**Министерство сельского хозяйства Российской Федерации**

**ФГБОУ ВО «КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ имени И.Т. Трубилина»**

**Факультет прикладной информатики**

**Кафедра информационных систем**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту

По дисциплине: Методы и средства проектирования информационных систем

На тему: Разработка информационной системы для автоматизации подготовки, хранения и выдачи на печать заказа покупателя на приобретение материальных ценностей.

Выполнил студент группы: ИТ1721 Рылов Михаил Леонидович

Допущен к защите

Руководитель проекта Тюнин Е.Б.

Нормоконтроллер

Защищен

Оценка

Члены комиссии Попова Елена Витальевна

Тюнин Евгений Борисович

Кондратьев Валерий Иванович

(подпись, расшифровка подписи)

Краснодар

2020

**Министерство сельского хозяйства Российской Федерации**

Федеральное государственное образовательное учреждение

высшего образования

**КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ имени И.Т. Трубилина**

**Кафедра информационных систем**

УТВЕРЖДАЮ

Заведующий кафедрой

Информационных систем

Попова Е.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

Студенту Рылов Михаил Леонидович группы ИТ1721 3 курса

По дисциплине Методы и средства проектирования информационных систем

Тема: Разработка информационной системы для автоматизации подготовки, хранения и выдачи на печать заказа покупателя на приобретение материальных ценностей

Утверждена на заседании кафедры от «\_\_»\_\_\_\_\_\_\_\_\_20\_\_г. Протокол №\_\_\_\_

Срок выполнения работы с «\_\_\_»\_\_\_\_\_\_\_\_2020г. по «\_\_\_»\_\_\_\_\_\_\_\_2020г.

Дата выдачи задания «\_\_\_»\_\_\_\_\_\_\_\_\_2020г.

Срок сдачи первого этапа «\_\_\_»\_\_\_\_\_\_\_\_\_2020г.

Срок сдачи второго этапа «\_\_\_»\_\_\_\_\_\_\_\_\_2020г.

Дата сдачи проекта на кафедру «\_\_\_»\_\_\_\_\_\_\_\_\_2020г.

Срок защиты «\_\_\_»\_\_\_\_\_\_\_\_\_2020г.

Задание к исполнению принял студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО, личная подпись, дата)

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, ФИО)

Краснодар

2020

**КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ имени И.Т. Трубилина**

**Факультет прикладной информатики**

**Кафедра информационных систем**

**КУРСОВАЯ РАБОТА**

**по дисциплине «Методы и средства проектирования информационных систем»**

На тему: Разработка информационной системы для автоматизации подготовки, хранения и выдачи на печать заказа покупателя на приобретение материальных ценностей

Выполнил: Рылов Михаил Леонидович

Группа: ИТ1721

Руководитель: Тюнин Е.Б.

Краснодар

2020

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc30986201)

[1. АНАЛИЗ ОБЪЕКТА ИССЛЕДОВАНИЯ 4](#_Toc30986202)

[1.1. Анализ предметной области 4](#_Toc30986203)

[1.2. Анализ аналогичных систем автоматизации 6](#_Toc30986204)

[1.3. Предъявляемые требования к разрабатываемой ИС. 7](#_Toc30986205)

[2.1. Обоснование выбора средства реализации 9](#_Toc30986206)

[2.2. Моделирование предметной области ИС 10](#_Toc30986207)

[2.4. Создание структуры базы данных 13](#_Toc30986208)

[2.5 Разработка форм ввода и вывода информации 17](#_Toc30986209)

[2.6. Разработка отчетов 29](#_Toc30986210)

[3. МЕТОДЫ РАБОТЫ С ИНФОРМАЦИОННОЙ СИСТЕМОЙ 33](#_Toc30986211)

[3.1. Требования к оборудованию и программному обеспечению 33](#_Toc30986212)

[3.2. Описание метода работы с информационной системой 35](#_Toc30986213)

[ЗАКЛЮЧЕНИЕ 36](#_Toc30986214)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 37](#_Toc30986215)

РЕФЕРАТ

Количество страниц пояснительной записки – 37;

Количество рисунков – 21;

Использованных источников – 7;

Ключевые слова: ORACLE, ИНФОРМАЦИОННАЯ СИСТЕМА, ЗАКАЗ, ОРДЕР, ФОРМА, ОТЧЕТ, ЗАПРОС, SQL .

Объект исследования – информационная система для автоматизации подготовки, хранения и выдачи на печать заказа покупателя на приобретение материальных ценностей.

Предмет исследования – разработка информационной системы для автоматизации подготовки, хранения и выдачи на печать заказа покупателя на приобретение материальных ценностей.

ВВЕДЕНИЕ

В условиях развития вычислительной техники и всеобщей информатизации общества особо остро встают проблемы автоматизации производственной деятельности путем внедрения современных инновационных способов взаимодействия с информационными системами. Их решение, в первую очередь, направлено на повышение производительности труда, снижение временных издержек, уменьшение объема монотонной, однообразной работы.

Важным этапом информатизации общества является постепенный переход от классической бумажной документации к электронным документам. Использование автоматизированных систем управления документацией имеет огромное значение в современном мире.

Цель данной курсовой работы состоит в проектировании и разработке информационной системы для автоматизации подготовки, хранения и выдачи на печать заказа покупателя на приобретение материальных ценностей.

Для достижения данной цели необходимо решить следующие задачи:

* проанализировать структуру объекта автоматизации;
* проанализировать существующие системы автоматизации;
* разработать концептуальную логическую модель «сущность –связь» в среде ERwin;
* разработать информационную систему;

Объектом курсовой работы является заказ покупателя на приобретение материальных ценностей.

Субъектом является создаваемая информационная система.

1. АНАЛИЗ ОБЪЕКТА ИССЛЕДОВАНИЯ

1.1. Анализ предметной области

Форма ТОРГ-26 Заказ является регламентированным с помощью Госкомстата бланком (соответствующее Постановление от 25.12.98 №132) и служит средством обеспечения оформления покупателем предварительного заказа на конкретный выбранный им товар, предлагаемый продавцом.  Бланк подразумевает обозначение необходимости доставки (или отсутствия таковой услуги) в заранее определенное время.

С помощью данного документа происходит процесс подтверждения факта покупки в розничной сети посредством реализации предварительного заказа. Форма ТОРГ-26 используется исключительно торговыми предприятиями, реализующими товар по предварительному заказу.

Процедура применения формы заказа предусматривает следующий алгоритм:

1. Покупатель заказывает товар у поставщика, направляя свою заявку в устной или письменной форме;
2. Сотрудник организации поставщика на основании полученной заявки заполняет заказ ТОРГ-26;
3. Бланк ТОРГ-26 передается на склад сборщику товара;
4. Проводится сборка указанных наименований;
5. Лицо, ответственное за стоимость, проверяет верность ее указания в документе;
6. Проводится ответственным сотрудником склада контрольная проверка правильности собранной заявки;
7. Товар направляется покупателю вместе с двумя экземплярами заказа ТОРГ-26;
8. Покупатель по факту получения ценностей сверяет содержание документа и реально доставленные товары, при согласии заверяет своей подписью бланк и один экземпляр возвращает поставщику.

