Лабораторная работа N 2. Диаграммы сущность–связь.

При создании моделей данных используется метод семантического моделирования. Семантическое моделирование основывается на значении структурных компонентов или характеристик данных, что способствует правильности их интерпретации (понимания, разъяснения). В качестве инструмента семантического моделирования используются различные варианты диаграмм сущность-связь (ER — Entity-Relationship). Модели сущность-связь основаны на выделении в предметной области, для которой осуществляется проектирование базы данных, различных типов объектов, информацию о которых требуется хранить в базе данных.

Набор однотипных объектов предметной области образует сущность. Между сущностями могут быть установлены информационные связи (зависимости), которые также могут быть учтены при проектировании схемы базыданных. Совокупность сущностей и связи между ними составляют информационную модель данных предметной области.

Для выполнения лабораторной работы мы будем использовать CASE-средство для проектирования и документирования баз данных

* CA ERwin Data Modeler (далее ERwin).

- или Astah Professional

Работа с программой начинается с создания новой модели, для которой нужно указать тип и целевую СУБД . ERwin позволяет создавать логическую, физическую модели и модель, совмещающую логический и физический уровни. На данном этапе нас интересует логический уровень.

Логический уровень - это абстрактный взгляд на данные, на нем данные представляются так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире.

Для создания нового файла необходимо нажать File-New и выбрать Logical(рис 1).

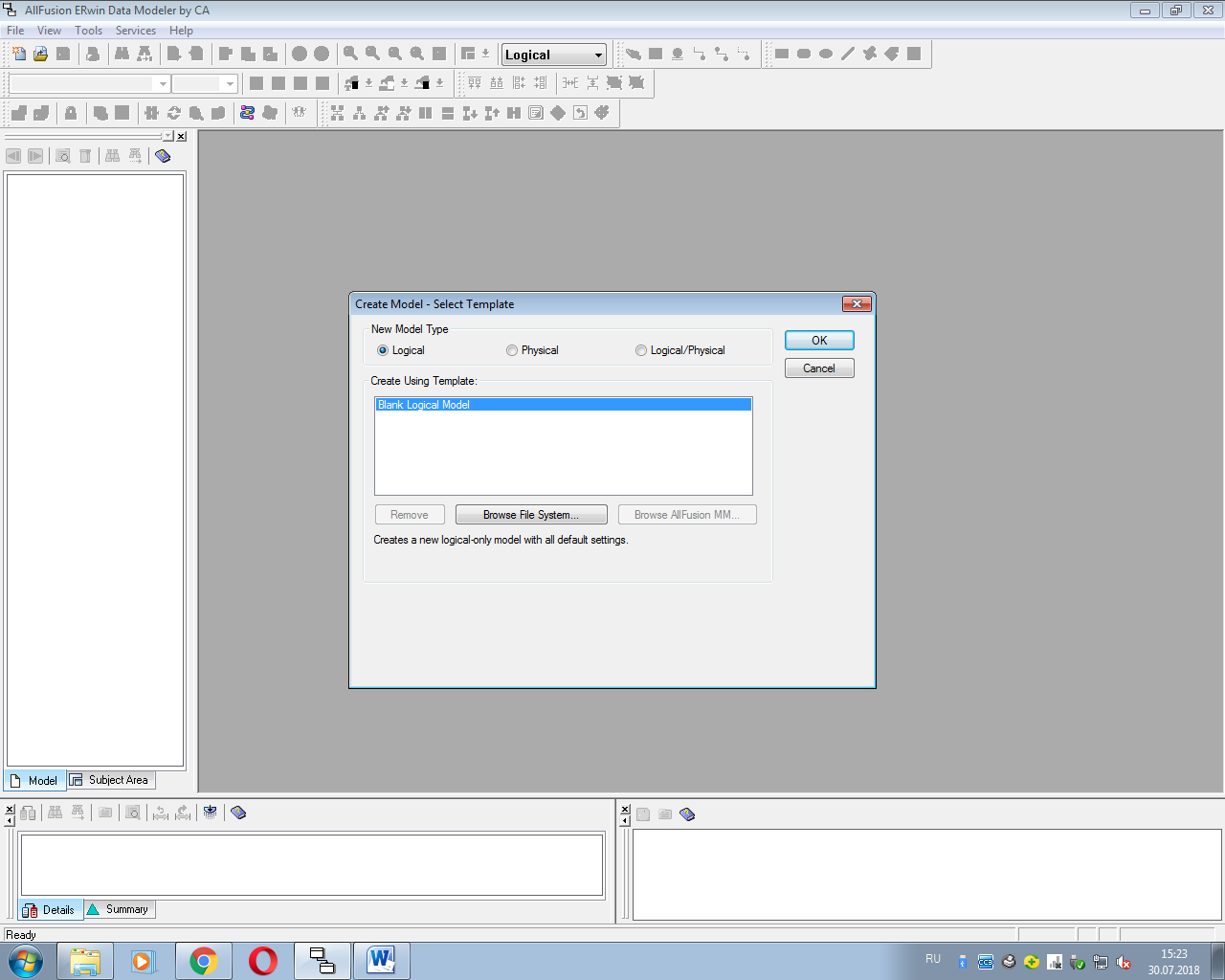


Рисунок 1 – создание нового файла Erwin.

