

## **Лабораторная работа № 7**

### **Тема: РАЗРАБОТКА СТРУКТУРЫ БАЗЫ ДАННЫХ И КОНТРОЛЬНОГО ПРИМЕРА ДЛЯ АРМ ИС**

**Цель:** Закрепить навыки создания структуры базы данных.

**Задание:** Разработать отчет, содержащий структуру базы данных и контрольный пример для АРМ ИС.

Должны быть определены:

- состав таблиц: по каждой таблице – поля, размерность полей, тип полей;
- взаимосвязь таблиц: ключевые атрибуты;
- структура: нарисовать структуру базы данных (рисунок рисовать в Inkscape).

Контрольный пример должен обеспечить проверку функционирования АРМ ИС, в том числе действий, выполняемых пользователями в процессе эксплуатации, и реакции АРМ на действия пользователей. Описание должно соответствовать требованиям стандартов «ГОСТ 19.301–79. ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению» и «ГОСТ 34.603–92. Информационная технология. Виды испытаний автоматизированных систем».

#### **Порядок сдачи лабораторной работы:**

Представить отчёт, содержащий структуру базы данных и контрольный пример для АРМ ИС.

**Общие требования к отчету** указаны в § 1.

#### **Дополнительные требования к отчёту:**

Отчет должен содержать следующую информацию:

- таблицы описаний атрибутов по каждому отношению базы данных, содержащие названия полей, размерность полей, тип полей;
- структуру базы данных в виде рисунка;
- контрольный пример в виде таблицы:

№ п/п	Входные данные	Реакция системы (выходные данные)	Описание проверяемой функциональности системы – что, собственно, проверяется (пункт требований ТЗ)
1.	...	...	...

## **Указания к выполнению работы**

### **1) Простейший алгоритм проектирования базы данных.**

1. Составляем перечень входных данных.

2. Разделяем данные на группы, описывающие конкретные сущности (объекты). Сущность (объект) – нечто целое, некоторый объект, информация о котором используется / обрабатывается в системе неделимо, совокупно. Например, объект «студент» в ИС «Деканат», или «абонент» в ИС учёта абонентов АТС, или «книга» в ИС «Библиотека», или «квартира» в ИС ЖЭУ, или «достопримечательность» в ИС учёта достопримечательностей, или понятие «вид подключения» в ИС учёта абонентов сотовой связи, или понятие «приход» в ИС торговой фирмы и т. д.

3. В каждой группе данных проверяем функциональную зависимость между ними. Те данные, которые находятся во взаимно однозначном соответствии, остаются в этой группе, иначе разносятся в разные группы, то есть, возможно, создаётся новая группа.

4. Каждая уточнённая группа данных формирует таблицу (отношение) базы данных. В каждую таблицу добавляется первичный ключ «с автоматическим увеличением самой СУБД» (с автоинкрементом). То есть, для каждой записи таблицы добавляется выделенное поле порядкового номера этой записи – это поле и будет первичным ключом в этой таблице. Так лучше делать даже в том случае, если одно из полей таблицы само является/формируется как порядковый номер (например, поле NAbonent формируется как порядковый номер абонента по мере появления новых абонентов; однако абоненты не вечны, приходят и уходят, номер NAbonent однажды освобождается и, если вы будете использовать это поле как первичный ключ, то 1) алгоритм обработки таблицы усложнится, 2) возникнут некоторые ограничения, например, вы не сможете легко сменить N абонента, поскольку первичный ключ не может быть изменён).

5. Проверяем связи между таблицами.

5.1. Если таблицы связаны между собой как «1\_к\_1», то они имеют один и тот же первичный ключ.

5.2. Если таблицы связаны как «1\_к\_∞», то первичный ключ таблицы со стороны «1» копируется в таблицу со стороны «∞». И этот ключ становится вторичным ключом в таблице, на стороне «многие».