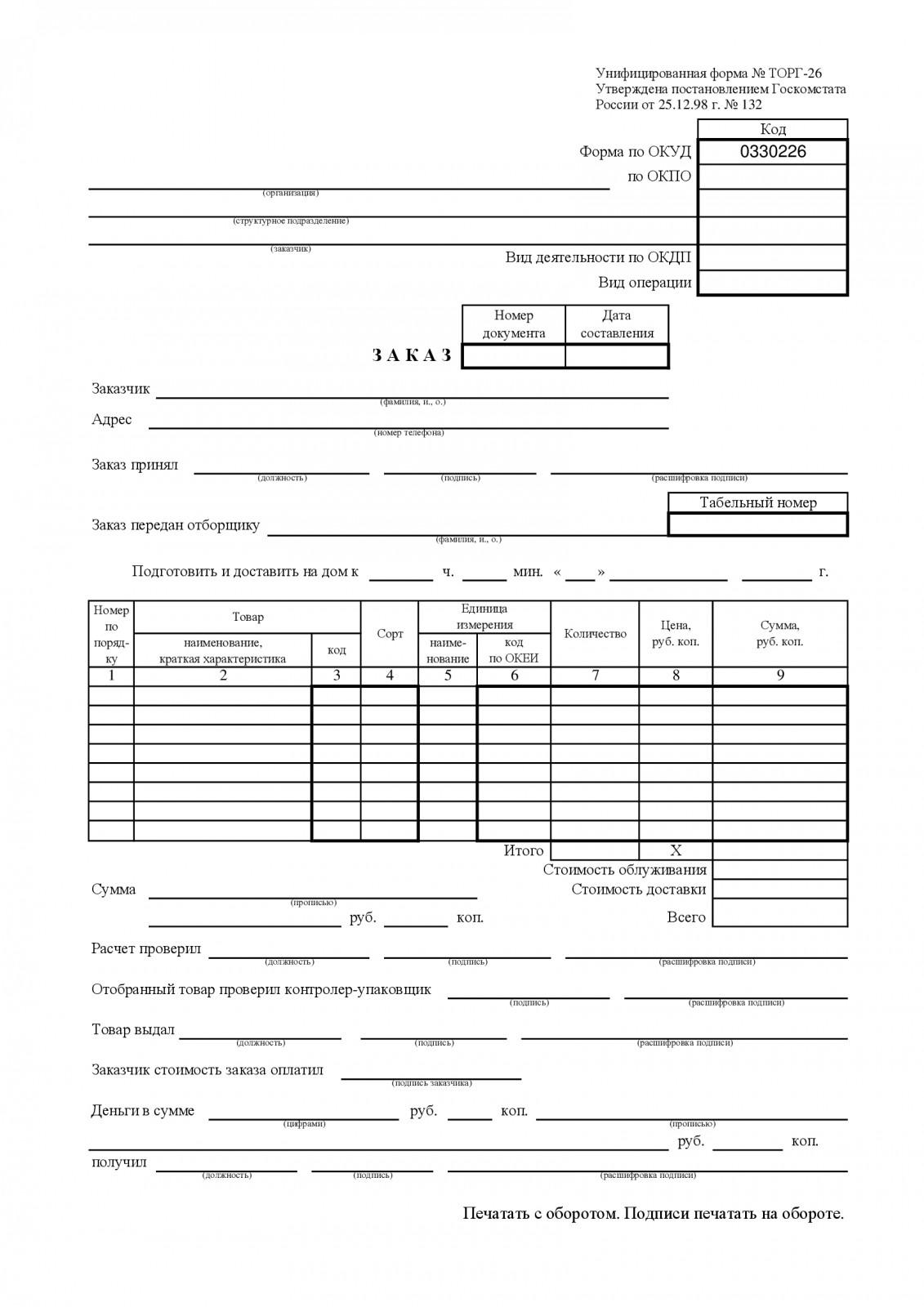


Рисунок 1 – Форма торг-26

В итоге один экземпляр ТОРГ-26 с подписями ответственных лиц, в том числе и со стороны покупателя, получает поставщик, документ хранится у сотрудника организации поставщика, отобравшего товар по заказу, второй – передается покупателю. Форма торг-26 изображена на рисунке 1.

1.2. Анализ аналогичных систем автоматизации

В результате поиска в Интернете было найдено множество сайтов, которые предоставляют услуги формирования документов на заказ материальных ценностей онлайн с последующей возможностью отправки готового заполненного документа на печать или сохранения в электронном виде.

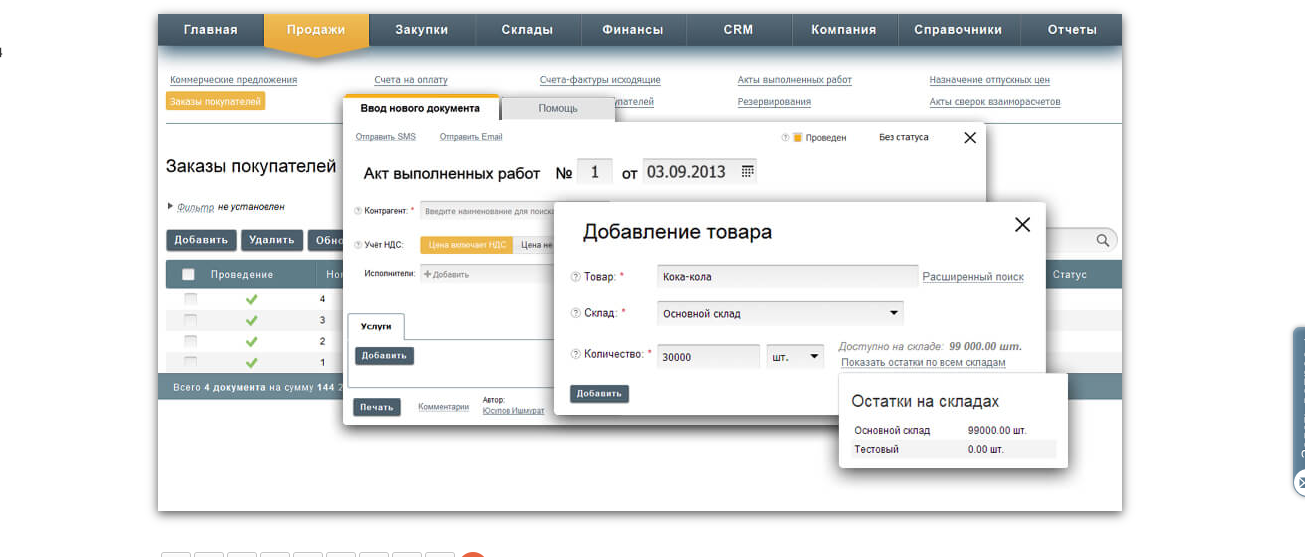


Рисунок 2 – страница создания документа на сайте class365.ru

На рисунке 2 показана страница создания документа на заказ материальных ценностей покупателем.

Данные сервисы предоставляют богатый функционал для автоматизации заказа материальных ценностей. Главным их преимуществом является возможность получить к ним доступ с любого устройства, имеющего выход в интернет, так как отсутствует необходимость скачивания и установки программного обеспечения на компьютер. Однако, это может быть и недостатком, так как если система безопасности окажется ненадежной, злоумышленники могут получить доступ к внутренней документации предприятия. Данными онлайн системами очень легко пользоваться, у них простой и понятный интерфейс, однако их функционал ограничен. Также отсутствует возможность реализации дополнительного специфичного функционала, необходимого для эффективной работы складов на различных предприятиях. Также к недостаткам стоит отнести необходимость использования баз данных, предоставляемых сайтом и невозможность интегрировать вебсайт с другими информационными системами.

1.3. Предъявляемые требования к разрабатываемой ИС.

Проанализировав уже существующие информационные системы, можно сформировать следующие требования к разрабатываемой информационной системе:

* использование веб технологий;
* высокий уровень надежности и отказоустойчивости;
* безопасность;
* легкая расширяемость;
* удобный интерфейс взаимодействия.

