

ДАТАЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БД

1. Подход к даталогическому проектированию и определение состава БД.
2. Метод проектирования реляционной БД на основе ИЛМ.
3. Пример проектирования РБД на основе ИЛМ и ДЛМ.

ВВЕДЕНИЕ

Цель даталогического проектирования (ДЛМ) заключается в *создании модели, которая отображает логические связи между элементами данных безотносительно к их смысловому содержанию и среде хранения.* Эта модель строится в терминах информационных единиц, предусмотренных в конкретной СУБД. Последовательность разработки ДЛМ показана на рис. 1, а расшифровка обозначений приводится на слайде 4.

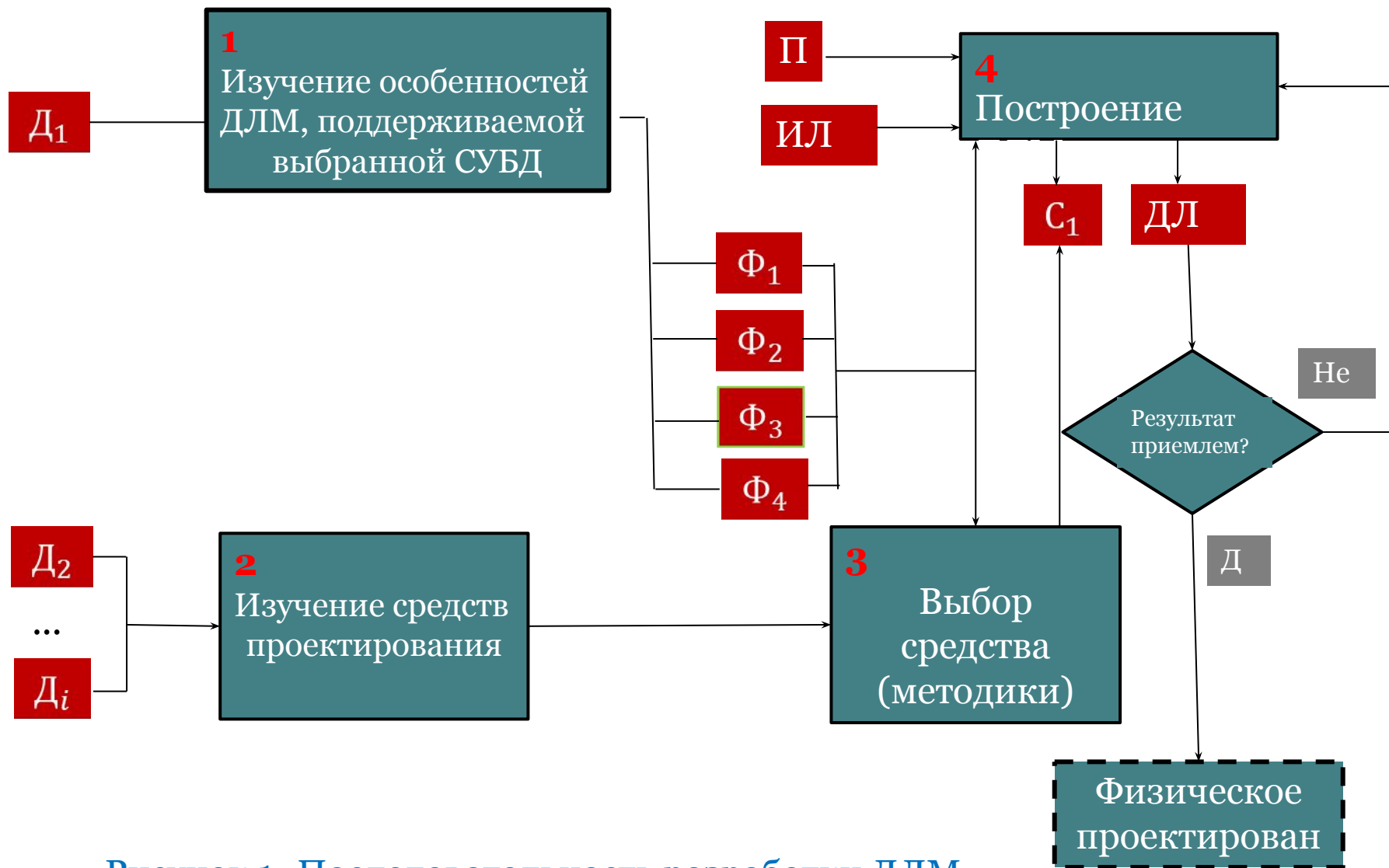


Рисунок 1- Последовательность разработки ДЛМ

Расшифровка обозначений на рисунке 1

- D_1 - документация по СУБД;
- D_2, \dots, D_i - документация по средствам проектирования;
- Π - перечень хранимых показателей (атрибутов);
- ДЛМ - даталогическая модель;
- ИЛМ – инфологическая модель;
- C_1 - средство проектирования;
- Φ_1 - набор допустимых даталогических конструкций;

- Φ_2 - операторы языка описания данных;
- Φ_3 - ограничения, налагаемые СУБД на ДЛМ;
- Φ_4 - возможности физической организации данных.

Здесь Φ_1 , Φ_2 , Φ_3 и Φ_4 - факторы, влияющие на ДЛМ и выбор средств проектирования. Прежде чем приступить к построению ДЛМ, необходимо детально изучить особенности СУБД, определить факторы, влияющие на выбор проектного решения.

На выбор проектных решений самое непосредственное влияние оказывает специфика предметной области, отражённая в ИЛМ. **Результатом** даталогического проектирования является **описание логической структуры БД на языке описания данных**. В спроектированной логической структуре БД должны быть определены все информационные единицы и связи между ними, заданы имена информационных единиц, их тип и количественные характеристики.

1. Подход к даталогическому проектированию и определение состава БД

Для любой предметной области существует множество проектных решений её отображения в ДЛМ. Методика проектирования должна обеспечивать выбор наиболее подходящего проектного решения. Минимальная логическая единица данных для всех СУБД семантически одинакова.

Связи между объектами предметной области, отражённые в ИЛМ, могут отображаться в ДЛМ либо посредством совместного расположения соответствующих им информационных элементов, либо путём объявления связи между ними. Не все виды связей, существующих в предметной области, могут быть непосредственно отображены в конкретной ДЛМ. Так, многие СУБД не поддерживают непосредственно степень связи М:М между объектами.

В этом случае в ДЛМ вводятся дополнительные элементы, отображающие эту связь. Связи могут быть представлены не только посредством структуры БД, но и программным путём. Принимаемое проектное решение зависит не только от специфики отображаемой предметной области, но и от характера обработки информации, хранимой в БД. Так, информацию, используемую часто, и информацию, частота обращения к которой мала, следует хранить в разных файлах.

При переходе от ИЛМ к ДЛМ следует иметь в виду, что ИЛМ включает в себя всю информацию, необходимую и достаточную для проектирования БД. Но это не означает, что все свойства, зафиксированные в ИЛМ, должны в явном виде отражаться в ДЛМ. Прежде, чем строить ДЛМ, необходимо решить, какая информация будет храниться в БД. Например, ИЛМ может содержать вычисляемые показатели, но вовсе не обязательно хранить их в БД.

Один из подходов к определению состава показателей, хранимых в БД, основан на **принципе синтезирования**: в БД должны храниться только исходные показатели, а все производные показатели должны вычисляться в момент выполнения запроса. Этот принцип имеет следующие достоинства: 1) простота и однозначность в принятии решения о том, «что хранить» в БД; 2) отсутствие явного дублирования информации; 3) потенциальная возможность получить любой расчётный показатель.

