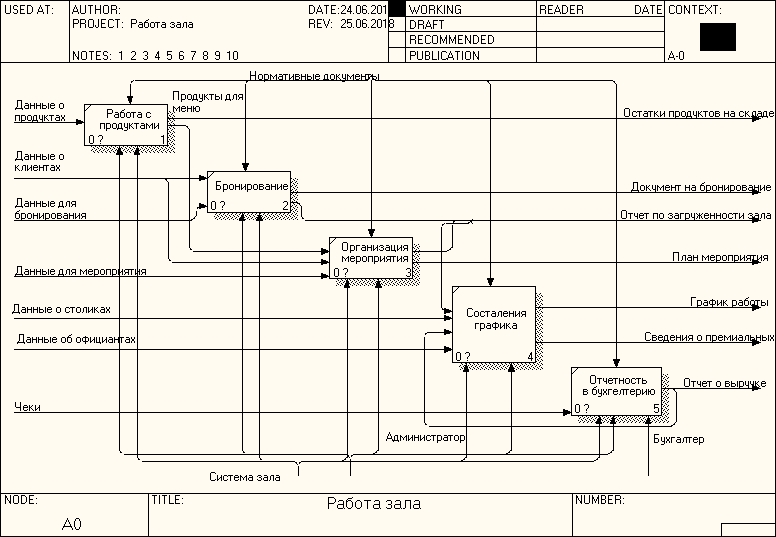


Рисунок – Контекстная диаграмма TO-BE, модель IDEF0

При сравнении контекстных диаграмм AS-IS и TO-BE особых изменений можно не наблюдать. Чтобы увидеть результат внедрения автоматизированной системы обратимся к рисунку 11, на котором представлена контекстная диаграмма TO-BE методологии DFD.

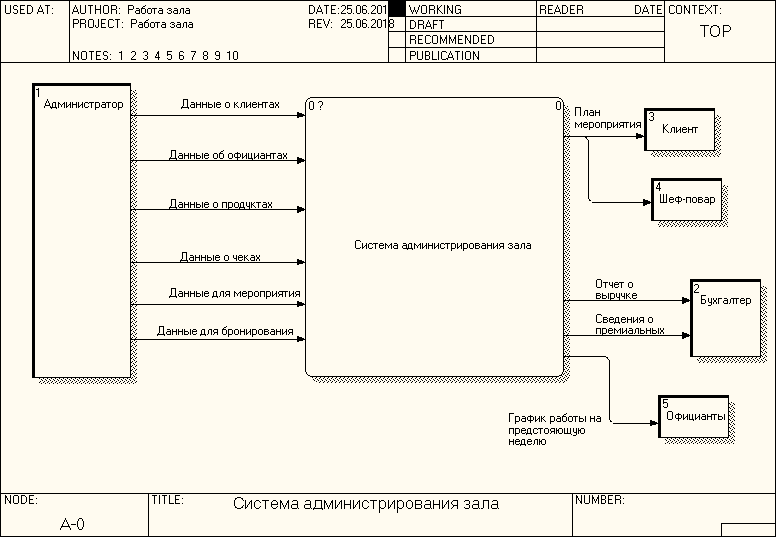


Рисунок – Контекстная диаграмма TO-BE, модель DFD

Внешние объекты, которые изображены на контекстной диаграмме не изменились после внедрения автоматизированной системы, так как ее задачей является не избавить пользователя от данных, а формализовать их, и декомпозиция диаграммы TO-BE это демонстрирует. (Рисунок 12.)

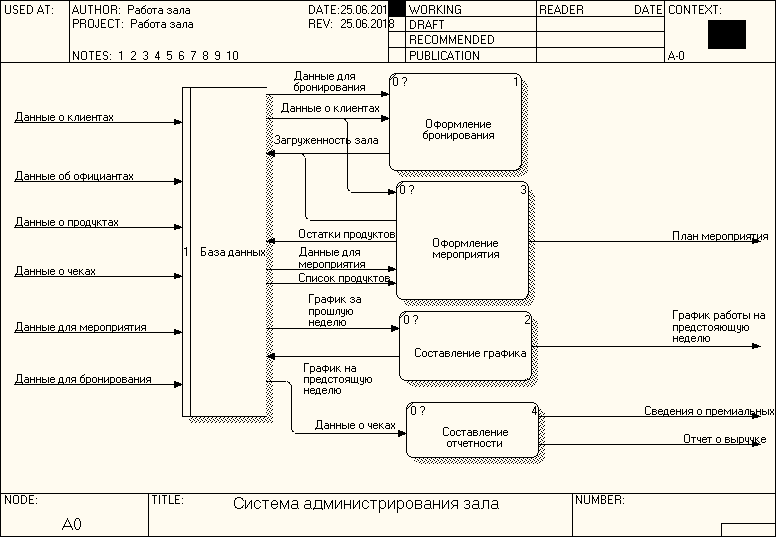


Рисунок – Декомпозиция контекстной диаграммы TO-BE, модель DFD

На декомпозиции контекстной диаграммы четко видны изменения после внедрения системы. Хранение и работа с данными упрощены в разы.

В итоге, после внедрения автоматизированной системы в работу зала, все данные будут объединены в общую базу данных, которая будет накапливать и хранить данные, а также производить с ними соответствующие действия, что облегчит в дальнейшем работу пользователей.

**4 Инфологическая модель проектируемой базы данных**

После анализа предметной области и формирования постановки задачи можно приступить к проектированию базы данных, которая является главной составляющей системы.

Для начала необходимо определить, как будут взаимосвязаны данные между собой и как буду хранится в будущей базе. Все это отражено на ER-диаграмме, составленная в соответствии с нотацией П. Чена, представленная на рисунке 13.



Рисунок – ER-диаграмма зала ресторана

В ходе анализа предметной области были выделены.

1) Стержневые сущности:

- меню для мероприятия;

- продукты;

- чек;

- вид мер (вид мероприятия);

- клиент;

- стол;

- график;

- официант.

2) Характеристики:

- примечание.

3) Ассоциативные сущности:

- список продуктов;

- мероприятие;

- обслуживание клиентов;

- бронирование;

- составление графика.

В таблице 1 приведены связи между объектами в ER-диаграмме, которые также определены на основе анализа предметной области.

Таблица 1 – Связи между объектами в ER-диаграмме

|  |  |  |
| --- | --- | --- |
| **Объект** | **Связь** | **Объект** |
| Составление\_Графика | Один к одному | График |
| Один ко многим | Официант |
| Один ко многим | Стол |
| Бронирование | Один ко многим | Стол |
| Один к одному | Клиент |
| Мероприятие | Один к одному | Вид\_Мероприятия |
| Один к одному | Клиент |
| Один ко многим | Стол |
| Один к одному | Меню\_Для\_Мероприятия |
| Обслуживание\_Клиентов | Один к одному | Чек |
| Один к одному | Официант |
| Один к одному | Стол |
| Список\_Продуктов | Один к одному | Меню\_Для\_Мероприятия |
| Один ко многим | Продукты |

Данная модель на языке инфологического моделирования будет иметь следующий вид:

Клиент (ID\_КЛ, ФИО, Номер\_Телефона)

Стол (ID\_СТ, Количество\_Мест)

График (ID\_ГР, ПРИМ)

Примечание (ПРИМ, Содержание) {График}

Официант (ID\_ГР, ФИО, Номер\_Телефона)

Вид\_Мер (ВМЕР, Наименование\_Вида)

Продукты (ID\_ПРОД, Наименование\_Продукта, Вес\_Кг, Цена\_За\_Продукт)

Меню\_Для\_Мероприятия (ID\_МЕН, Наименование\_Блюда, Количество, Вес\_Кг, Цена\_За\_Блюдо)

Чек (ID\_Чек, Сумма\_Заказа)

Составление\_Графика [График, Официант M, Стол N] (СОСТГР, ID\_СТ, ID\_ГР, ID\_ОФ, Дата)

Бронирование [Клиент, Стол N] (БРО, ID\_СТ, ID\_ОФ, Дата, Время, Длительность)

Мероприятие [Клиент, Стол N, Вид\_Мер K, Меню\_Для\_Мероприятия] (МЕР, ID\_СТ, ID\_КЛ, ВМЕР, ID\_МЕН, Дата, Время, Длительность, Количество\_Гостей)

Список\_Продуктов [Продукты P, Меню\_Для\_Мероприятия] (ID\_СППРОД, ID\_ПРОД, ID\_МЕН)

Обслуживание\_Клиентов [Чек C, Официант M, Стол N] (ОБСЛ, ID\_ОФ, ID\_КЛ, ID\_Чек, Дата)

Расшифровка аббревиатур, использованных при создании ER-диаграммы представлена в таблице 2.

Таблица 2 – Расшифровка аббревиатур

|  |  |
| --- | --- |
| **Аббревиатура** | **Расшифровка** |
| ID\_КЛ | Идентифицирующий номер клиента |
| ID\_ОФ | Идентифицирующий номер официанта |
| ID\_СТ | Идентифицирующий номер стола |
| ID\_ГР | Идентифицирующий номер графика |
| ПРИМ | Идентифицирующий номер примечания |
| ВМЕР | Идентифицирующий номер вида мероприятия |
| ID\_ПРОД | Идентифицирующий номер продукта |
| ID\_МЕН | Идентифицирующий номер меню |
| СОСТГР | Идентифицирующий номер составления графика |
| БРО | Идентифицирующий номер бронирования |
| МЕР | Идентифицирующий номер мероприятия |
| ОБСЛ | Идентифицирующий номер обслуживания |
| ID\_СППРОД | Идентифицирующий номер списка продуктов |

Следующий этап проектирования – создание логической модели проектируемой базы данных, которая строится на основе созданной ранее инфологической модели.

**5 Логическая модель данных**

Логическая модель данных по методологии IDEF1X – отображение структуры базы данных, в которой имеются таблицы, их атрибуты, типы данных и связи между таблицами, представлена на рисунке 14.

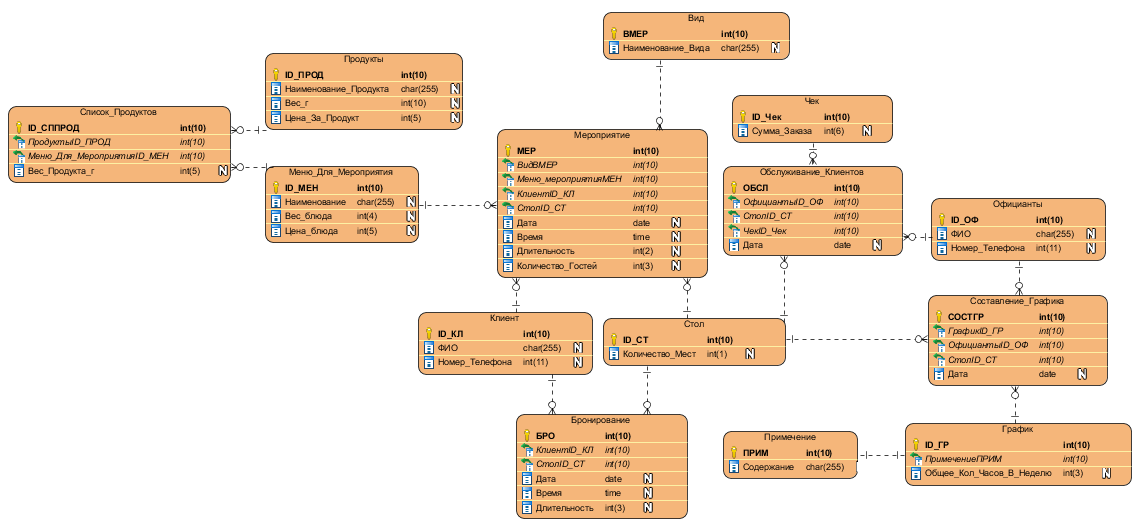


Рисунок – Логическая модель данных, модель IDEF1X

В результате, объекты ER-диаграммы были отображены в логической модели данных, при чем с более детальным пояснением, такими как, типы данных, и удобочитаемыми первичные и внешние ключи.

**6 Физическая модель базы данных и ее реализация**

Физическая модель данных описывает реализацию объектов логической модели на уровне объектов конкретной базы данных.

Реализация осуществлялась с помощью программного средства 1С: Предприятие 8.3 (8.3.4.437).

В ходе реализации было принято решение расширить некоторые из объектов логической модели данных. Соответствие объектов физической и логической моделей базы данных приведено в таблице 3. Более подробное соответствие и описание приведено в приложении В.

Таблица 3 – Соответствие объектов

|  |  |
| --- | --- |
| **Логическая модель** | **Физическая модель** |
| Клиент | Справочник «Клиенты» |
| Стол | Справочник «Столы» |
| Официанты | Справочник «Официанты» |
| График | Документ «График», перечисление «Время\_Работы», перечисление «Виды\_Графиков», Перечисление «Виды\_Смен» |
| Бронирование | Документ «Бронирование», регистр сведений «Загруженностьпо залу», отчет «Загруженность зала (подробно)» |
| Обслуживание\_Клиентов | Документ «Чек», отчет «Сведения для выдачи премии» |
| Мероприятие | Документ «Мероприятие», регистр сведений «Загруженностьпо залу», отчет «Загруженность зала (подробно)» |
| Меню\_Для\_Мероприятия | Включено в документ «Мероприятие» |
| Продукты | Справочник «Продукты», документ «Приходная накладная», регистр накопления «Остаток продуктов», отчет «Продукты на складе» |
| Список\_Продуктов | Справочник «Продукты», справочник «Общее меню», регистр накопления «Остаток продуктов», отчет «Продукты на складе» |

Далее подробнее о физической модели.

6.1 Роли и подсистемы

Для начала стоит отметить, что система разбита на подсистемы, рисунок 15, в каждой из которых имеются необходимые для работы справочники, документы, регистры и отчеты.

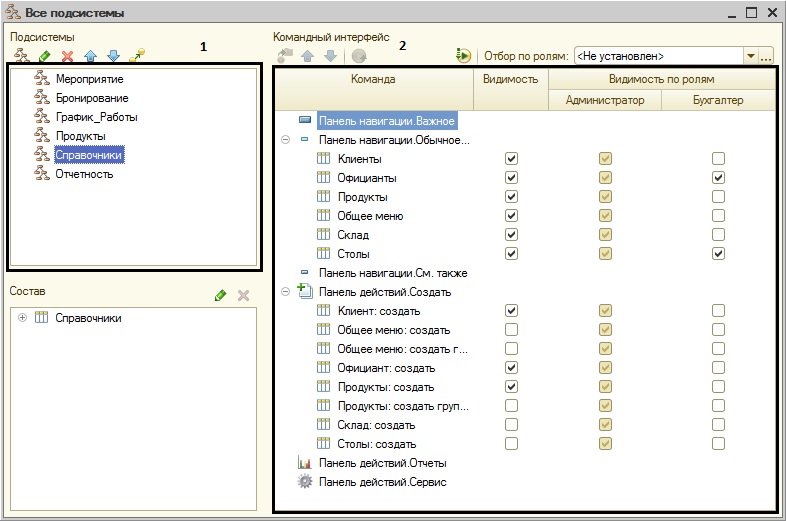


Рисунок – Подсистемы (1), функции подсистемы «Справочники» (2)

В системе имеются две роли (рисунок 16), каждая их которых имеет свой функционал, например, доступные права, имеющиеся у администратора приведены на рисунке 17, он также является и системным администратором, как об этом уже упоминалось выше.



Рисунок – Список ролей системы

Роль «Администратора» используют два администратора посменно, а роль «Бухгалтер» использует один бухгалтер.

Для входа в систему используется соответствующий логин и пароль, который пользователь может получить или изменить у администратора системы.

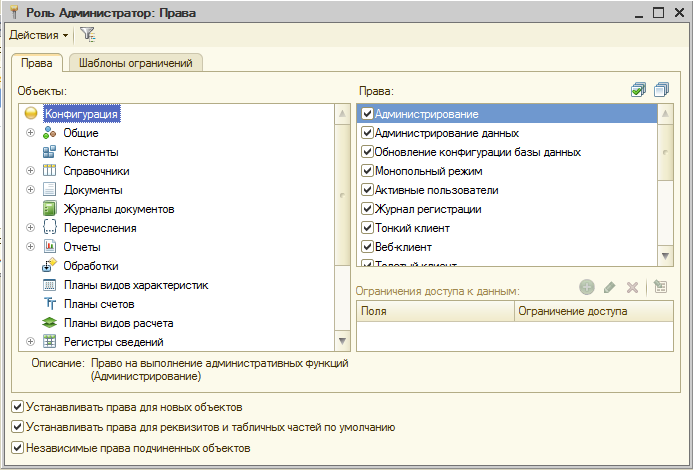


Рисунок – Настройка права «Администратора»

На рисунке 18, приведен полный список пользователей системы.

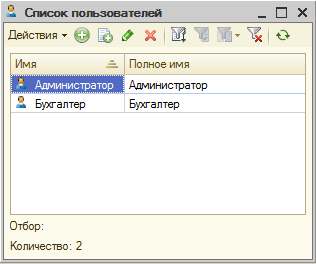


Рисунок – Список пользователей системы

На рисунках 19-24 приведена интерфейсные команды в соответствии с подсистемами, которые доступны администратору в режиме конфигурации (1C: Предприятие).

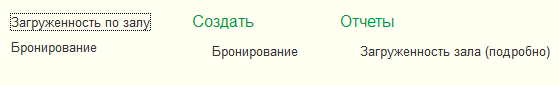


Рисунок – Подсистема «Бронирование»

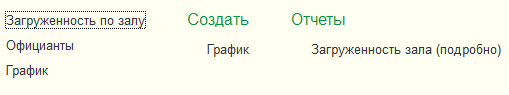


Рисунок – Подсистема «График»

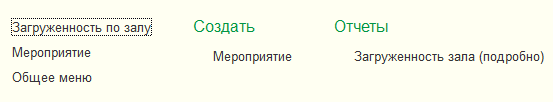


Рисунок – Подсистема «Мероприятие»

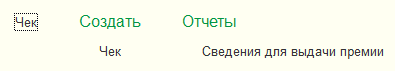


Рисунок – Подсистема «Отчетность»

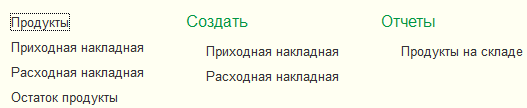


Рисунок – Подсистема «Продукты»

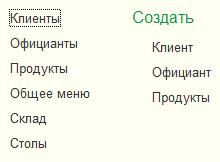


Рисунок – Подсистема «Справочники»

В режиме конфигурации (1C: Предприятие) бухгалтер вправе только просматривать документ «График работы», документ «Чеки», отчет «Сведения для выдачи премии», справочники «Официанты» и «Столы».

6.2 Справочники системы

На рисунках 25 представлены все справочники системы с их реквизитами и табличными частями.

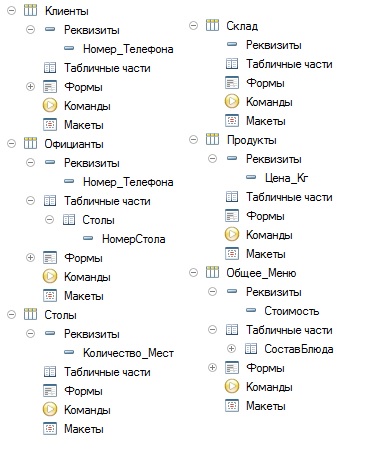


Рисунок – Справочники системы

Справочники по умолчанию имеют такие реквизиты, как код и наименование, что облегчает проектирования базы данных. Кроме стандартных реквизитов можно добавить свои реквизиты, например, в справочники «Клиенты» имеет дополнительный реквизит «Номер\_Телефона», или создать табличную часть, которая реализует связь «многие ко многим», например, в справочнике «Официанты», за каждым работник закреплено несколько столов.

Проверка на дублирование назначенных столов в справочнике «Официанты» реализована в модуле формы. Код представлен в приложении Б «Справочник «Официанты». Модуль формы элемента».

Для справочника «Общее меню» реализована проверка на дублирование продуктов, при выборе продукта ему автоматически проставляется цена за кг, также рассчитывается общая стоимость блюда на основе общей суммы поставки всех продуктов с наценкой в 700%. Код модуля формы приведен в приложении Б «Справочник «Общее меню». Модуль формы элемента».