Система должна поддерживать такую схему взаимодействия между модулями и автоматизированными рабочими местами, которая отвечала бы требованиям и техническим возможностям пользователя. Важнейшими параметрами информационной системы являются надёжность, масштабируемость, безопасность, поэтому при создании таких систем используется архитектура клиент-сервер. Эта архитектура позволяет распределить работу между клиентской и серверной частями системы, предусматривает развитие и совершенствование в соответствии с особенностями решаемых задач. В последние годы наблюдается устойчивая тенденция увеличения спроса на клиент - серверные приложения, которые обладают в области учёта и управления большими возможностями, чем файл-серверные системы при обработке больших объёмов данных, возможностью создания распределённых систем, а также достаточной интеграцией с другими системами.

Несмотря на переход к электронному документообороту, все еще остается необходимость печати документов. Для этого в разрабатываемой информационной системе должен быть реализован функционал печати отчетов.

2. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1. Обоснование выбора средства реализации

Реализация информационной системы будет произведена в инструментальной среде СУБД Oracle.Database 10g.

Данная СУБД представляет собой огромный программный комплекс, позволяющий создавать приложения любых уровней.

Ядро СУБД Oracle представляет из себя базу данных, в которой производится хранение информации, объем которой не может выходить за рамки предоставленной физической памяти, но также может в теории быть расширено до почти безграничных пределов.

Количество пользователей, которые могут одновременно работать с данной СУБД, строго упирается в аппаратные возможности той системы, на которой данный комплекс развернут. Стоит отметить, что производительность системы от количества пользователей не зависит.

Механизмы масштабирования в СУБД Oracle последней версии позволяют безгранично увеличивать мощность и скорость работы сервера Oracle и своих приложений, просто добавляя новые и новые узлы кластера. Выход из строя отдельных узлов кластера не приводит к остановке приложения.

В следствии этого проявляется высокая надежность всего программного-аппаратного комплекса, что положительно сказывается на безопасности как всей системы, так и отдельных ее компонентов.

СУБД Oracle в одинаковой степени оптимизирована и для приложений оперативной обработки транзакций, и для аналитических приложений, причем их можно выполнять одновременно на одном и том же компьютере, не задумываясь о дополнительных блокировках, режимах изоляции и других технических деталях. На практике это означает, что один и тот же продукт (например, Oracle Database Enterprise Edition) можно с успехом использовать и как сервер оперативных баз данных, обрабатывающий интенсивный поток относительно простых и коротких транзакций, поступающих от множества пользователей, так и в качестве сервера хранилища данных, который позволяет концентрировать большие объемы данных и выполнять над ними сложные аналитические вычисления.

Все это в совокупности позволяется СУБД Oracle не только оставаться востребованной на рынке, но и производить обновления всего программно-аппаратного комплекса, расширяя старые и добавляя новые возможности в арсенал системы.

2.2. Моделирование предметной области ИС

Данный этап позволить определить организацию разрабатываемой информационной системы, а также проследить за потоками данных в ней.

Построении модели позволяет определить свойства системы, характер ее поведения и пределы исполнения.

Моделирование как процесс в свою очередь позволит избежать значительных как финансовых затрат, так и лишних человеко-часов работы при проведении эксперимента в реальных условиях, что также убережет от убытков и полной ликвидации в случае провала.

Определим список диаграмм и программных средств, необходимые для анализа разрабатываемой ИС:

* Модель «сущность-связь» или ER-диаграмма;
* Даталогическая модель.

Таблица 1 – Датологическая модель базы данных

|  |  |
| --- | --- |
| Orders | |
| Id\_or | Number |
| OKYD | Number |
| OKPO | Number |
| Datep | Date |
| Mind | Number |
| Hourd | Number |
| Dayd | Number |
| Monthd | Varchar2 |
| Yeard | Number |
| Sum\_b | Varchar2 |
| Sum\_n | Number |
| Id\_cust | Number |
| Id\_str | Number |
| Recipient | Number |
| Payer | Number |
| Issuer | Number |
| Product\_list | |
| Id\_list | Number |
| Id\_pr | Number |
| Count | Number |
| Sum | Number |
| Struct\_sub | |
| Id\_str | Number |
| Name\_str | Varchar2 |
| Org | |
| Id\_org | Number |
| Name\_org | Varchar2 |
| Product | |
| Id\_pr | Number |
| Namep | Varchar2 |
| Sortp | Number |
| Unit | Varchar2 |
| OKEI | Number |
| Price | Number |
| Employee | |
| Id\_em | Number |
| FIO | Varchar2 |
| Customer | |
| Id\_cust | Number |
| Address | Varchar2 |
| Tel\_num | Varchar2 |
| FIO | Varchar2 |

2.3. Построение инфологической модели базы данных

ERwin – это инструмент проектирования баз данных. Его огромным преимуществом является графический пользовательский интерфейс, базируемый на операционной системе Windows со встроенным мощным инструментом для работы с ER-диаграммами (сущность-связь). Также ERwin включает в себя редакторы для создания логического и физического описания модели данных и поддержку ведущих реляционных СУБД и настольных баз данных.

Инфологическая модель базы данных, показанная на рисунке 3, была разработана с помощью программы ERwin. Выявлены основные атрибуты, сущности, и связи между ними.

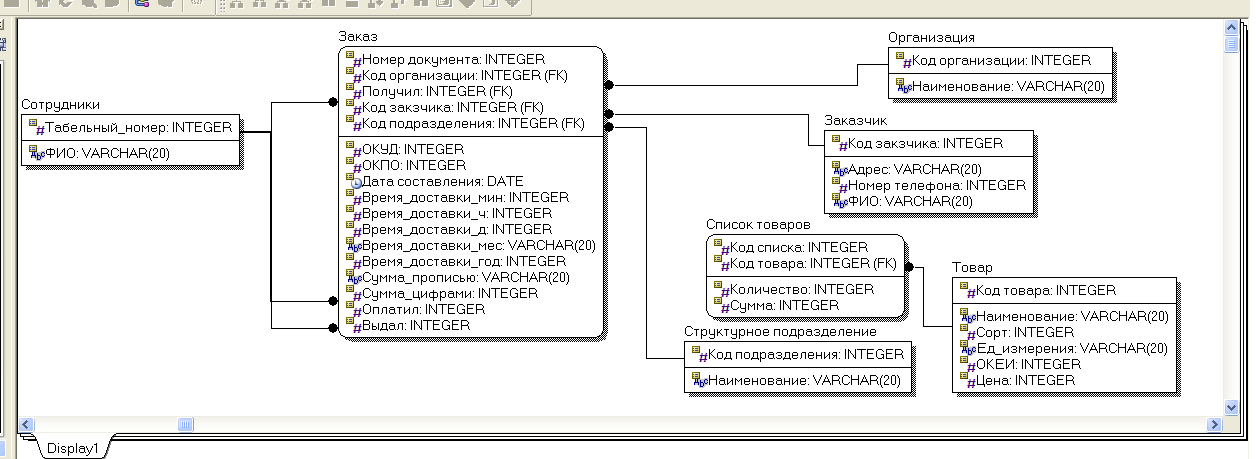


Рисунок 3 – инфологическая модель БД

Инфологическая модель базы данных, изображенная на рисунке 3 представляет собой описание объектов (сущностей), с набором атрибутов и связей между ними, которые выявляются в процессе исследования как входных, так и выходных данных. Она предназначается для структурного образования предметной области, с ориентированием на информационное внимание пользователей, разрабатываемой системы.