Несмотря на эти достоинства, в каждом конкретном случае нужно оценивать целесообразность хранения вычисляемых показателей в БД с учётом частоты использования этих показателей.

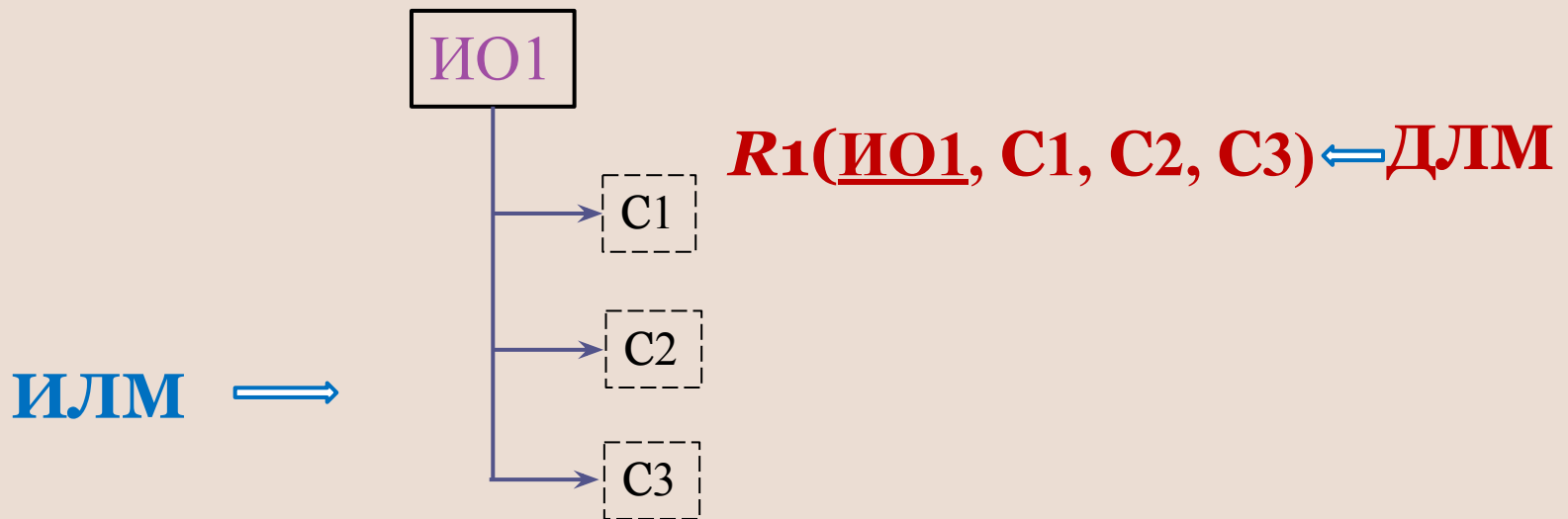
При отображении объекта в БД идентификатор объекта будет атрибутом, который в большинстве случаев используется для однозначной идентификации объекта. Однако может появиться необходимость введения искусственных идентификаторов или кодов. Это нужно, когда:

- 1) в предметной области наблюдается омо-
нимия, например, среди студентов могут быть
полные тёзки. В этом случае для обеспечения
однозначной идентификации объектов необ-
ходимо использовать коды;
- 2) если объект участвует во многих связях, то для
идентификации связи удобнее исполь-
зовать короткий код объекта, нежели длинный
идентификатор объекта;
- 3) если естественный идентификатор может со
временем изменяться, то при отсутствии кода
это может вызвать проблемы с поиском
нужных сведений.

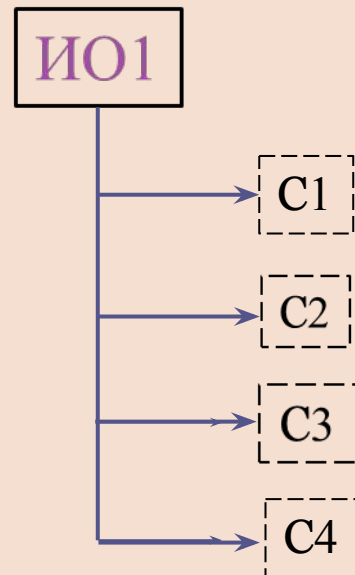
2. Метод проектирования реляционной БД на основе ИЛМ

Рассмотрим метод проектирования реляционной БД, основанной на анализе ИЛМ и переходе от неё к отношениям ДЛМ. Этот метод является достаточно простым и наглядным и в то же время даёт хорошие результаты. Для перехода от ИЛМ к реляционной ДЛМ используются следующие **правила**, в которых идентификаторы объектов обозначены через ИО1, ИО2, ИО3, а их свойства – через С1, С2 и т.д.

2.1. Для каждого простого объекта и его **единичных** свойств строится отношение, атрибутами которого являются идентификатор объекта и реквизиты, соответствующие каждому из единичных свойств:



2.2. Если у объекта имеются **множественные** свойства, то каждому из них ставится в соответствие отдельное отношение. Ключом этого отношения будет идентификатор соответствующего объекта и реквизит, отражающий множественное свойство:



$R_1(\underline{\text{ИО1}}, C1, C2)$

$R_2(\underline{\text{ИО1}}, C3)$

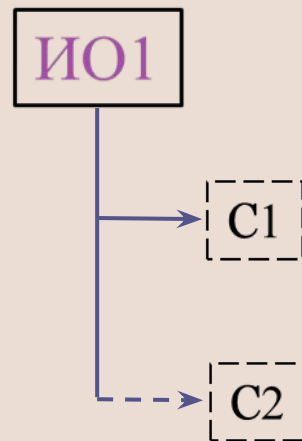
$R_3(\underline{\text{ИО1}}, C4)$

2.3. Если между объектом и его свойством имеется **условная связь** (условное свойство), то при отображении в ДЛМ возможны следующие варианты:

а) если **многие** из объектов обладают **условным свойством**, то его можно хранить в БД так же, как и обычное свойство;

б) если только **незначительное** число объектов обладает **условным свойством**, то можно выделить отношение, которое будет включать идентификатор объекта и атрибут, соответствующий условному свойству.

Это отношение будет содержать столько строк, сколько объектов имеют условное свойство:



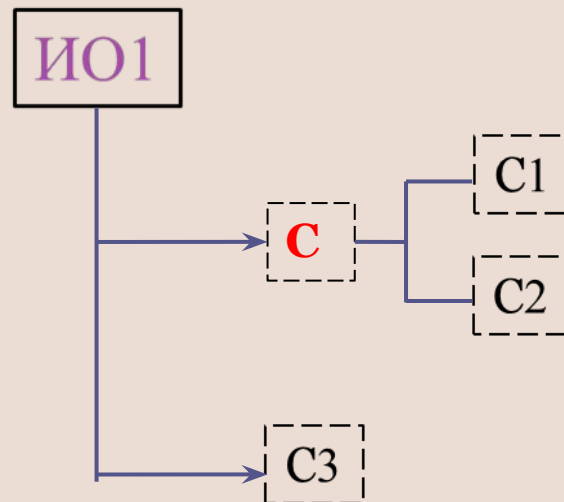
$R_1(\underline{\text{ИО1}}, \text{С1}, \text{С2})$

