МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт информационных технологий |
| наименование института (факультета) |
| Кафедра математического и программного обеспечения ЭВМ |
| наименование кафедры  Проектирование баз данных |

наименование дисциплины в соответствии с учебным планом

ЛАБОРАТОРНАЯ РАБОТА №2

«Физическое проектирование»

|  |  |
| --- | --- |
| Исполнитель |  |
| студент | 1ПИб-02-3оп-22 |
|  | группа |
|  | Маркелов С. А. |
|  | Фамилия, имя, отчество |
| Руководитель | Селяничев О. Л. |
|  | Ф.И.О. преподавателя |
| Оценка |  |
| Подпись |  |

2025 год

1. Ознакомиться с понятием «валидация» и научиться применять его при проектировании.

Валидация базы данных – это процесс проверки данных, которые вводятся в базу данных, на соответствие определённым правилам и ограничениям, установленным для обеспечения корректности, надёжности и целостности информации. Валидация помогает защитить систему от некорректных, противоречивых или вредоносных данных и гарантирует, что данные соответствуют требованиям и техническим ограничениям.

ERwin позволяет задавать правила валидации. Для этого на панели вкладок необходимо нажать на вкладку «Model» («Модель»), а затем в выпадающем меню выбрать пункт «Validation Rules» («Правила валидации») (рис. 1).

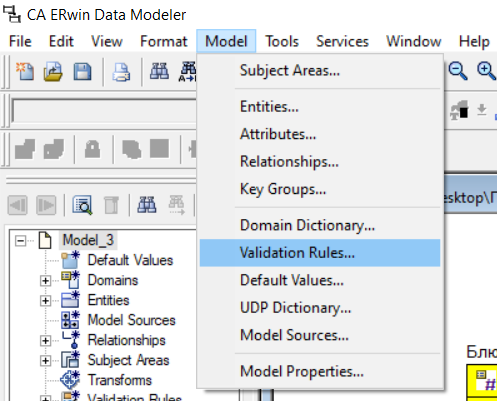


Рис. 1. Выпадающее меню «Model»

После этого откроется окно «Validation Rules» («Правила валидации»), в котором эти правила необходимо задать (рис. 2).

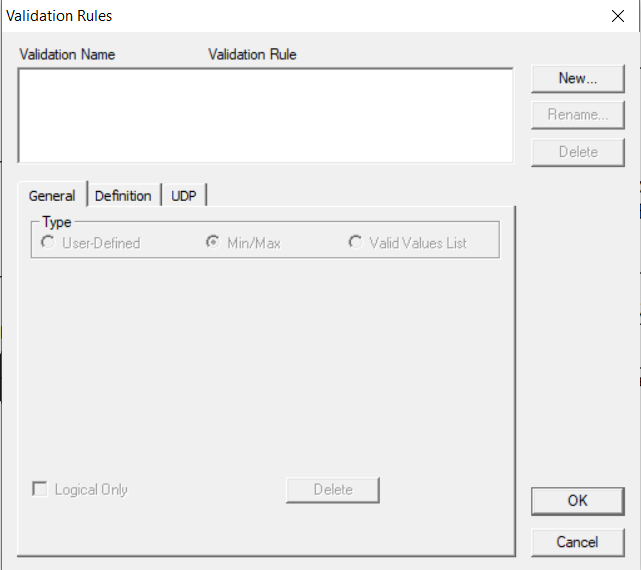


Рис. 2. Окно «Validation Rules»

Ознакомимся подробнее с этим окном.

В верхней части окна – список всех правил. Справа от него 3 кнопки:

* «New» («Новое») – для добавления нового правила;
* «Rename» («Переименовать») – для переименования выбранного правила;
* «Delete» («Удалить») – для удаления выбранного правила.

Ниже находится область с возможностью выбора одной из трех вкладок:

* «General» («Главная») – раздел, в котором задаются основные правила валидации;
* «Definition» («Описание») – раздел, в котором можно описать правило текстом;
* «UDP» («User Defined Properties» – «Пользовательские свойства») – раздел, в котором можно прикрепить к правилу собственные дополнительные свойства, создаваемые пользователем.

Выберем вкладку «General». В данной вкладке программа предлагает выбрать один из трех типов правил:

* «User-Defined» («Определяемое пользователем») – пользовательское выражение, написанное вручную;
* «Min/Max» («Минимум/Максимум») – правило, позволяющее задать минимальное и/или максимальное допустимое значение;
* «Valid Values List» («Список допустимых значений») – правило, позволяющее задать фиксированный список допустимых значений.

Создадим правило типа «Min/Max». Для этого в верхней части окна нажимаем на кнопку «New». После этого появится еще одно окно «New Validation Rule» («Новое правило валидации»). В этом окне в поле «Name» («Имя») необходимо ввести название нового правила – назовем его «Вес» (рис. 3).



Рис. 3. Окно «New Validation Rule»

Выбираем тип правила «Min/Max». После этого в окне появятся два поля – «Min» – для минимального значения и «Max» – для максимального. Правее также есть два флажка – «Quote» («Кавычка») – для заключения значения в кавычки при формировании SQL-запросов и «NOT» («НЕ») – для проверки на НЕсоответствие правилу.

Создадим правило, что значение атрибута «Вес» в сущностях «Состав» и «Поставки» обязательно должно быть положительным. Укажем значение «Min» = 1. Также ограничим максимальное значение – зададим, что вес не может быть более 1000, указав значение «Max» = 1000 (рис. 4).

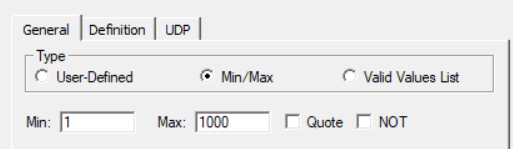


Рис. 4. Создание правила типа «Min/Max»

Правило создано и записано как «%AttFieldName BETWEEN 1 AND 1000» («%ИмяАтрибута МЕЖДУ 1 И 1000») (рис. 5).



Рис. 5. Правило типа «Min/Max»

Теперь данное правило необходимо привязать к самим атрибутам, т. к. до этого был создан лишь шаблон правила. Для этого нажимаем правой клавишей мыши по сущности «Состав» и в контекстном меню выбираем пункт «Attributes» (рис. 6).

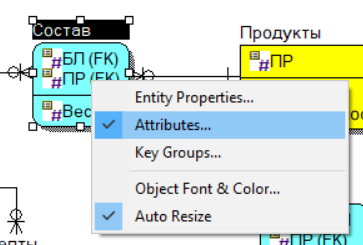


Рис. 6. Контекстное меню сущности «Состав»

Откроется контекстное меню «Attributes» («Атрибуты»). Слева – список атрибутов. Выбираем атрибут «Вес». Справа – область из нескольких вкладок. Нас интересует вкладка «Constraint» («Ограничение»). Ставим флажок «Validation Constraint» («Ограничение валидации») после чего из выпадающего списка «Valid» выбираем правило «Вес» (рис. 7). Нажимаем на кнопку «ОК».