2.4. Создание структуры базы данных

Создание справочника «Организация»:

CREATE TABLE ORG

(ID\_ORG NUMBER NOT NULL,

NAME\_ORG VARCHAR2(70))

/

ALTER TABLE ORG

ADD CONSTRAINT PK\_ID\_ORG PRIMARY KEY (ID\_ORG)

USING INDEX

/

Создание справочника «Товары»:

CREATE TABLE Product

(ID\_PR NUMBER NOT NULL,

NAMEP VARCHAR2(70),

SORTP NUMBER,

UNIT VACHAR2(70),

OKEI NUMBER,

PRICE NUMBER))

/

ALTER TABLE

ADD CONSTRAINT PK\_ID\_PR PRIMARY KEY (ID\_PR)

USING INDEX

Создание справочника «Заказчик»:

CREATE TABLE CUSTOMER

(ID\_CUST NUMBER NOT NULL,

ADDRESS VARCHAR2(70),

TEL\_NUM NUMBER,

FIO VARCHAR2(70))

/

ALTER TABLE CUSTOMER

ADD CONSTRAINT PK\_ID\_CUST PRIMARY KEY (ID\_CUST)

USING INDEX

/

Создание справочника «Сотрудники»:

CREATE TABLE EMPLOYEE

(ID\_EM NUMBER NOT NULL,

FIO VARCHAR2(70))

/

ALTER TABLE EMPLOYEE

ADD CONSTRAINT PK\_ID\_EM PRIMARY KEY (ID\_EM)

USING INDEX

/

Создание справочника «Структурное подразделение»:

CREATE TABLE STRUCT\_SUB

(ID\_STR NUMBER NOT NULL,

NAME\_STR VARCHAR2(70))

/

ALTER TABLE STRUCT\_SUB

ADD CONSTRAINT PK\_ID\_STR PRIMARY KEY (ID\_STR)

USING INDEX

/

Создание документа «Список товаров»:

CREATE TABLE PRODUCT\_LIST

(ID\_LIST NUMBER NOT NULL,

ID\_PR NUMBER,

COUNT NUMBER,

SUM NUMBER))

/

ALTER TABLE PRODUCT\_LIST

ADD CONSTRAINT PK\_ID\_LIST PRIMARY KEY (ID\_LIST)

USING INDEX

/

Создание документа «Заказ»:

CREATE TABLE ORDERS

(ID\_OR NUMBER NOT NULL,

OKYD NUMBER,

OKPO NUMBER,

DATEP DATE,

MIND NUMBER,

HOURD NUMBER,

DAYD NUMBER,

MONTHD NUMBER,

YEARD NUMBER,

SUM\_B VARCHAR2(70),

SUM\_N NUMBER,

ID\_CUST NUMBER,

ID\_STR NUMBER,

ID\_ORG NUMBER,

RECIPIENT NUMBER,

PAYER NUMBER,

ISSUER NUMBER)

/

ALTER TABLE ORDERS

ADD CONSTRAINT PK\_ID\_OR PRIMARY KEY (ID\_OR)

USING INDEX

/

Создание реляционной связи между таблицами «Список товаров» и «Товары»:

ALTER TABLE PRODUCT\_LIST

ADD CONSTRAINT FK\_ID\_PR FOREIGN KEY (ID\_PR)

REFERENCES PRODUCT (ID\_PR);

/

Создание реляционной связи между таблицами «Заказ» и «Сотрудники»:

ALTER TABLE ORDERS

ADD CONSTRAINT FK\_RECIPIENT FOREIGN KEY (RECIPIENT)

REFERENCES EMPLOYEE (ID\_EM);

/

ALTER TABLE ORDERS

ADD CONSTRAINT FK\_RECIPIENT FOREIGN KEY (PAYER)

REFERENCES EMPLOYEE (ID\_EM);

/

ALTER TABLE ORDERS

ADD CONSTRAINT FK\_RECIPIENT FOREIGN KEY (ISSUER)

REFERENCES EMPLOYEE (ID\_EM);

/

Создание реляционной связи между таблицами «Заказ» и «Заказчик»:

ALTER TABLE ORDERS

ADD CONSTRAINT FK\_ID\_CUST FOREIGN KEY (ID\_CUST)

REFERENCES CUSTOMER (ID\_CUST);

/

Создание реляционной связи между таблицами «Заказ» и «Организация»:

ALTER TABLE ORDERS

ADD CONSTRAINT FK\_ID\_ORG FOREIGN KEY (ID\_ORG)

REFERENCES ORG (ID\_ORG);

/

Создание реляционной связи между таблицами «Заказ» и «Структурное подразделение»:

ALTER TABLE ORDERS

ADD CONSTRAINT FK\_ID\_STR FOREIGN KEY (ID\_STR)

REFERENCES STRUCT\_SUB (ID\_STR);

/

2.5 Разработка форм ввода и вывода информации

Создание форм было произведено с помощью программы Oracle Forms Developer. Для этого были проделаны следующие шаги:

* создание новой формы ADFORM;
* ее сохранение с помощью File > Save с именем ADFORM.fmb;

Чтобы подключить таблицу из базы данных к созданной нами форме необходимо в окне Object Navigator навести курсор на узел DataBlocks, создать блок данных и с помощью DataBlock Wizard задать параметры таблицы. В открывшемся мастере необходимо нажать кнопку «Далее» и выбрать тип блока данных Table or View. На следующем шаге мастера, нужно указать по какой таблице будет строиться блок данных. Выбор таблицы осуществляется нажатием кнопки Browse (обзор), после чего следует ввести данные соединения с БД. В окне Tables осуществляется выбор таблиц. Чтобы таблицы текущей схемы были видны, следует выбрать пункты Current User, Tables и Views, где нужно выбрать таблицу ORFERS. Чтобы отображать в блоке данных все колонки выбранной таблицы необходимо названия всех колонок переместить из поля Available Columns в поле Database Items, что позволит взаимодействовать с полями таблиц из самой формы.

На следующем шаге в поле DataBlock Name вводится имя блока данных: ORDERS. После чего завершаем работу мастера нажатием на кнопку «Готово».

Для планирования расположения полей на форме только что созданного блока данных сразу после мастера DataBlock Wizard запускается мастер Layout Wizard. На двух первых его шагах следует просто нажать кнопку «Далее». На третьем шаге, будет предложено выбрать, какие поля следует отображать на форме. В рассматриваемой задаче будут отображаться все поля, поэтому их все и нужно переместить в поле Displayed Items.

Следующий шаг, где предлагается установить размеры полей, остается без изменения.

На следующем шаге выбирается тип отображения формы как Form и завершается работа мастера нажатием кнопки «Готово». Далее появляется форма с полями таблицы ORDERS – документ «Заказ».

Таким же методом создаются и другие справочники и документы.

Таким образом, каждая таблица имеет свою собственную отдельную форму.

Для всех созданных форм был создан триггер открытия формы – WHEN-NEW-FORM-INSTANCE – со следующим кодом:

EXECUTE\_QUERY;

В каждую форму следует добавить предупреждение SAVE\_ALERT на случай, если пользователь попытается закрыть форму с еще не сохраненными изменениями в ней, и задать следующие параметры:

* Name: SAVE\_ALERT
* Title: Сообщение
* Message: Вы хотите сохранить сделанные Вами изменения?
* Button 1 Label: Да
* Button 2 Label: Нет
* Button 3 Label: Отмена
* Default Alert Button: Button 3

До этого мы создавали триггер открытия формы, сейчас создадим триггер закрытия формы – WHEN-WINDOW-CLOSED:

DECLARE

BUT NUMBER;

STAT VARCHAR2(20);

BEGIN

GO\_BLOCK(AD);

STAT := :SYSTEM.BLOCK\_STATUS;

IF STAT <> `QUERYQUERY` THEN

BUT := SHOW\_ALERT(`SAVE\_ALERT`);

IF BUT = ALERT\_BUTTON3 THEN

RETURN;

END IF;

IF BUT = ALERT\_BUTTON1 THEN

SAVE\_FORM;

END IF;

END IF;

EXIT\_FORM(NO\_VALIDATE);

END;

Код триггера означает, что при закрытии формы AD в случае несохраненных данных будет отображаться SAVE\_ALERT, а в случае соблюдения всех условий форма будет успешно закрыта.