или

$R_1(\underline{\text{ИО1}}, \text{С1})$

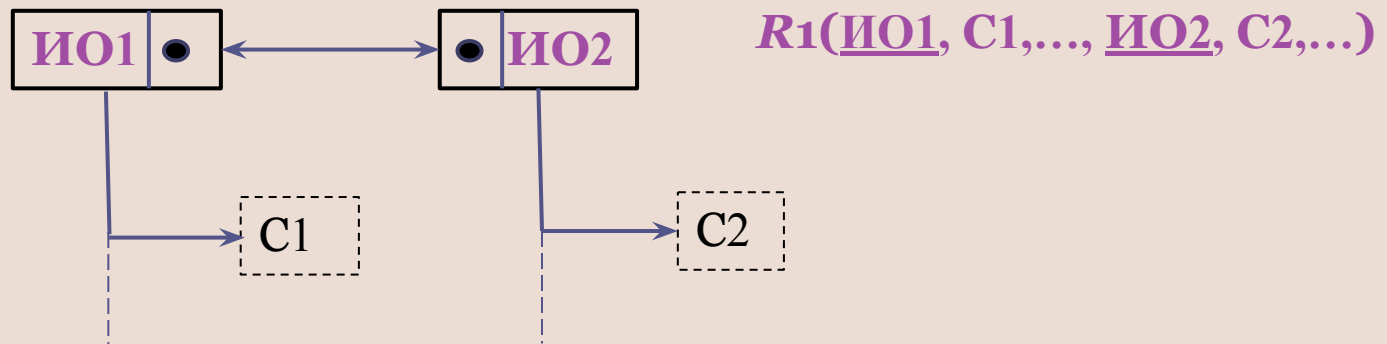
$R_1(\underline{\text{ИО1}}, \text{С2})$

2.4. Если объект имеет **составное свойство С**, то оно представляется в отношении набором атрибутов С1, С2, соответствующих элементам составного свойства:



$R_1(\underline{ИО1}, C1, C2, C3)$

2.5. Если между объектами имеется **степень связи 1:1**, то ДЛМ определяется классом принадлежности объектов. Если класс принадлежности обоих объектов является **обязательным**, то ДЛМ задаётся одним отношением, в котором атрибутами будут идентификаторы объектов и свойства обоих объектов:

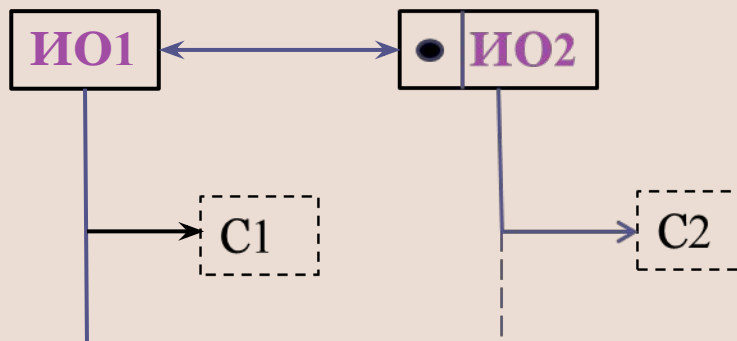


Такая модель потребует меньше всего памяти. Однако если в запросах часто требуется информация отдельно по каждому из объектов, то для ускорения поиска информации целесообразно каждый объект представить отдельным отношением, а связь объектов представить в ДЛМ, указав идентификатор одного объекта в качестве атрибута в отношении, соответствующем другому объекту, например:

$R_1(\underline{ИО1}, C_1, \dots, \underline{ИО2})$

$R_1(\underline{ИО2}, C_2)$

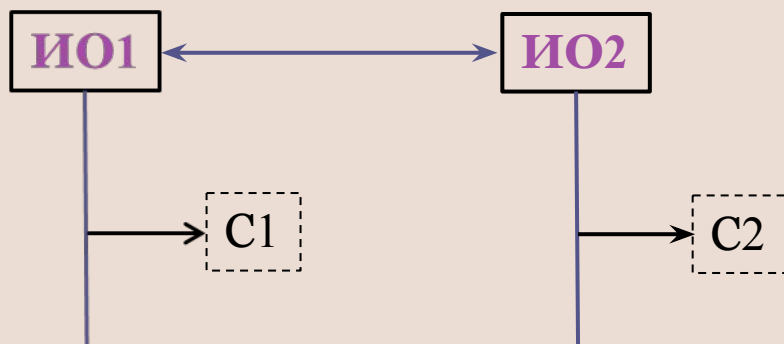
Если класс принадлежности одного из объектов *необязательный*, то ДЛМ задаётся двумя отношениями, причём идентификатор объекта, для которого класс принадлежности является *необязательным*, добавляется в качестве атрибута в отношение, соответствующее объекту с *обязательным* классом принадлежности:



$R_1(\underline{\text{ИО1}}, C1, \dots)$

$R_2(\underline{\text{ИО2}}, C2, \dots, \text{ИО1})$

Если класс принадлежности обоих объектов является **необязательным**, то ДЛМ задаётся тремя отношениями - по одному для каждого объекта и одно для отображения связи между объектами:



$R_1(\underline{ИО1}, C1, \dots)$

$R_2(\underline{ИО2}, C2, \dots)$

$R_3(\underline{ИО1}, \underline{ИО2})$

2.6. Если между объектами имеется **степень связи 1:М** и класс принадлежности **много-связного** объекта является **обязательным**, то независимо от класса принадлежности односвязного объекта ДЛМ задаётся **двумя отношениями** – по одному для каждого объекта. Отношение, соответствующее много-связному объекту можно дополнить атрибутом, являющимся идентификатором (ключом) односвязного объекта:

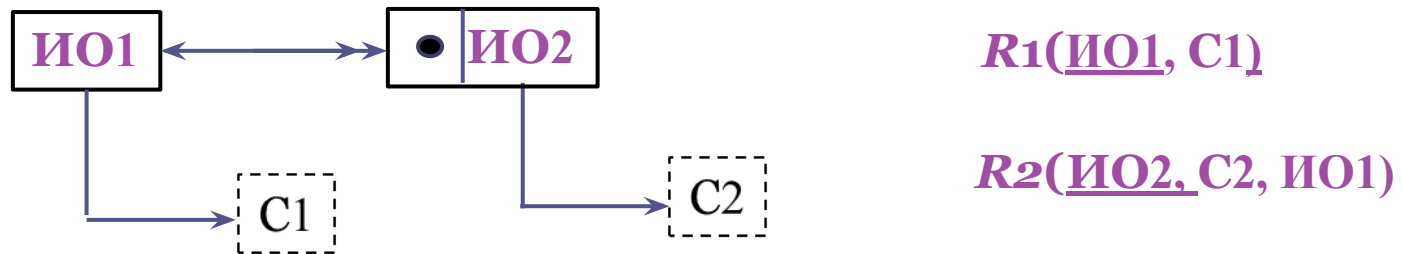
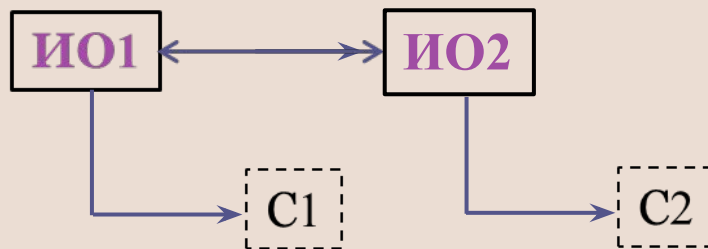


Рисунок к п. 2.6, когда степень связи имеет вид 1:M, а класс принадлежности многосвязного объекта является обязательным

Если класс принадлежности **многосвязного** объекта является **необязательным**, то ДЛМ задаётся тремя отношениями – по одному для каждого объекта и одно для отображения связи между объектами. Связь объектов задаётся атрибутами, являющимися идентификаторами (ключами) объектов:

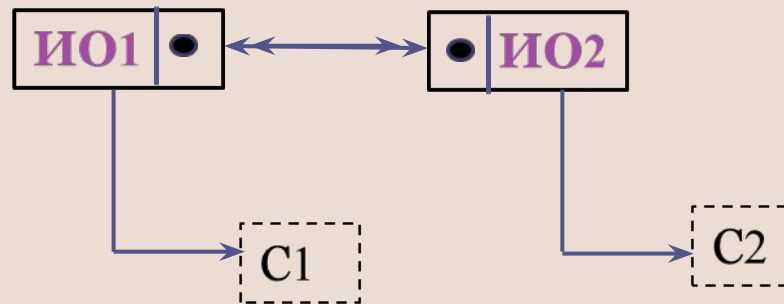


$R_1(\underline{\text{ИО1}}, C_1)$

$R_2(\underline{\text{ИО2}}, C_2)$

$R_3(\underline{\text{ИО1}}, \underline{\text{ИО2}})$

2.7. Если между объектами предметной области имеется **степень связи М:М**, то ДЛМ задаётся тремя отношениями независимо от класса принадлежности объектов:

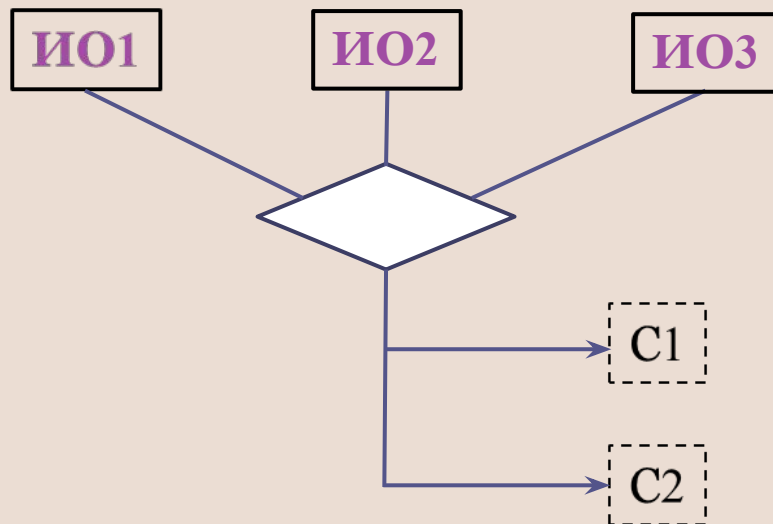


$R_1(\underline{IO1}, C1)$

$R_2(\underline{IO2}, C2)$

$R_3(\underline{IO1}, \underline{IO2})$

2.8. Каждому агрегированному объекту, имеющемуся в предметной области, в ДЛМ соответствует отдельное отношение. Атрибутами этого отношения будут идентификаторы всех объектов, входящих в агрегированный объект, а также свойства этого объекта:



$R_1(\underline{ИО1}, \underline{ИО2}, \underline{ИО3}, C1, C2)$

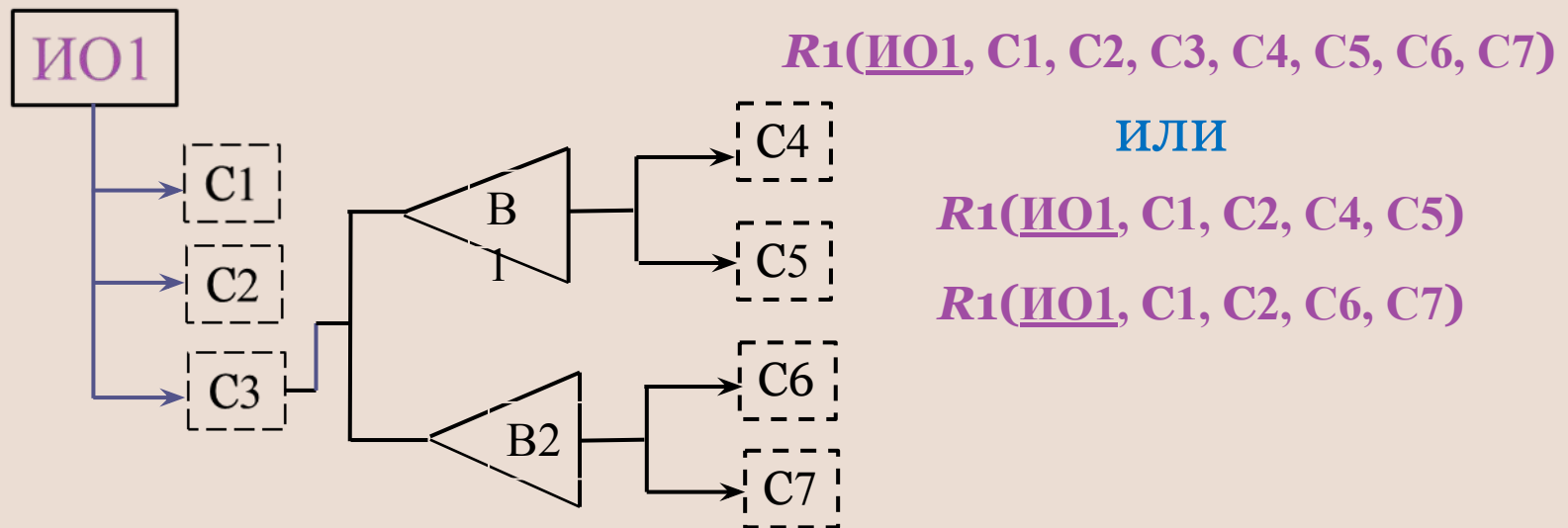
2.9. При отображении **обобщённых** объектов возможны разные решения.

Во-первых, всему обобщённому объекту может быть поставлено в соответствие одно отношение.

Во-вторых, каждой категории объектов **нижнего уровня** ставится в соответствие отдельное отношение.

В первом случае атрибутами отношения будут все единичные свойства, присущие объектам хотя бы одной категории плюс идентификатор объекта.

Во втором случае каждое отношение включает в себя идентификатор объекта, те свойства, которые присущи объектам данной категории, а также свойства, которыми обладают родовые объекты, стоящие выше его по иерархии:



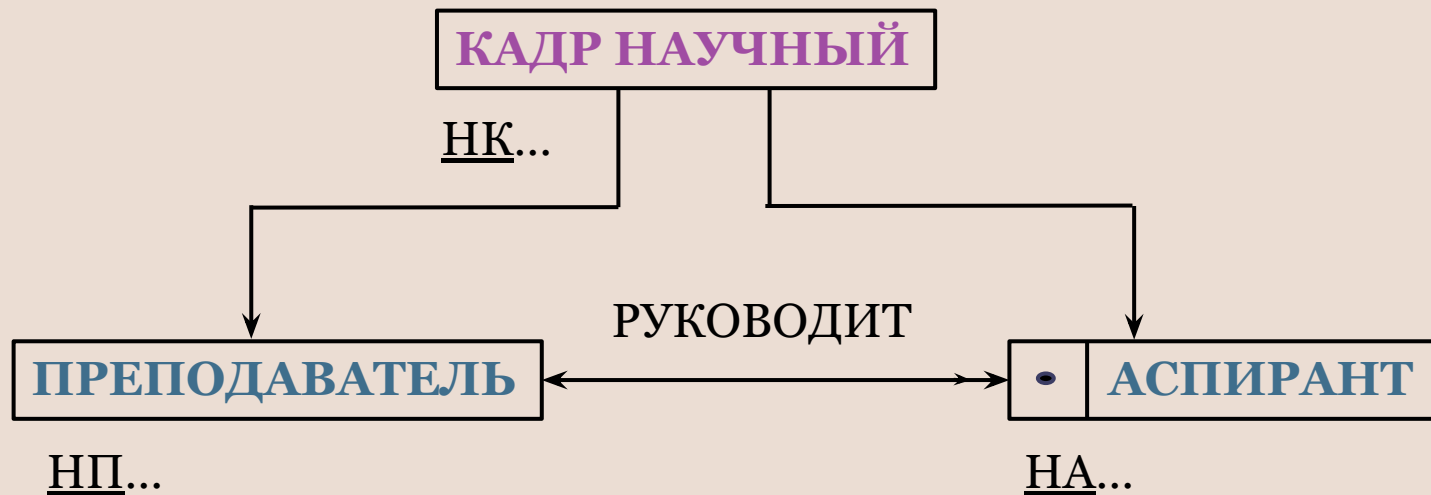
Кроме рассмотренных двух случаев возможны и комбинированные варианты.