6.3 Документы и регистры системы

По мимо справочников для системы был создан ряд документов, представленных на рисунке 26.

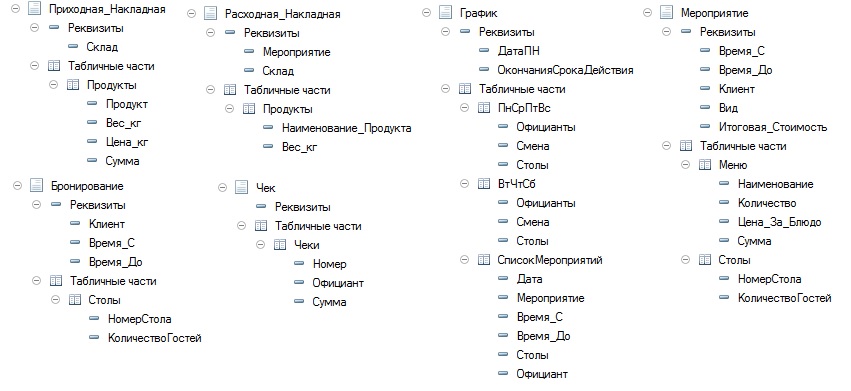


Рисунок – Документы системы

Документы, также, как и справочники, имеют стандартные реквизиты – дата и номер, дополнительные реквизиты и табличные части. Все документы, за исключением документа «Бронирования», можно распечатать.

Документ «Приходная накладная», рисунок 26, используется для учета продуктов на складе. В документе производится проверка на дублирование наименований продукта, то есть администратор не может дважды выбрать один и тот же продукт. Также производится автоматическое сопоставление цены продукта за кг из справочника «Продукты» и расчет суммы по продуктам при изменении количества. Код модуля формы приведен в приложении Б «Документ «Приходная накладная». Модуль формы элемента».

При проведении документ «Приходная накладная» записывается в регистр накопления «Остаток продуктов», рисунки 27 (структура) и 28 (форма), чтобы учитывать приход продуктов на склад. Код объекта приведен в приложении Б «Документ «Приходная накладная». Модуль объекта».

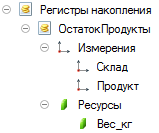


Рисунок – Структура регистр накопления «Остаток продуктов»

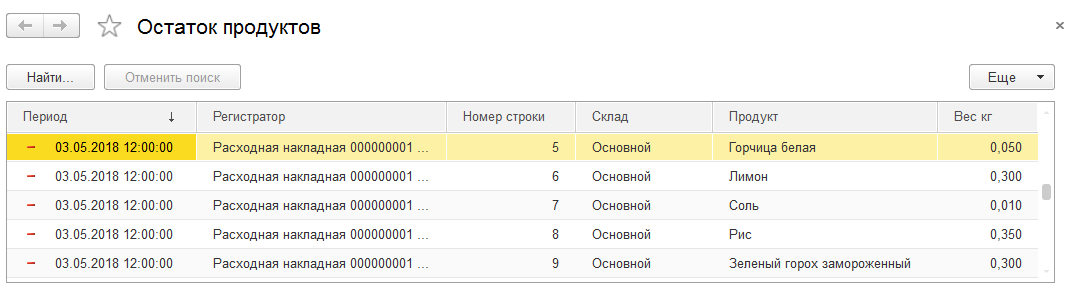


Рисунок – Форма регистра накопления «Остаток продуктов»

Также имеется печатная форма данного документа, рисунок 29.

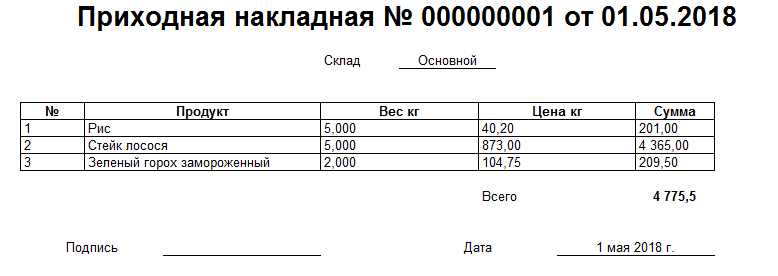


Рисунок – Печатная форма документа «Приходная накладная»

Макет документа «Приходная накладная» представлен на рисунке 30.

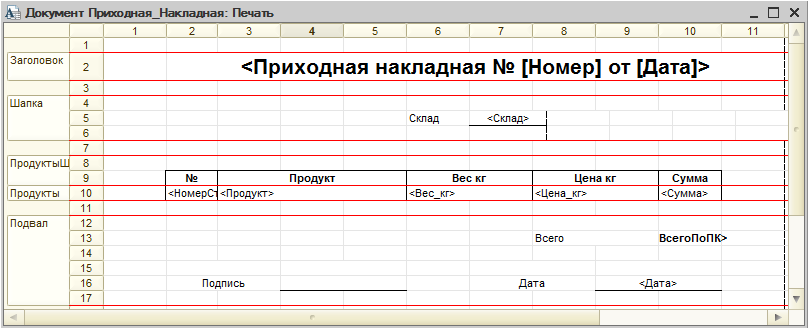


Рисунок – Макет документа «Приходная накладная»

Процедура печати документа представлена в приложении Б «Документ «Приходная накладная». Модуль менеджера».

Документ «Расходная накладная», рисунок 26, используется для учета продуктов на складе. Документ при проведении записывается в регистр накопления «Остаток продуктов», рисунки 27 (структура) и 28 (форма), чтобы учитывать расход продуктов, а также проводятся проверки на наличие продукта на складе. Код объекта приведен в приложении Б «Документ «Расходная накладная». Модуль объекта». Для проверки на дубликаты наименований продуктов используется код приведенный в приложении Б «Документ «Расходная накладная». Модуль формы элемента».

Также имеется печатная форма данного документа, рисунок 31.

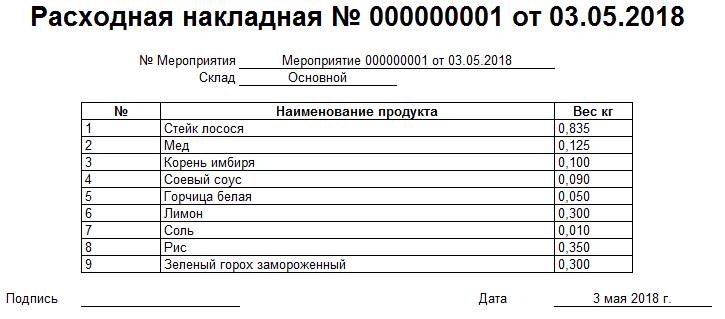


Рисунок – Печатная форма документа «Расходная накладная»

Макет документа «Расходная накладная» представлен на рисунке 32.

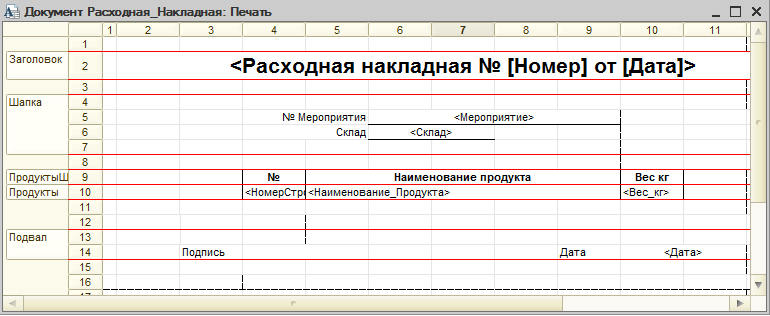


Рисунок – Макет документа «Расходная накладная»

Процедура печати документа представлена в приложении Б «Документ «Расходная накладная». Модуль менеджера».

Документ «Бронирование», рисунок 26, используется для того, чтобы создать бронь на стол (-ы) в системе. В документе производится проверка на дублирование наименований столов, то есть администратор не может дважды выбрать один и тот же стол. Также администратор не может записать в реквизите «Количество гостей» людей больше, чем мест за выбранным столом. Код модуля формы приведен в приложении Б «Документ «Бронирование». Модуль формы элемента».

Документ при проведении записывается в регистр сведений «Загруженность по залу», рисунки 33 (структура) и 34 (форма), чтобы учитывать загрузку ресторана. Код объекта приведен в приложении Б «Документ «Бронирование». Модуль объекта».

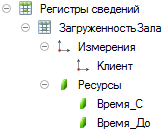


Рисунок – Структура регистра сведений «Загруженность по залу»

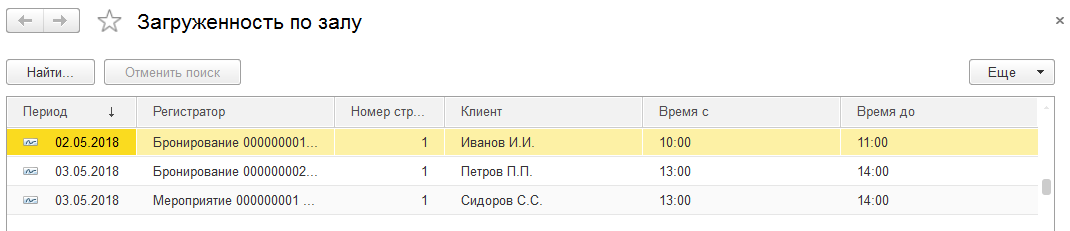


Рисунок – Форма регистра сведений «Загруженность по залу»

Документ «Мероприятие», рисунок 26, используется для того, чтобы зарегистрировать мероприятие в системе. В документе производится проверка на дублирование наименований блюд и столов, то есть администратор не может дважды выбрать один и тот же наименование. Также администратор не может записать в реквизите «Количество гостей» людей больше, чем мест за выбранным столом. Также система автоматически сопоставляет цену блюда из общего меню и автоматически заносит ее при составлении меню, когда администратор выбирает нужное ему блюдо. При изменении количества блюд, рассчитывается общая стоимость каждого их них. По итогу, администратор может увидеть автоматически посчитанную сумму по всем блюдам, это и есть итоговая стоимость мероприятия. Код модуля формы приведен в приложении Б «Документ «Бронирование». Модуль формы элемента».

Документ при проведении записывается в регистр сведений «Загруженность по залу», рисунки 33 (структура) и 34 (форма), чтобы учитывать загрузку ресторана. Код объекта приведен в приложении Б «Документ «Мероприятие». Модуль объекта».

Также имеется печатная форма данного документа, рисунок 35.

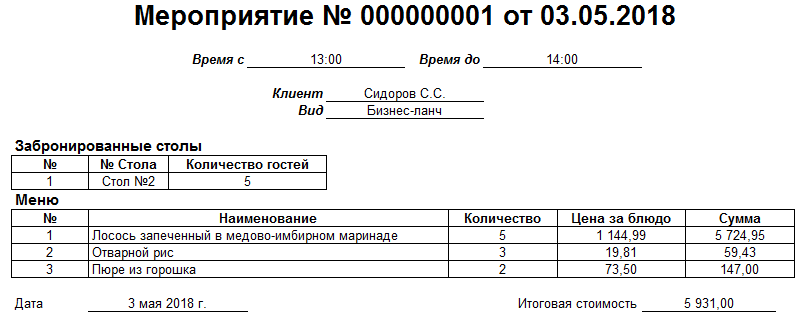


Рисунок – Печатная форма документа «Мероприятие»

Макет документа «Мероприятие» представлен на рисунке 36.

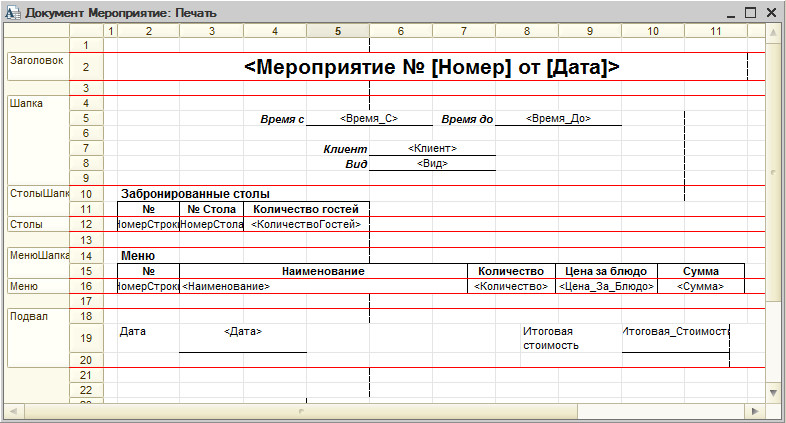


Рисунок – Макет документа «Мероприятие»

Процедура печати документа представлена в приложении Б «Документ «Мероприятие». Модуль менеджера».

Документ «Чек», рисунок 26, использует для учета выручки за день. Документ при проведении записывается в регистр накопления «Сведения для выдачи премии», рисунки 37 (структура), чтобы учитывать загрузку ресторана. Код объекта приведен в приложении Б «Документ «Чек». Модуль объекта».

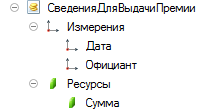


Рисунок – Структура регистра сведений «Сведения для выдачи премии»

Также имеется печатная форма данного документа, в которой система автоматически рассчитывает итоговую выручку по чекам за день, рисунок 38.

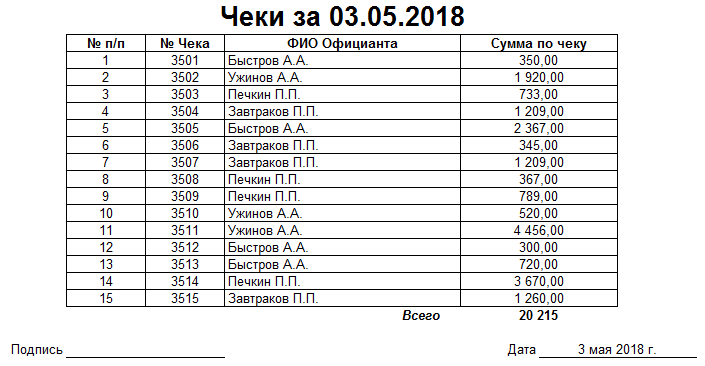


Рисунок – Печатная форма документа «Чек»

Макет документа «Чек» представлен на рисунке 39.

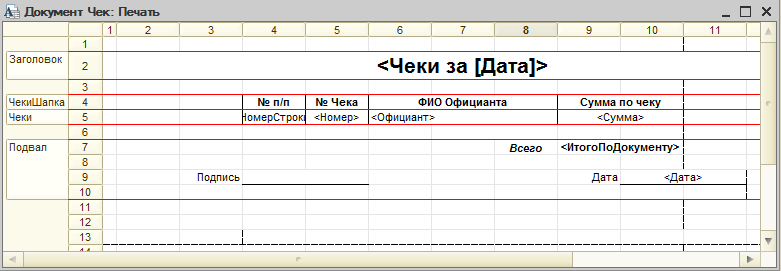


Рисунок – Макет документа «Чек»

Процедура печати документа представлена в приложении Б «Документ «Чек». Модуль менеджера».

Документ «График», рисунок 26, используется для создания графика работы для официантов в системе и дальнейшего его использования. В документе производится проверка на дублирование ФИО официантов, то есть администратор не может дважды выбрать одного и того же официанта. Автоматически осуществляется привязка столов к каждому официанту по справочнику «Официанты» и мероприятий, которые назначены на предстоящую неделю (выбор официанта для участия мероприятия администратор выбирает сам). Код модуля формы приведен в приложении Б «Документ «График». Модуль формы элемента».

Также имеется печатная форма данного документа, рисунок 40.

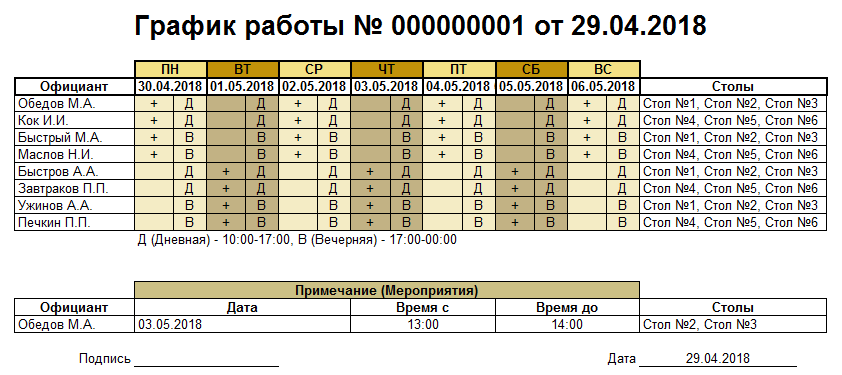


Рисунок – Печатная форма документа «График»

Макет документа «График» представлен на рисунке 41.

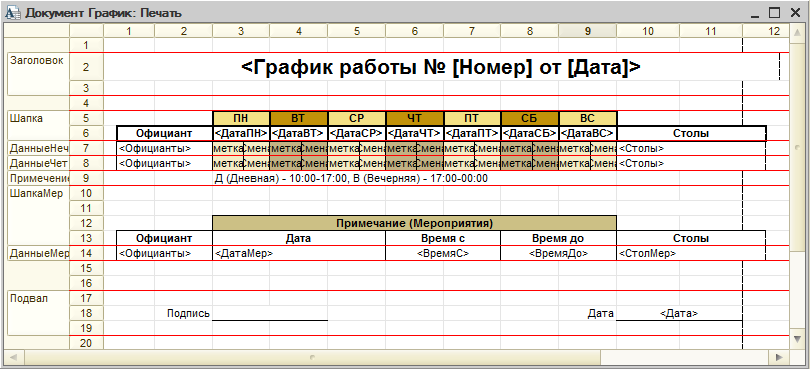


Рисунок – Макет документа «График»

Процедура печати документа представлена в приложении Б «Документ «График». Модуль менеджера».

6.4 Перечисления, используемые в системе

Для хранения постоянной информации таких, как время работы (ресторана) видов смен, видов мероприятий и видов графиков (работы официантов) были использованы перечисления, которые имеют определенный системным администратором неизменяемые значения, рисунки 42.

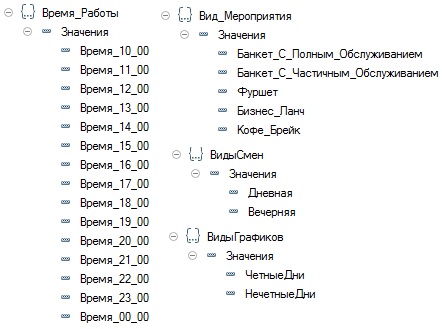


Рисунок – Перечисления, использующиеся в системе

6.5 Отчеты, создаваемые системой

В системе имеется ряд отчетов, которые созданы системой при помощи схемы компоновки данных, рисунок 43, в которых информация формализована и приведена в удобочитаемый вид.

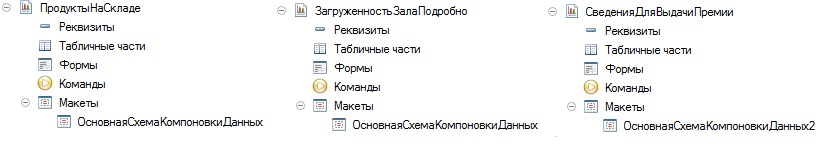


Рисунок – Отчеты, создаваемые системой

Отчет «Продукты на складе», рисунок 44, формируется на основе регистра накопления «Остаток продуктов», рисунки 27 (структура) и 28 (форма), которые используется для учета движения продуктов.

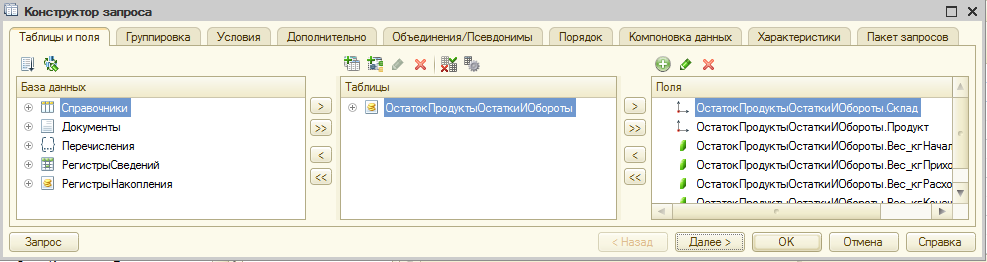


Рисунок – Конструктор запроса отчета «Продукты на складе»

Отчет «Загруженность зала (подробно)», рисунок 45, формируется на основе регистра сведений «Загруженность по залу», рисунки 33 (структура) и 34 (форма), документов «Бронирование» и «Мероприятие». По данному отчету можно отследить занятость ресторана.