5.3. Если таблицы связаны как «∞\_к\_∞», то создаётся дополнительная таблица, в которую копируются первичные ключи обеих таблиц. Оба эти поля образуют составной первичный ключ дополнительной таблицы. В

эту таблицу вносятся также данные, относящиеся одновременно к обоим этим отношениям (таблицам).

Примеры структуры базы данных приведены на рис. 3, 4.

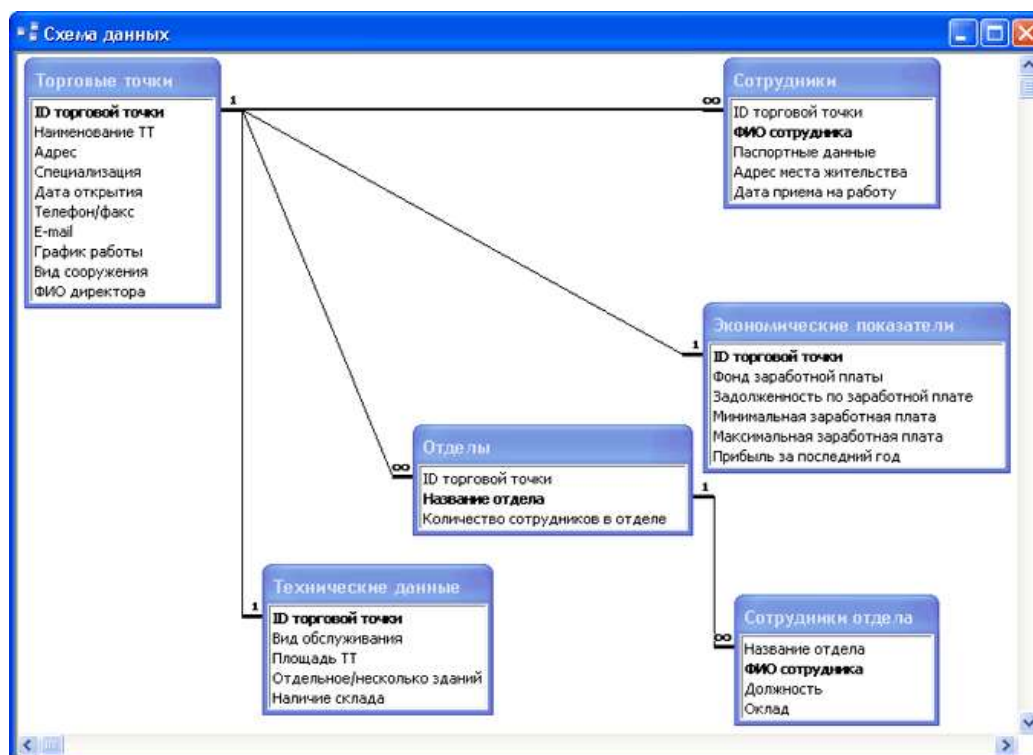


Рис. 3. Пример структуры простой базы данных

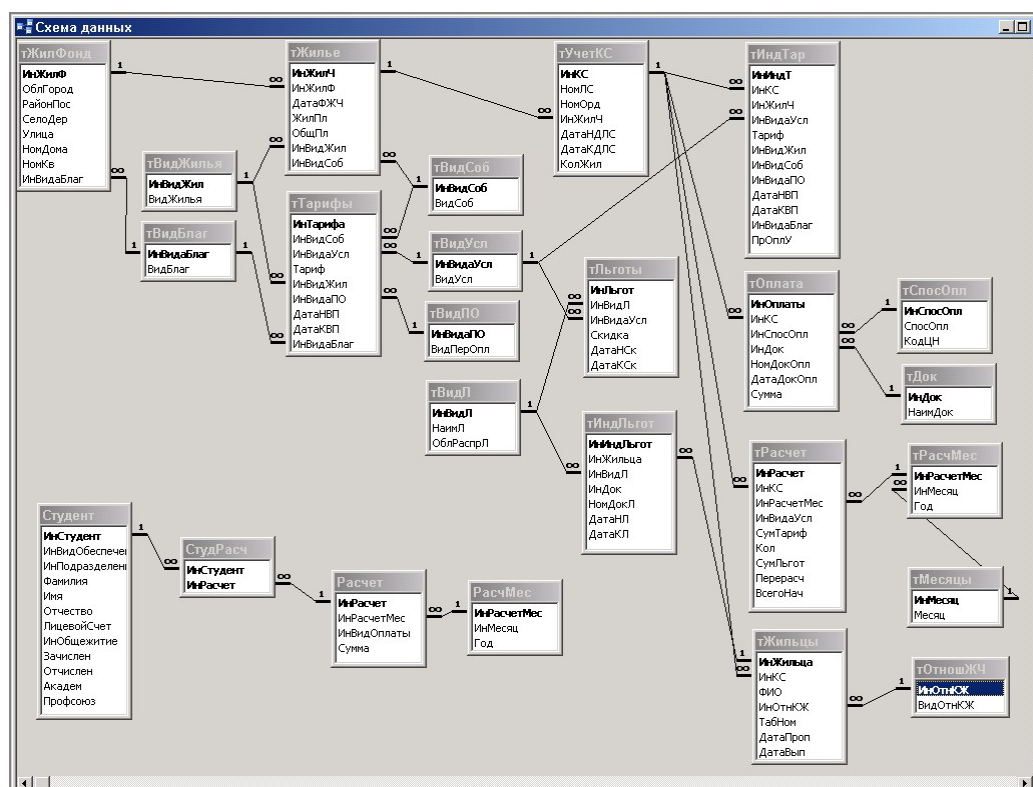


Рис. 4. Пример структуры более сложной базы данных

**2) Контрольный пример** – документ с описанием конкретного теста. Основное требование к контрольному примеру – описание проверки и ожидаемых результатов четко определенной самостоятельной части функциональности (или свойств) программного обеспечения, которое должно быть однозначно понятно вам. Разработка контрольных примеров очень сильно завязана на требования, которые и проверяются описанными в них тестами. Вся суть таких контрольных примеров в том, что их можно потом структурировать, превращая в наборы таблиц контроля (при автоматизации тестирования это будут наборы):

- таблица контроля (Check-list, он же Suite-набор). Это табличный документ, объединяющий в себе набор контрольных примеров с отметками о результате их исполнения и примечаниями;
- отчет о тестировании – результирующий документ, содержащий в себе ссылки на таблицы контроля и выводы о работоспособности релиза с подписями тестера и руководителя проекта.