Аналогичным образом эти триггеры задаются и в остальных формах.

Так же необходимо создать кнопки «ОК» для сохранения изменения и закрытия, «ОТМЕНА» для закрытия формы.

Код кнопки «ОК»;

SAVE\_FORM;

EXIT\_FORM(NO\_VALIDATE);

Код кнопки «ОТМЕНА»:

EXIT\_FORM(NO\_VALIDATE);

Процедура сохранения формы:

Save\_Form;

PROCEDURE SAVE\_FORM IS

BEGIN

Post;

Commit\_Form;

END;

Код кнопки «ПЕЧАТЬ»:

GO\_PRINT('10.0.2.2','8889','rep\_student','REPORT\_ORD','NUM\_OR='||

To\_Char(:ORDERS.ID\_OR));

Код процедуры «GO\_PRINT»:

PROCEDURE GO\_PRINT (srv\_host VARCHAR2, srv\_port VARCHAR2, srv\_name VARCHAR2, rep VARCHAR2, params VARCHAR2)

IS

repid REPORT\_OBJECT;

v\_rep VARCHAR2(100);

rep\_status VARCHAR2(20);

other VARCHAR2(1024);

printer\_name VARCHAR2(80);

BEGIN

repid := find\_report\_object(rep);

-- Устанавливаем свойства отчета

SET\_REPORT\_OBJECT\_PROPERTY(repid, REPORT\_EXECUTION\_MODE,BATCH);

SET\_REPORT\_OBJECT\_PROPERTY(repid, REPORT\_COMM\_MODE,SYNCHRONOUS);

SET\_REPORT\_OBJECT\_PROPERTY(repid, REPORT\_DESTYPE,CACHE);

SET\_REPORT\_OBJECT\_PROPERTY(repid, REPORT\_DESFORMAT,'pdf');

SET\_REPORT\_OBJECT\_PROPERTY(repid, REPORT\_SERVER,srv\_name);

-- Передаем отчету дополнительную строку параметров

SET\_REPORT\_OBJECT\_PROPERTY(repid, REPORT\_OTHER, params);

v\_rep := RUN\_REPORT\_OBJECT(repid);

rep\_status := REPORT\_OBJECT\_STATUS(v\_rep);

-- Ожидаем выполнение и формирование отчета

WHILE rep\_status in ('RUNNING','OPENING\_REPORT','ENQUEUED') LOOP

rep\_status := Report\_Object\_Status(v\_rep);

END LOOP;

-- Если отчет сформировался, показываем его в браузере

IF rep\_status = 'FINISHED' THEN

WEB.SHOW\_DOCUMENT('http://'|| srv\_host ||':'|| srv\_port||

'/reports/rwservlet/getjobid'|| substr(v\_rep,instr(v\_rep,'\_',-1)+1)||

'?'||'server=' || srv\_name,'\_blank');

ELSE

message('Ошибка вывода на печать!!!');

END IF;

END;

В формах для выбора и заполнения уже существующими записями были созданы списки List of Values (LOV).

Добавление списка значений производится в окне Object Navigator в узле LOVs.

Получившиеся формы представлены на рисунках 4-9:

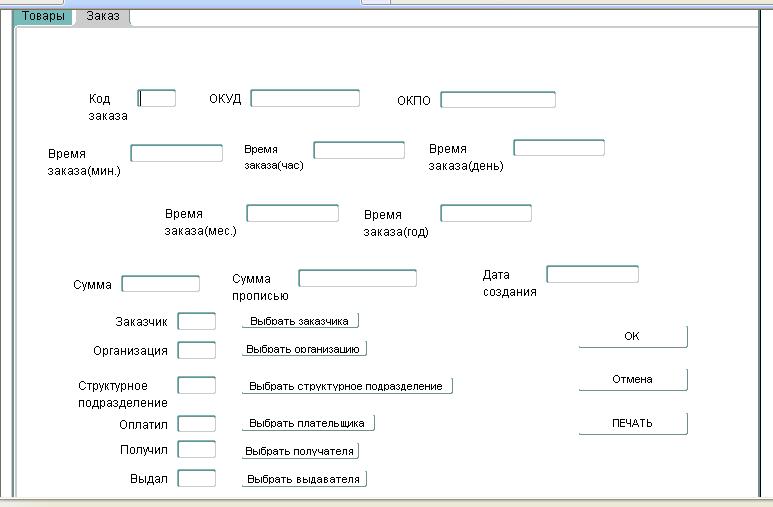


Рисунок 4 – Форма «Заказ»

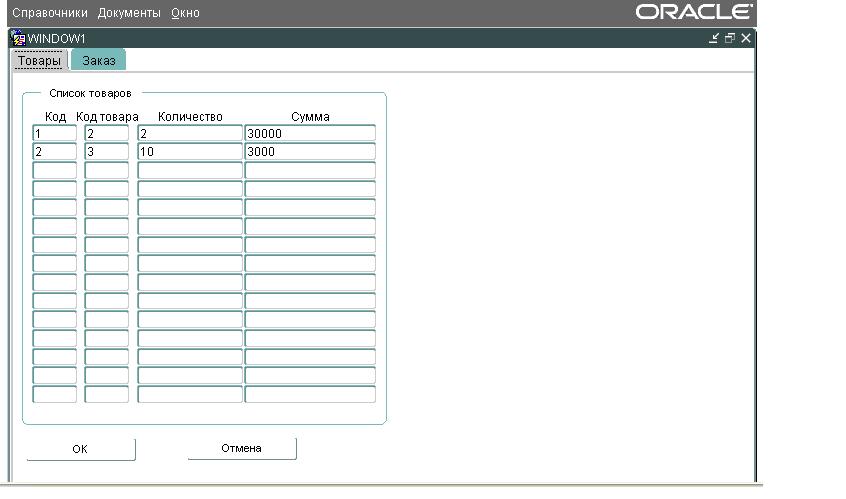


Рисунок 5 – Табличная форма «Список товаров»

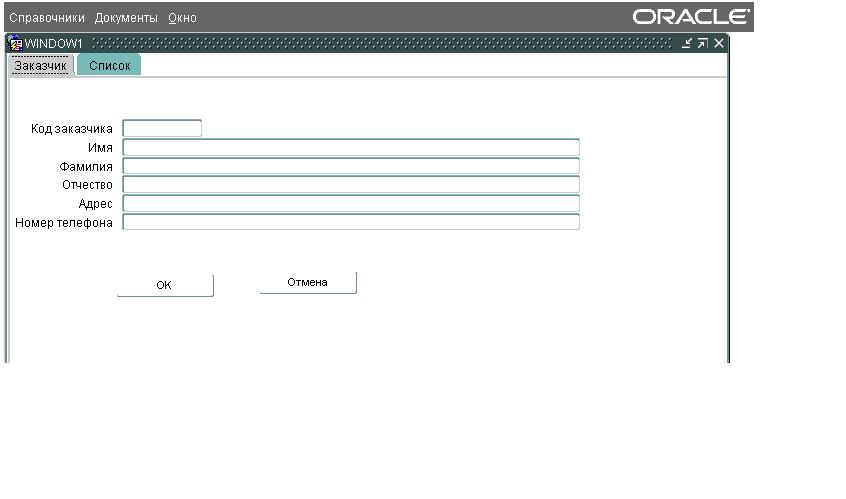


Рисунок 6 – Форма «Заказчик»