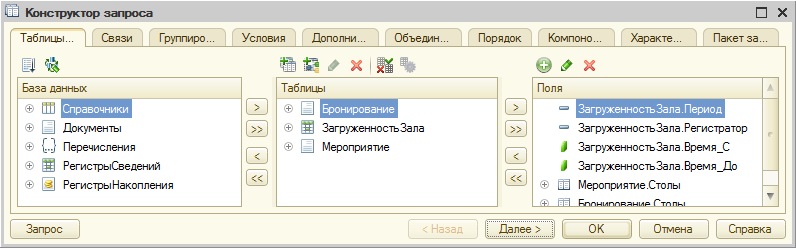


Рисунок – Конструктор запроса отчета «Загруженность зала (подробно)»

Отчет «Сведения для выдачи премии», рисунок 46, формируется на основе регистра накопления «Сведения для выдачи премии», рисунки 37 (структура). Он содержит список официантов, у которых выручка по всем их чекам за неделю составляет больше 14000 включительно.

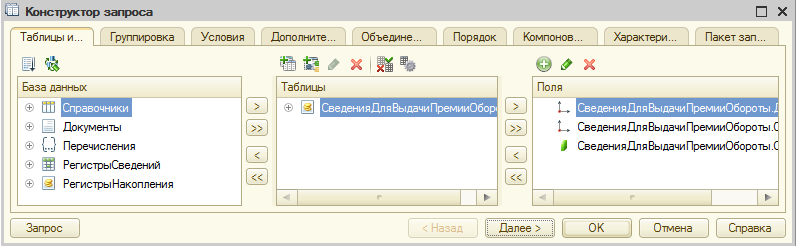


Рисунок – Конструктор запроса отчета «Сведения для выдачи премии»

6.6 Формы

В режиме конфигурации (1C: Предприятие) доступны различные формы для удобства работы пользователей системы, например, форма для оформления брони, рисунок 47, и форма для создания графика, рисунок 48.

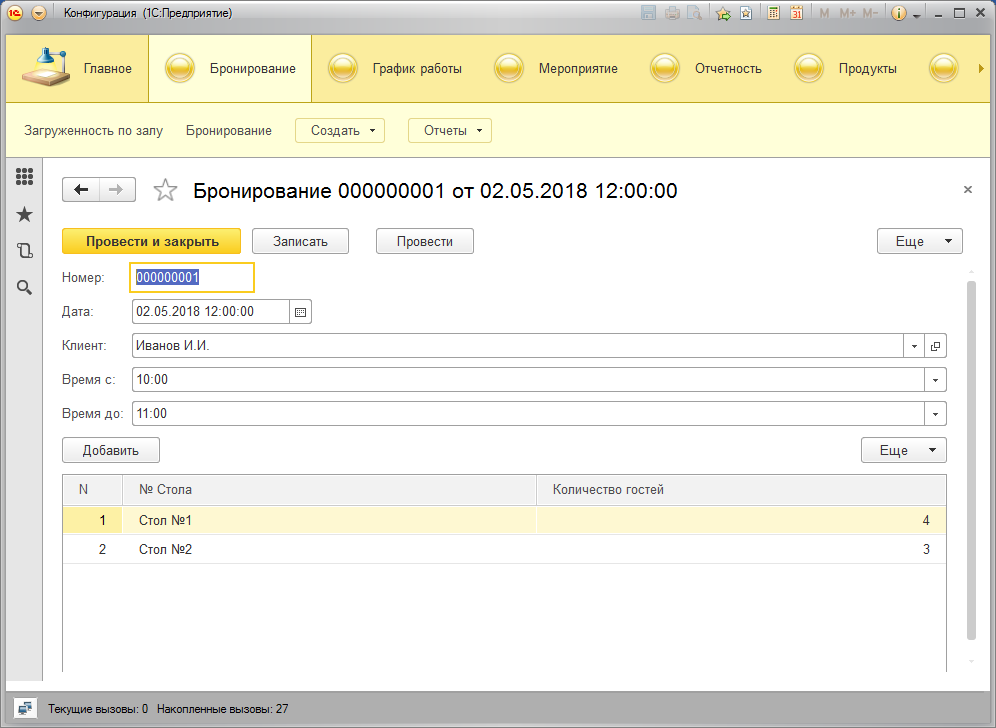


Рисунок – Форма для оформления бронирования

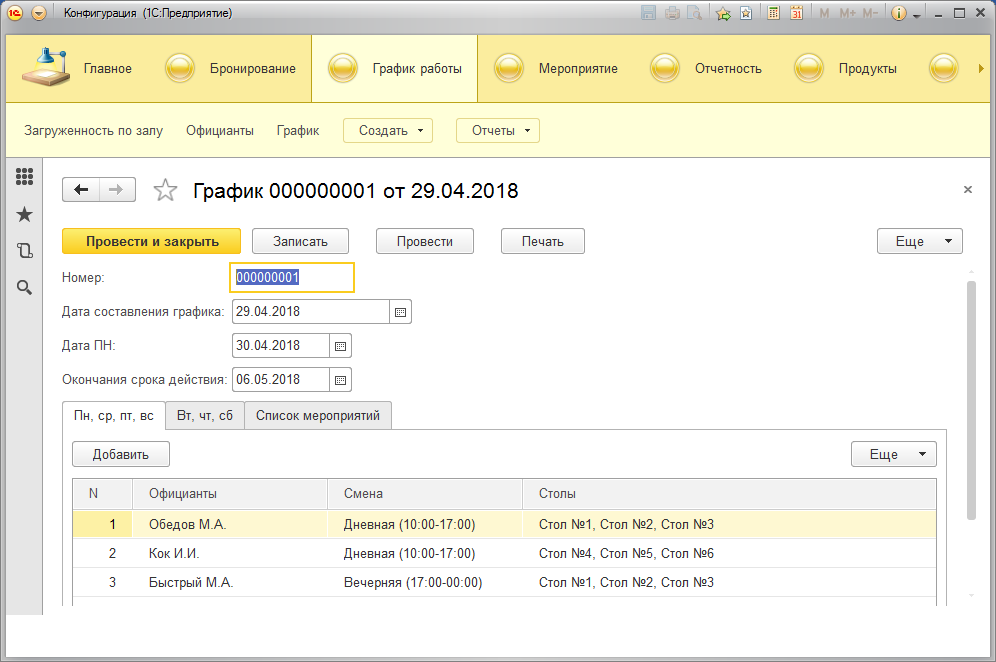


Рисунок – Форма для составления графика

Все созданные формы автоматизированной системы интуитивно понятны, просты в использовании, что не затруднит переход от бумажных носителей к системе.

Система, как уже говорилось ранее, располагает рядом отчетов.

Подробный отчет по загрузке зала приведен на рисунке 49.

Данный отчет помогает администратору быстро найти свободное время для оформления брони или мероприятия, а также составить график работы официантов.

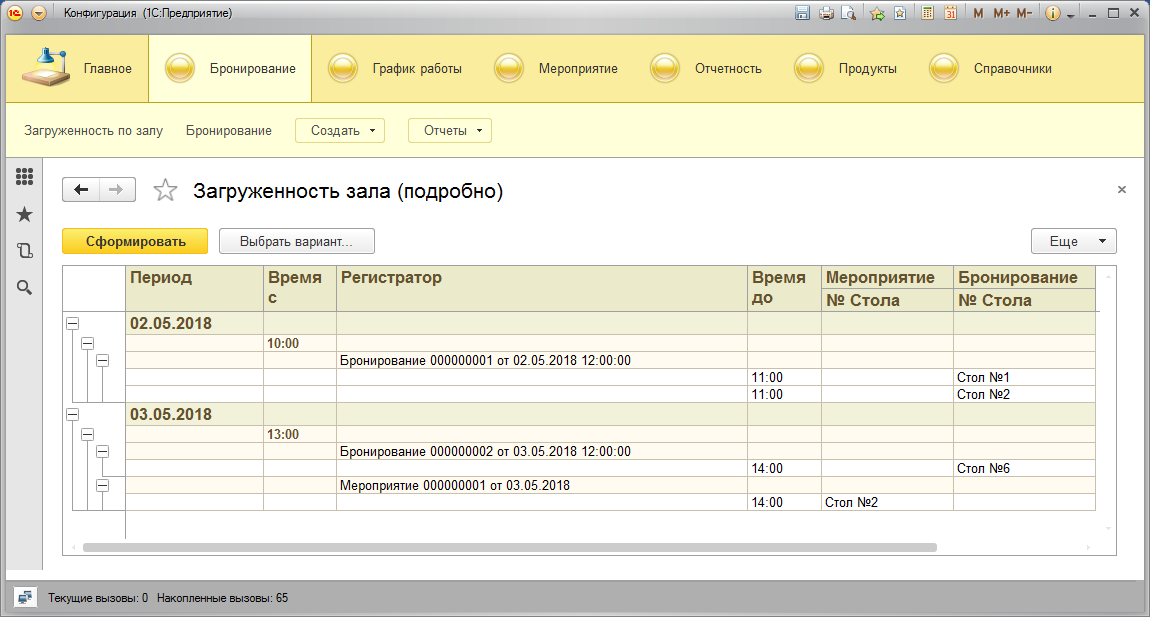


Рисунок – Отчет «Загруженность зала (подробно)

Отчет «Сведения для выдачи премии», приведен на рисунке 50.

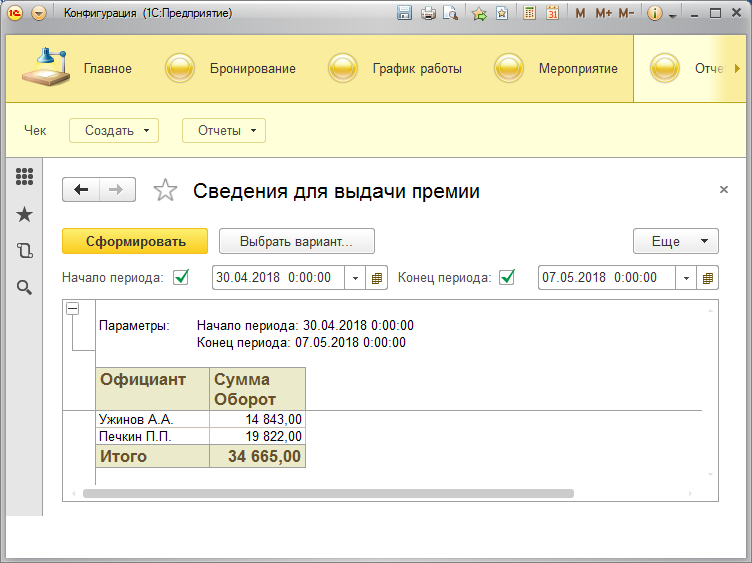


Рисунок – Отчет «Сведения для выдачи премии»

И последний отчет, приведен на рисунке 51, это отчет, который помогает администратору увидеть движение продуктов на складе за определенный период времени: начальный остаток, приход, расход, конечный остаток.

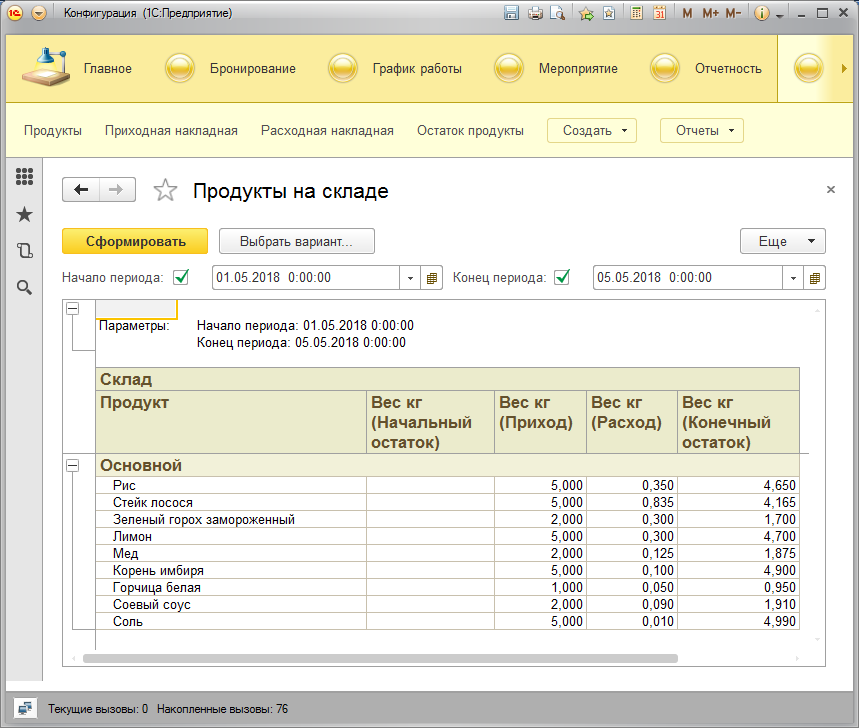


Рисунок – Отчет «Продукты на складе»

**Заключение**

По итогу курсового проекта была создана автоматизированная система для улучшения работы ресторана, а именно деятельности администратора, при помощи программного средства 1С: Предприятие 8.3.

При анализе предметной области были определены основные бизнес-процессы, на основе которых спроектированы модели для отображения работы зала ресторана как есть и как будет, а также модели будущей базы данных.

Разработанная база данных позволяет решать ряд следующих поставленных задач в полном объеме:

1) автоматизирована организация бронирования;

2) автоматизировано проведение мероприятия;

3) автоматизировано создание отчетности;

4) автоматизированы расчеты;

5) создана единую базу данных.

Решен ряд следующих поставленных задач в неполном объеме:

1) автоматизировано составление графика работы официантов;

2) автоматизирована работа с данными о продуктах для мероприятий.

В итоге была создано единое хранилище данных, а также будущие пользователи системы будут избавлены:

1) от большого количества бумажных носителей, что ускорит их работу;

2) от ручных расчетов;

3) от лишнего дублирования информации;

4) от допущений ошибок.

**Список литературы**

1) Бойко Э.В. 1С: Предприятие 8.0. Универсальный самоучитель / Э.В. Бойко. - М.: Омега-Л, 2011. - 232 c.

2) Габец А.П. 1С: Предприятие 8.1. Простые примеры разработки / А.П. Габец, Д.И. Гончаров. - М.: 1С: Паблишинг; СПб: Питер, 2008. – 383 с.

3) Дейт, К.Дж. Введение в системы баз данных. / К.Дж. Дейт. – К.; М.; Спб: Изд. дом «Вильямс», 2005.

4) Красюк Л.В. Методические указания по выполнению курсовой работы по дисциплине «Базы данных» - Владивосток, 2016 – 20 с.

5) Пирогов В. Информационные системы и базы данных: организация и проектирование. - СПб: БХВ-Петербург, 2010. - 528 с.

6) Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений/ Под ред. Проф. А.Д. Хомоненко.-6-е изд.- СПб.:КОРОНА-Век, 2010.-736 с.

**Приложение А**

**Руководство пользователя**

**Руководство пользователя для роли «Администратор»**

Перед началом работы пользователь должен осуществить вход в систему, рисунок А1.

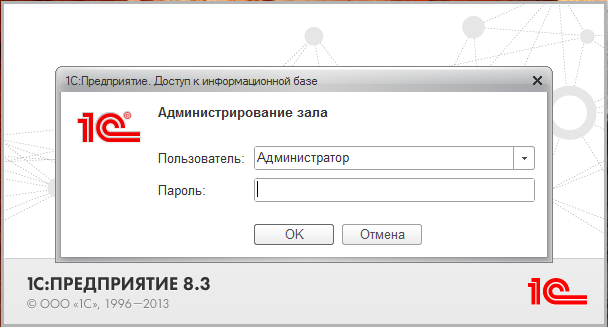


Рисунок А1 – Вход в систему

При успешной идентификации запускается главное окно системы, рисунок А2.

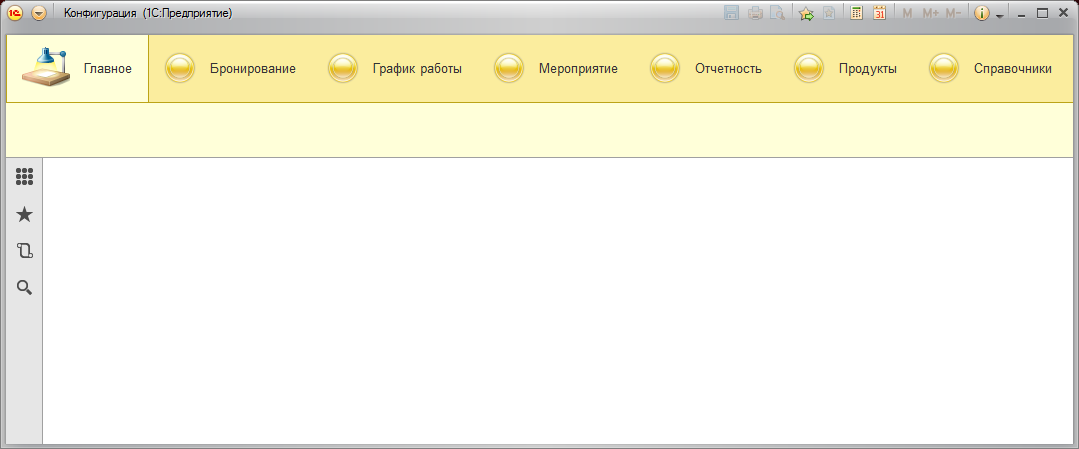


Рисунок А2 – Главное окно системы

Перед началом работы пользователю необходимо в подсистеме «Справочники» (1) заполнить соответствующие справочники «Официанты», «Столы», «Общее меню», «Продукты», «Клиенты» (2), рисунок А3. Справочник «Склад» имеет уже предопределенное наименование «Основной» и поэтому его заполнение можно отложить, пока не это потребуется.

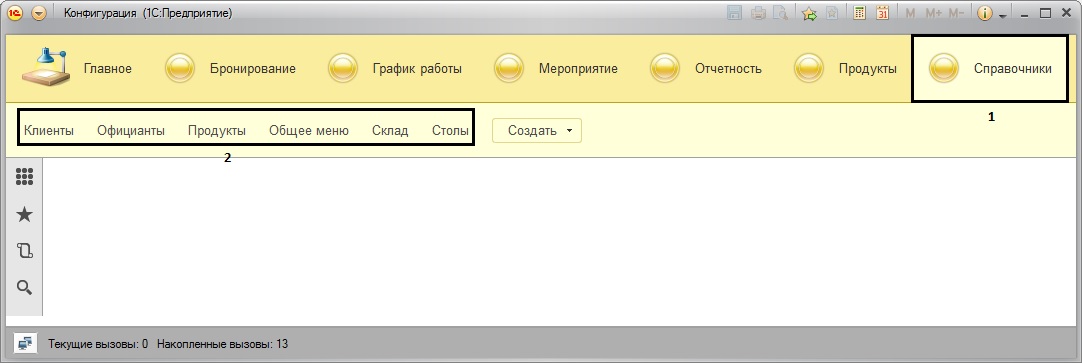


Рисунок А3 – Подсистема «Справочники»

Заполнить справочники можно тремя способами, либо использовать кнопку создать и выбрать нужный объект (1), либо непосредственно запустить нужный справочник (2) и выбрать кнопку «Создать» (3) или использовать горячую клавишу Ins. Удалить какие-либо данные можно при помощи кнопки «Еще» (4) или сочетании клавиш Shift+Del. Отредактировать данные можно использовав двойной щелчок мыши по записи, при помощи кнопки «Еще» (4) или клавиши F2, рисунок А4.