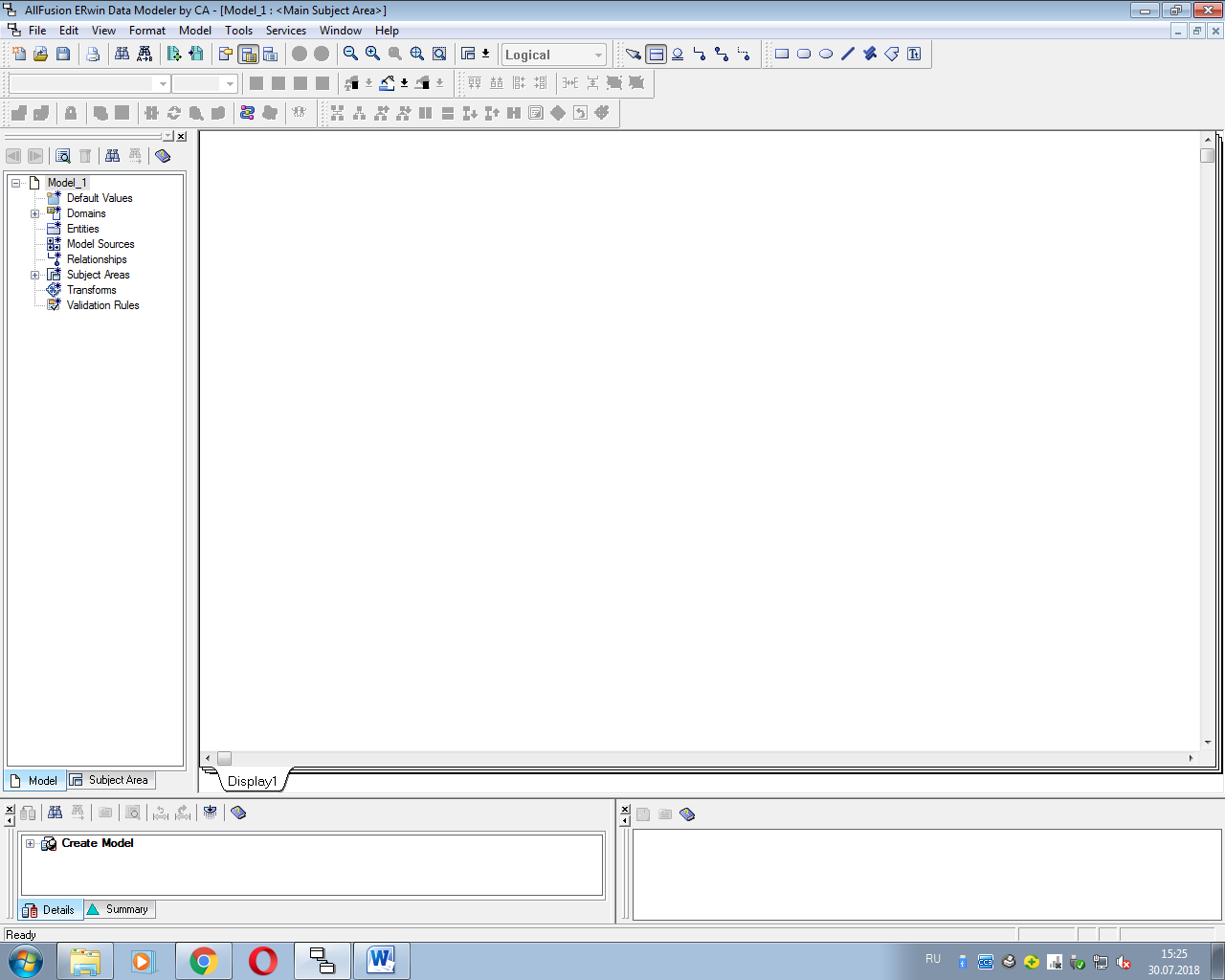


Рисунок 2 ­– рабочая область программы.

Для создания на логическом уровне сущностей и связей между ними предназначена панель Toolbox:

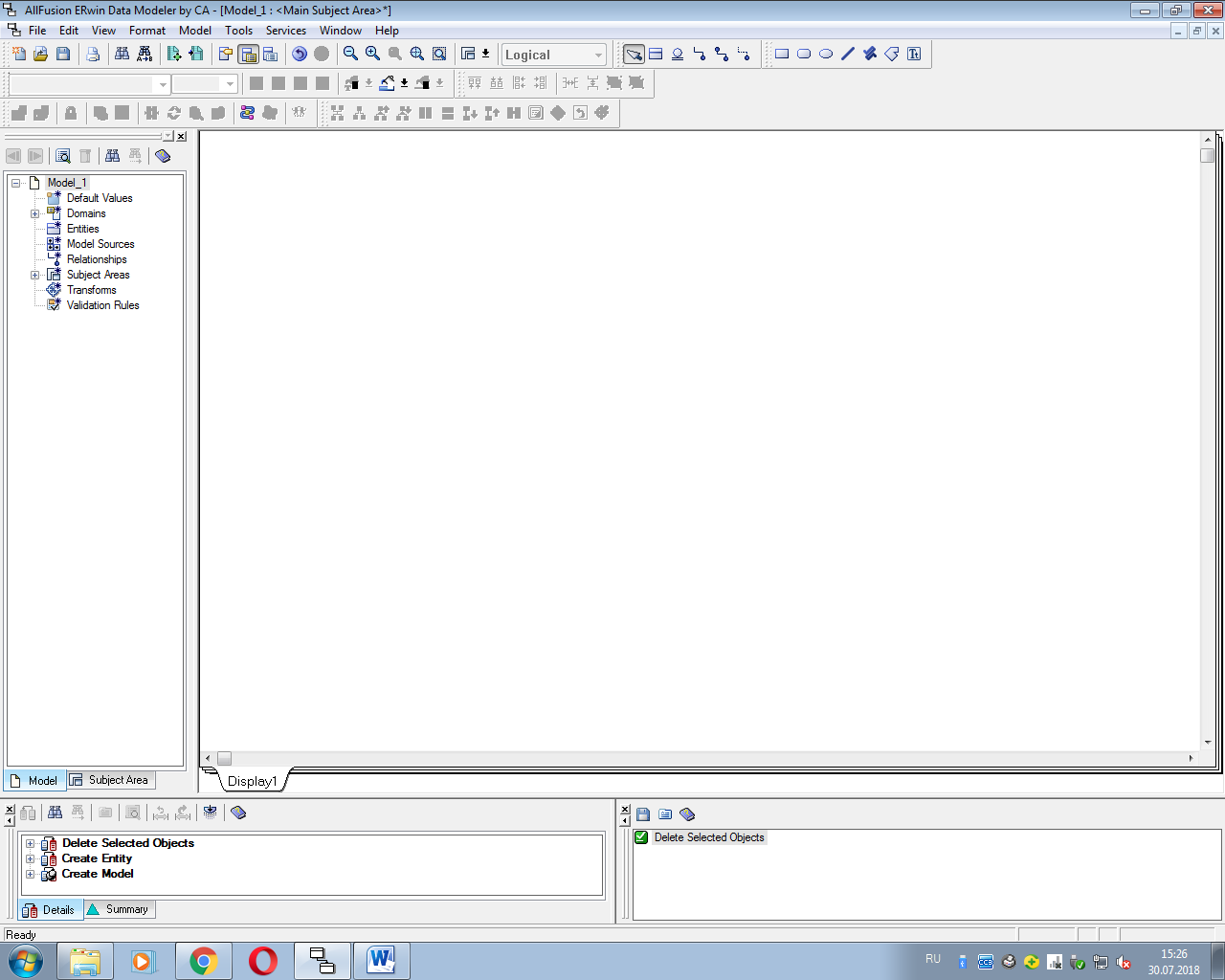
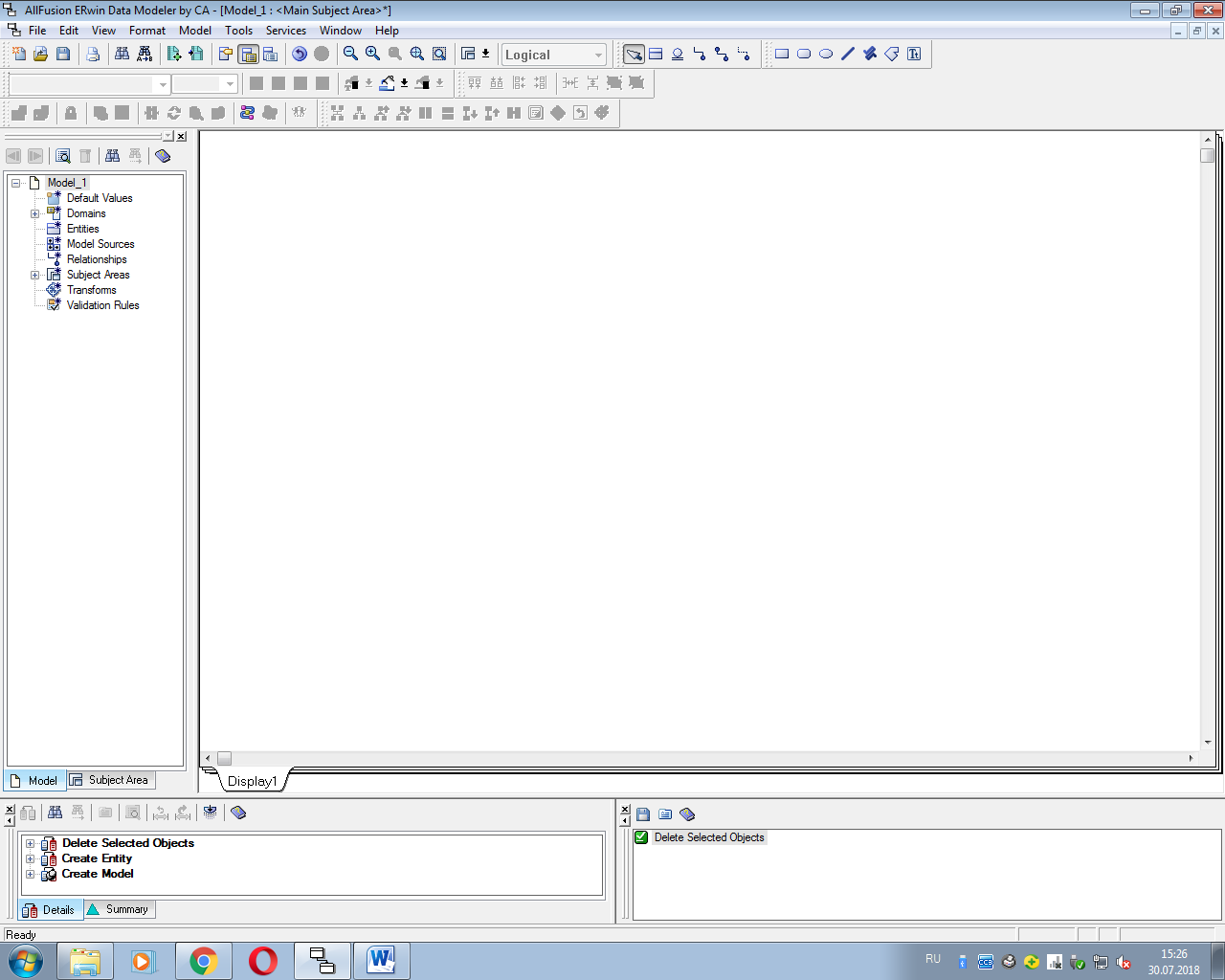


Рисунок 3 – Toolbox

Для создания сущности этого нужно щелкнуть по кнопке entity( ) и затем по свободному месту на модели.

Сущность (entity) - это объект, который может быть идентифицирован неким способом, отличающим его от других объектов(рис. 4). Примеры: конкретный человек, предприятие, событие и т.д.