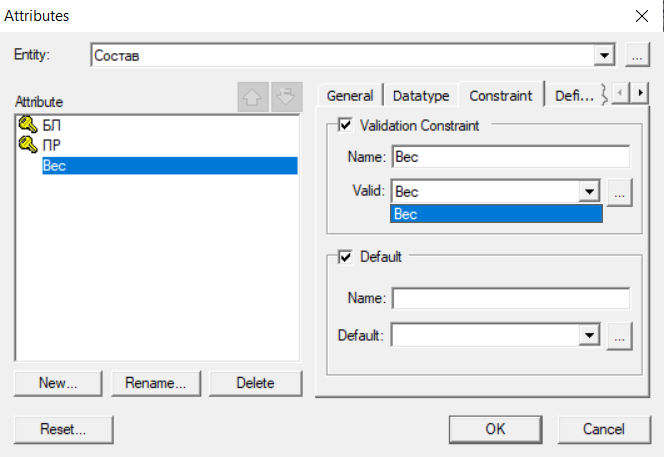


Рис. 7. Окно «Attributes» для сущности «Состав»

Аналогичные действия проделываем и для сущности «Поставки» (рис. 8).

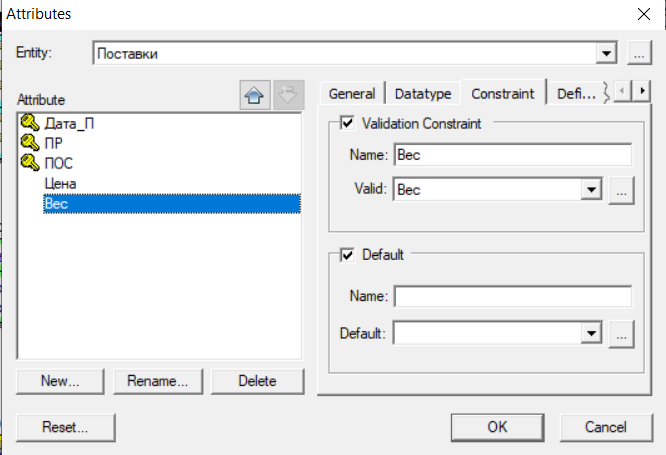


Рис. 8. Окно «Attributes» для сущности «Поставки»

Возвращаемся к окну «Validation Rules». Теперь создадим правило типа «Valid Values List». Выбираем соответствующий тип, после чего появится таблица «Valid Value» («Допустимые значения») с 3 столбцами:

* «Valid Value» («Действительное значение») – фактическое значение, которое допустимо для данного атрибута;
* «Display Value» («Отображаемое значение») – значение, которое увидит пользователь в интерфейсе (отчете или форме);
* «Definition» («Описание») – текстовое описание, поясняющее, что значит это значение.

Создадим новое правило, назовем его Поставщики и выберем тип «Valid Values List». Создадим перечень допустимых поставщиков, которые могут быть значениями атрибута «Поставщик» в сущности «Поставщики». Их наименования прописываем в столбец «Valid Value». При этом сами наименования будут также содержать организационно-правовые формы юридических лиц (ООО, ПАО, АО и т. п.). Для пользователей эта информация будет излишней, поэтому в «Display Value» указываем названия поставщиков уже без организационно-правовой формы (рис. 9).

Поле «Definition» никак не влияет на проверку значений, SQL-код и работу самой БД. Оно используется лишь для пояснений. Поэтому оно не обязательно для заполнения. В данном случае дополнительные пояснения не требуются – из названия поставщика сразу становится понятно, кто поставляет продукцию.

Правило создано и записано как «%AttFieldName IN (‘ООО «Овощи»’, ‘АО «Кубанские фругты»’, ‘ПАО «Гринфилд»’, ‘ООО «Белорусский картофель»’)» («%ИмяАтрибута В (…)»).

После создания правила его необходимо привязать к полю «Поставщик» сущности «Поставщики», аналогично тому, как это делалось для сущностей «Состав» и «Поставки» (рис. 10).

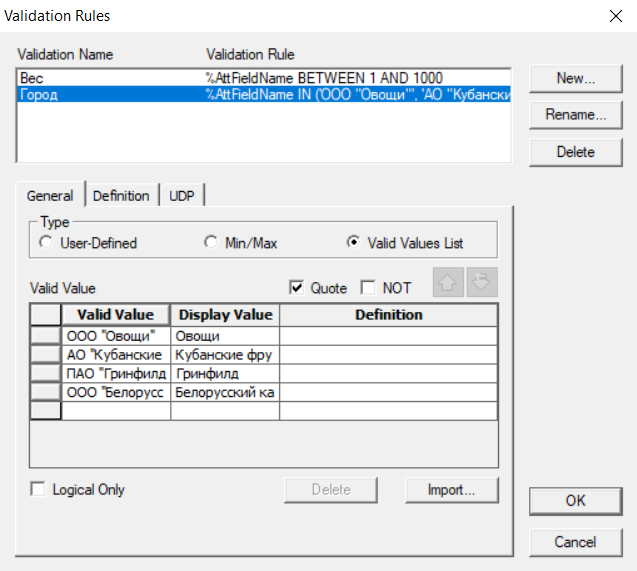


Рис. 9. Правило типа «Validation Values List»

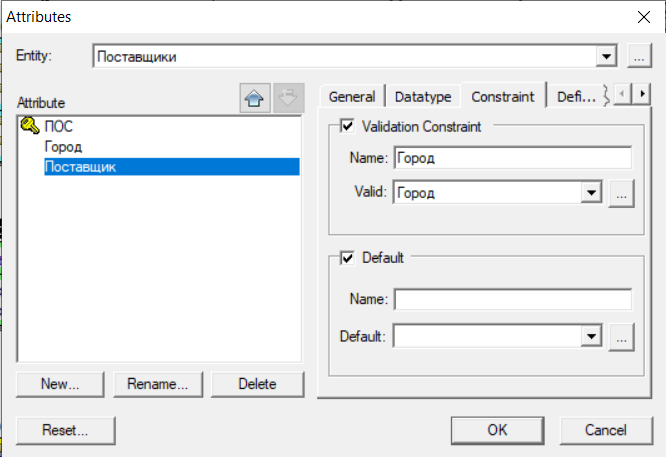


Рис. 10. Окно «Attributes» для сущности «Поставки»

Снова возвращаемся к окну «Validation Rules». Теперь создадим правило типа «Used-Defined». Правило такого типа необходимо записывать пользователю. Для предыдущих типов программа составляла правила сама («%AttFieldName BETWEEN…», «%AttFieldName IN…»). Здесь необходимо прописать такое же правило, но уже вручную.

Создадим новое правило и назовем его Город. Зададим список городов, которые могут быть значениями атрибута «Город» сущности «Города».

Правило будет иметь вид «%AttFieldName IN (‘Москва’, ‘Санкт-Петербург’, ‘Череповец’, ‘Вологда’, ‘Ярославль’)» (рис. 11).