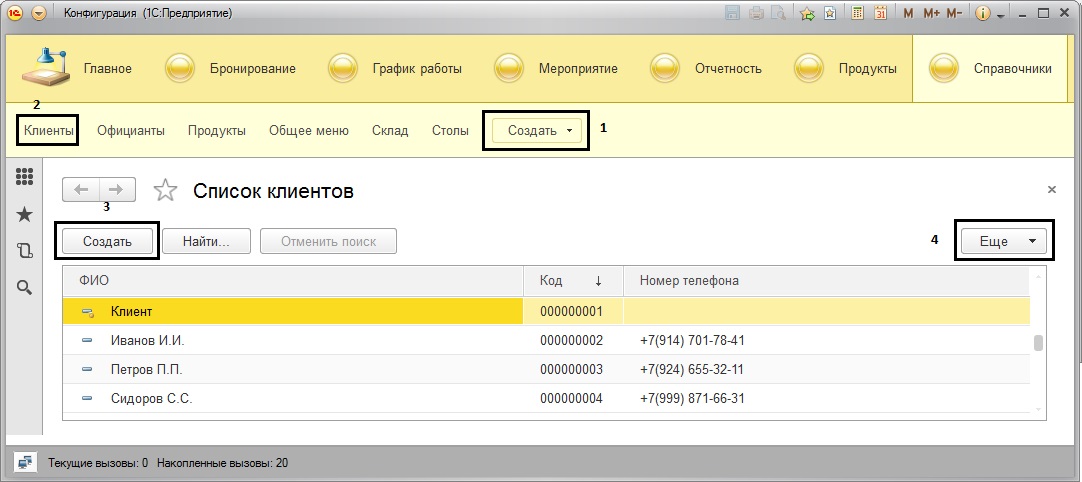


Рисунок А4 – Кнопки «Создать»

Рассмотрим заполнение справочника «Клиенты».

По мимо созданных пользователем посетителей, имеется также предопределенное наименование – «Клиент», оно используется для тех клиентов, которые не бронировали столики заранее, но для полного отражения загруженности зала, они оформляются пользователем, как обычная бронь, но без дополнительных данных, таких как номер телефона и ФИО клиента.

Добавление нового клиента, рисунок А5:

1) Заполните поле «ФИО» (1);

2) Заполните поле «Номер телефона» (2);

3) Нажать на кнопку «Записать и закрыть» (3).

Код присевается системой автоматически.

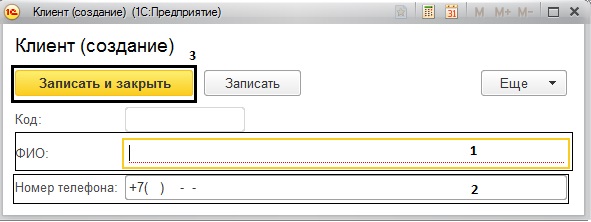


Рисунок А5 – Форма «Клиент (создание)»

Справочник «Продукты» имеет разделение на группы, в каждую из групп добавляются определенные продукты, рисунок А6. Группу можно создать, редактировать или удалять при помощи кнопки «Еще» или щелчком правой кнопки мыши.

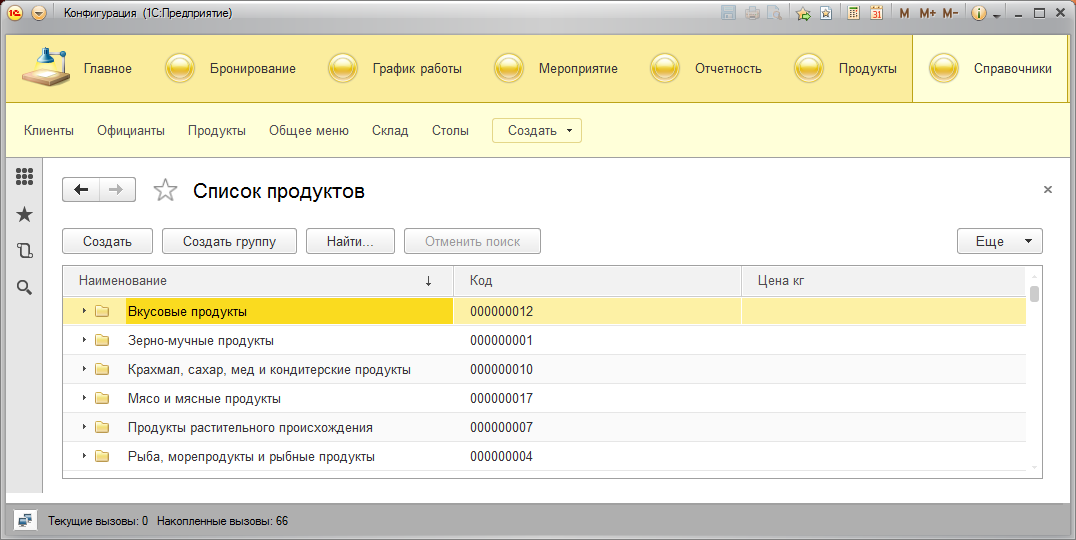


Рисунок А6 – Форма «Список продуктов»

Добавление нового продукта, рисунок А7, и нового стола, рисунок А8, аналогично добавлению клиента.

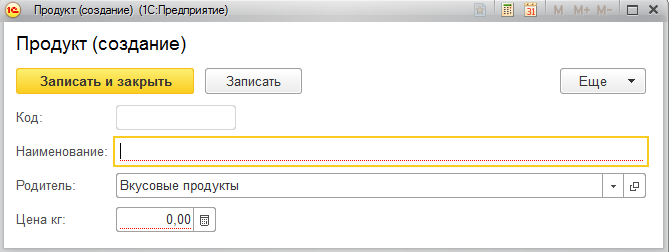


Рисунок А7 – Форма «Продукт (создание)»

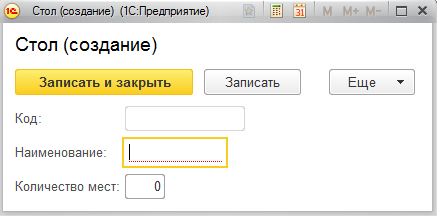


Рисунок А8 – Форма «Стол (создание)»

Добавление нового официанта, рисунок А9:

1) Заполните поле «ФИО» (1);

2) Заполните поле «Номер телефона» (2);

3) Нажать кнопку добавить (3);

4) Выбрать стол, который необходимо закрепить за официантом (4);

5) Нажать на кнопку «Записать и закрыть» (5).

Примечание. В табличной части предусмотрена возможность добавления списка столов, а также проверка на дублирование, то есть пользователь не может выбрать два одинаковых стола, в противном случае, система выведет сообщение об ошибке и не позволит записать уже использующийся стол, рисунок А10.

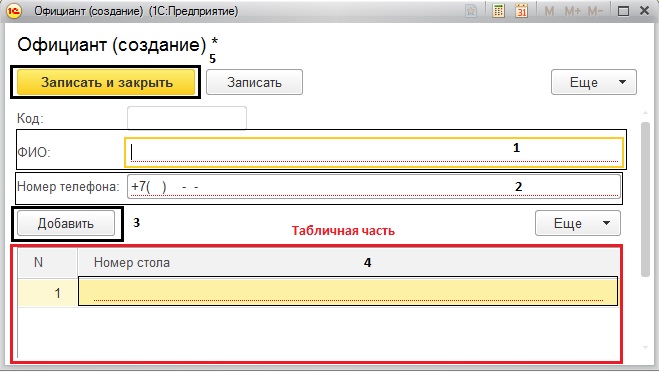


Рисунок А9 – Форма «Официант (создание)»

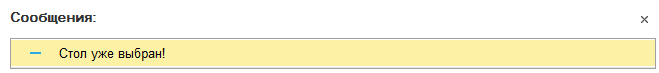


Рисунок А10 – Сообщение об ошибке

Справочник «Общее меню» имеет разделение на группы, в каждую из групп добавляются определенные блюда, рисунок А11. Группу можно создать, редактировать или удалять при помощи кнопки «Еще» или щелчком правой кнопки мыши.

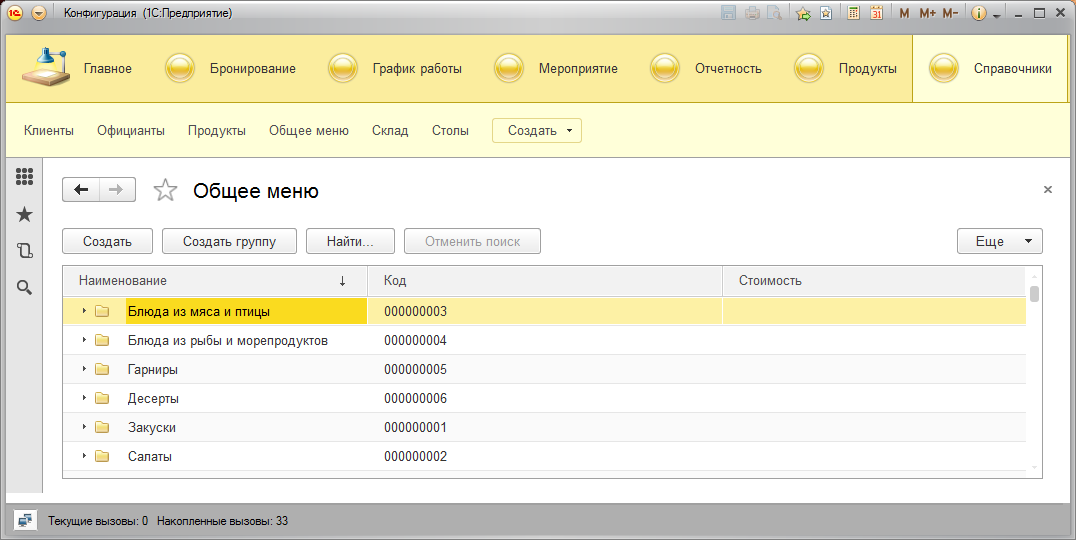


Рисунок А11 – Форма «Общее меню»

Добавление нового блюда, рисунок А12:

1) ввести «Наименование» блюда (1);

2) выбрать родителя, то есть группу (2), если блюдо не было создано в определенной группе;

3) нажать на кнопку «Добавить» для добавления строк в табличную часть, что является списком продуктов для блюда;

4) выбрать «Продукт» (3);

5) ввести «Вес кг» блюда (4);

6) нажать на кнопку «Записать и закрыть».

Примечание. При выборе продукта из справочника «Продукты», ему будет автоматически присваиваться «Цена кг» в зависимости от выбранного продукта (5). При попытке добавления продукта, уже имеющегося в меню, на экран будет выведено сообщение об ошибке, рисунок А13. По ходу добавления продуктов, «Стоимость» (6) блюда будет рассчитываться автоматически.

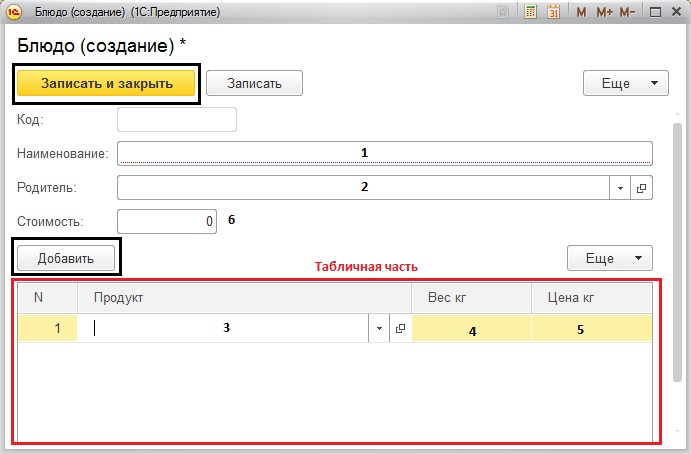


Рисунок А12 – Форма «Блюдо (создание)»

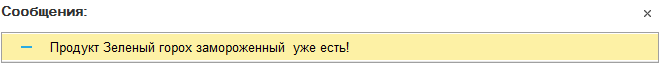


Рисунок А13 – Сообщение об ошибке

После заполнения справочников в подсистеме «Справочники», пользователь необходимо перейти в подсистему «Бронирование». Данная подсистема содержит регистры сведений «Загруженность по залу», документ «Бронирование» и отчет «Загруженность зала (подробно)».

Перейдя на вкладку «Бронирование» (1), в распоряжении пользователя будет список документов, связанных с бронированием столиков (2), рисунок А14.

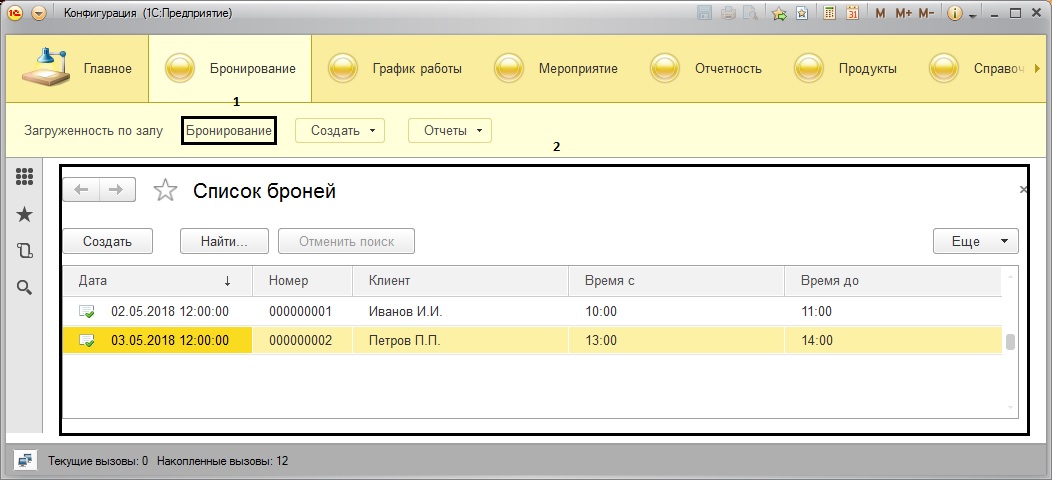


Рисунок – Форма «Список броней»

Создание, редактирование и удаление документа осуществляется по тому же принципу, что и в справочниках системы.

Добавления новой брони, рисунок А15:

1) «Номер» документа заполняется автоматически (1);

2) занести «Дату» бронирования (2);

3) выбрать «Клиента» из списка клиентов, который был создан в справочнике «Клиенты» (3);

4) выбрать «Время с» (4);

5) выбрать «Время до» (5);

6) нажать на кнопку добавить;

7) выбрать «№ стола» из списка столов, который был создан в справочнике «Столы» (6) и указать все столы, которые необходимо забронировать;

8) количество гостей заполняется автоматически в соответствии со справочником «Столы» (7);

9) нажать на кнопку «Провести и закрыть».

Примечание. При попытке добавления стола, уже имеющего в текущей брони, на экран будет выведено сообщение об ошибке, рисунок А16.

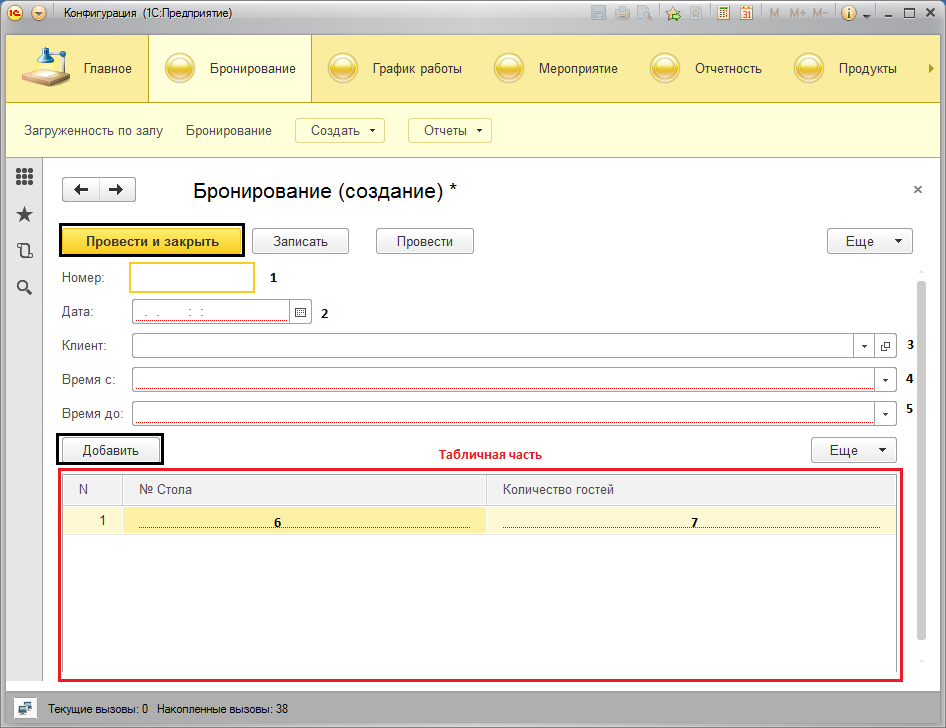


Рисунок А15 – Форма «Бронирование (создание)»

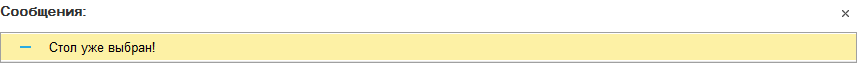


Рисунок А16 – Сообщение об ошибке

После создания документа «Бронирование», его данные автоматически (2) заносятся в регистр сведений «Загруженность по залу» (1), рисунок А17.

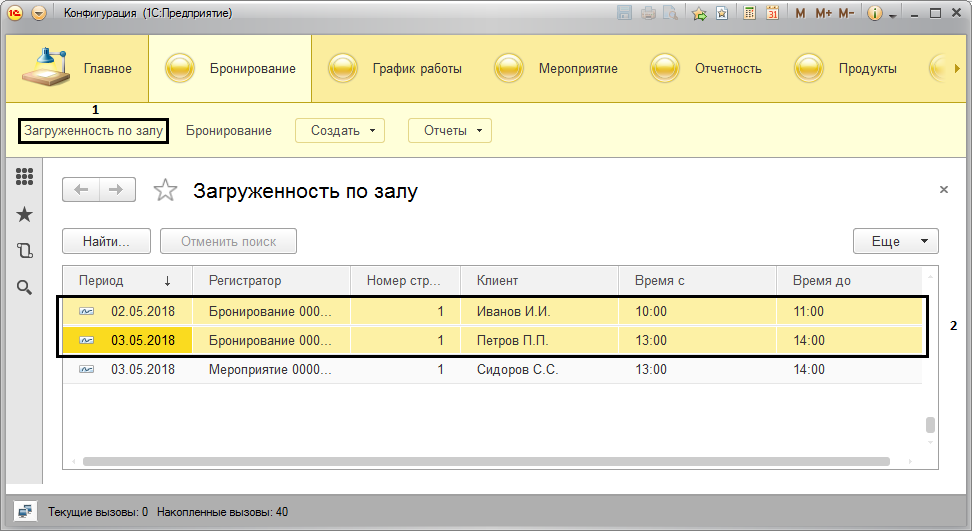


Рисунок А17 – Регистр сведений «Загруженность по залу»

Подробный отчет о загруженность зала пользователь может получить при нажатии кнопки «Отчеты» – «Загруженность зала (подробно)», далее нажав на кнопку «Сформировать», внесенные данные о бронировании будут отображаться в отчете (1), рисунок А18.