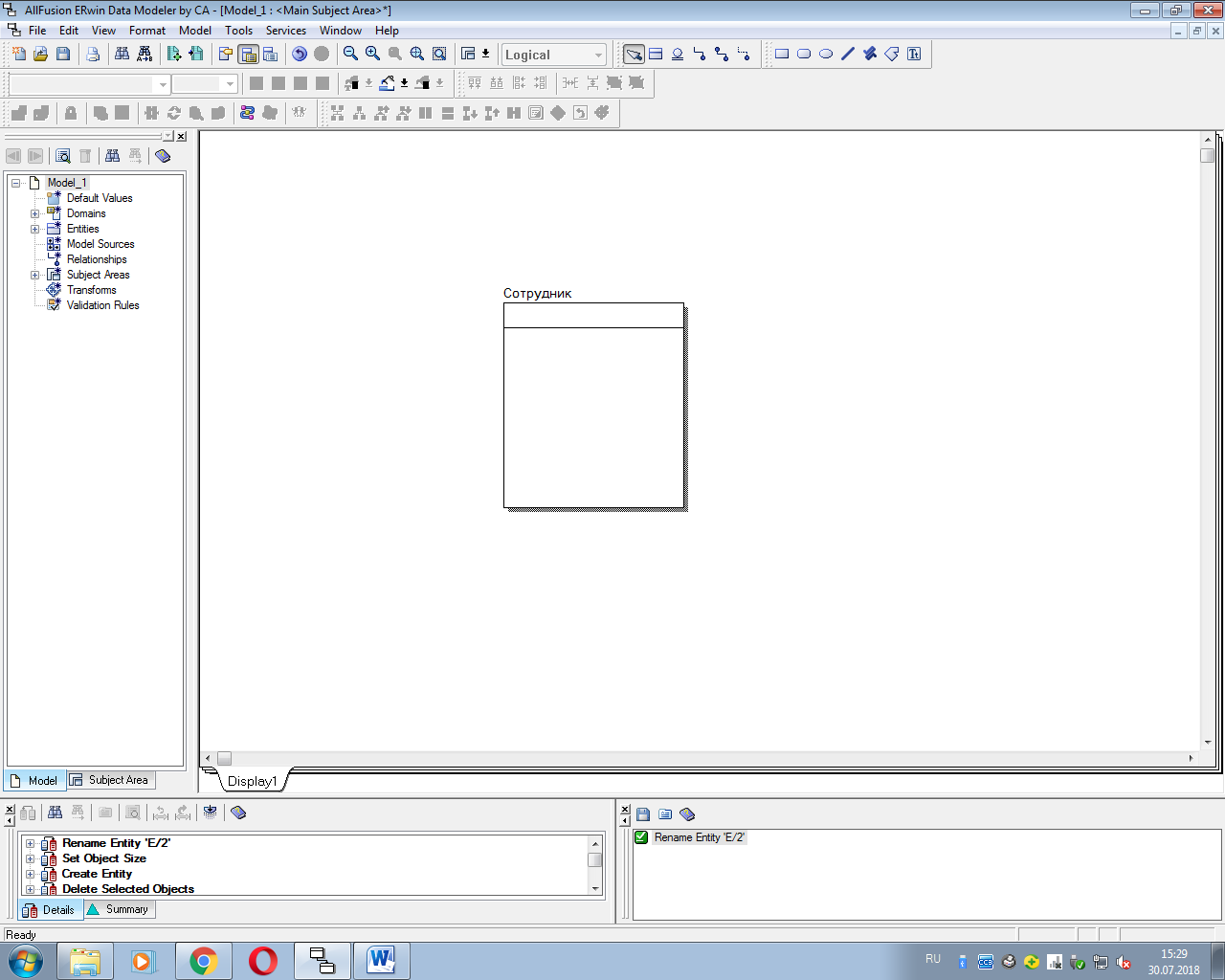


Рисунок 4 – сущность сотрудник.

Набор сущностей (entity set) - множество сущностей одного типа (обладающих одинаковыми свойствами). Примеры: все люди, предприятия, праздники и т.д. Наборы сущностей не обязательно должны быть непересекающимися.

Атрибут сущности - это именованная характеристика, являющаяся некоторым свойством сущности.

 Ключ сущности - это неизбыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности. Неизбыточность заключается в том, что удаление любого атрибута из ключа нарушается его уникальность.

Для создания атрибутов в сущности необходимо щелкнуть по полю внутри прямоугольника, далее нажать New и в поле Atribute Name ввести имя.После ввода имени, что бы атрибут стал ключевым, поставьте галочку в поле Primary Key(рис. 5).