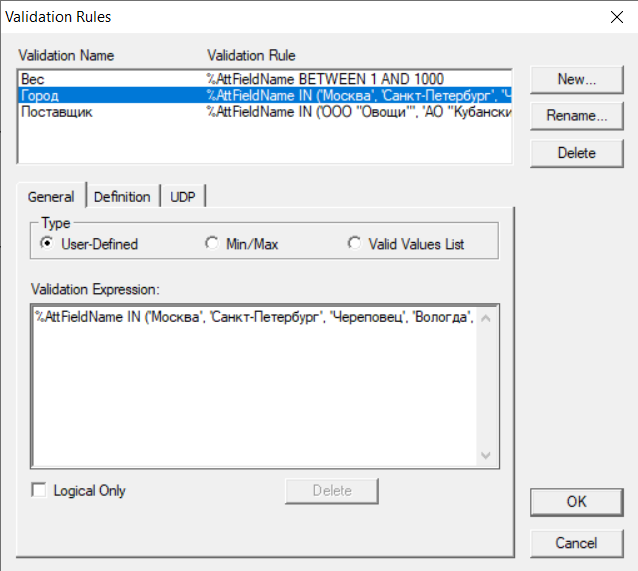


Рис. 11. Правило типа «Used-Defined»

Теперь применяем данное правило к атрибуту «Город» сущности «Города» (рис. 12).

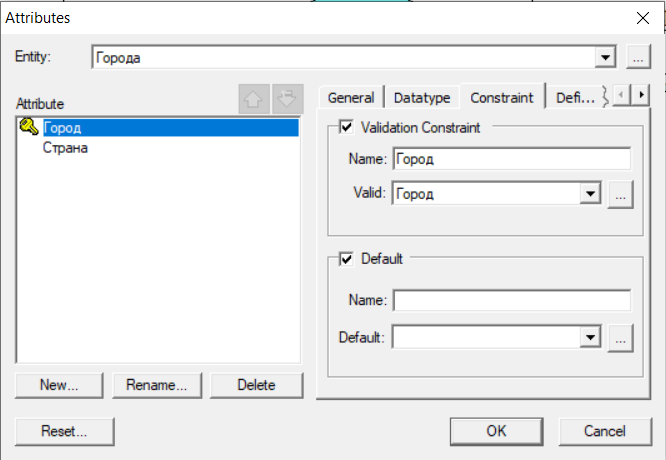


Рис. 12. Окно «Attributes» для сущности «Города»

Теперь создадим правило, проверяющее на несоответствие условию (с флагом «NOT»). Возвращаемся к окну «Validation Rules», создаем правило типа «Min/Max» с названием «Калорийность». Зададим правило, что калорийность НЕ может быть меньше или равна 0. Для этого в поле «Max» вводим значение 0 и ставим флаг «NOT». Без флага «NOT» допустимыми значениями были бы все, что меньше или равны 0. А с флагом «NOT» будет все наоборот – допустимыми значениями станут те, которые НЕ меньше и НЕ равны (т. е. больше) 0. Правило будет иметь вид «%AttFieldName > 0» (рис. 13). Далее применяем его к атрибуту «Калорийность» сущности «Продукты» (рис. 14).

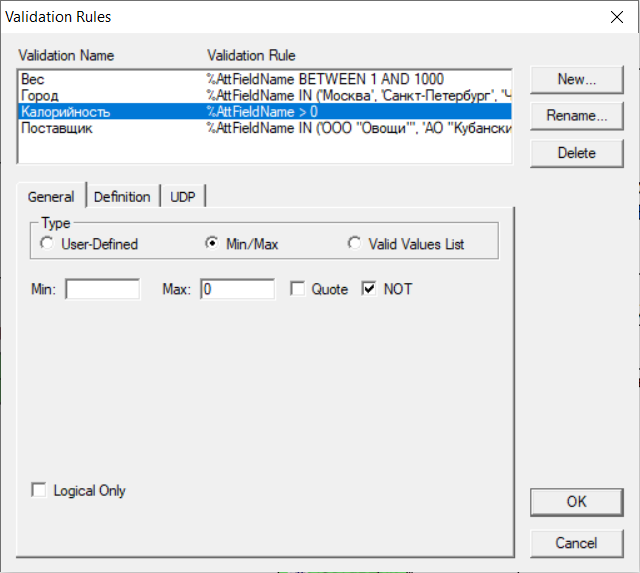


Рис. 13. Правило типа «Min/Max» с флагом «NOT»

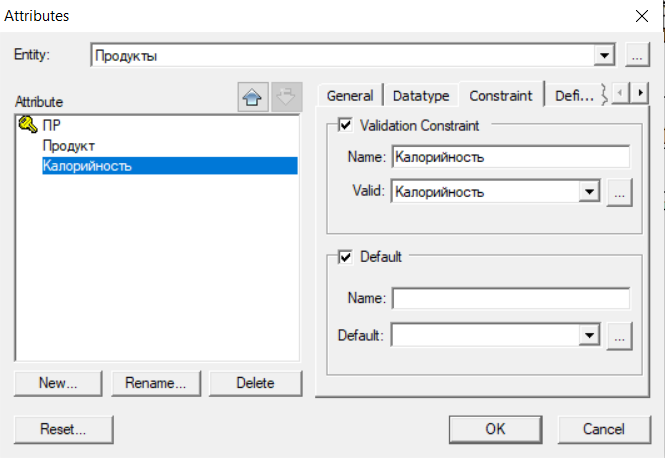


Рис. 14. Окно «Attributes» для сущности «Продукты»

ERwin также позволяет задавать значения атрибутов по умолчанию, если пользователь их не заполнит самостоятельно. Для этого на панели вкладок необходимо нажать на вкладку «Model» («Модель»), а затем в выпадающем меню выбрать пункт «Default Values» («Значения по умолчанию») (рис. 15).

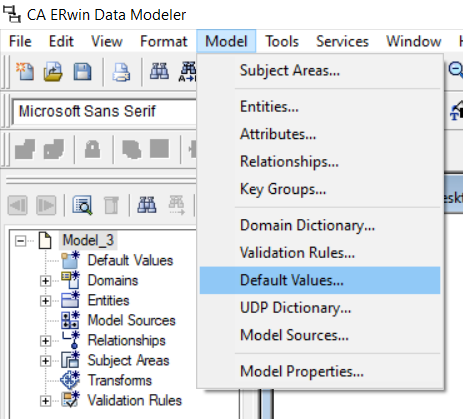


Рис. 15. Выпадающее меню «Model»

Откроется окно «Default Values» («Значения по умолчанию»), интерфейс которого схож с окном «Validation Rules» (рис. 16).

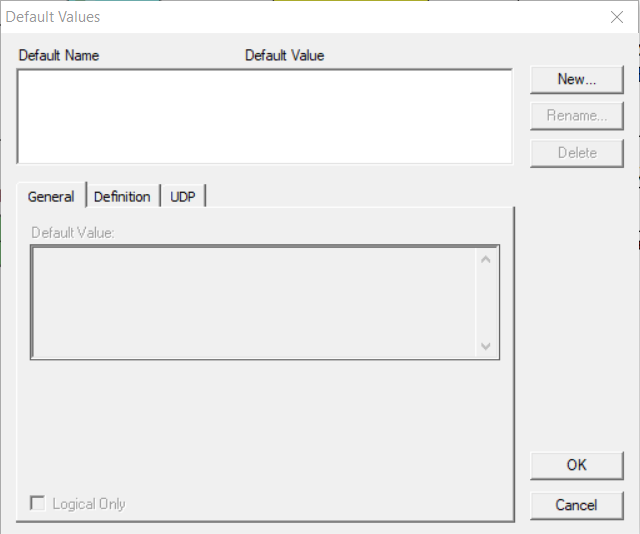


Рис. 16. Окно «Default Values»

Для создания нового значения по умолчанию необходимо нажать на кнопку «New» («Новое»). Откроется окно «New Default Value» («Новое значение по умолчанию»), в котором необходимо ввести название нового значения по умолчанию. Назовем новое значение по умолчанию «Город» (рис. 17).