2.10. Составной объект, для которого характерно наличие связи **«целое-часть»**, может быть отображён в ДЛМ по-разному. Если рассматриваемая связь реализована на однородном множестве объектов, то для отображения связи в этом случае можно использовать два отношения. Первое из них будет содержать информацию о самих объектах, а второе – информацию о связи между ними, а также дополнительную информацию, характеризующую эту связь. В большинстве случаев для построения ДЛМ можно использовать рекомендации **правила 6**.

2.11. В некоторых случаях одних объектов (сущностей) и связей может оказаться недостаточно для всестороннего моделирования предметной области. Один из таких случаев возникает тогда, когда экземпляры некоторой сущности должны играть разные роли в деятельности организации. Например, предположим, что для кафедры института необходимо хранить информацию о процессе подготовки научных кадров. Различают две категории объектов в этом процессе: преподаватели и аспиранты.

И те и другие являются научными кадрами, но играют разные роли в процессе подготовки научных кадров: преподаватели передают свои знания, а аспиранты приобретают эти знания.

С учётом указанных ролей диаграмма ER-типа будет иметь следующий вид:



При разработке ДЛМ можно следовать такому правилу. Исходная сущность представляется одним отношением, причём ключ сущности служит первичным ключом. Ролевые объекты и связи, их соединяющие, представляются в ДЛМ таким числом отношений, которое определяется ранее описанными правилами, причём каждая роль трактуется как обычная сущность. Согласно этому правилу для нашего примера ДЛМ задаётся тремя отношениями:

$R1(\underline{НК}, \dots)$ $R2(\underline{НЦ}, \dots)$ $R3(\underline{НА}, \dots, \underline{НЦ})$

В этих отношениях множества значений атрибутов НП и НА не пересекаются, а множество значений атрибута НК состоит из объединения значений атрибутов НП и НА ($НК = НП \cup НА$). В отношении $R1$ помимо атрибута НК попадают атрибуты, общие для преподавателей и аспирантов, а в отношении $R2$ и $R3$ помимо атрибутов НП и НА соответственно попадают атрибуты, специфичные для преподавателей и аспирантов. Полученная таким способом ДЛМ напоминает результат применения правила 9 для **обобщённого** объекта.

2.12. До сих пор в ИЛМ присутствовали связи только между парами объектов. Но могут возникнуть и такие ситуации, в которых между объектами существуют связи более высокого порядка. Например, рассмотрим ситуацию приёма экзаменов преподавателями в группах, представив её в виде диаграмм **ER-экземпляров** и **ER-типа** (на рисунке 3 следующего слайда введены следующие обозначения: НП – номер преподавателя, ИЭ – идентификатор экзамена, ШГ – шифр группы, М:N, М:L – связи более высокого порядка).