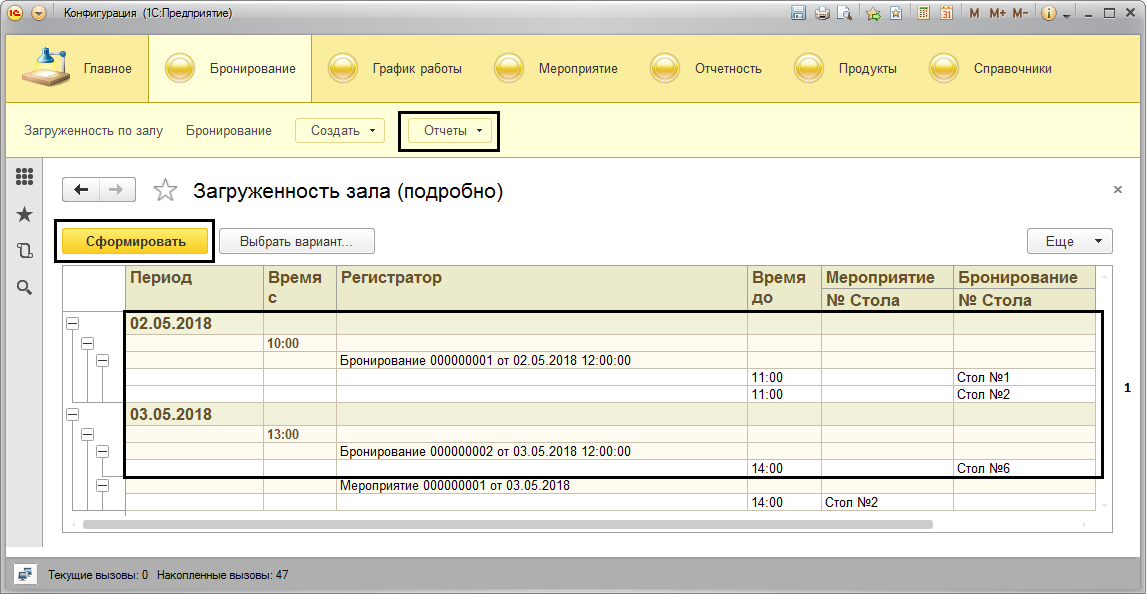


Рисунок А18 – Отчет «Загруженность зала (подробно)»

Теперь пользователю необходимо перейти в подсистему «Мероприятие». Данная подсистема содержит регистры сведений «Загруженность по залу» и отчет «Загруженность зала (подробно)», такие же как в подсистеме «Бронирование», документ «Мероприятие», справочник «Общее меню».

Перейдя на вкладку «Мероприятие» (1), в распоряжении пользователя будет список документов, связанных с проведением мероприятий (2), рисунок А19.

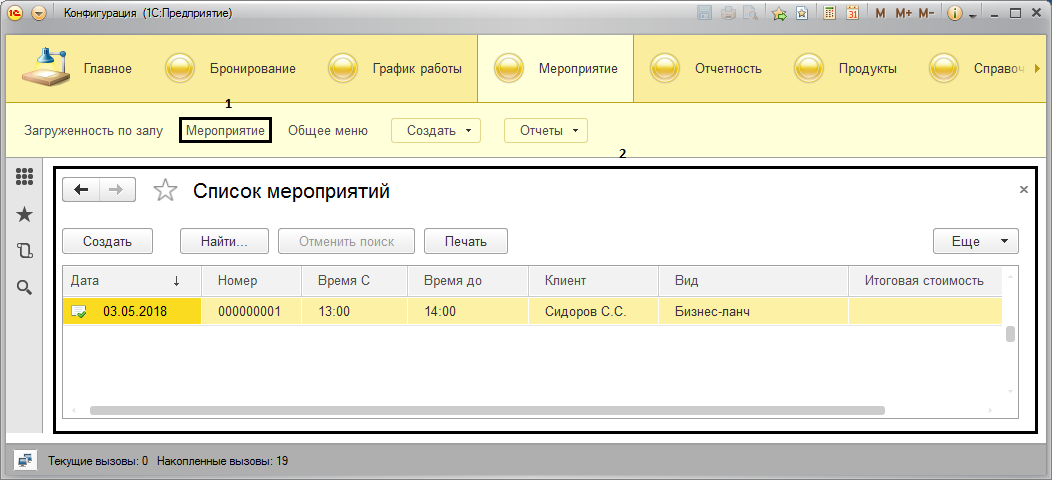


Рисунок А19 – Форма «Список мероприятий»

Создание, редактирование и удаление документа осуществляется по тому же принципу, что и в справочниках системы.

Добавления нового мероприятия, рисунок А20:

1) «Номер» документа заполняется автоматически (1);

2) занести «Дату» мероприятия (2);

3) выбрать «Время с» (3);

4) выбрать «Время до» (4);

5) выбрать «Клиента» из списка клиентов, который был создан в справочнике «Клиенты» (5);

6) выбрать «Вид» мероприятия (6);

7) нажать на кнопку добавить на вкладе «Столы»;

7) выбрать «№ стола» из списка столов, который был создан в справочнике «Столы» (9) и указать все столы, которые необходимо забронировать;

8) количество гостей заполняется автоматически в соответствии со справочником «Столы» (10);

9) перейти на вкладку «Меню», рисунок А21;

10) нажать на кнопку добавить;

11) выбрать «Наименование» блюда (11) из списка блюд, который был создан в справочнике «Общее меню» и указать все блюда, которые необходимо включить в меню текущего мероприятия;

12) указать «Количество» конкретного блюда (12);

13) «Цена за блюдо» заносится системой автоматически при выборе блюда (13);

14) система автоматически рассчитает общую «Сумму» по каждому блюду (14);

15) «Итоговая стоимость» мероприятия рассчитывается также автоматически при составлении меню (7);

16) нажать на кнопку «Провести и закрыть».

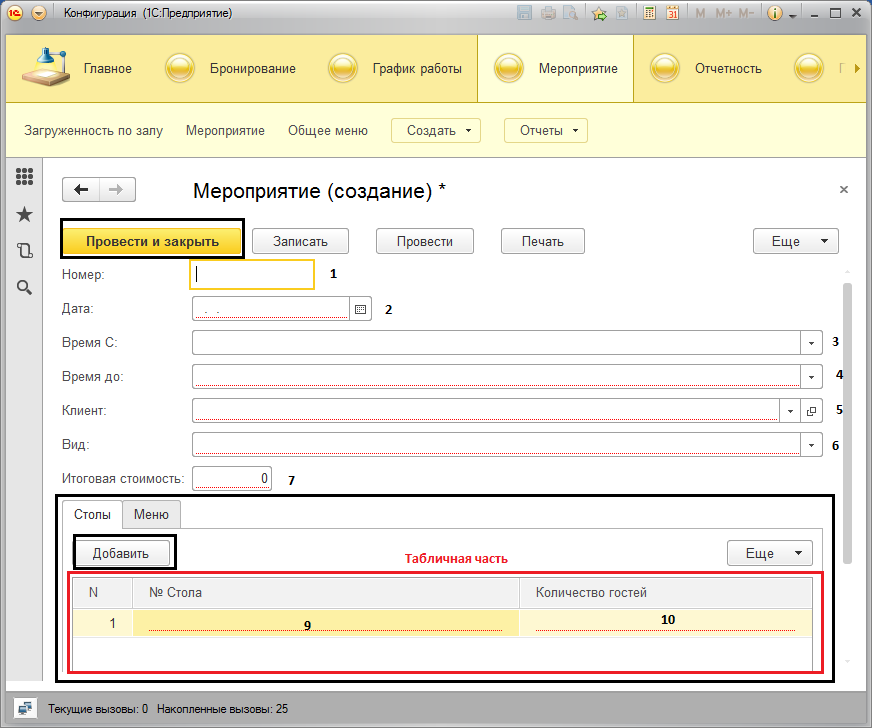


Рисунок А20 – Форма «Мероприятие (создание»)

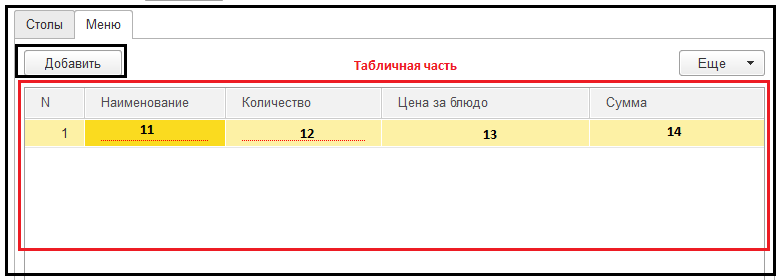


Рисунок А21 – Форма «Мероприятие (создание)», вкладка «Меню»

Примечание. При попытке добавления стола, уже имеющего в текущей брони, на экран будет выведено сообщение об ошибке, рисунок А22.

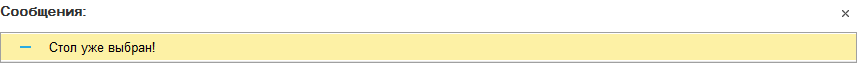


Рисунок А22 – Сообщение об ошибке

После создания документа, его можно распечатать, нажав на соответствующую кнопку «Печать», рисунок А23.

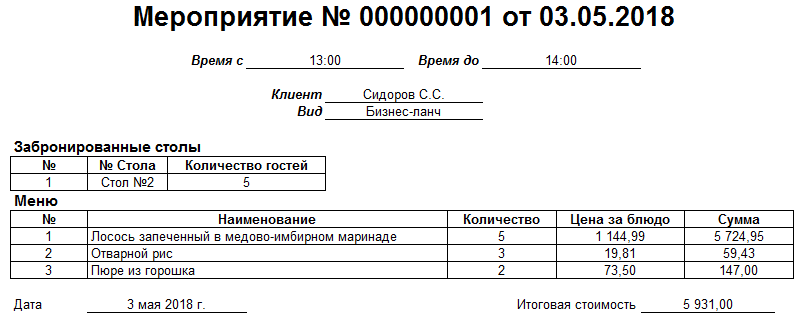


Рисунок А23 – Печатаная форма документа «Мероприятие»

Также пользователь не может записать «Количество гостей» больше, чем мест за выбранным столом, на экран будет выведено соответствующее сообщение об ошибке, рисунок А24.

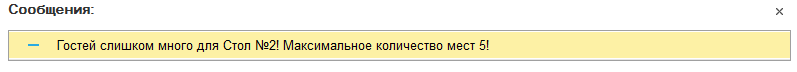


Рисунок А24 – Сообщение об ошибке

Если пользователь пытается добавить одно и тоже блюдо, то появляется сообщение об ошибке, рисунок А25.

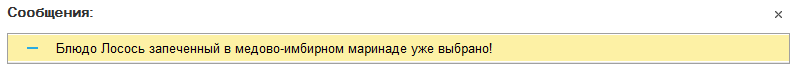


Рисунок А25 – Сообщение об ошибке

В регистр сведений «Загруженность по залу» и отчет «Загруженность зала (подробно)» данные о добавленном мероприятии также заносятся автоматически.

Следующая подсистема для заполнения – подсистема «Продукты». Данная подсистема содержит справочник «Продукты», документы «Приходная накладная» и «Расходная накладная», регистр накопления «Остаток продуктов» и отчет «Продукты на складе».

Первым делом пользователь должен создать документ «Приходная накладная».

Создание новой приходной накладной, рисунок А26:

1) «Номер» документа заполняется автоматически (1);

2) занести «Дату» приходной накладной (2);

3) «Склад» по умолчанию является «Основным» (3);

4) нажать на кнопку «Добавить»;

5) выбрать «Продукт» из списка продуктов, который был создан в справочнике «Продукты» (4) и указать все продукты, которые были поставлены;

6) указать «вес кг» (5);

7) «Цена кг» заносится автоматически при выборе продукта (6);

8) «Сумма» рассчитывается автоматически (7);

9) нажать на кнопку «Провести и закрыть».

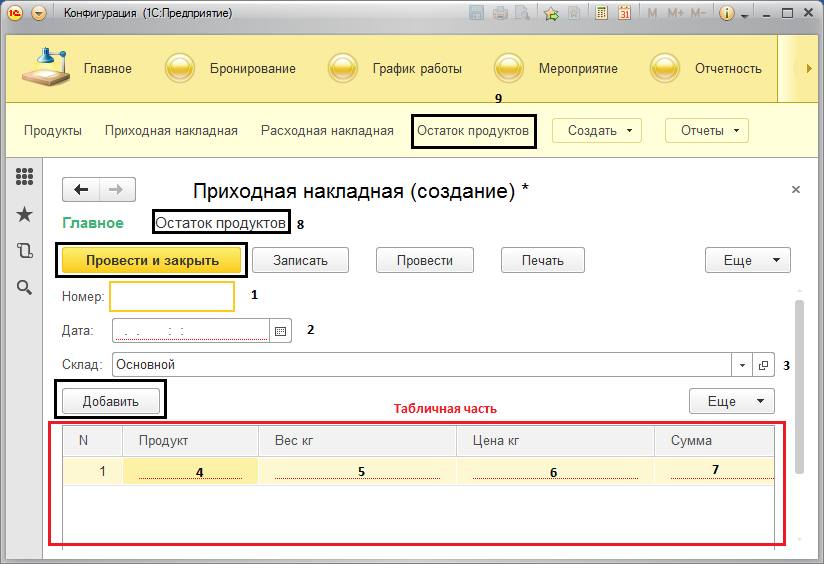


Рисунок А26 – Форма «Приходная накладная (создание)»

Примечание. Исключено дублирование продукта, сообщение об ошибке приведено на рисунке А27.

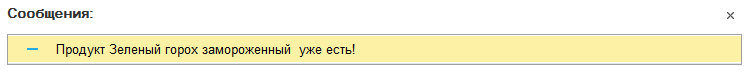


Рисунок А27 – Сообщение об ошибке

После создания документа, его можно распечатать, нажав на соответствующую кнопку «Печать», рисунок А28.

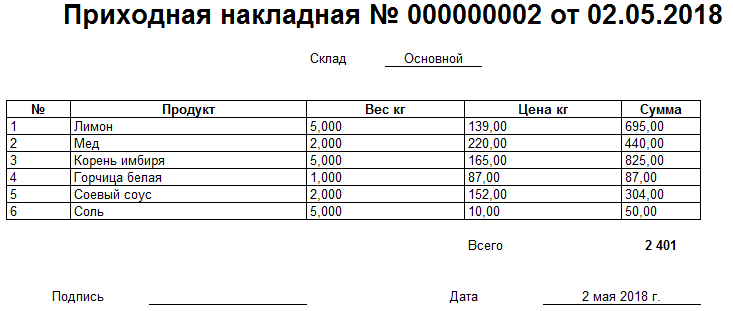


Рисунок А28 – Печатная форма документа «Приходная накладная»

Перейти в регистр накопления «Остаток продуктов», рисунок А29, можно нажав на соответствующую кнопку «Остаток продуктов» (8) или (9), рисунок А26. Данный регистр учитывает приход (+) и расход (-) продукта по соответствующим документам.

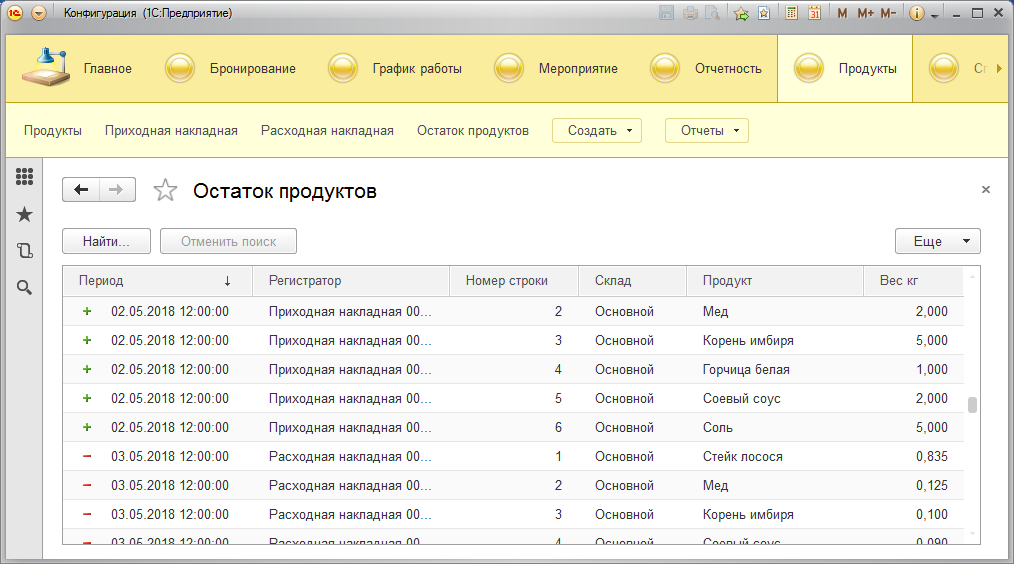


Рисунок А29 – Форма «Остаток продукта»

Создание новой расходной накладной, рисунок А30:

1) «Номер» документа заполняется автоматически (1);

2) занести «Дату» расходной накладной (2);

3) «Склад» по умолчанию является «Основным» (3);

4) выбрать «Мероприятие», к которому относится данная расходная накладная (4);

5) нажать на кнопку «Добавить»;

6) выбрать «Продукт» из списка продуктов, который был создан в справочнике «Продукты» (5) и указать все продукты, которые были поставлены;

7) указать «вес кг» (6);

9) нажать на кнопку «Провести и закрыть».

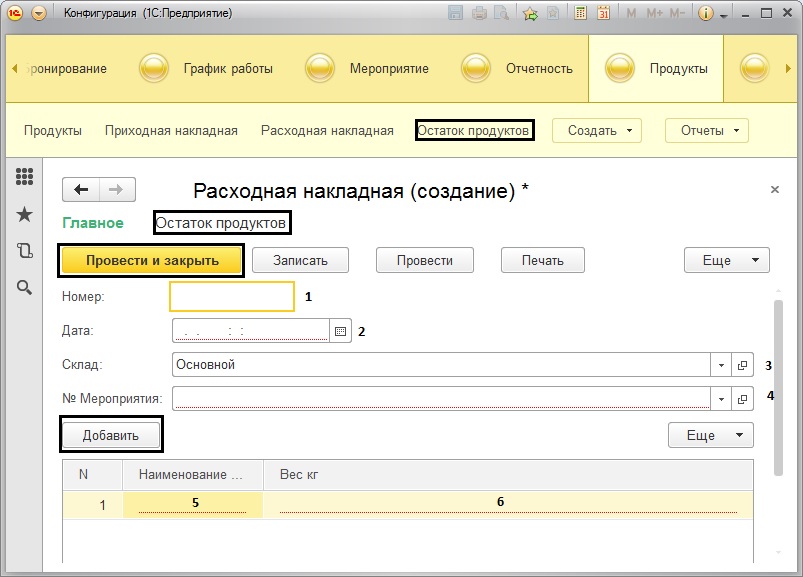


Рисунок А30 – Форма «Расходная накладная (создание)»

Примечание. Системой проводится проверка на наличие продукта на складе, если его не хватает, то выводится соответствующее сообщение об ошибке, рисунок А31.

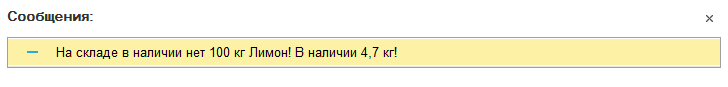


Рисунок А31 – Сообщение об ошибке

А также проверка на дублирование наименований продуктов, если пользователь внесет один и тот же продукт дважды, то выведется сообщение об ошибке, рисунок А32.

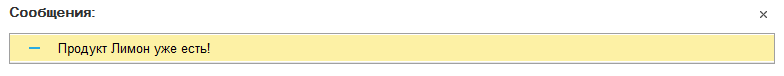


Рисунок А32 – Сообщение об ошибке

После создания документа, его можно распечатать, нажав на соответствующую кнопку «Печать», рисунок А33.

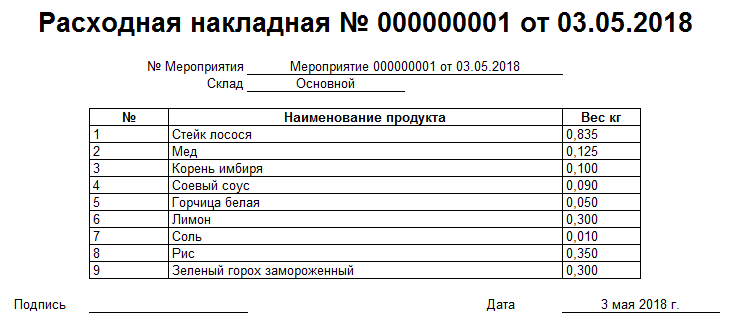


Рисунок А33 – Печатная форма документа «Расходная накладная»

Данные из расходной накладной также учитываются в регистре накопления «Остатки продуктов», рисунок А29.

В данной подсистеме присутствует отчет «Продукты на складе», рисунок А34. Пользователь может задать начало период (1) и конец период (2).