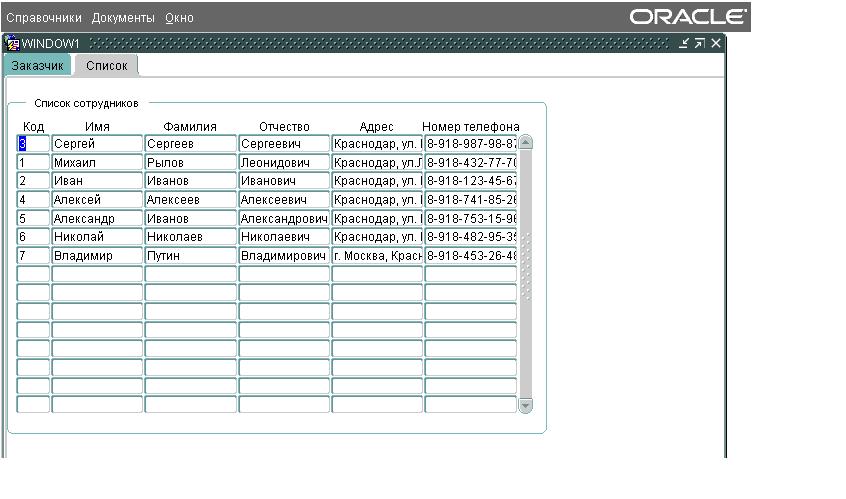


Рисунок 7 – Табличная форма «Список заказчиков»

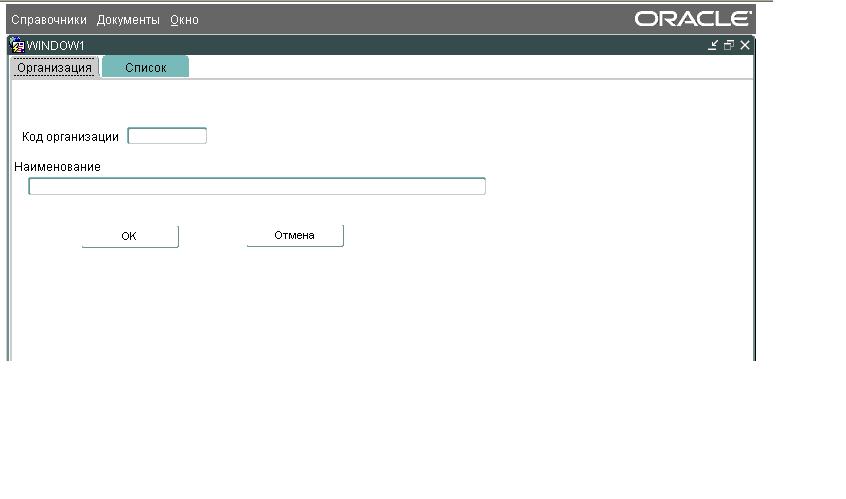


Рисунок 8 – Форма «Организация»



Рисунок 9 – Табличная форма «Список организаций»

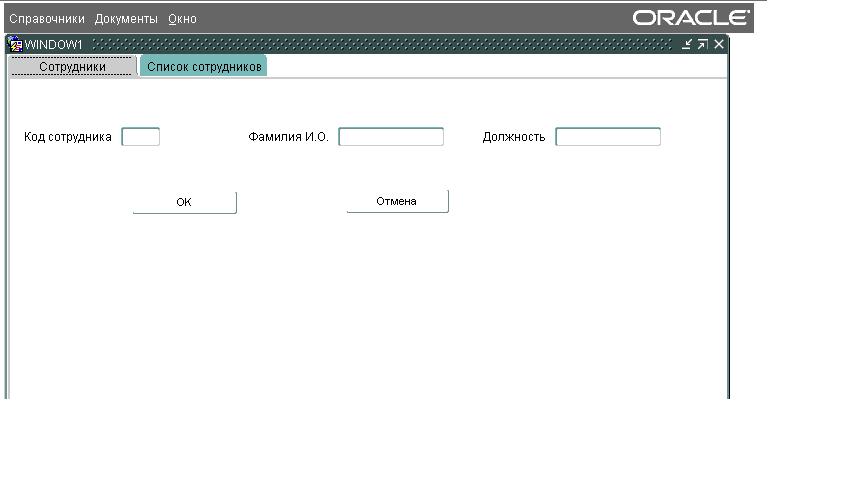


Рисунок 10 – Форма «Сотрудники»

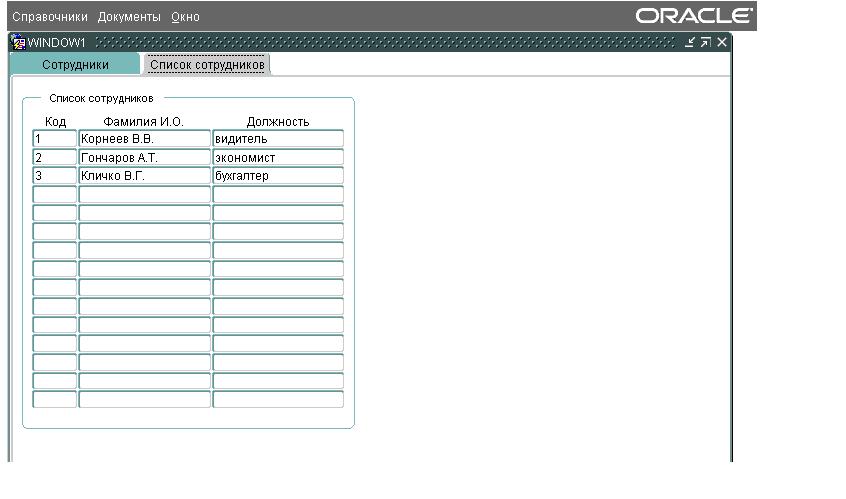


Рисунок 11 – Табличная форма «Список сотрудников»

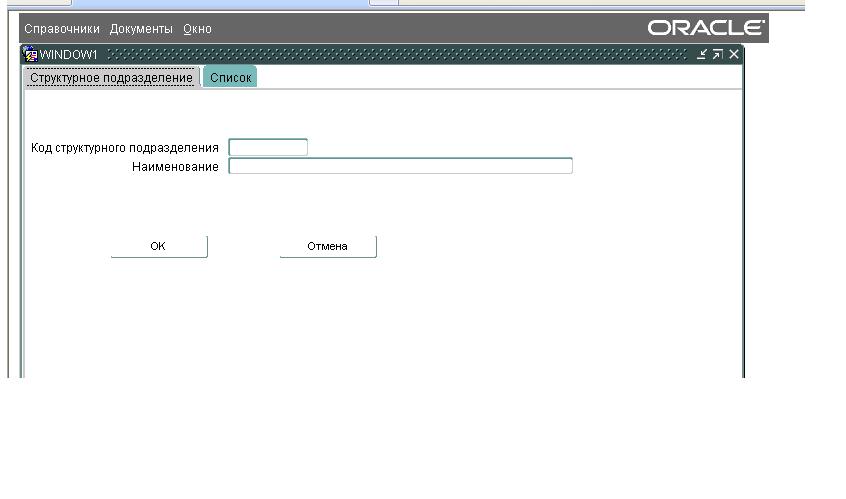


Рисунок 12 – Форма «Структурное подразделение»