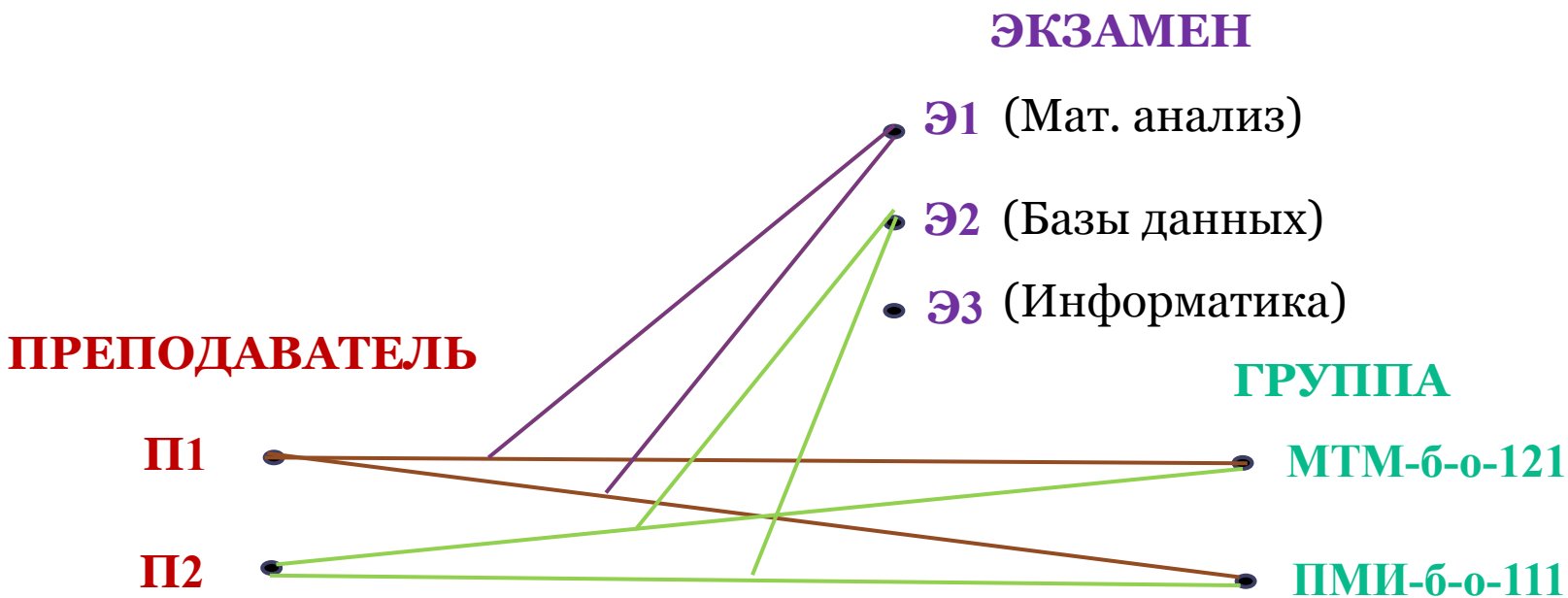


Рисунок 2 - Диаграмма ER-экземпляров

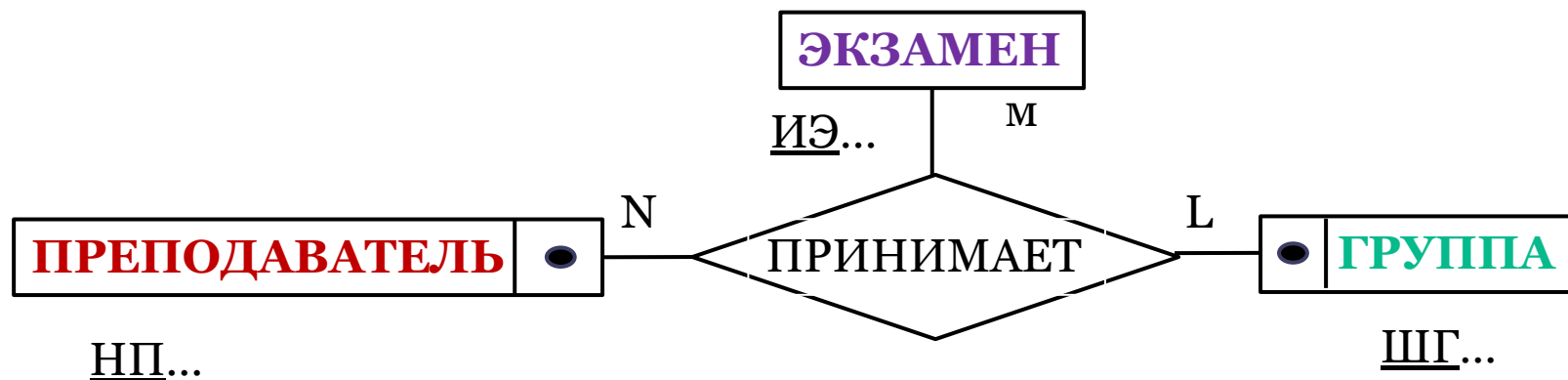


Рисунок 3 - Диаграмма ER-типа

В случае **трёхсторонних связей** в ИЛМ даталогическая модель состоит из четырёх отношений: по одному для каждой сущности, причём ключ каждой сущности должен служить первичным ключом для соответствующего отношения, и одно отношение для связи сущностей. Отношение, порождённое связью, будет иметь среди своих атрибутов ключи сущностей от каждой сущности. Для рассматриваемого примера ДЛМ состоит из таких отношений:

$R1(\underline{НЦ}, \dots)$

$R2(\underline{ИЭ}, \dots)$

$R3(\underline{ШГ}, \dots)$

$R4(\underline{НЦ}, \underline{ИЭ}, \underline{ШГ}, \dots)$

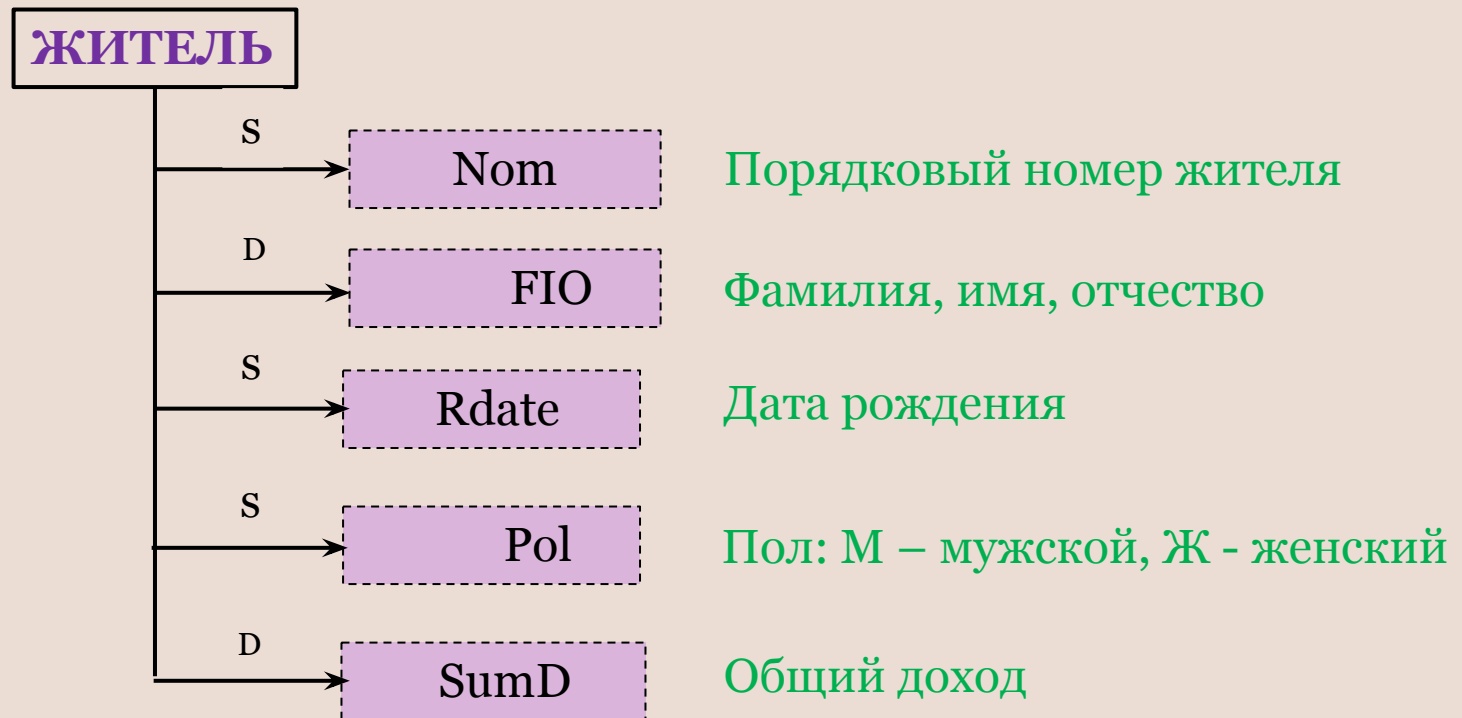
При наличии n -сторонней связи требуется $n+1$ отношение: n отношений для сущностей и одно отношение для связи. Полученная таким образом ДЛМ содержит отношение для связи, которое напоминает результат применения правила 8 для агрегированного объекта. Это не случайно, поскольку ИЛМ отображает один процесс, в который вовлечено несколько объектов, то есть предметную область можно трактовать как агрегированный объект, именуемый ПРИЁМОМ ЭКЗАМЕНОВ.

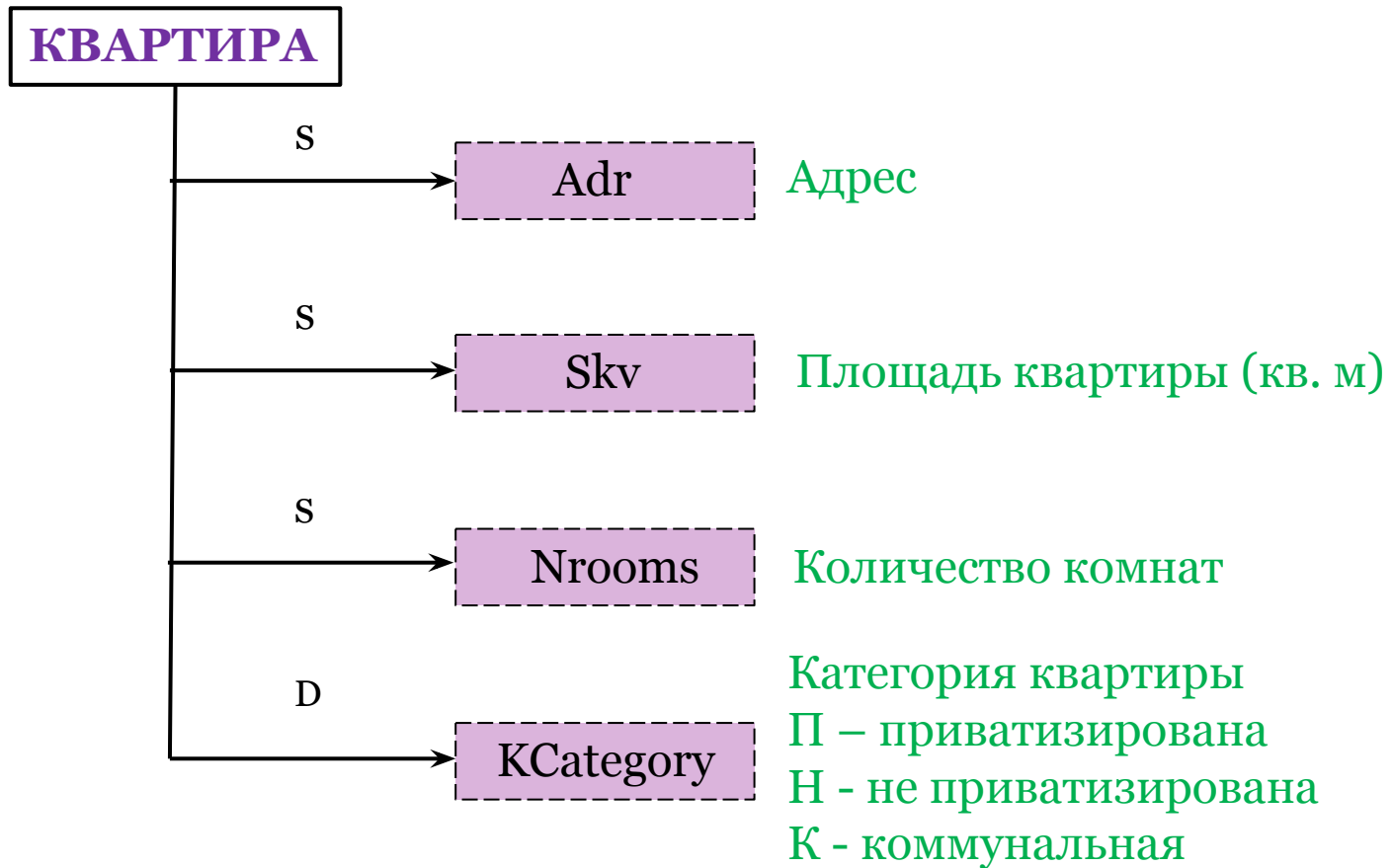
3. ПРИМЕР ПРОЕКТИРОВАНИЯ РБД НА ОСНОВЕ ИЛМ и ДЛМ