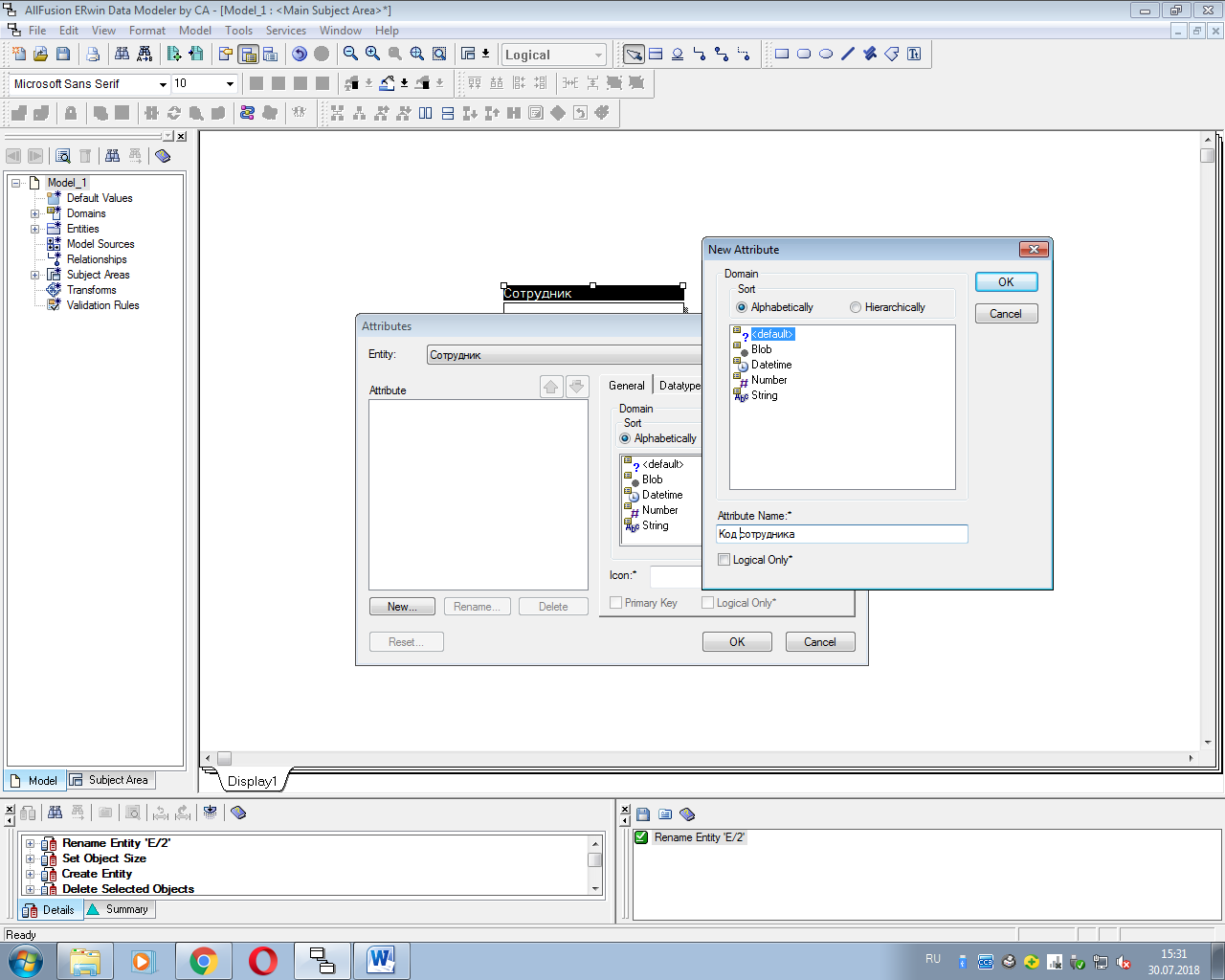


Рисунок 5 – создание атрибутов сущности Сотрудник.

Связь - это некоторая ассоциация между *двумя* сущностями. Одна сущность может быть связана с другой сущностью или сама с собою.

Связи позволяют по одной сущности находить другие сущности, связанные с нею.

Каждая связь может иметь один из следующих типов связи(рис.6):

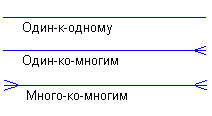


Рисунок 6 – типы связи

Связь типа один-к-одному означает, что один экземпляр первой сущности (левой) связан с одним экземпляром второй сущности (правой). Связь один-к-одному чаще всего свидетельствует о том, что на самом деле мы имеем всего одну сущность, неправильно разделенную на две.

Связь типа один-ко-многим означает, что один экземпляр первой сущности (левой) связан с несколькими экземплярами второй сущности (правой). Это наиболее часто используемый тип связи. Левая сущность (со стороны "один") называется родительской, правая (со стороны "много") - дочерней. Характерный пример такой связи приведен на Рис. 4.

Связь типа много-ко-многим означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Тип связи много-ко-многим является временным типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи должен быть заменен двумя связями типа один-ко-многим путем создания промежуточной сущности.

Каждая связь может иметь одну из двух модальностей связи(рис.7).

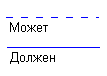
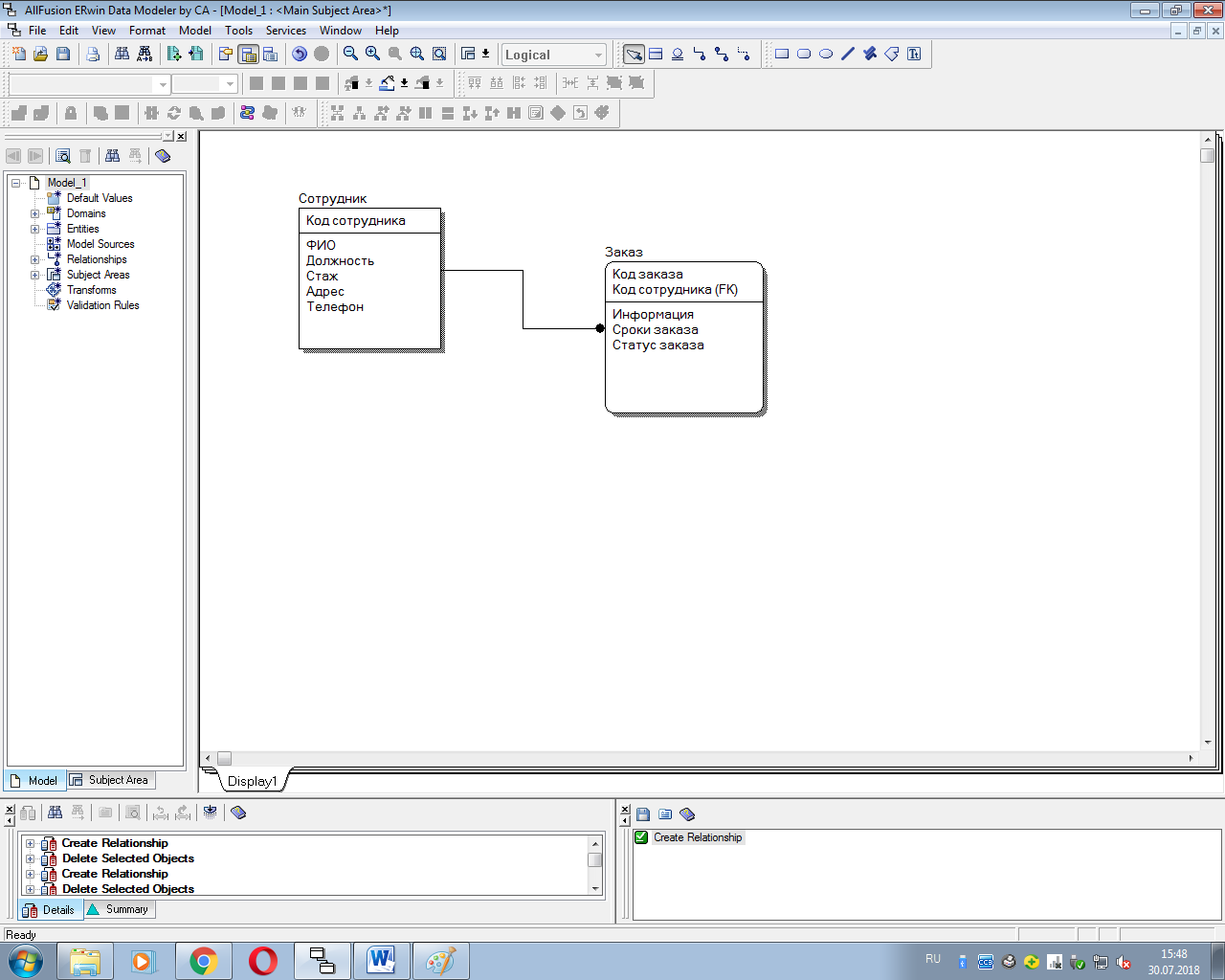


Рисунок 7 – модальности связи.