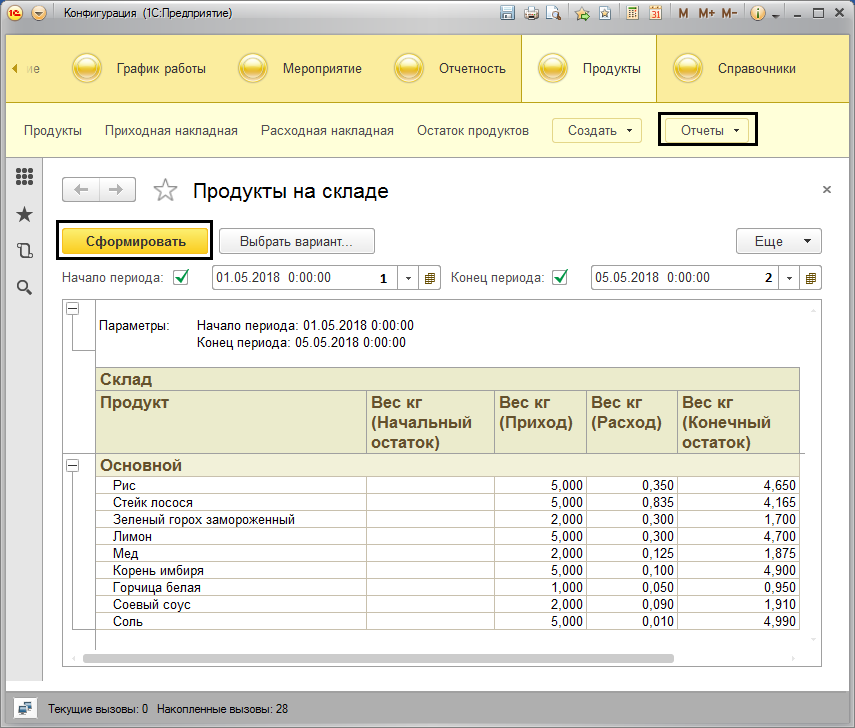


Рисунок А34 – Отчет «Продукты на складе»

Подсистема «График работы» содержит в себе регистр сведений «Загруженность по залу», справочник «Официанты», документ «График работы для официантов», отчет «Загруженность зала (подробно)».

Перед составление нового графика работы для официантов, пользователь должен просмотреть подробный отчет по загруженности зала и график работы официантов за прошлую неделю, а также учесть их предпочтения.

Составление нового графика работы официантов, рисунок А35:

1) «Номер» документа заполняется автоматически (1);

2) указать «дату составления графика» (2);

3) «Дата ПН» рассчитывается системой (3), как и «окончания срока действия» (4);

4) нажать на кнопку «Добавить» на вкладе «Пн, ср, пт, вс»;

5) выбрать «Официантов» из списка официантов, который был создан в справочнике «Официанты» (5) и внести всех официантов, с учетом требований, указанных выше;

6) выбрать соответствующую «Смену» для официанта (6);

7) закрепленный «Столы» автоматически указываются системой (7);

8) нажать на кнопку «Добавить» на вкладе «Вт, чт, сб», рисунок А36;

9) проделать аналогичные действия с полями (8), (9) и (10);

10) нажать на кнопку «Добавить» на вкладе «Список мероприятий», рисунок А37;

11) при нажатии на кнопку добавить система автоматически заполнит поля (11), (12), (13), (14) и (15), если в период действия графика назначено какое-либо мероприятие;

12) если система сформировала мероприятие, то необходимо внести официанта (16).

13) нажать на кнопку «Провести и закрыть».

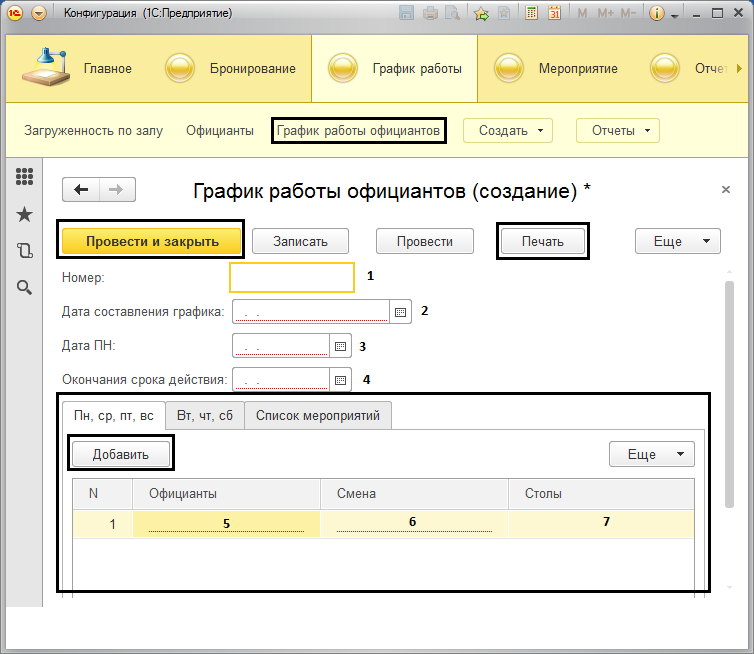


Рисунок А35 – Форма «График работы официантов (создание)»

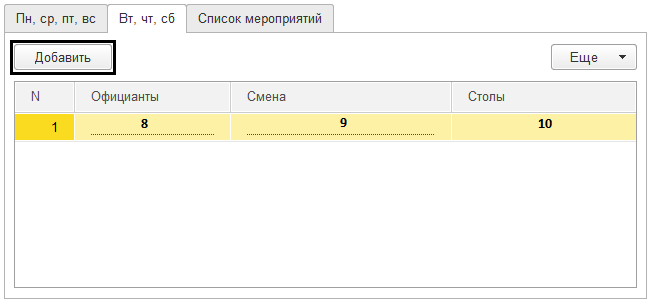


Рисунок А36 – Форма «График работы официантов (создание)», вкладка «Вт, чт, сб»

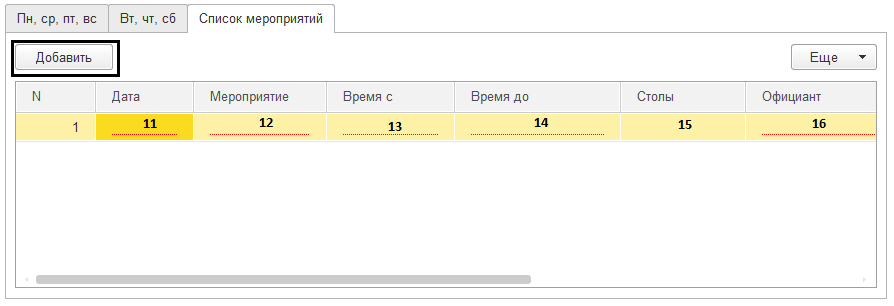


Рисунок А37 – Форма «График работы официантов (создание)», вкладка «Список мероприятий»

Примечание. При составлении графика действует проверка на дублирование официантов, сообщение об ошибке приведено на рисунке А38.

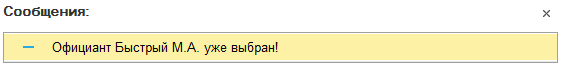


Рисунок А38 – Сообщение об ошибке

Если официант поставлен по нечетным дням недели, то его нельзя поставить на четные дни недели, и наоборот. Сообщение об ошибке представлено на рисунке А39.

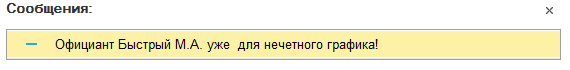


Рисунок А39 – Сообщение об ошибке

После создания документа, его можно распечатать, нажав на соответствующую кнопку «Печать», рисунок А40.

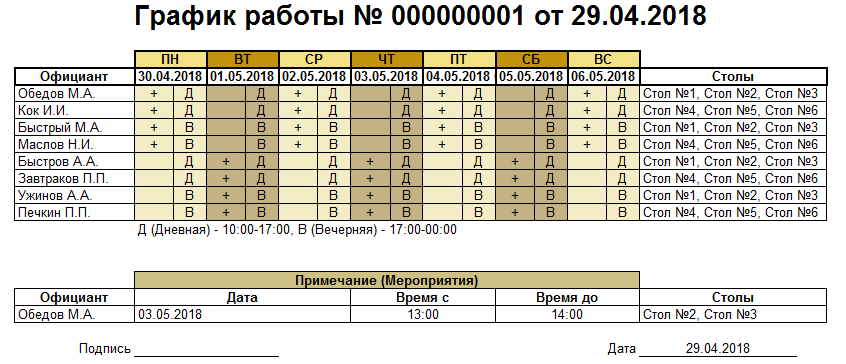


Рисунок А40 – Печатная форма документа «График работы для официантов»

И последняя подсистема – подсистема «Отчетность». В ней содержатся документ «Чек», отчет «Сведений для выдачи премиальных».

Для того, чтобы учитывать выручку за день пользователю необходимо внести все чеки в один документ.

Создание документа «Чек», рисунок А41:

1) «Номер» документа присваивается автоматически (1);

2) указать «Дату» (2);

3) нажать на кнопку «Добавить»;

4) указать «Номер» чека (3);

5) выбрать «Официанта» (4);

5) указать итоговую «Сумму» по чеку (5);

6) проделать вышеуказанные действия (пункт 3-5) по мере поступления чеков;

7) нажать на кнопку «Провести и закрыть».

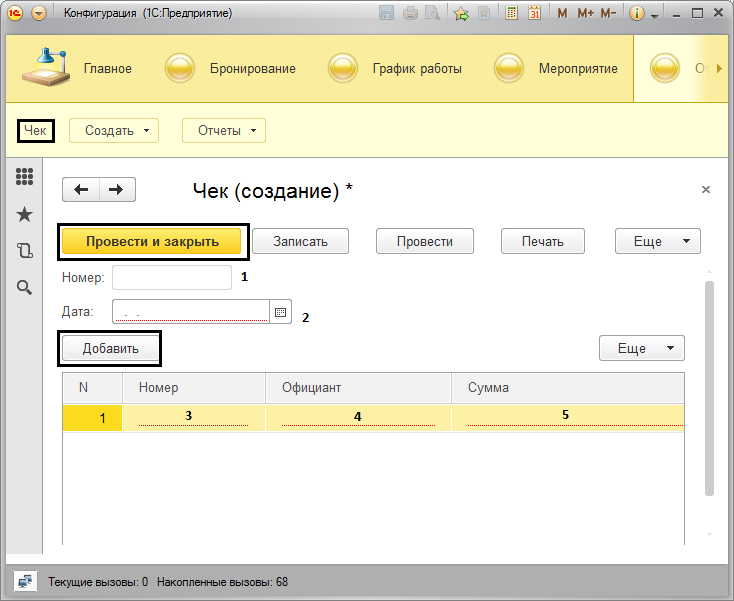


Рисунок А41 – Форма «Чек (создание)»

После создания документа, его можно распечатать, нажав на соответствующую кнопку «Печать», рисунок А42.



Рисунок А42 – Печатная форма документа «Чек»

При формировании отчета «Сведения для выдачи премии», рисунок А43, при выборе начала (1) и конце периодов (2), пользователь должен помнить, что эти сведения составляются в конце каждой недели.

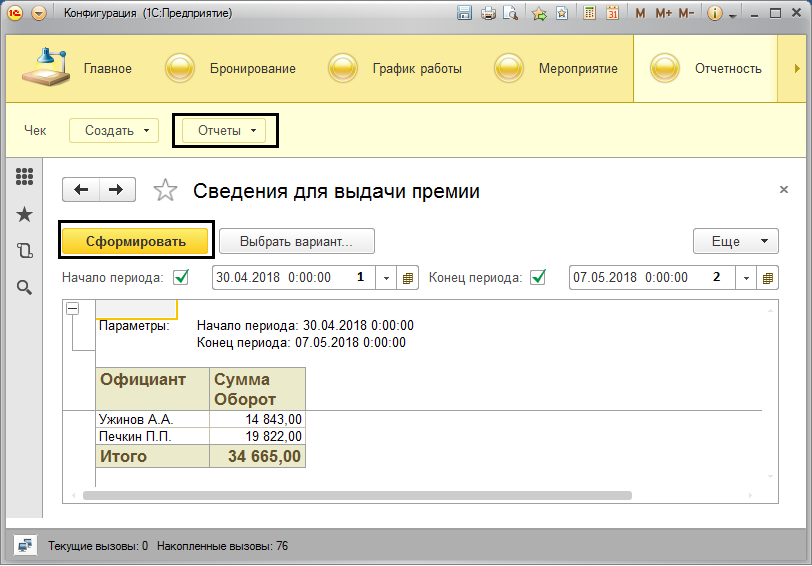


Рисунок А43 – Отчет «Сведения для выдачи премии»

Руководство пользователя для роли «Бухгалтер»

Перед началом работы бухгалтер должен осуществить вход в систему, рисунок А44.

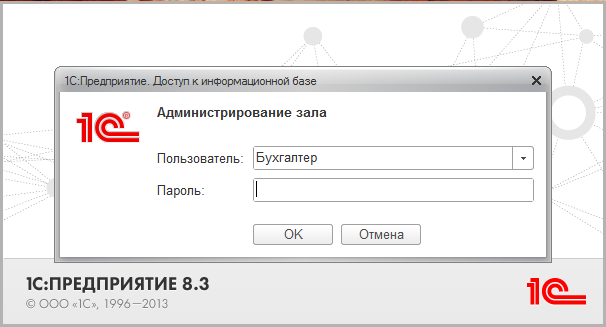


Рисунок А44 – Вход в систему

При успешной идентификации запускается главное окно системы, рисунок А45.

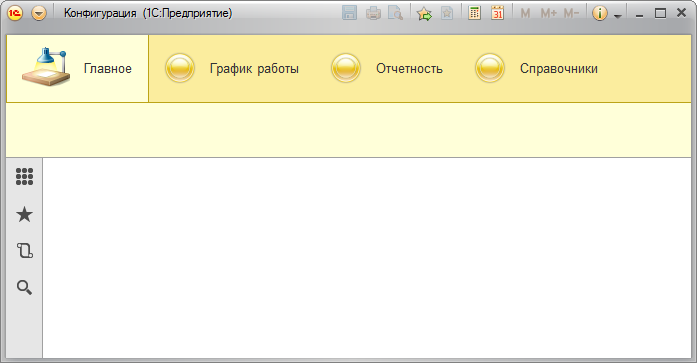


Рисунок А45 – Главное окно системы

Для получения отчета о выручке за день, пользователю необходимо, рисунок А46:

1) зайти в подсистему «Отчетность» (1);

2) выбрать «Чек» (2);

3) списка документов выделить нужный (3);

4) нажать на кнопку «Печать» (4).

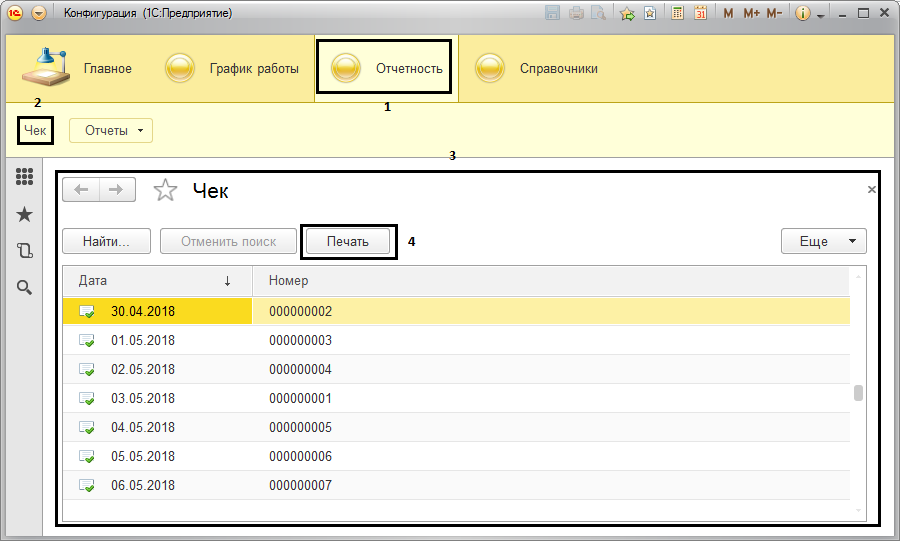


Рисунок А46 – Форма «Чек»

В итоге пользователь получит следующий печатный документ, рисунок А47.



Рисунок А47 – Печатная форма документа «Чек»

В этой же подсистеме пользователь моет обратить к отчету «Сведения для выдачи премии», рисунок А48, чтобы увидеть список официантов, у которых выручка за указанную неделю была больше 14000 рублей:

1) нажать на кнопку «Отчеты»;

2) выбрать «Сведения для выдачи премии»;

3) указать начало периода (1);

4) указать конец периода (2);

5) нажать на кнопку «Сформировать».

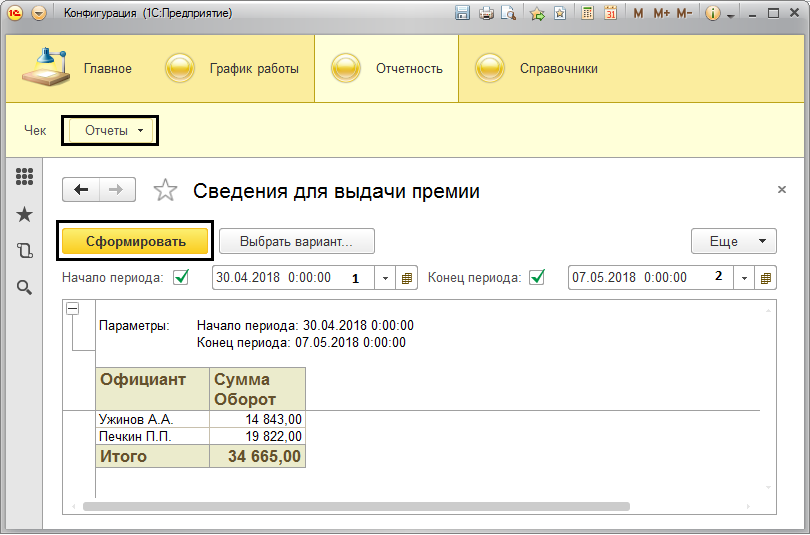


Рисунок А48 – Отчет «Сведений для выдачи премии»

Для того, чтобы узнать какие официанты участвовали в мероприятиях, пользователю необходимо, рисунок А49:

1) перейти на подсистему «График работы»;

2) выделить документ с нужными датами;

3) нажать на кнопку «Печать».

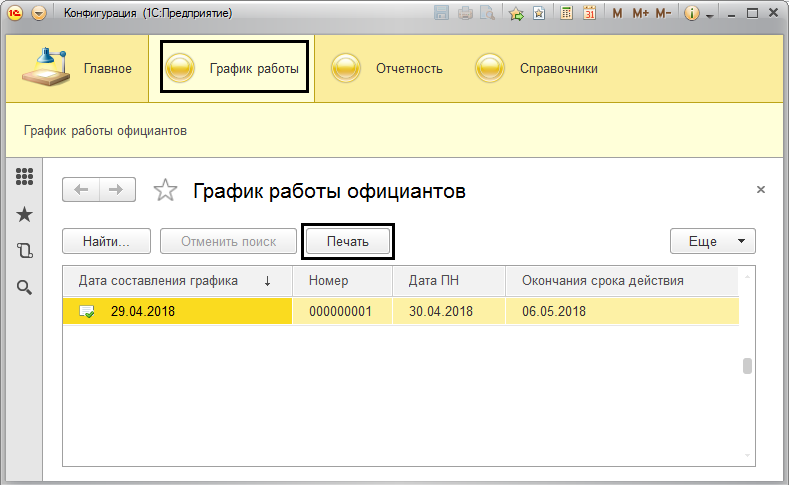


Рисунок А49 – Форма «График работы официантов»

В результате на экран будет выведет документ на печать, в котором пользователь может увидеть нужных официантов в примечании, рисунок А50.