Возьмём в качестве предметной области жителей некоторого города и спроектируем простейшую реляционную базу данных, в которой будут храниться сведения, представляющие интерес для налоговой инспекции. Проектирование начинается с разработки ИЛМ, которая включает в себя компоненты, изображённые на рисунке 1 предыдущей темы.

3.1. Описание объектов и связей между ними

Объектами, представляющими интерес, будут ЖИТЕЛЬ, КВАРТИРА, ТЕЛЕФОН, ДОХОД. Опишем каждый из них и связи между ними.





ТЕЛЕФОН

S

Ntel

Номер
телефона

D

TCategory

Категория
телефона:
О – обычный
Д – спаренный
С – сотовый

ДОХОД

S

Id

Идентифи-
катор дохода

S

Source

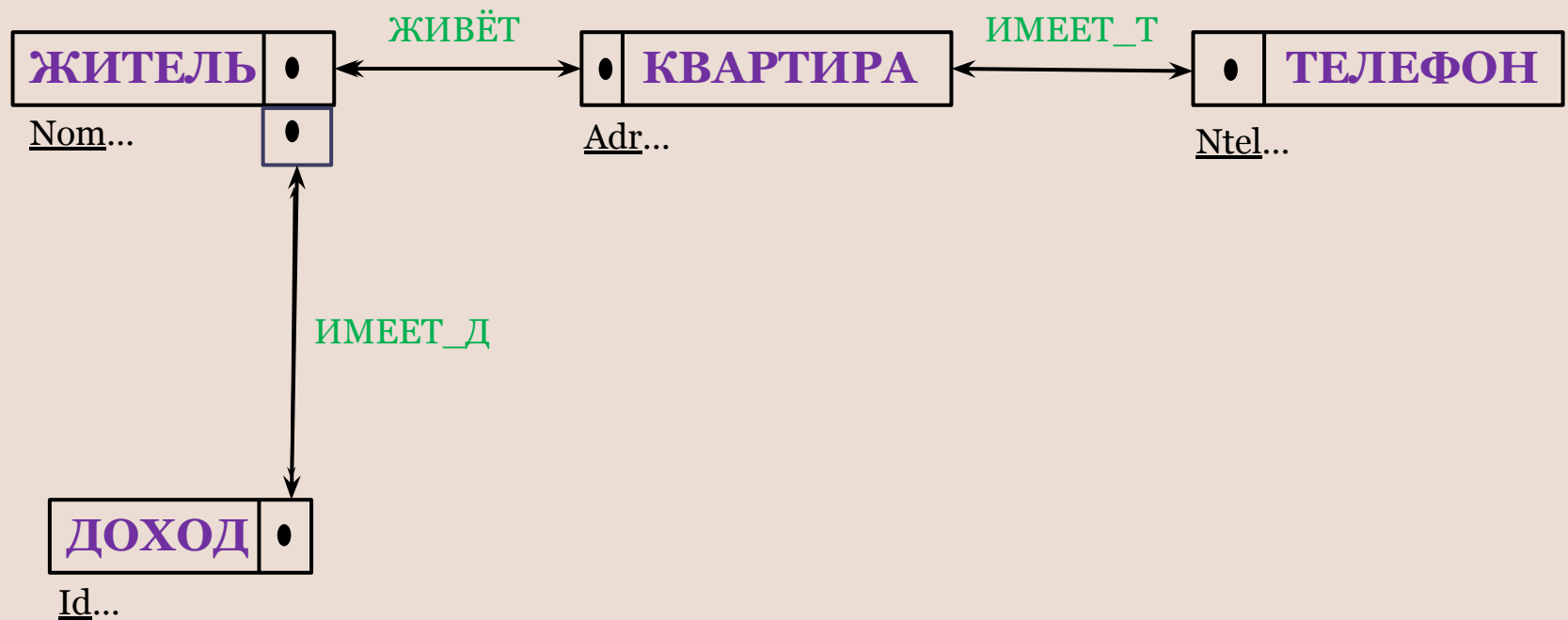
Название
источника
дохода

D

Money

Размер
дохода

Связи между объектами отражаются на диаграмме ER-типа:



3.2. Лингвистические отношения

В лингвистических отношениях должно быть дано толкование используемых в ИЛМ терминов и понятий, например:

Nom – уникальный номер жителя данного города, в качестве которого используется целое число;

Adr – адрес жителя; задаётся указанием улицы, номера дома (корпуса) и квартиры;

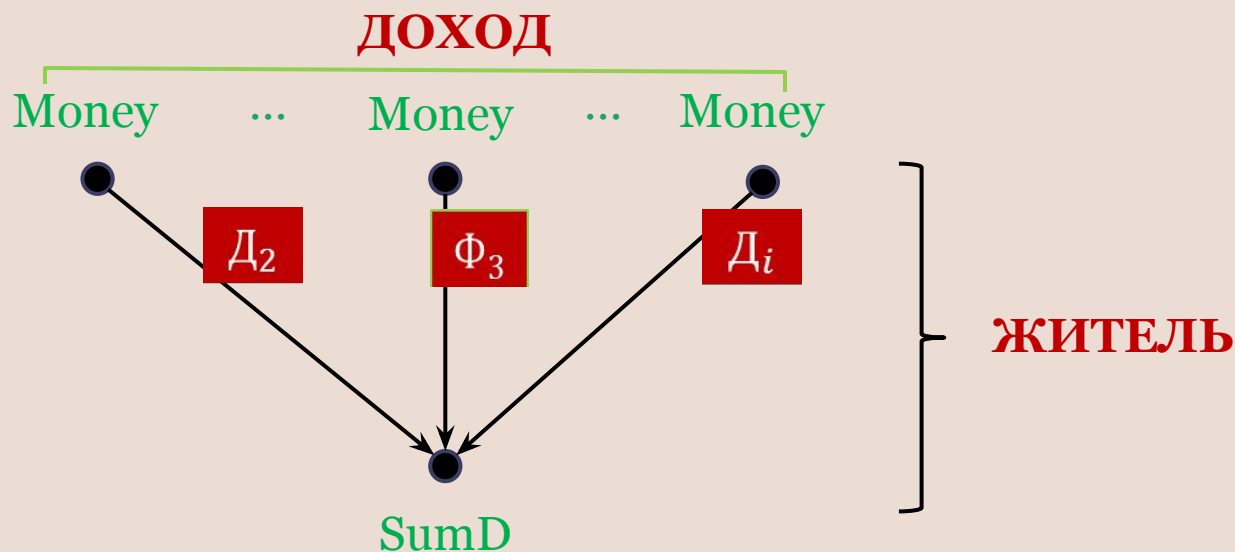
Source – обозначение источника дохода:

Пенсия1 – пенсия по старости; **Пособие1** – пособие на ребёнка; **Стипендия1** – повышенная стипендия; **Работа1** – работа в банке и т.д.