Рис. 17. Окно «New Default Value»

Далее переходим во вкладку «General» («Главная») и в текстовом поле вводим само значение по умолчанию. Например, пусть значением города по умолчанию будет «Череповец» (рис. 18).

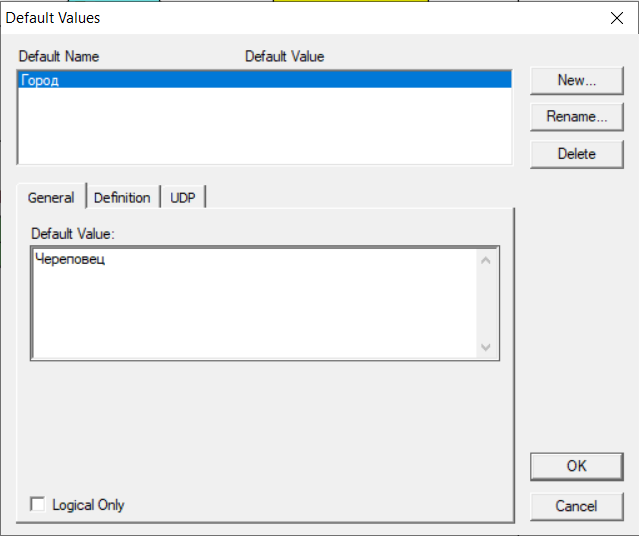


Рис. 18. Ввод значения города по умолчанию

Теперь значение по умолчанию необходимо связать с атрибутом. Для этого нажимаем правой клавишей мыши по сущности «Города» и в контекстном меню выбираем пункт «Attributes» (рис. 19).

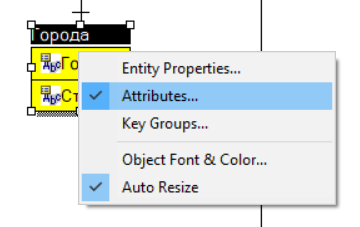


Рис. 19. Контекстное меню сущности «Города»

Откроется контекстное меню «Attributes» («Атрибуты»). Слева – список атрибутов. Выбираем атрибут «Город». Справа – область из нескольких вкладок. Нас интересует вкладка «Constraint» («Ограничение»). Ставим флажок «Default» («По умолчанию») после чего из выпадающего списка «Default» выбираем правило «Город» (рис. 20). Нажимаем на кнопку «ОК».

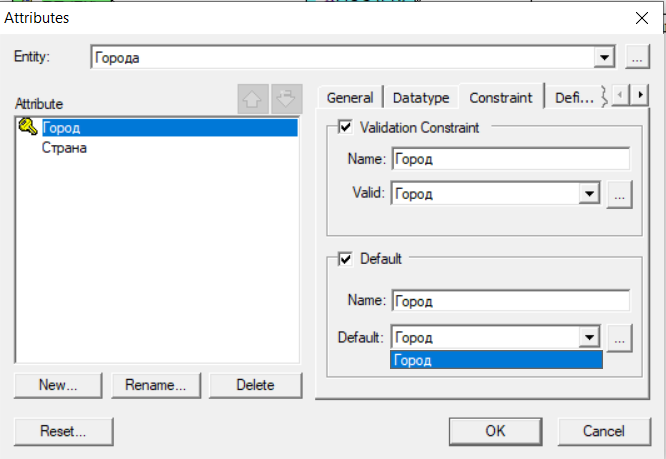


Рис. 20. Окно «Attributes» для сущности «Города»

Аналогично создадим еще 3 значения по умолчанию (рис. 21):

* «Вес» = 1;
* «Калорийность» = 1;
* «Поставщик» = ООО «Овощи».

После этого также привязываем их к соответствующим атрбиутам.

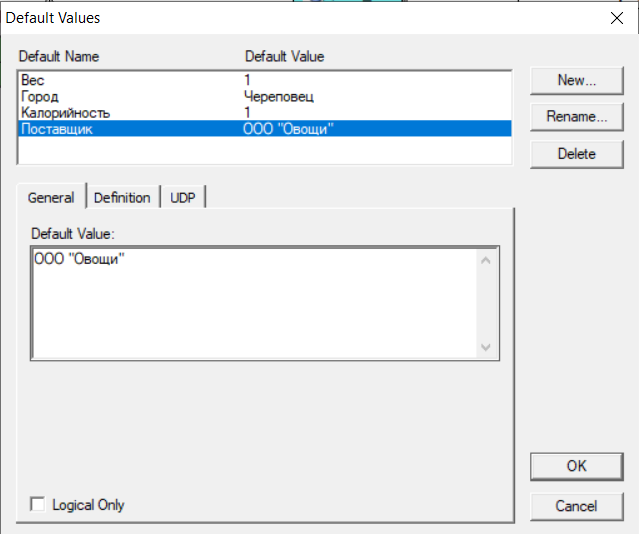


Рис. 21. Значения по умолчанию

1. Изучить правила обращения ERwin’а с индексами.

Индексация – это процесс создания индекса по одному или нескольким полям таблицы, чтобы ускорить доступ к данным.

Индекс – это объект, который содержит отсортированные значения этих полей и ссылки на соответствующие записи таблицы. Индексы позволяют значительно ускорить выполнение запросов, особенно при поиске, фильтрации, сортировке и соединении таблиц.

ERwin позволяет проводить индексацию. Она проводится на уровне физической модели. Для ее создания необходимо на панели вкладок нажать на вкладку «Tools» («Инструменты»), а затем в выпадающем меню выбрать пункт «Derive New Model» («Вывести новую модель») (рис. 22).

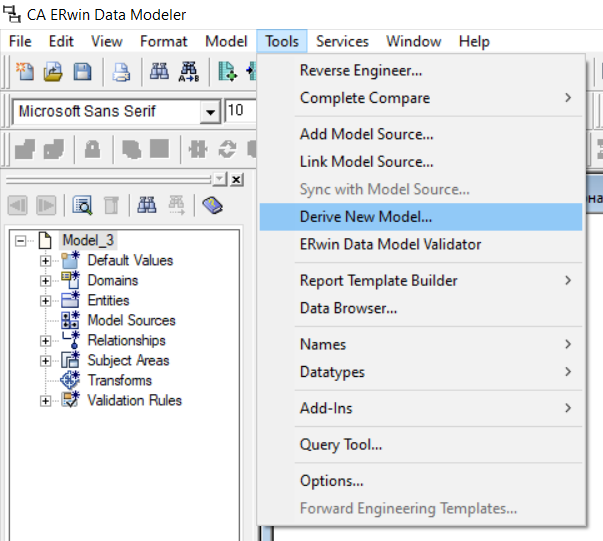


Рис. 22. Выпадающее меню «Tools»

Откроется окно «Derive Model» («Вывести модель»). В левом меню выбираем раздел «Target Model» («Целевая модель»), значение «New Model Type» («Тип новой модели») выбираем «Logical/Physical» («Логическая/Физическая») (рис. 23). В выпадающем списке «Database» выбираем «Access». Далее нажимаем кнопку «Derive» («Вывести»).