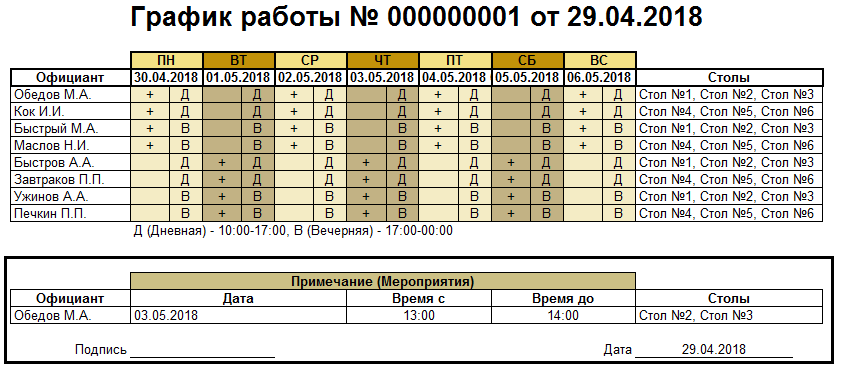


Рисунок А50 – Печатная форма документа «График работы официантов»

**Приложение Б**

**Модули системы**

**Справочник «Официанты». Модуль формы элемента**

//Избавление от дублирования столов

&НаКлиенте

Процедура СтолыНомерСтолаОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

Стол = Объект.Столы;

Для каждого Строка Из Стол Цикл

Если ВыбранноеЗначение = Строка.НомерСтола тогда

СтандартнаяОбработка = ложь;

Сообщить("Стол уже выбран!");

возврат;

КонецЕсли;

КонецЦикла;

КонецПроцедуры

Справочник «Общее меню». Модуль формы элемента

//Получение цены за кг определенного продукта

Функция ПолучитьЦенуКгПродукта(Продукт)

Возврат Справочники.Продукты.НайтиПоНаименованию(Продукт).Цена\_Кг;

КонецФункции

//Избавление от дублирования продуктов, отображение цены продукта за кг

&НаКлиенте

Процедура СоставБлюдаПродуктОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

ПродТаб = Объект.СоставБлюда;

Для каждого Стр ИЗ ПродТаб Цикл

Если ВыбранноеЗначение = Стр.Продукт тогда

СтандартнаяОбработка = Ложь;

Сообщить("Продукт " + ВыбранноеЗначение + " уже есть!");

Возврат;

КонецЕсли;

конецЦикла;

//Получение цены за кг выбранного продукта

Цена = Элементы.СоставБлюда.ТекущиеДанные;

Цена.Цена\_кг = ПолучитьЦенуКгПродукта(ВыбранноеЗначение);

КонецПроцедуры

//Расчет стоимости блюда без наценки для поставки

&НаКлиенте

Процедура СоставБлюдаВес\_кгПриИзменении(Элемент)

СтрСостав = Элементы.СоставБлюда.ТекущиеДанные;

СтрСостав.Стоимость = СтрСостав.Вес\_кг\*СтрСостав.Цена\_Кг;

КонецПроцедуры

//Установление цены блюда с учетом цены поставленных продуктов

&НаКлиенте

Процедура СтоимостьАвтоПодбор(Элемент, Текст, ДанныеВыбора, Параметры, Ожидание, СтандартнаяОбработка)

СтрСостав = Объект.СоставБлюда;

Объект.Стоимость = 0;

Для каждого стр из СтрСостав цикл

Объект.Стоимость = Объект.Стоимость + стр.Стоимость;

конеццикла;

Объект.Стоимость = Объект.Стоимость \* 7;

КонецПроцедуры

Документ «Приходная накладная». Модуль формы элемента

//Расчет суммы по продукту при изменении веса

&НаКлиенте

Процедура ПродуктыВес\_кгПриИзменении(Элемент)

СтрТабЧасти = Элементы.Продукты.ТекущиеДанные;

СтрТабЧасти.Сумма = СтрТабЧасти.Вес\_кг \*СтрТабЧасти.Цена\_кг;

КонецПроцедуры

//Избавление от дублирования наименований продуктов

&НаКлиенте

Процедура ПродуктыПродуктОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

ПродТаб = Объект.Продукты;

Для каждого Стр ИЗ ПродТаб Цикл

Если ВыбранноеЗначение = Стр.Продукт тогда

СтандартнаяОбработка = Ложь;

Сообщить("Продукт " + ВыбранноеЗначение + " уже есть!");

Возврат;

КонецЕсли;

конецЦикла;

//Получение цены за кг выбранного продукта

Цена = Элементы.Продукты.ТекущиеДанные;

Цена.Цена\_кг = ПолучитьЦенуКгПродукта(ВыбранноеЗначение);

КонецПроцедуры

//Получение цены за кг определенного продукта

Функция ПолучитьЦенуКгПродукта(Продукт)

Возврат Справочники.Продукты.НайтиПоНаименованию(Продукт).Цена\_Кг;

КонецФункции

Документ «Приходная накладная». Модуль объекта

//регистр ОстатокПродукты Приход

Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ОстатокПродукты.Записывать = Истина;

Для Каждого ТекСтрокаПродукты Из Продукты Цикл

Движение = Движения.ОстатокПродукты.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение.Период = Дата;

Движение.Склад = Склад;

Движение.Продукт = ТекСтрокаПродукты.Продукт;

Движение.Вес\_кг = ТекСтрокаПродукты.Вес\_кг;

КонецЦикла;

КонецПроцедуры

Документ «Приходная накладная». Модуль менеджера

Процедура Печать(ТабДок, Ссылка) Экспорт

Макет = Документы.Приходная\_Накладная.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Приходная\_Накладная.Дата,

| Приходная\_Накладная.Номер,

| Приходная\_Накладная.Склад,

| Приходная\_Накладная.Продукты.(

| НомерСтроки,

| Продукт,

| Вес\_кг,

| Цена\_кг,

| Сумма

| )

|ИЗ

| Документ.Приходная\_Накладная КАК Приходная\_Накладная

|ГДЕ

| Приходная\_Накладная.Ссылка В (&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ОбластьПродуктыШапка = Макет.ПолучитьОбласть("ПродуктыШапка");

ОбластьПродукты = Макет.ПолучитьОбласть("Продукты");

ОбластьПодвал = Макет.ПолучитьОбласть("Подвал");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ОбластьЗаголовок.Параметры.Номер = Выборка.Номер;

ОбластьЗаголовок.Параметры.Дата = Формат(Выборка.Дата,"ДЛФ=D");

ТабДок.Вывести(ОбластьЗаголовок);

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());

ТабДок.Вывести(ОбластьПродуктыШапка);

СуммаИтог = 0;

ВыборкаПродукты = Выборка.Продукты.Выбрать();

Пока ВыборкаПродукты.Следующий() Цикл

ОбластьПродукты.Параметры.Заполнить(ВыборкаПродукты);

ТабДок.Вывести(ОбластьПродукты, ВыборкаПродукты.Уровень());

СуммаИтог = СуммаИтог + ВыборкаПродукты.Сумма;

КонецЦикла;

ОбластьПодвал.Параметры.ВсегоПоПК = СуммаИтог;

ОбластьПодвал.Параметры.Дата = Формат(Выборка.Дата,"ДЛФ=DD");

ТабДок.Вывести(ОбластьПодвал);

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

КонецПроцедурым

Документ «Расходная накладная». Модуль объекта

//Проверка наличия продукта на складе

Функция ПродуктаНеХватает(Продукт,Вес)

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ОстатокПродуктыОстаткиИОбороты.Вес\_кгКонечныйОстаток

|ИЗ

| РегистрНакопления.ОстатокПродукты.ОстаткиИОбороты КАК ОстатокПродуктыОстаткиИОбороты

|ГДЕ

| ОстатокПродуктыОстаткиИОбороты.Продукт = &Продукт

| И ОстатокПродуктыОстаткиИОбороты.Вес\_кгКонечныйОстаток - &Вес > 0";

Запрос.УстановитьПараметр("Вес", Вес);

Запрос.УстановитьПараметр("Продукт", Продукт);

РезультатЗапроса = Запрос.Выполнить();

Выборка = РезультатЗапроса.Выбрать();

Если Выборка.Следующий() Тогда

Возврат Ложь;

КонецЕсли;

Возврат Истина;

КонецФункции

Процедура ОбработкаПроведения(Отказ, Режим)

Для каждого Строка ИЗ Продукты Цикл

Если ПродуктаНеХватает(Строка.Наименование\_Продукта,Строка.Вес\_кг) Тогда

Сообщить("На складе в наличии нет " + Строка.Вес\_кг + " кг " + Строка.Наименование\_Продукта + "! В наличии " + КолПродНаСкладе(Строка.Наименование\_Продукта)+ " кг!");

Отказ = Истина;

Возврат;

КонецЕсли;

КонецЦикла;

// регистр ОстатокПродукты Расход

Движения.ОстатокПродукты.Записывать = Истина;

Для Каждого ТекСтрокаПродукты Из Продукты Цикл

Движение = Движения.ОстатокПродукты.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Склад = Склад;

Движение.Продукт = ТекСтрокаПродукты.Наименование\_Продукта;

Движение.Вес\_кг = ТекСтрокаПродукты.Вес\_кг;

КонецЦикла;

КонецПроцедуры

//Наличие на складе определенного количества продукта

Функция КолПродНаСкладе(Продукт)

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

|ОстатокПродуктыОстаткиИОбороты.Продукт,

|ОстатокПродуктыОстаткиИОбороты.Вес\_кгКонечныйОстаток КАК Вес\_кгКонечныйОстаток

|ИЗ

|РегистрНакопления.ОстатокПродукты.ОстаткиИОбороты КАК ОстатокПродуктыОстаткиИОбороты

|ГДЕ

| ОстатокПродуктыОстаткиИОбороты.Продукт = &Продукт";

Запрос.УстановитьПараметр("Продукт",Продукт);

РезультатЗапроса = Запрос.Выполнить().Выгрузить();

Выборка = РезультатЗапроса.ВыгрузитьКолонку(1);

возврат Выборка.Получить(0);

КонецФункции

Документ «Расходная накладная». Модуль формы элемента

//Избавление от дублирования продуктов

&НаКлиенте

Процедура ПродуктыНаименование\_ПродуктаОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

ПродТаб = Объект.Продукты;

Для каждого Стр ИЗ ПродТаб Цикл

Если ВыбранноеЗначение = Стр.Наименование\_Продукта тогда

СтандартнаяОбработка = Ложь;

Сообщить("Продукт " + ВыбранноеЗначение + " уже есть!");

Возврат;

КонецЕсли;

конецЦикла;

КонецПроцедуры

Документ «Расходная накладная». Модуль менеджера

Процедура Печать(ТабДок, Ссылка) Экспорт

Макет = Документы.Расходная\_Накладная.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Расходная\_Накладная.Дата,

| Расходная\_Накладная.Мероприятие,

| Расходная\_Накладная.Номер,

| Расходная\_Накладная.Склад,

| Расходная\_Накладная.Продукты.(

| НомерСтроки,

| Наименование\_Продукта,

| Вес\_кг

| )

|ИЗ

| Документ.Расходная\_Накладная КАК Расходная\_Накладная

|ГДЕ

| Расходная\_Накладная.Ссылка В (&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ОбластьПродуктыШапка = Макет.ПолучитьОбласть("ПродуктыШапка");

ОбластьПродукты = Макет.ПолучитьОбласть("Продукты");

Подвал = Макет.ПолучитьОбласть("Подвал");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ОбластьЗаголовок.Параметры.Номер = Выборка.Номер;

ОбластьЗаголовок.Параметры.Дата = Формат(Выборка.Дата,"ДЛФ=D");

ТабДок.Вывести(ОбластьЗаголовок);

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());

ТабДок.Вывести(ОбластьПродуктыШапка);

ВыборкаПродукты = Выборка.Продукты.Выбрать();

Пока ВыборкаПродукты.Следующий() Цикл

ОбластьПродукты.Параметры.Заполнить(ВыборкаПродукты);

ТабДок.Вывести(ОбластьПродукты, ВыборкаПродукты.Уровень());

КонецЦикла;

Подвал.Параметры.Дата = Формат(Выборка.Дата,"ДЛФ=DD");

ТабДок.Вывести(Подвал);

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

КонецПроцедуры

Документ «Бронирование». Модуль формы элемента

//Функция возвращает количество мест бронируемого стола

&НаСервере

Функция ПолучитьКоличествоМест(Стол)

Возврат Справочники.Столы.НайтиПоНаименованию(Стол).Количество\_Мест;

КонецФункции

//Количество гостей не должно превышать максимальное количество мест за столом

&НаКлиенте

Процедура СтолыКоличествоГостейПриИзменении(Элемент)

СтрСтолы = Элементы.Столы.ТекущиеДанные;

МаксКолМест = ПолучитьКоличествоМест(СтрСтолы.НомерСтола);

КолГостей = Элементы.Столы.ТекущиеДанные;

Если КолГостей.КоличествоГостей > МаксКолМест Тогда

Сообщить("Гостей слишком много для " + СтрСтолы.НомерСтола + "! Максимальное количество мест " + МаксКолМест + "!");

Возврат;

КонецЕсли;

КонецПроцедуры

//Проверка на дублирование столов

&НаКлиенте

Процедура СтолыНомерСтолаОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

Стол = Объект.Столы;

Для каждого Строка Из Стол Цикл

Если ВыбранноеЗначение = Строка.НомерСтола тогда

СтандартнаяОбработка = ложь;

Сообщить("Стол уже выбран!");

возврат;

КонецЕсли;

КонецЦикла;

КонецПроцедуры

Документ «Бронирование». Модуль объекта

Процедура ОбработкаПроведения(Отказ, Режим)

// регистр ЗагруженностьЗала

Движения.ЗагруженностьЗала.Записывать = Истина;

Движение = Движения.ЗагруженностьЗала.Добавить();

Движение.Период = Дата;

Движение.Клиент = Клиент;

Движение.Время\_С = Время\_С;

Движение.Время\_До = Время\_До;

КонецПроцедуры

Документ «Бронирование». Модуль формы элемента

//Расчет итоговой стоимости по каждому блюду

&НаКлиенте

Процедура МенюКоличествоПриИзменении(Элемент)

СтрМеню = Элементы.Меню.ТекущиеДанные;

СтрМеню.Сумма = СтрМеню.Количество\*СтрМеню.Цена\_За\_Блюдо;

Объект.Итоговая\_Стоимость = Объект.Итоговая\_Стоимость + СтрМеню.Сумма;

КонецПроцедуры

//Проверка на дублирование блюд и сопосталение цен с общим меню

&НаКлиенте

Процедура МенюНаименованиеОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

//Проверка на наличие блюда в меню для мероприятия

Мен = Объект.Меню;

Для каждого Строка Из Мен Цикл

Если ВыбранноеЗначение = Строка.Наименование тогда

СтандартнаяОбработка = ложь;

Сообщить("Блюдо "+ВыбранноеЗначение+" уже выбрано!");

возврат;

КонецЕсли;

КонецЦикла;

//Сопоставление блюду его цены из общего меню

СтрМеню = Элементы.Меню.ТекущиеДанные;

СтрМеню.Цена\_За\_Блюдо = ПолучитьСтоимость(ВыбранноеЗначение);

КонецПроцедуры

//Функция для получения стоимости блюда из общего меню

&НаСервере

Функция ПолучитьСтоимость (Блюдо)

Возврат Справочники.Общее\_Меню.НайтиПоНаименованию(Блюдо).Стоимость;

КонецФункции

//Пользователь не может дважды выбрать один и тот же стол

&НаКлиенте

Процедура СтолыНомерСтолаОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

Стол = Объект.Столы;

Для каждого Строка Из Стол Цикл

Если ВыбранноеЗначение = Строка.НомерСтола тогда

СтандартнаяОбработка = ложь;

Сообщить("Стол уже выбран!");

возврат;

КонецЕсли;

КонецЦикла;

КонецПроцедуры

//Функция возвращает количество мест бронируемого стола

&НаСервере

Функция ПолучитьКоличествоМест(Стол)

Возврат Справочники.Столы.НайтиПоНаименованию(Стол).Количество\_Мест;

КонецФункции

//Проверка для вместимости стола (чтобы гостей было меньше или равно количеству мест бронируемого стола)

&НаКлиенте

Процедура СтолыКоличествоГостейПриИзменении(Элемент)

СтрСтолы = Элементы.Столы.ТекущиеДанные;

МаксКолМест = ПолучитьКоличествоМест(СтрСтолы.НомерСтола);

КолГостей = Элементы.Столы.ТекущиеДанные;

Если КолГостей.КоличествоГостей > МаксКолМест Тогда

Сообщить("Гостей слишком много для " + СтрСтолы.НомерСтола + "! Максимальное количество мест " + МаксКолМест + "!");

Возврат;

КонецЕсли;

КонецПроцедуры

Документ «Бронирование». Модуль объекта

Процедура ОбработкаПроведения(Отказ, Режим)

// регистр ЗагруженностьЗала

Движения.ЗагруженностьЗала.Записывать = Истина;

Движение = Движения.ЗагруженностьЗала.Добавить();

Движение.Период = Дата;

Движение.Клиент = Клиент;

Движение.Время\_С = Время\_С;

Движение.Время\_До = Время\_До;

КонецПроцедуры

Документ «Бронирование». Модуль менеджера

Процедура Печать(ТабДок, Ссылка) Экспорт

Макет = Документы.Мероприятие.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Мероприятие.Вид,

| Мероприятие.Время\_До,

| Мероприятие.Время\_С,

| Мероприятие.Дата,

| Мероприятие.Итоговая\_Стоимость,

| Мероприятие.Клиент,

| Мероприятие.Номер,

| Мероприятие.Столы.(

| НомерСтроки,

| НомерСтола,

| КоличествоГостей

| ),

| Мероприятие.Меню.(

| НомерСтроки,

| Наименование,

| Количество,

| Цена\_За\_Блюдо,

| Сумма

| )

|ИЗ

| Документ.Мероприятие КАК Мероприятие

|ГДЕ

| Мероприятие.Ссылка В (&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ОбластьСтолыШапка = Макет.ПолучитьОбласть("СтолыШапка");

ОбластьСтолы = Макет.ПолучитьОбласть("Столы");

ОбластьМенюШапка = Макет.ПолучитьОбласть("МенюШапка");

ОбластьМеню = Макет.ПолучитьОбласть("Меню");

Подвал = Макет.ПолучитьОбласть("Подвал");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ОбластьЗаголовок.Параметры.Номер = Выборка.Номер;

ОбластьЗаголовок.Параметры.Дата = Формат(Выборка.Дата,"ДЛФ=D");

ТабДок.Вывести(ОбластьЗаголовок);

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());

ТабДок.Вывести(ОбластьСтолыШапка);

ВыборкаСтолы = Выборка.Столы.Выбрать();

Пока ВыборкаСтолы.Следующий() Цикл

ОбластьСтолы.Параметры.Заполнить(ВыборкаСтолы);

ТабДок.Вывести(ОбластьСтолы, ВыборкаСтолы.Уровень());

КонецЦикла;

ТабДок.Вывести(ОбластьМенюШапка);

ВыборкаМеню = Выборка.Меню.Выбрать();

Пока ВыборкаМеню.Следующий() Цикл

ОбластьМеню.Параметры.Заполнить(ВыборкаМеню);

ТабДок.Вывести(ОбластьМеню, ВыборкаМеню.Уровень());

КонецЦикла;

Подвал.Параметры.Дата = Формат(Выборка.Дата,"ДЛФ=DD");

Подвал.Параметры.Итоговая\_Стоимость = Выборка.Итоговая\_Стоимость;

ТабДок.Вывести(Подвал);

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

КонецПроцедуры

Документ «Чек». Модуль объекта

Процедура ОбработкаПроведения(Отказ, Режим)

//регистр накопления сведения для выдачи премии

Движения.СведенияДляВыдачиПремии.Записывать = Истина;

Для Каждого ТекСтрокаЧеки Из Чеки Цикл

Движение = Движения.СведенияДляВыдачиПремии.Добавить();

Движение.Период = Дата;

Движение.Дата = Дата;

Движение.Официант = ТекСтрокаЧеки.Официант;

Движение.Сумма = ТекСтрокаЧеки.Сумма;

КонецЦикла;

КонецПроцедуры

Документ «Чек». Модуль менеджера

Процедура Печать(ТабДок, Ссылка) Экспорт

Макет = Документы.Чек.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Чек.Дата,

| Чек.Чеки.(

| НомерСтроки,

| Сумма,

| Номер,

| Официант

| )

|ИЗ

| Документ.Чек КАК Чек

|ГДЕ

| Чек.Ссылка В(&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

ОбластьЧекиШапка = Макет.ПолучитьОбласть("ЧекиШапка");