На каждую сборку системы создаются все указанные документы (кроме, естественно, тест-плана). В таком виде их уже достаточно, чтобы по окончании этапа разработки знать, что вся основная функциональность системы была протестирована, и утверждать, что данная сборка работоспособна.

Контрольный пример должен обеспечить проверку функций задач во всех режимах, а также проверку информационных связей между задачами. Разработчик должен проверить исходные данные, провести анализ результатов выполнения контрольного примера и исправить ошибки, допущенные при доработке программ.

Контрольный пример должен охватывать ошибочные ситуации, которые могут возникнуть при эксплуатации программ в результате некорректности исходной информации. Для этого следует использовать искусственно подобранные ошибочные данные для проверки работоспособности программ.

**Примечание.** Хранилище данных и версионность. На файл-сервере может формироваться общий ресурс, в котором создаются папки на каждый проект. Каждая такая папка содержит следующие элементы:

- файл с тест-планом;
- файл с шаблоном отчета о тестировании;

- каталог TestCase с набором контрольных примеров по данному проекту;
- каталог Builds, в котором в отдельных папках хранятся отработанные контрольные примеры по данной сборке (практически, копия папки TestCase, документы из которой используются в качестве шаблонов) и отчет о тестировании.

Для начала и дальнейшего использования подобной структуры вполне хватит для контроля процесса тестирования. Желательно в папку каждой сборки системы вкладывать файл со списком требований на данную итерацию.