Модальность "может" означает, что экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, а может быть и не связан ни с одним экземпляром.

Модальность "должен" означает, что экземпляр одной сущности обязан быть связан не менее чем с одним экземпляром другой сущности.

Для обозначения связи в программе Erwin необходимо выбрать нужную связь () и щелкнуть сначала по родительской сущности, а далее по дочерней(рис.8).

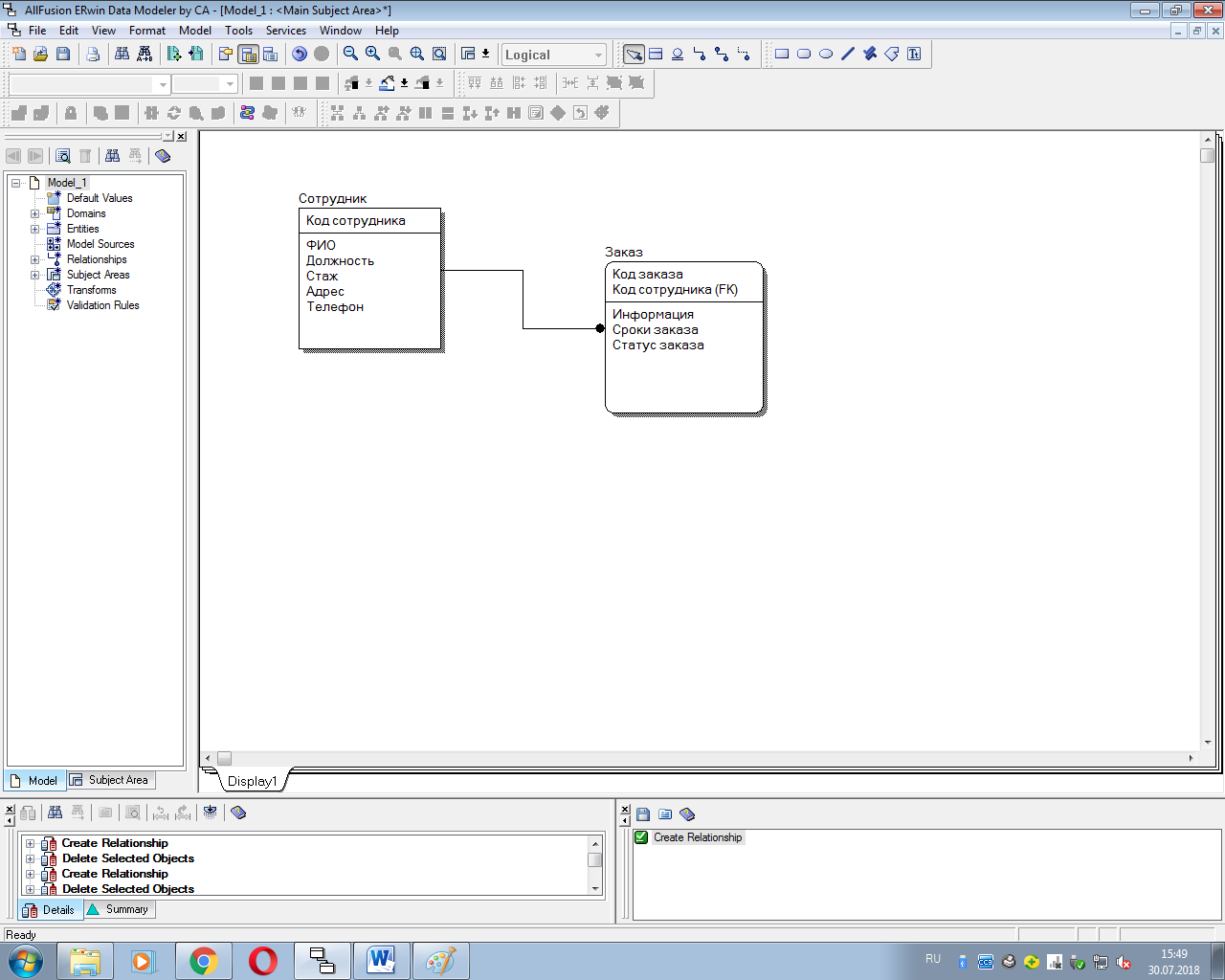


Рисунок 8 – связанные сущности в программе Erwin.

При разработке ER-моделей мы должны получить следующую информацию о предметной области:

1. Список сущностей предметной области.
2. Список атрибутов сущностей.
3. Описание взаимосвязей между сущностями

Для разработки базы для библиотеки, мы должны изучить предметную область и процессы, происходящие в ней.

В ходе изучения предметной области, выяснилось, что проектируемая БД должна выполнять следующие функции:

1. Хранить информацию о всех книгах
2. Хранить информацию о читателях
3. Хранить информацию о взятых книгах

Выделим сущности из этих предложений: книга, читатель, взятая книга.