Money – размер дохода в рублях.

3.3. Алгоритмические связи показателей

Из всех показателей, отражённых в ИЛМ, алгоритмически связанным является общий доход (SumD). Его вычисление описывается следующим графом взаимосвязи показателей:



3.4. Описание информационных потребностей пользователей

Здесь должны быть определены все запросы, которые будут поступать от пользователей БД, например:

- 1) вывести список всех жителей с указанием общего дохода;
- 2) вывести список жителей, у которых общий доход не меньше облагаемого минимума;
- 3) подсчитать налоги отдельных жителей и общую сумму налогов и т.д.

Этот компонент служит основой для разработки БД.

3.5. Ограничения целостности

Ограничения целостности используются при разработке БД, чтобы контролировать правильность данных, вводимых в БД, и корректность вычислений. Например, размер дохода не должен быть отрицательным, источник дохода должен выбираться из известного списка и т.п. Таким образом, в данном компоненте ДЛМ формулируются условия, которым должны удовлетворять отдельные показатели и группы показателей, чтобы информация, хранимая в БД, имела смысл.

3.6. Описание структуры БД на основе языка ДЛМ

Будем хранить в этой БД все показатели, в том числе и вычисляемый показатель **SumD**, поскольку его значение требуется при выполнении многих запросов. Введём теперь отношения, включаемые в БД. Для этой цели применим соответствующие правила из **п.2** к каждой паре связанных объектов, изображённых на диаграмме ER-типа (рисунок **п. 3.1**).

Согласно правилу 6 связанные объекты **ЖИТЕЛЬ-КВАРТИРА** представляются двумя отношениями:

PERSON'(Nom,..., Adr)

FLAT'(Adr, ...)

Согласно правилу 7 связанные объекты **ЖИТЕЛЬ-ДОХОД** представляются тремя отношениями:

PERSON''(Nom,...)

PROFIT''(Id, ...)

HAVE_D''(Nom, Id)

Согласно правилу 5 связанные объекты **КВАРТИРА-ТЕЛЕФОН** представляются двумя отношениями:

FLAT'''(Adr, ...)

TFONE'''(Ntel, ..., Adr)

Анализ этих семи отношений позволяет установить, что для отображения ИЛМ предметной области в ДЛМ достаточно пяти отношений:

PERSON(Nom, Fio, Rdate, Pol, SumD, Adr)

FLAT(Adr, Skv, Nrooms, Kcategory)

HAVE_D(Nom, Id)

TFONE(Ntel, Tcategory, Adr)

PROFIT(Id, Sourse, Money)

Даталогическое проектирование завершается *описанием логической структуры БД на языке конкретной СУБД*. Это описание называется **схемой БД** и помимо всего прочего содержит такие характеристики атрибутов отношений как тип и длина (размер) атрибута.

Ввод информации в БД и получение нужной информации из БД осуществляются либо непосредственно средствами СУБД, либо с помощью специально разработанной прикладной системы, использующей команды СУБД.

Можно легко проверить, что спроектированная РБД не создаёт проблем вставки, обновления и удаления информации при работе с ней, то есть аномалий вставки, обновления и удаления не возникает.

Пусть, например, появился новый житель, у которого отсутствуют источники дохода. Тогда информация о жителе будет занесена только в отношении **PERSON**, причём даже атрибут **SumD** будет иметь определённое (нулевое) значение, соответствующее действительности и *аномалии вставки не возникает*.

Аномалии обновления не возникает при изменении адреса у конкретного жителя, поскольку в этом случае будет обновляться атрибут **Adr** в кортеже отношения **PERSON** и, возможно, появится новый кортеж с описанием адреса и характеристик квартиры в отношении **FLAT**, если в БД не было сведений об этой квартире.

Не проявляется аномалия обновления и при изменении номера телефона, установленного в квартире. Если об этом сообщит любой из жителей такой квартиры, то изменения отразятся только в одном кортеже отношения **TFONE** (обновится только номер телефона).

Аномалия удаления. Пусть налоговый инспектор узнаёт, что житель Иванов П.Р. лишился своего источника дохода, условно названного РАБОТА2, и удаляет этот кортеж из отношения **HAVE_D**. После этого сведения о существовании Иванова остаются в отношении **PERSON** без изменений и аномалии удаления не возникает.

ЗАКЛЮЧЕНИЕ

Попытаемся свести в некоторый перечень основные действия, которые обычно приходится выполнять в процессе проектирования реляционной БД с использованием метода «**сущность-связь**» при нормализации проектного решения до третьей нормальной формы.

- детально изучить предметную область в целях составления её словесного описания и конкретизации необходимых бизнес-правил. Сформулировать функциональные требования к БД, выявив основные запросы и транзакции;

- ❑ выявить объекты (сущности), которые необходимо включить в модель, и определить первичные ключи каждого из выявленных объектов;
- ❑ выявить и описать смысл, тип и возможные значения всех представляющих интерес атрибутов для каждого объекта. Определить домены (области значений) для каждого атрибута;
- ❑ определить типы связей между выявленными объектами;
- ❑ построить графическое описание логической модели предметной области в виде ER-диаграмм с указанием степеней связи и классов принадлежности;

- ❑ построить графическое описание функциональных зависимостей между атрибутами и выявить минимальное покрытие;
- ❑ для построенной диаграммы ER-типа получить предварительные отношения и разместить в них выявленные атрибуты объектов;
- ❑ выполнить нормализацию модели, используя минимальное покрытие;
- ❑ принять обоснованные решения по введению в модель контролируемой избыточности и способов её контроля;
- ❑ получить окончательное проектное решение в виде наборов отношений с присвоенными им

смысловыми именами, указанными ключами и перечнем атрибутов для каждого отношения. Проверить, все ли атрибуты нашли своё место;

- убедиться в возможности реализации необходимой функциональности БД на полученном проектном решении.

Разумеется, на практике эти пункты выполняются не в жёсткой последовательности. Опытные проектировщики обычно подразумевают наличие диаграммы ER-типа или концептуальной модели и зачастую ограничиваются изображением логической модели БД в некоторой автоматизированной среде проектирования БД.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. *Дейт, К., Дж.* Ведение в системы баз данных, 7-е издание: пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 1072 с.
2. *Илющечкин, В.М.* Основы использования и проектирования баз данных: учеб. пособие/В. М. Илющечкин. – М.: Высшее образование, 2009. -213 с. – (Основы наук).
3. *Мирошниченко, Г.А.* Реляционные базы данных: практические приёмы оптимальных решений. – СПб.: БХВ-Петербург, 2005. – 400 с.