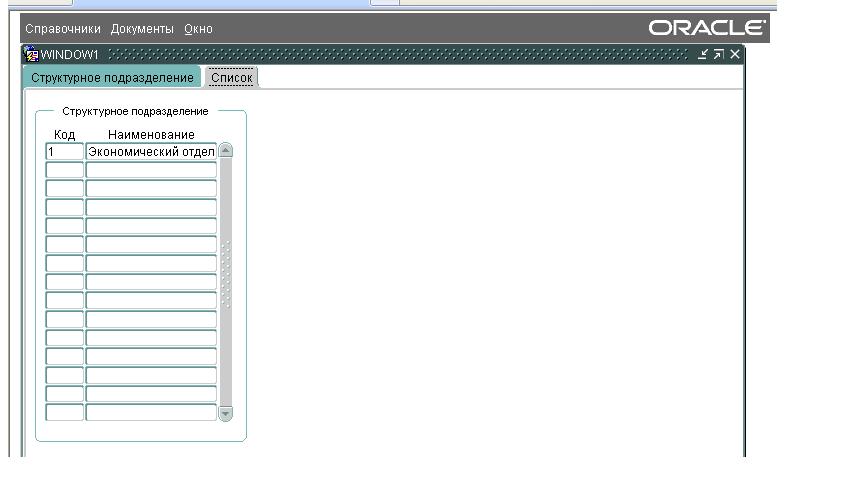


Рисунок 13 – Табличная форма «Список структурных подразделений»



Рисунок 14 – Форма «Товар»

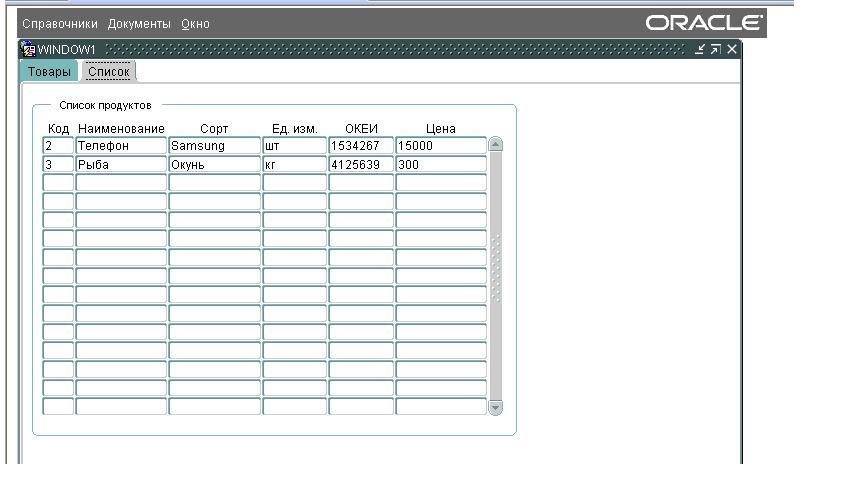


Рисунок 15 – Табличная форма «Список товаров»

Создание меню FORM\_MENU для упорядочивания и более удобного доступа к формам.

В окне Object Navigator нужно навести курсор на узел Menu и создать новое меню с именем FORM\_MENU, а для доступа к формам необходимо прописать следующий код:

OPEN\_FORM(`ORDERS`);

В результате выполнения этого кода при нажатии в меню на пункт «Заказ» откроется соответствующая форма.

Такие коды необходимо прописать для каждой формы.

Общий вид FORM\_MENU:



Рисунок 16 – Главное меню

2.6. Разработка отчетов

Oracle Report Builder – средство разработки отчетов для информационных систем на базе СУБД Oracle.

На рисунке 17 показана схема запросов для отчета. Каждая из изображенных схем является результатом SQL запроса в базу данных.

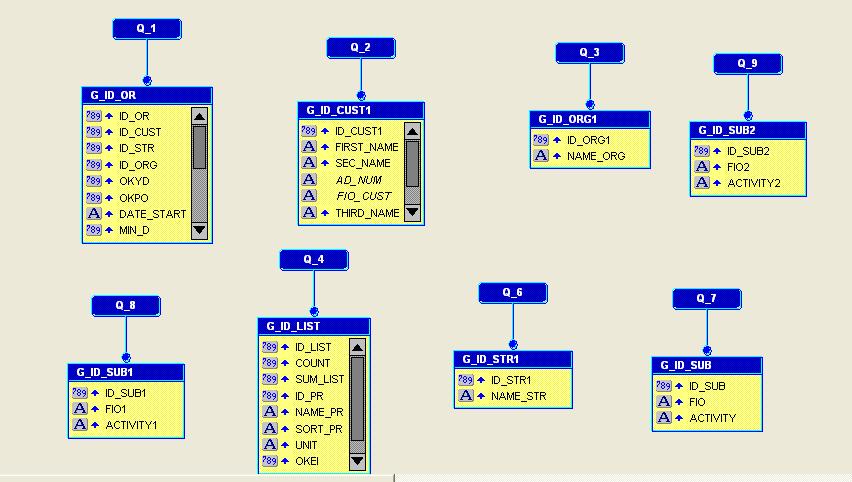


Рисунок 17 – схема запросов данных из БД

На основе запросов, схема которых расположена на рисунке 17, составляется шаблон отчета, который по требованию печати заполняется необходимыми данными и отображается пользователю в удобном виде. Пример получившейся накладной показан на рисунках 18-20. Вызвать его можно нажав на кнопку «ПЕЧАТЬ» на странице «Заказ» в разделе управления накладными. Данная накладная была составлена на основе документа «ТОРГ-26», пример которой был показан в первой главе. Новая требование-накладная соответствует всем необходимым критериям, которые были представлены раньше, чтобы иметь юридическую силу.