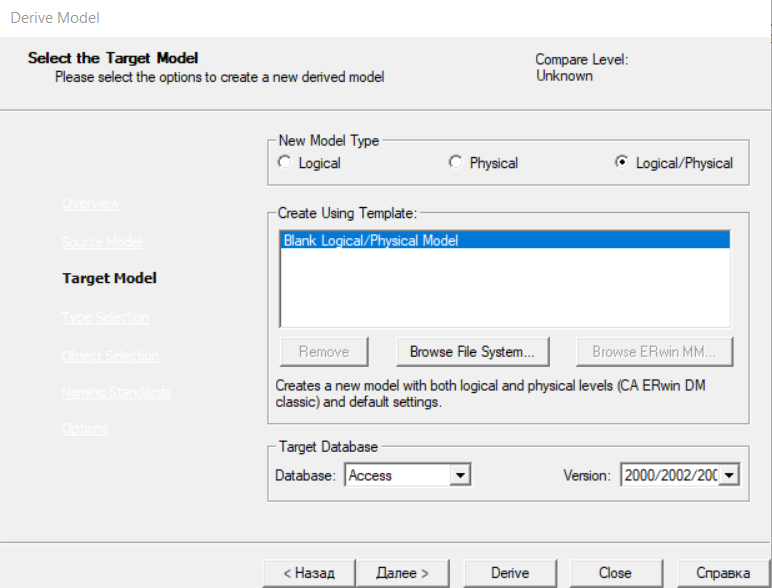


Рис. 23. Окно «Derive Model»

Далее на панели инструментов в выпадающем списке выбираем «Physical» («Физическая») (рис. 24).

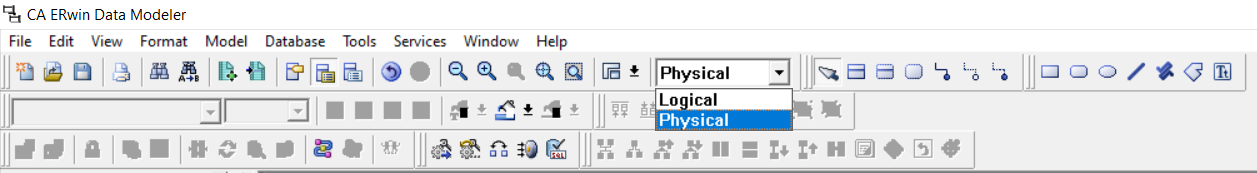


Рис. 24. Выпадающий список типа модели

Физическая модель представлена на рис. 25.

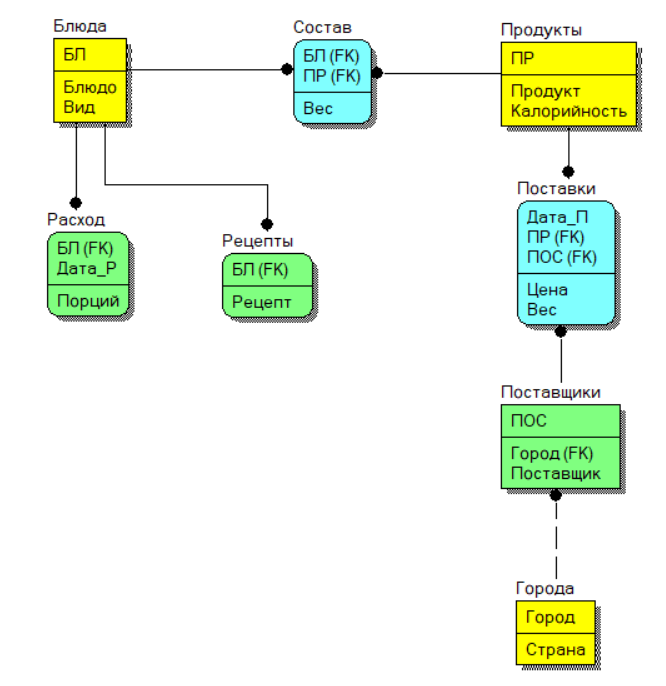


Рис. 25. Физическая модель

Теперь можно проводить индексацию. Для этого на панели вкладок необходимо нажать на вкладку «Model» («Модель»), а затем в выпадающем меню выбрать пункт «Indexes» -> «Table» («Индексы» -> «Таблица») (рис. 26).

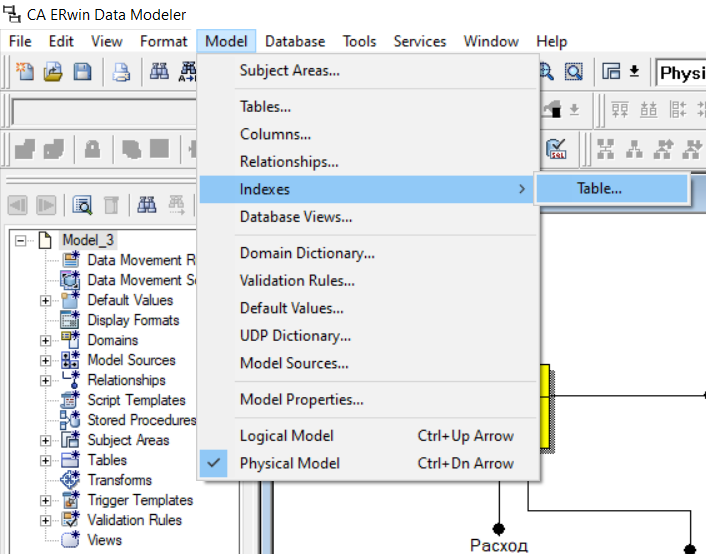


Рис. 26. Выпадающее меню «Model»

Откроется окно «Access Indexes» («Индексы Access»). В верхней части этого окна – выпадающий список «Table» («Таблица») со списком всех таблиц. Ниже – список индексов для выбранной таблицы. Справа от списка три кнопки:

* «New» («Новый») – для создания нового индекса;
* «Rename» («Переименовать») – для переименования выбранного индекса;
* «Delete» («Удалить») – для удаления выбранного индекса.

Еще ниже находится область из нескольких вкладок. Во вкладке «General» («Главная») находится 2 списка:

* «Avaible Columns» («Доступные столбцы») – список полей, которые можно добавить в индекс;
* «Index Members» («Состав индекса») – поля, уже включенные в индекс.