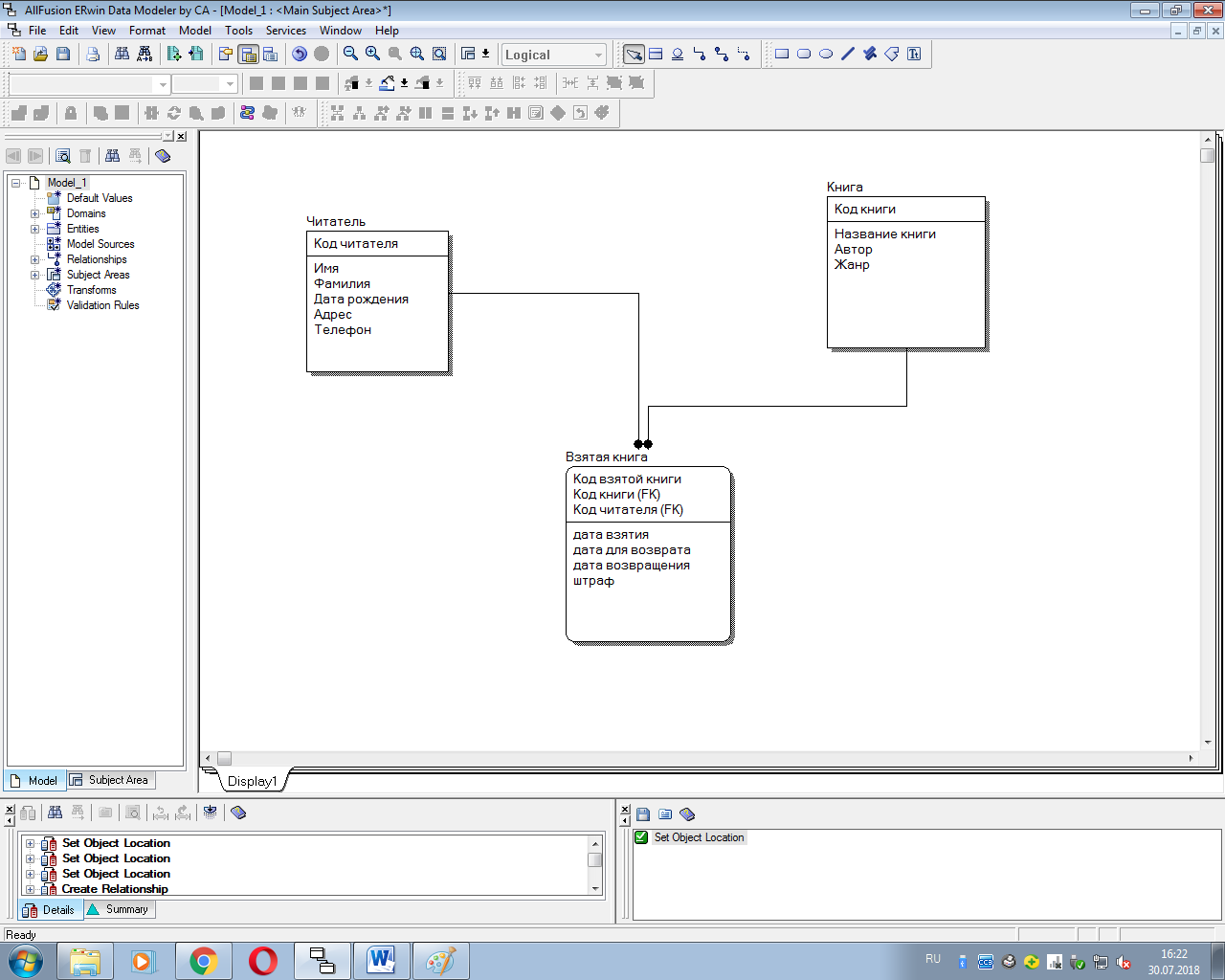


Рисунок 9 – ER диаграмма БД библиотека.

Разработанный выше пример ER-диаграммы является примером концептуальной диаграммы. Это означает, что диаграмма не учитывает особенности конкретной СУБД. По данной концептуальной диаграмме можно построить физическую диаграмму, которая уже будут учитываться такие особенности СУБД, как допустимые типы и наименования полей и таблиц, ограничения целостности

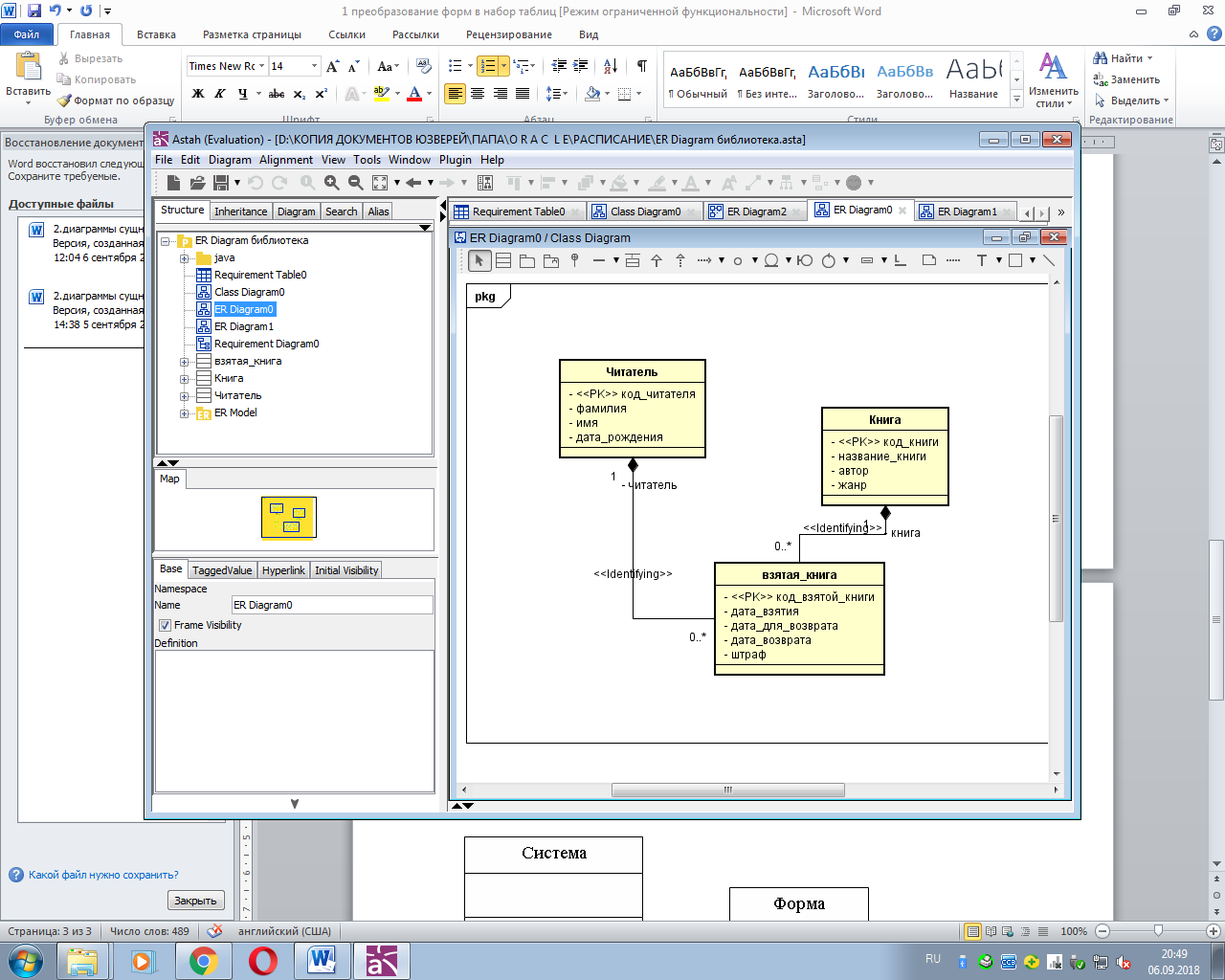
Задания для лабораторной работы:

В программе Astah Professional  или Erwin выполнить создание концептуальной модели с описанием предметной области согласно своему варианту.

1. Рекламное агентство
2. Турфирма
3. [Страховая компания](https://accesshelp.ru/baza-dannyh-access-strahovaja-kompanija/)
4. [Гостиница](https://accesshelp.ru/baza-dannyh-access-gostinica/)
5. [Ломбард](https://accesshelp.ru/baza-dannyh-access-lombard/)
6. [Реализация готовой продукции](https://accesshelp.ru/baza-dannyh-access-realizacija-gotovoj-produkcii/)
7. [Ведение заказов](https://accesshelp.ru/baza-dannyh-access-vedenie-zakazov/)
8. [Бюро по трудoустройству](https://accesshelp.ru/baza-dannyh-access-bjuro-po-trudoustrojstvu/)
9. [Нотариальная контора](https://accesshelp.ru/baza-dannyh-access-notarialnaja-kontora/)
10. [Фирма по продаже запчастей](https://accesshelp.ru/baza-dannyh-access-firma-po-prodazhe-zapchastej/)
11. [Курсы повышения квалификации](https://accesshelp.ru/baza-dannyh-access-kursy-povyshenija-kvalifikacii/)
12. [Определение факультативов для студентов](https://accesshelp.ru/baza-dannyh-access-opredelenie-fakultativov-dlja-studentov/)
13. [Распределение учебной нагрузки](https://accesshelp.ru/baza-dannyh-access-raspredelenie-uchebnoj-nagruzki/)
14. [Распределение дополнительных обязанностей](https://accesshelp.ru/baza-dannyh-access-raspredelenie-dopolnitelnyh-objazannostej/)
15. [Техническое обслуживание станков](https://accesshelp.ru/baza-dannyh-access-tehnicheskoe-obsluzhivanie-stankov/)
16. [Туристическая фирма](https://accesshelp.ru/baza-dannyh-access-turisticheskaja-firma/)
17. [Грузовые перевозки](https://accesshelp.ru/baza-dannyh-access-gruzovye-perevozki/)
18. [Учет телефонных переговоров](https://accesshelp.ru/baza-dannyh-access-uchet-telefonnyh-peregovorov/)
19. [Учет внутриофисных расходов](https://accesshelp.ru/baza-dannyh-access-uchet-vnutriofisnyh-rashodov/)
20. Прокат автомобилей

Запуск инструмента проектирования системы, поддерживающим UML (Унифицированный язык моделирования) - Astah Professional

1. Создание <NEW PROJECT>
2. <Diagram>
3. <ER Diagram>
4. Создание ER Диаграммы (см. Лабораторная работа № 2)



1. <Tools>
2. Провести операцию экспорта <ER Diagram> <Export Sql> <Export>

CREATE TABLE Книга (

код\_книги CHAR(10) NOT NULL,

название\_книги CHAR(10),

автор CHAR(10),

жанр CHAR(10)

);

ALTER TABLE Книга ADD CONSTRAINT PK\_Книга PRIMARY KEY (код\_книги);

CREATE TABLE Читатель (

код\_читателя CHAR(10) NOT NULL,

фамилия CHAR(10),

имя CHAR(10),

дата\_рождения CHAR(10)

);

ALTER TABLE Читатель ADD CONSTRAINT PK\_Читатель PRIMARY KEY (код\_читателя);

CREATE TABLE взятая\_книга (

код\_взятой\_книги CHAR(10) NOT NULL,

код\_читателя CHAR(10) NOT NULL,

код\_книги CHAR(10) NOT NULL,

дата\_взятия CHAR(10),

дата\_для\_возврата CHAR(10),

дата\_возврата CHAR(10),

штраф CHAR(10)

);

ALTER TABLE взятая\_книга ADD CONSTRAINT PK\_взятая\_книга PRIMARY KEY (код\_взятой\_книги,код\_читателя,код\_книги);

ALTER TABLE взятая\_книга ADD CONSTRAINT FK\_взятая\_книга\_0 FOREIGN KEY (код\_читателя) REFERENCES Читатель (код\_читателя);

ALTER TABLE взятая\_книга ADD CONSTRAINT FK\_взятая\_книга\_1 FOREIGN KEY (код\_книги) REFERENCES Книга (код\_книги);

1. Просмотреть возможности кодирования проекта в форматы RTF, XML. Проанализировать полученные результаты.

*Introduction*

*This Manual, “Astah Reference Manual”, briefly explains the functions of Astah and how to use them.*

*Astah Professional is a system design tool that supports UML (Unified Modeling Language) 2.x (partly), UML1.4, Flowchart, Data Flow Diagram, ER diagram, CRUD, Requirement diagram and Mind Map.*

*Astah UML is* a modeling tool that supports UML (Unified Modeling Language) 2.x (partly), UML1.4 and Mind Map.