ОбластьЧеки = Макет.ПолучитьОбласть("Чеки");

Подвал = Макет.ПолучитьОбласть("Подвал");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ОбластьЗаголовок.Параметры.Дата = Формат(Выборка.Дата,"ДЛФ=D");

ТабДок.Вывести(ОбластьЗаголовок);

СуммаИтог = 0;

ТабДок.Вывести(ОбластьЧекиШапка);

ВыборкаЧеки = Выборка.Чеки.Выбрать();

Пока ВыборкаЧеки.Следующий() Цикл

ОбластьЧеки.Параметры.Заполнить(ВыборкаЧеки);

ТабДок.Вывести(ОбластьЧеки, ВыборкаЧеки.Уровень());

СуммаИтог = СуммаИтог + ВыборкаЧеки.Сумма;

КонецЦикла;

Подвал.Параметры.ИтогоПоДокументу = СуммаИтог;

Подвал.Параметры.Дата = Формат(Выборка.Дата,"ДЛФ=DD");

ТабДок.Вывести(Подвал);

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

КонецПроцедуры

Документ «График». Модуль формы элемента

//Привязка закрепленных столов

&НаСервере

Функция ПолучитьСтолы(Официант)

Поиск = Справочники.Официанты.НайтиПоНаименованию(Официант).Столы;

ВсеСтолы = "";

Для каждого стр из поиск цикл

Если ВсеСтолы = "" тогда

ВсеСтолы = ВсеСтолы+стр.НомерСтола;

иначе

ВсеСтолы = ВсеСтолы + ", " + стр.НомерСтола;

конецесли;

КонецЦикла;

Возврат ВсеСтолы;

КонецФункции

//Избавление от дублирования,привязка столов, мероприятия

&НаКлиенте

Процедура ПнСрПтВсОфициантыОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

ОфЧет = Объект.ВтЧтСб;

Оф = Объект.ПнСрПтВс;

для каждого Стр из Оф цикл

Для каждого Стр2 из ОфЧет цикл

Если ВыбранноеЗначение = Стр2.Официанты тогда

СтандартнаяОбработка = ложь;

Сообщить("Официант " + ВыбранноеЗначение + " уже для четного графика!");

возврат;

конецЕсли;

конецЦикла;

конецЦикла;

Для каждого Стр Из Оф Цикл

Если ВыбранноеЗначение = Стр.Официанты тогда

СтандартнаяОбработка = ложь;

Сообщить("Официант " + ВыбранноеЗначение + " уже выбран!");

возврат;

КонецЕсли;

КонецЦикла;

//Привязка столов

Ст = Элементы.ПнСрПтВс.ТекущиеДанные;

Ст.Столы = "";

Ст.Столы = ПолучитьСтолы(ВыбранноеЗначение);

КонецПроцедуры

//Избавление от дублирования, привязка столов, мероприятия

&НаКлиенте

Процедура ВтЧтСбОфициантыОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

ОфНечет = объект.ПнСрПтВс;

Оф = Объект.ВтЧтСб;

для каждого Стр из Оф цикл

Для каждого Стр2 из ОфНечет цикл

Если ВыбранноеЗначение = Стр2.Официанты тогда

СтандартнаяОбработка = ложь;

Сообщить("Официант " + ВыбранноеЗначение + " уже для нечетного графика!");

возврат;

конецЕсли;

конецЦикла;

конецЦикла;

Для каждого Стр Из Оф Цикл

Если ВыбранноеЗначение = Стр.Официанты тогда

СтандартнаяОбработка = ложь;

Сообщить("Официант " + ВыбранноеЗначение + " уже выбран!");

возврат;

КонецЕсли;

КонецЦикла;

//Привязка столов

Ст = Элементы.ВтЧтСб.ТекущиеДанные;

Ст.Столы = "";

Ст.Столы = ПолучитьСтолы(ВыбранноеЗначение);

КонецПроцедуры

//Автоматическое заполнение даты для ПН (Дата составления + один день)

&НаКлиенте

Процедура ДатаПриИзменении(Элемент)

Объект.ДатаПН = Объект.Дата + 60 \* 60 \* 24;

Объект.ОкончанияСрокаДействия = Объект.Дата + 60\*60\*24\*7;

КонецПроцедуры

//Автоматическое заполнение нужного мероприятия

&НаКлиенте

Процедура СписокМероприятийПередНачаломДобавления(Элемент, Отказ, Копирование, Родитель, Группа, Параметр)

ПолучитьМероприятие();

КонецПроцедуры

&НаСервере

//Поиск мероприятий для текущего графика

Функция ПолучитьМероприятие()

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Мероприятие.Ссылка,

| Мероприятие.Дата,

| Мероприятие.Номер,

| Мероприятие.Время\_С,

| Мероприятие.Время\_До,

| Мероприятие.Столы

|ИЗ

| Документ.Мероприятие КАК Мероприятие,

| Документ.График КАК График

|ГДЕ

| Мероприятие.Дата <= График.ДатаПН

| И Мероприятие.Дата >= НЕДЕЛЯ(График.ДатаПН)";

Выборка = Запрос.Выполнить().Выбрать();

МерГраф = Объект.СписокМероприятий;

ВсеСтолы = "";

Пока Выборка.Следующий() цикл

стр = МерГраф.Добавить();

стр.Мероприятие = Выборка.Ссылка;

стр.Дата = Выборка.Дата;

стр.Время\_С = Выборка.Время\_С;

стр.Время\_До = Выборка.Время\_До;

ВыборкаСтолов = Документы.Мероприятие.НайтиПоНомеру(Выборка.Номер).Столы.Выгрузить();

Для каждого Стол из ВыборкаСтолов Цикл

Если ВсеСтолы = "" тогда

ВсеСтолы = ВсеСтолы + Стол.НомерСтола;

иначе

ВсеСтолы = ВсеСтолы + ", " + Стол.НомерСтола;

конецесли;

стр.Столы = ВсеСтолы;

КонецЦикла;

выборка.Уровень();

КонецЦикла;

КонецФункции

Документ «График». Модуль менеджера

Процедура Печать(ТабДок, Ссылка) Экспорт

Макет = Документы.График.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| График.Дата,

| График.ДатаПН,

| График.Номер,

| График.ПнСрПтВс.(

| НомерСтроки,

| Официанты,

| Смена,

| Столы

| ),

| График.ВтЧтСб.(

| НомерСтроки,

| Официанты,

| Смена,

| Столы

| ),

| График.СписокМероприятий.(

| Дата,

| Время\_С,

| Время\_До,

| Столы,

| Официант

| )

|ИЗ

| Документ.График КАК График

|ГДЕ

| График.Ссылка В(&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

Заголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ДанныеНечет = Макет.ПолучитьОбласть("ДанныеНечет");

ДанныеЧет = Макет.ПолучитьОбласть("ДанныеЧет");

ПримечениеГрафик = Макет.ПолучитьОбласть("ПримечениеГрафик");

ШапкаМер = Макет.ПолучитьОбласть("ШапкаМер");

ДанныеМер = Макет.ПолучитьОбласть("ДанныеМер");

Подвал = Макет.ПолучитьОбласть("Подвал");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

//ЗАГОЛОВОК

Заголовок.Параметры.Номер = Выборка.Номер;

Заголовок.Параметры.Дата = Формат(Выборка.Дата,"ДФ=dd.MM.yyyy");

ТабДок.Вывести(Заголовок);

//ШАПКА

Шапка.Параметры.ДатаПН = Выборка.ДатаПН;

Шапка.Параметры.ДатаВТ = Шапка.Параметры.ДатаПН + 60 \* 60 \* 24;

Шапка.Параметры.ДатаСР = Шапка.Параметры.ДатаВТ + 60 \* 60 \* 24;

Шапка.Параметры.ДатаЧТ = Шапка.Параметры.ДатаСР + 60 \* 60 \* 24;

Шапка.Параметры.ДатаПТ = Шапка.Параметры.ДатаЧТ + 60 \* 60 \* 24;

Шапка.Параметры.ДатаСБ = Шапка.Параметры.ДатаПТ + 60 \* 60 \* 24;

Шапка.Параметры.ДатаВС = Шапка.Параметры.ДатаСБ + 60 \* 60 \* 24;

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка);

//ДАННЫЕ НЕЧЕТ НЕДЕЛИ

ВыборкаПнСрПтВс = Выборка.ПнСрПтВс.Выбрать();

Пока ВыборкаПнСрПтВс.Следующий() Цикл

ДанныеНечет.Параметры.Официанты = ВыборкаПнСрПтВс.Официанты;

//Заполнение отметок

ДанныеНечет.Параметры.ОтметкаПН = "+";

ДанныеНечет.Параметры.ОтметкаСР = "+";

ДанныеНечет.Параметры.ОтметкаПТ = "+";

ДанныеНечет.Параметры.ОтметкаВС = "+";

//Сокращение наименования вида смены

Если ВыборкаПнСрПтВс.Смена = Перечисления.ВидыСмен.Дневная тогда

ДанныеНечет.Параметры.Смена = "Д";

Иначе ДанныеНечет.Параметры.Смена = "В";

КонецЕсли;

//Указание закрепленных столов

ДанныеНечет.Параметры.Столы = ВыборкаПнСрПтВс.Столы;

ТабДок.Вывести(ДанныеНечет, ВыборкаПнСрПтВс.Уровень());

КонецЦикла;

//ДАННЫЕ ЧЕТ НЕДЕЛИ

ВыборкаВтЧтСб = Выборка.ВтЧтСб.Выбрать();

Пока ВыборкаВтЧтСб.Следующий() Цикл

ДанныеЧет.Параметры.Официанты = ВыборкаВтЧтСб.Официанты;

//Заполнение отметок

ДанныеЧет.Параметры.ОтметкаВТ = "+";

ДанныеЧет.Параметры.ОтметкаЧТ = "+";

ДанныеЧет.Параметры.ОтметкаСБ = "+";

//Сокращение наименования вида смены

Если ВыборкаВтЧтСб.Смена = Перечисления.ВидыСмен.Дневная тогда

ДанныеЧет.Параметры.Смена = "Д";

Иначе ДанныеЧет.Параметры.Смена = "В";

КонецЕсли;

//Указание закрепленных столов

ДанныеЧет.Параметры.Столы = ВыборкаВтЧтСб.Столы;

ТабДок.Вывести(ДанныеЧет, ВыборкаВтЧтСб.Уровень());

КонецЦикла;

//Примечание для графика

ТабДок.Вывести(ПримечениеГрафик);

//ПРИМЕЧАНИЕ

ТабДок.Вывести(ШапкаМер);

ВыборкаСписокМероприятий = Выборка.СписокМероприятий.Выбрать();

Пока ВыборкаСписокМероприятий.Следующий() цикл

ДанныеМер.Параметры.Официанты = ВыборкаСписокМероприятий.Официант;

ДанныеМер.Параметры.ДатаМер = Формат(ВыборкаСписокМероприятий.Дата,"ДФ=dd.MM.yyyy");

ДанныеМер.Параметры.ВремяС = ВыборкаСписокМероприятий.Время\_С;

ДанныеМер.Параметры.ВремяДо = ВыборкаСписокМероприятий.Время\_До;

ДанныеМер.Параметры.СтолМер = ВыборкаСписокМероприятий.Столы;

ТабДок.Вывести(ДанныеМер, ВыборкаСписокМероприятий.Уровень());

КонецЦикла;

//ПОДВАЛ

Подвал.Параметры.Дата = Формат(Выборка.Дата,"ДФ=dd.MM.yyyy");

ТабДок.Вывести(Подвал);

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

КонецПроцедуры

**Приложение В**

**Таблица описания реквизитов, участвующих в системе**

Таблица В1 – Описание реквизитов «Клиенты»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя** | **Тип в IDEF1X** | **Тип в 1C** | **Значение по умолчанию** | **Ограничение** |
| Код | Intеger (10) | Число (9) | Автоматическое | Инкрементируемое |
| ФИО | Char (255) | Строка (100) | Пустое | Обязательное к заполнению |
| Номер\_Телефона | Intеger (11) | Строка (17) | Пустое | Обязательное к заполнению |

Таблица В2 – Описание реквизитов «Столы»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя** | **Тип в IDEF1X** | **Тип в 1C** | **Значение по умолчанию** | **Ограничение** |
| Код | Intеger (10) | Число (9) | Автоматическое | Инкрементируемое |
| Наименование |  | Строка (7) | Пустое | Обязательное к заполнению |
| Количество\_Мест | Intеger (1) | Число (1) | Пустое | Обязательное к заполнению, неотрицательное |

Таблица В3 – Описание реквизитов «Официанты»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя** | **Тип в IDEF1X** | **Тип в 1C** | **Значение по умолчанию** | **Ограничение** |
| Код | Intеger (10) | Число (9) | Автоматическое | Инкрементируемое |
| ФИО | Char (255) | Строка (100) | Пустое | Обязательное к заполнению |
| Номер\_Телефона | Intеger (11) | Строка (17) | Пустое | Обязательное к заполнению |

Таблица В4 – Описание реквизитов «График»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя** | **Тип в IDEF1X** | **Тип в 1C** | **Значение по умолчанию** | **Ограничение** |
| Код | Intеger (10) | Число (9) | Автоматическое | Инкрементируемое |
| Общее\_Количество\_  Часов\_В\_Неделю | Intеger (3) |  | Пустое | Обязательное к заполнению |
| Дата | Дата | Дата | Пустое | Обязательное к заполнению |
|  |  |  |  |  |
|  |  |  |  |  |
| *Окончание таблицы В4* | | | | |
| **Имя** | **Тип в IDEF1X** | **Тип в 1C** | **Значение по умолчанию** | **Ограничение** |
| ДатаПН |  | Дата | Пустое | Обязательное к заполнению |
| ОкончаниеСрокаДействия |  | Дата | Пустое | Обязательное к заполнению |
| ПнСрПтВс.Официанты |  | СправочникСсылка.  Официанты | Пустое | Обязательное к заполнению |
| ПнСрПтВс.Смена |  | ПеречислениеСсылка.  ВидыСмен | Пустое | Обязательное к заполнению |
| ПнСрПтВс.Столы |  | СправочникСсылка.  Столы | Пустое | Обязательное к заполнению |
| ВтЧтСб.Официанты |  | СправочникСсылка.  Официанты | Пустое | Обязательное к заполнению |
| ВтЧтСб.Смена |  | ПеречислениеСсылка.  ВидыСмен | Пустое | Обязательное к заполнению |
| ВтЧтСб.Столы |  | СправочникСсылка.  Столы | Пустое | Обязательное к заполнению |
| СписокМероприятий.Дата |  | Дата | Пустое | Обязательное к заполнению |
| СписокМероприятий.  Мероприятие |  | ДокументСсылка.  Мероприятие | Пустое | Нет |
| СписокМероприятий.  Дата\_С |  | ПеречислениеСсылка.  Время\_Работы | Пустое | Нет |
| СписокМероприятий.  Дата\_До |  | ПеречислениеСсылка.  Время\_Работы | Пустое | Нет |
| СписокМероприятий.  Столы |  | Строка (255) | Пустое | Нет |
| СписокМероприятий.  Официант |  | СправочникСсылка.  Официанты | Пустое | Нет |

Таблица В5 – Описание реквизитов «Бронирование»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя** | **Тип в IDEF1X** | **Тип в 1C** | **Значение по умолчанию** | **Ограничение** |
| Код | Intеger (10) | Число (9) | Автоматическое | Инкрементируемое |
| Клиент | Intеger (10) | СправочникСсылка.  Клиенты | Пустое | Обязательное к заполнению |
| Время\_С | Time | ПеречислениеСсылка.  Время\_Работы | Пустое | Обязательное к заполнению |
| Время\_До | Time | ПеречислениеСсылка.  Время\_Работы | Пустое | Обязательное к заполнению |
|  |  |  |  |  |
|  |  |  |  |  |
| *Окончание таблицы В5* | | | | |
| **Имя** | **Тип в IDEF1X** | **Тип в 1C** | **Значение по умолчанию** | **Ограничение** |
| НомерСтола | Intеger (10) | СправочникСсылка.  Столы | Пустое | Обязательное к заполнению |
| Количество  Гостей |  | Число (5) | Пустое | Обязательное к заполнению, неотрицательное |

Таблица В6 – Описание реквизитов «Мероприятие»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя** | **Тип в IDEF1X** | **Тип в 1C** | **Значение по умолчанию** | **Ограничение** |
| Код | Intеger (10) | Число (9) | Автоматическое | Инкрементируемое |
| Вид | Intеger (10) | ПеречислениеСсылка.  Вид\_Мероприятия | Пустое | Обязательное к заполнению |
| Меню | Intеger (10) |  | Пустое | Обязательное к заполнению |
| Клиент | Intеger (10) | СправочникСсылка.  Клиенты | Пустое | Обязательное к заполнению |
| Стол | Intеger (10) |  | Пустое | Обязательное к заполнению |
| Дата |  | Дата | Пустое | Обязательное к заполнению |
| Время |  |  | Пустое | Обязательное к заполнению |
| Длительность |  |  | Пустое | Обязательное к заполнению |
| КоличествоГостей |  |  | Пустое | Обязательное к заполнению |
| Время\_С |  | ПеречислениеСсылка.  Время\_Работы | Пустое | Обязательное к заполнению |
| Время\_До |  | ПеречислениеСсылка.  Время\_Работы | Пустое | Обязательное к заполнению |
| Итоговая\_  Стоимость |  | Число (6) | Пустое | Обязательное к заполнению, неотрицательное |
| Меню.  Наименование |  | СправочникСсылка.  Общее\_Меню | Пустое | Обязательное к заполнению |
| Меню.Количество |  | Число (3) | Пустое | Обязательное к заполнению, неотрицательное |
| Меню.Цена\_За  \_Блюдо |  | Число (10), точность (2) | Пустое | Обязательное к заполнению, неотрицательное |
| Меню.Сумма |  | Число (10), точность (2) | Пустое | Обязательное к заполнению, неотрицательное |
|  |  |  |  |  |
| *Окончание таблицы В6* | | | | |
| **Имя** | **Тип в IDEF1X** | **Тип в 1C** | **Значение по умолчанию** | **Ограничение** |
| Столы.  НомерСтола |  | СправочникСсылка.  Столы | Пустое | Обязательное к заполнению |
| Столы.  КоличествоГостей |  | Число (5) | Пустое | Обязательное к заполнению, неотрицательное |

Таблица В7 – Описание реквизитов «Чек»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя** | **Тип в IDEF1X** | **Тип в 1C** | **Значение по умолчанию** | **Ограничение** |
| Код | Intеger (10) | Число (9) | Автоматическое | Инкрементируемое |
| Сумма\_Заказа | Intеger (6) |  | Пустое | Обязательное к заполнению |
| Чеки.Номер |  | Строка (5) | Пустое | Обязательное к заполнению |
| Чеки.Официант |  | СправочникСсылка.  Официанты | Пустое | Обязательное к заполнению |
| Чеки.Сумма |  | Число (7), точность (2) | Пустое | Обязательное к заполнению, неотрицательное |

Таблица В8 – Описание реквизитов «Список продуктов» (idef1x) и «Расходная накладная» (1c)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя** | **Тип в IDEF1X** | **Тип в 1C** | **Значение по умолчанию** | **Ограничение** |
| Код | Intеger (10) | Число (9) | Автоматическое | Инкрементируемое |
| Продукты | Intеger (10) |  | Пустое | Обязательное к заполнению |
| Вес\_  Продукта\_г | Intеger (5) |  | Пустое | Обязательное к заполнению |
| Цена\_За\_  Продукт | Intеger (5) |  | Пустое | Обязательное к заполнению |
| Мероприятие |  | ДокументСсылка.  Мероприятие | Пустое | Обязательное к заполнению |
| Склад |  | СправочникСсылка.  Склад | Есть | Обязательное к заполнению |
| Продукты.  Наименование\_  Продукта |  | СправочникСсылка.  Продукты | Пустое | Обязательное к заполнению |
| Продукты.  Вес\_кг |  | Число (10), точность (3) | Пустое | Обязательное к заполнению, неотрицательное |