В качестве примера рассмотрим индексы таблицы «Поставки» (рис. 27-28). У нее имеется 2 индекса – первичный ключ «ХРКПоставки», состоящий из полей «Дата\_П», «ПР» и «ПОС» и альтернативный ключ ХАК1Поставки, состоящий из поля «Вес». При необходимости в индекс можно добавлять и другие поля или исключать уже добавленные.

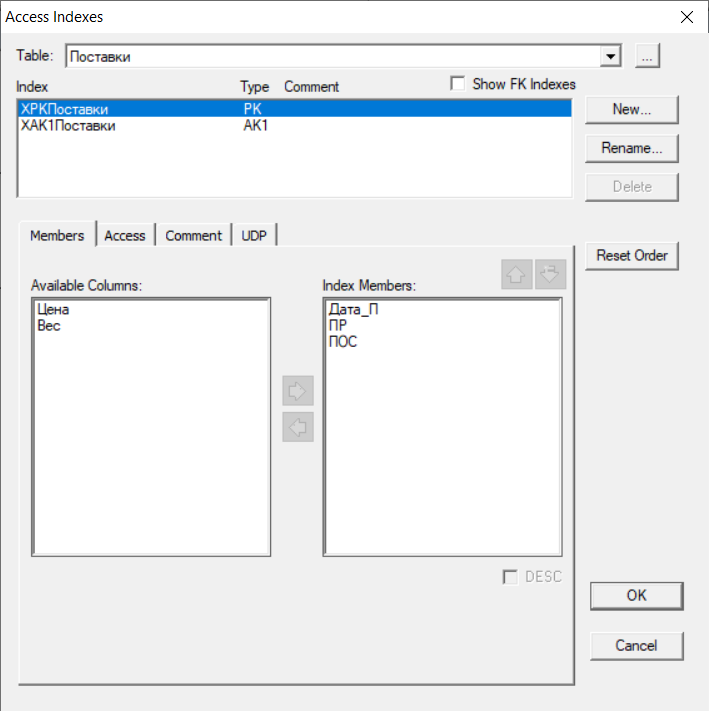


Рис. 27. Индексы таблицы «Поставки»

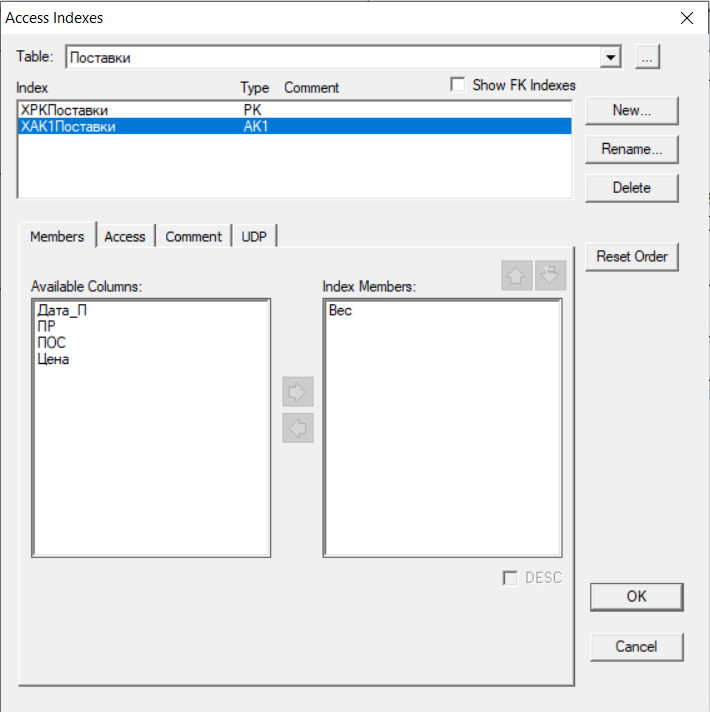


Рис. 28. Индексы таблицы «Поставки»

Создадим новый индекс. Для этого необходимо нажать на кнопку «New». Появится окно «New Index» («Новый индекс»). В поле «Index» («Индекс») необходимо указать название нового индекса. Оставим название по умолчанию – «ХАК2Поставки» (рис. 29).

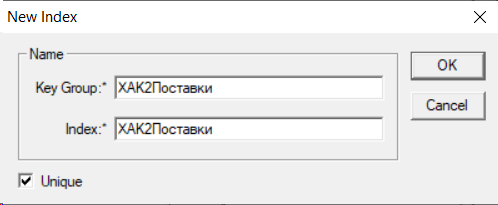


Рис. 29. Окно «New Index»

После этого необходимо вернутся в окно «Access Indexes» и во вкладке «Members» выбрать поля, которые будут использоваться в данном индексе. Выберем для индекса поля «ПР» и «ПОС» (рис. 30).

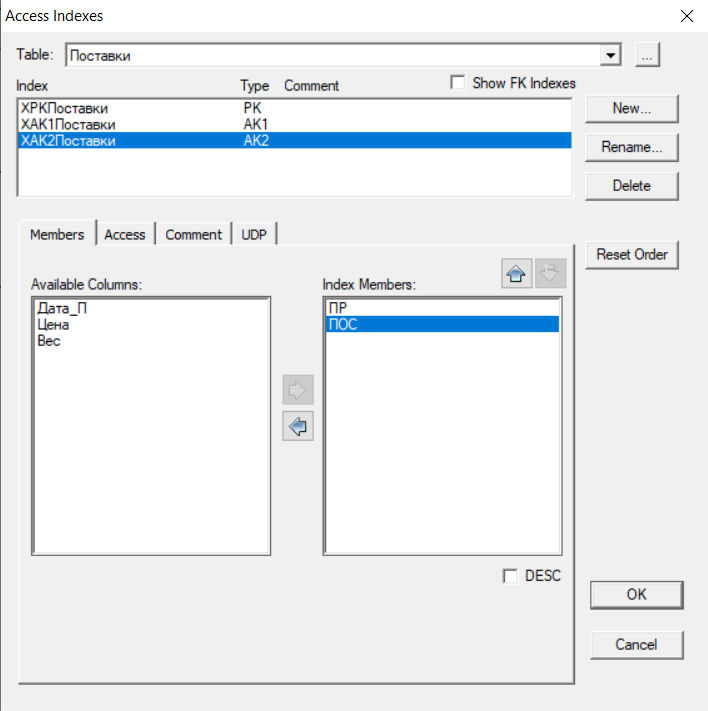


Рис. 30. Создание нового индекса

1. Рассмотреть работу на этапах логического и физического проектирования, уделив особое внимание последнему.

Логическая модель данных – это абстрактное представление структуры данных, которое определяет сущности, атрибуты, связи между сущностями и ограничения. Она не зависит от конкретной СУБД и служит основой для проектирования физической модели данных.

Физическая модель данных – это конкретная реализация логической модели данных с учётом особенностей конкретной СУБД. Она определяет физическое представление данных на диске, включая структуру таблиц, типы данных полей, связи между таблицами, индексы, ограничения целостности и другие параметры хранения.

Средствами ERwin была создана логическая модель. Позднее, с помощью инструмента «Derive New Model» была создана физическая модель для СУБД Access.