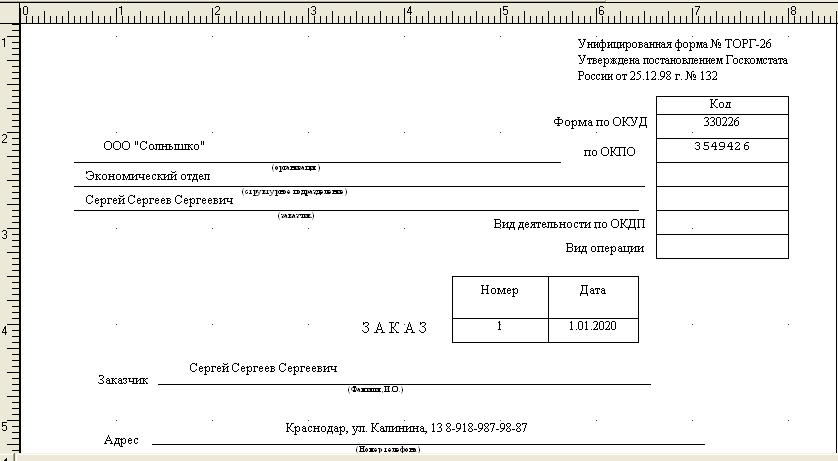


Рисунок 18 – Заказ часть 1

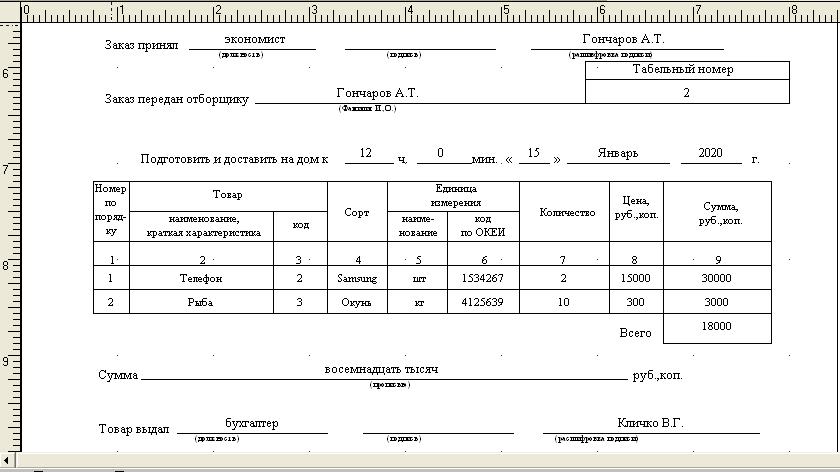


Рисунок 19 – Заказ часть 2

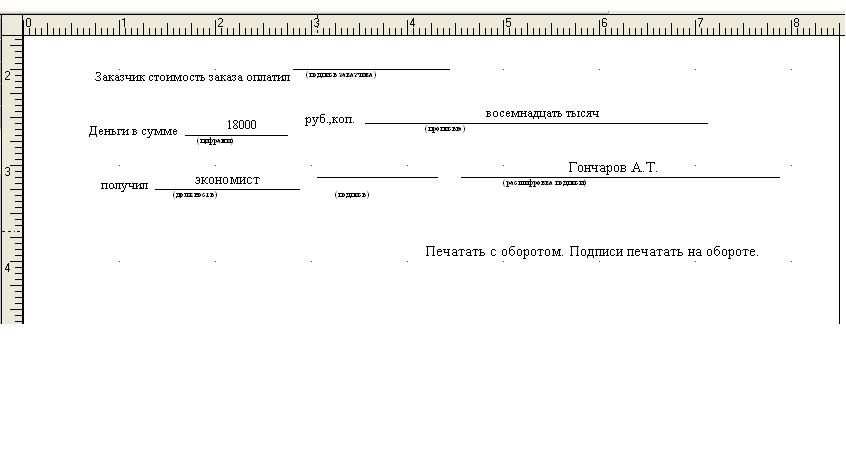


Рисунок 20 – Заказ часть 3

Отчеты позволяют избежать ручного заполнения документов, что существенно ускорит и упростит данный процесс, а также поможет избежать ошибок, связанных с человеческим фактором.

3. МЕТОДЫ РАБОТЫ С ИНФОРМАЦИОННОЙ СИСТЕМОЙ

3.1. Требования к оборудованию и программному обеспечению

Для использования разработанной системы автоматизации необходимо установить Oracle Database 10g.

Для установки ПО Oracle требуется войти под учетной записью имеющей привилегии администратора.

Системные требования для Oracle Database10*g* Release 2 (10.2) для Microsoft Windows (x64) представлены в таблице 2.

Таблица 2 – Системные требования для Oracle Database10*g****.***

|  |  |
| --- | --- |
| Параметр | Значение |
| Оперативная память (RAM) | минимум 512 MB, рекомендовано 2GB  Для Windows Vista и Windows Server 2008 минимум 1GB  Виртуальная память должна быть минимум вдвое больше |
| Свободное место на диске | 2.57 GB |
| Видеокарта | 256 цветов (минимум) |
| Процессор | AMD 64 или Intel Extended Memory 64 Technology (EM64T) |
| Операционная система | ■ Windows Server 2003, Standard x64 Edition  ■ Windows Server 2003, Enterprise x64 Edition  ■ Windows Server 2003, Datacenter x64 Edition  ■ Windows XP Professional x64 Edition  ■ Windows Vista x64, Service Pack 1 - Business, Enterprise,  and Ultimate editions  ■ Windows Server 2008 x64 - Standard, Enterprise,  Datacenter, Web, Standard without Hyper-V, Enterprise  without Hyper-V, and Datacenter without Hyper-V editions  Windows Multilingual User Interface Pack поддерживается на Windows Server 2003, Windows XP Professional, Windows Vista and Windows Server 2008. |

Также необходимо наличие установленного Developer Suite 10g (9.0.4.0.0) for Microsoft Windows.

В таблице 3 представлены минимальные системные требования к установке

Таблица 3 – Системные требования к установке:

|  |  |
| --- | --- |
| **Требование** | **Минимальное значение** |
| Оперативная память | 128Мб (рекомендуется 256Мб) |
| Виртуальная память | 384Мб |
| Место на жестком диске | 943 Мб |
| Видеокарта | 256 цветов |
| Процессор | 500 МГц |
| Операционная система | Windows Server 2003  Windows XP Professional SP1 |

3.2. Описание метода работы с информационной системой

Для начала необходимо запустить нашу информационную систему в браузере. Первое, что нас встретит, будет FORM\_MENU, как было показано на рисунке 10 ранее. Нажав на пункт «Заказ», нам откроется форма таблицы «Заказ», как было показано на рисунке 4. Теперь нажмем на кнопку «Выбрать заказчика» и нам откроется список значений LOV для формы «Заказчик»:

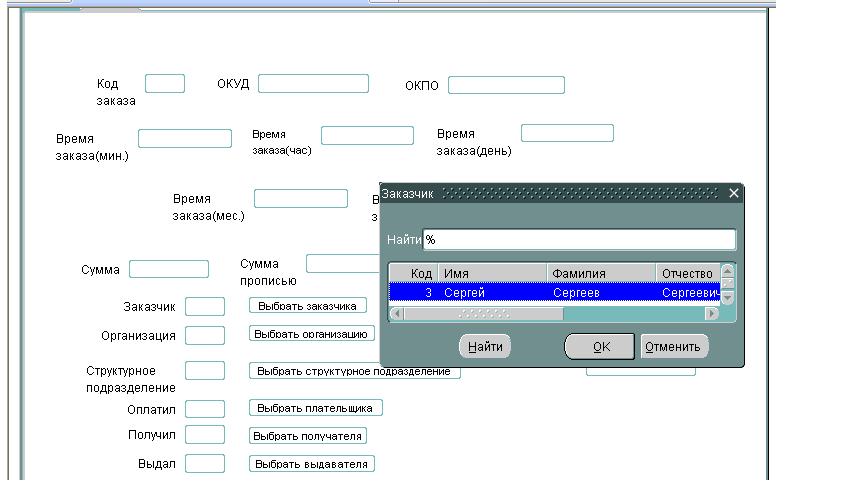


Рисунок 21 – LOV Заказчик

Аналогично делаем с остальными кнопками и заполняем поля. После надо нажать на кнопку «ОК» для сохранения данных.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы пошагово была достигнута поставленная цель – разработана информационная система для автоматизации подготовки, хранения и выдачи на печать заказа покупателя на приобретение материальных ценностей.

В ней мы реализовали следующие функции:

* внесение данных в справочники;
* подстановка существующих записей с помощью списка значений LOV;
* вывод на печать отчета.

В ходе выполнения работы были достигнуты поставленные задачи:

* проанализирована предметная область;
* создана модель информационной системы;
* реализована информационная система;
* разработано руководство пользователя.

В ходе реализации информационной системы были получены навыки и опыт работы с СУБД Oracle.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем. М, 2011
2. Кузин А.В., Левонисова С.В. Базы данных (5-е изд.). Изд: Академия, 2012- 317 с.
3. Марков А.С. Базы данных. Введение в теорию и методологию - М.: Финансы и статистика, 2012- 512 с.
4. Перри Джеймс, Пост Джеральд. Введение в Oracle 10g. 2006 – 704 c.
5. Тиори Т., Фрай Дж. Проектирование структур баз данных. М, 2010
6. Хаббард Дж. Автоматизированное проектирование баз данных. М, 2011
7. Фейерштейн С. Oracle PL/SQL. Для профессионалов/ Б. Прибыл. – Питер,2011 г. – 800 стр.