Следующим этапом будет создание схемы отношений для выбранной СУБД. Для этого на панели вкладок необходимо нажать на вкладку «Tools» («Инструменты»), а затем в выпадающем меню выбрать пункт «Forward Engineer» -> «Schema Generation» («Прямое проектирование» -> «Генерация схемы») (рис. 31).

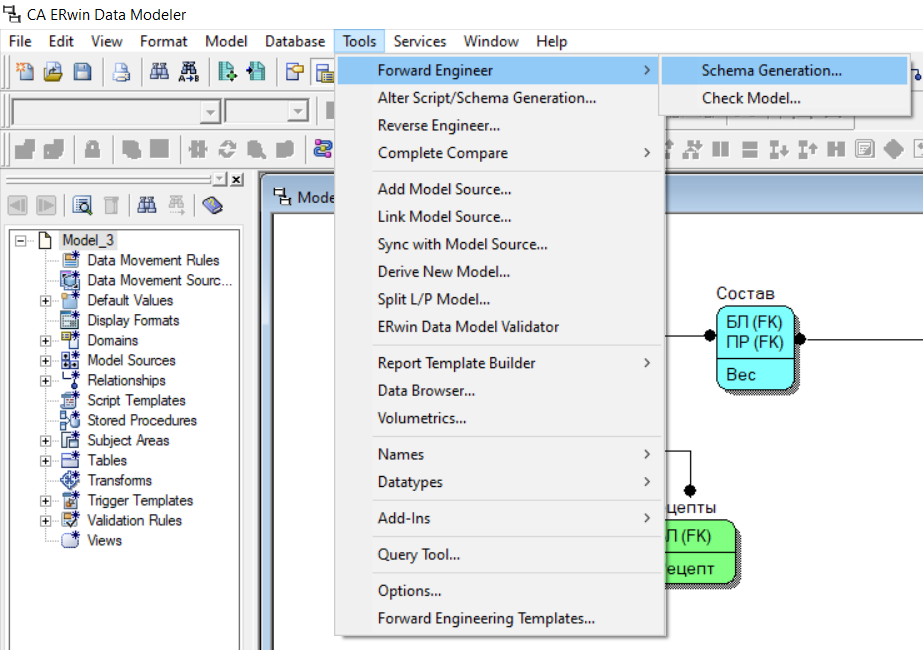


Рис. 31. Выпадающее меню «Tools»

После этого откроется окно «Forward Engineer Schema Generation» («Прямое проектирование – Генерация схемы») (рис. 32). Здесь можно выбрать параметры генерации схемы. Во вкладках «Options» («Опции») и «Schema» («Схема») можно выбрать, какие элементы будут перенесены в СУБД. Менять их не будем, используем те, что по умолчанию. Нажимаем на кнопку «Generate» («Сгенерировать»).

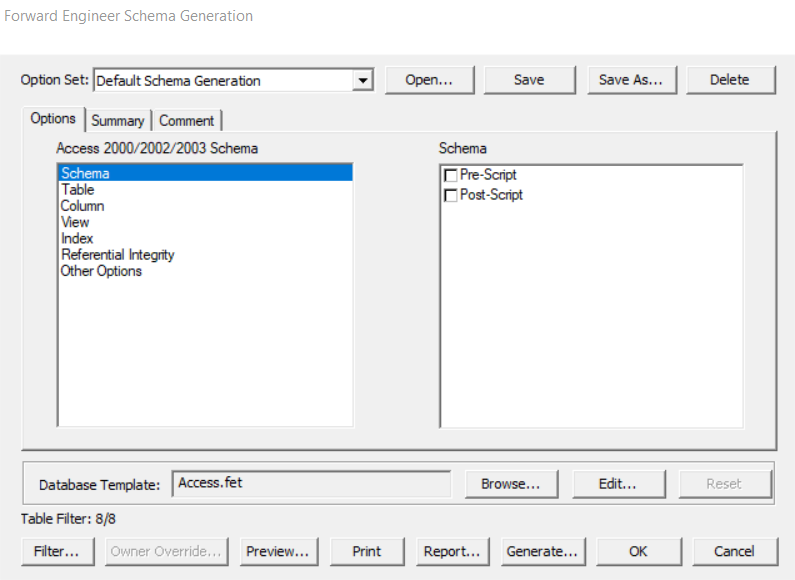


Рис. 32. Окно «Forward Engineer Schema Generation»

После этого появится еще одно окно подключения к базе данных (рис. 33). В середине данного окна находится таблица. В строке «Database» («База данных») необходимо вызвать окно выбора файла и выбрать файл базы данных формата .mdb, в который будет импортирована схема отношений (рис. 34). Если такого файла еще нет – его нужно создать. После этого нужно нажать на кнопку «Connect».

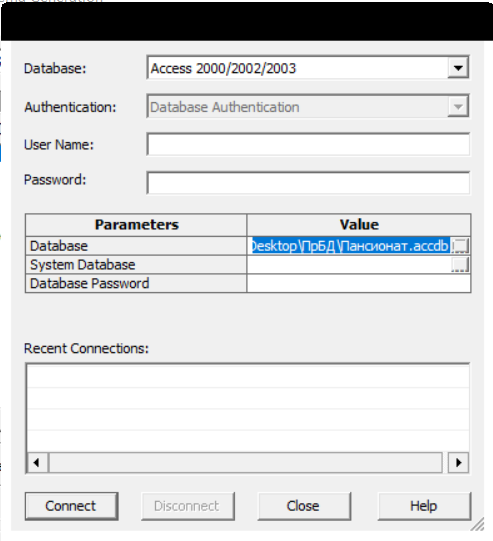


Рис. 33. Окно подключения к БД

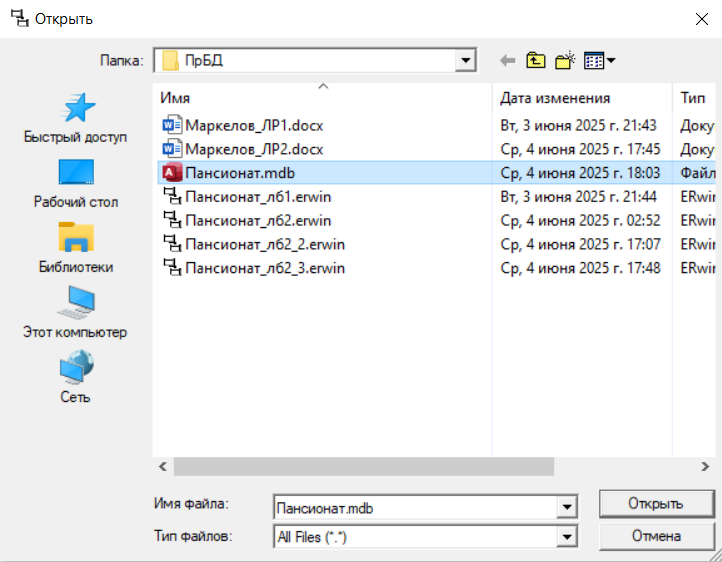


Рис. 34. Выбор файла БД

После этого начнется процесс генерации схемы данных. Информация о процессе генерации будет выводиться в окне «Generate Database Schema» («Генерация схемы базы данных») (рис. 35).

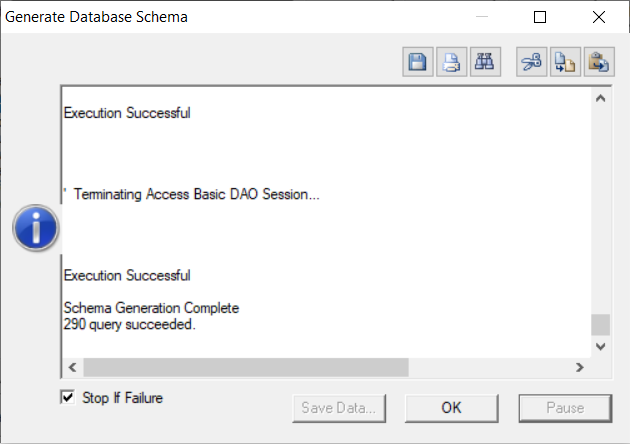


Рис. 35. Окно «Generate Database Schema»

Теперь открываем сам файл БД в Access. На панели вкладок выбираем вкладку «Работа с базами данных», после чего нажимаем на кнопку «Схема данных» (рис. 36). Откроется схема данных (рис. 37).

Для проверки корректности генерации откроем одну из таблиц (например, «Поставщики») (рис. 38) и затем откроем ее же в Конструкторе, чтобы проверить типы данных (рис. 39) и правила валидации (рис. 40).

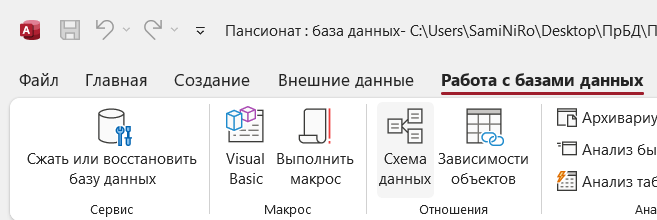


Рис. 36. Вкладка «Работа с базами данных»

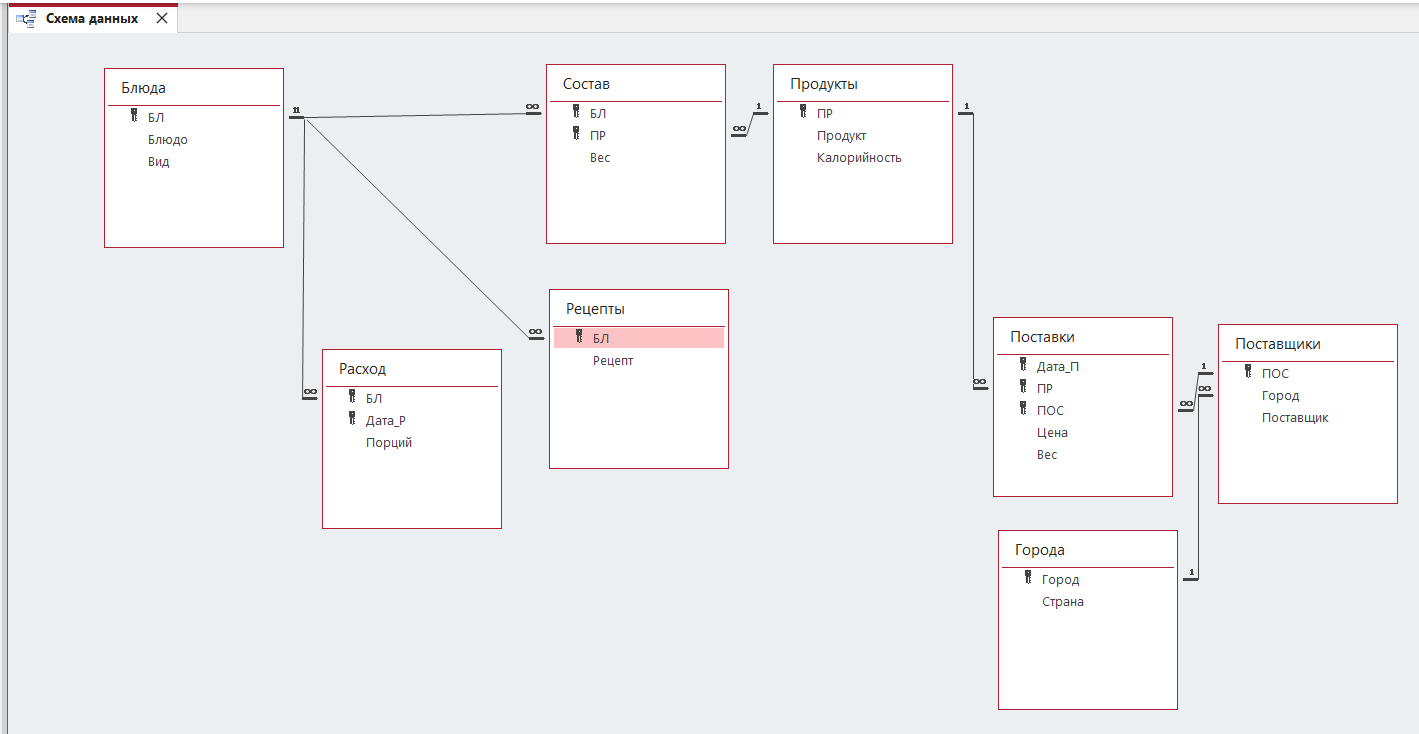


Рис. 37. Схема данных в Access

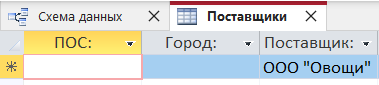


Рис. 38. Таблица «Поставщики» со значением по умолчанию

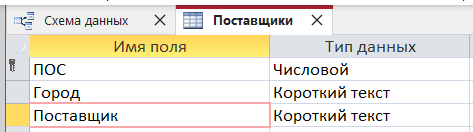


Рис. 39. Таблица «Поставщики» в конструкторе

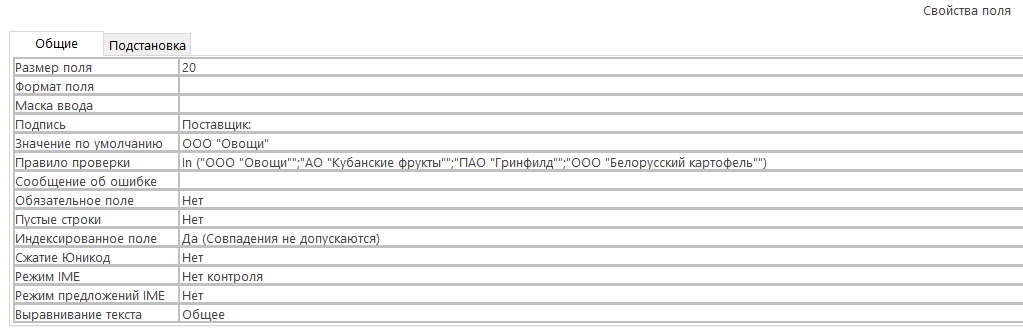


Рис. 40. Свойства поля «Поставщик» со значением по умолчанию и правилом проверки