

V. СЕМАНТИЧЕСКАЯ МЕТОДИКА ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННЫХ СХЕМ БД

СЕМАНТИЧЕСКАЯ МЕТОДИКА ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННЫХ СХЕМ БД

Этапы проектирования

- **Функциональное моделирование предметной области (деловая модель)**
- **Семантическое моделирование данных (ER-модель)**
- **Логическое проектирование данных (реляционная модель)**
- **Физическое проектирование данных**

5.1. Функциональное моделирование предметной области

Функциональные модели и их назначение

- **Structured Analysis and Design Technique (SADT-модель) – моделирование сложных организационных бизнес-процессов, их реинжиниринг**
- **Data Flow Diagrams (DFD-модель) – моделирование функционирования проектируемых информационных систем**
- **Деловая модель – простейшее средство функционального моделирования, достаточное для задачи проектирования схемы БД**

Деловая модель для процесса лечения пациента

<div> <div>КЛАССЫ ДАННЫХ</div> <div>ФУНКЦИИ</div> </div>	ПАЦИЕНТ	БОЛЬНИЦА	ВРАЧ	ЛАБОРАТОРИЯ	АНАЛИЗ	ДИАГНОЗ	ПАЛАТА	ПЕРСОНАЛ
Провести первичный осмотр	X	X	X					
Обследовать больного	X		X	X	X			
Поставить диагноз	X		X			X		
Провести лечение в стационаре	X	X	X			X	X	X

5.2. Семантическое моделирование данных

Семантическое моделирование данных (ER-модель)

Этапы построения ER-схемы предметной области

- 1. Построение подсхем для каждой функции в отдельности:**
 - 1. А. Определение множеств сущностей**
 - 1. Б. Определение множеств связей**
 - 1. В. Определение ограничений целостности**
- 2. Интеграция подсхем в общую ER-схему предметной области**

Семантическое моделирование данных

Этап 1. Построение подсхем для каждой функции в отдельности

1.А. Определение множеств сущностей

- 1) Каким множествам сущностей соответствует каждый класс данных?**
- 2) Каковы семантика и имя каждого множества сущностей?**
- 3) Какие атрибуты этих множеств сущностей представляют интерес с точки зрения функции?**
- 4) Каковы их семантика и имена?**

1.А. Определение множеств сущностей (пример)

Провести осмотр

ПАЦИЕНТ (Регистрационный номер (Р/Н), Фамилия, Адрес, Дата рождения (Д/Р), Пол, Номер медицинского полиса (НМП))

БОЛЬНИЦА (Название, Адрес, Телефон)

ВРАЧ (Фамилия, Специальность)

Обследовать больного

ПАЦИЕНТ (Регистрационный номер (Р/Н), Фамилия, Возраст)

ВРАЧ (Фамилия, Специальность)

ЛАБОРАТОРИЯ (Название, Адрес, Телефон)

АНАЛИЗ (Тип анализа (Т/А), Назначенная дата (Н/Д), Назначенное время (Н/В), Номер направления (Н/Н), Состояние)

Поставить диагноз

ПАЦИЕНТ (Регистрационный номер (Р/Н), Фамилия, Возраст)

ВРАЧ (Фамилия, Специальность)

ДИАГНОЗ (Тип диагноза (Т/Д), Осложнения)

Провести лечение

ПАЦИЕНТ (Регистрационный номер (Р/Н), Фамилия, Возраст, Пол)

БОЛЬНИЦА (Название, Адрес, Телефон, Число коек (Ч/К))

ВРАЧ (Фамилия, Специальность)

ДИАГНОЗ (Тип диагноза (Т/Д), Осложнения, Предупреждающая информация)

ПАЛАТА (Номер палаты (Н/П), Название, Число коек (Ч/К))

ПЕРСОНАЛ (Фамилия, Должность, Смена, Зарплата (З/П))

Семантическое моделирование данных

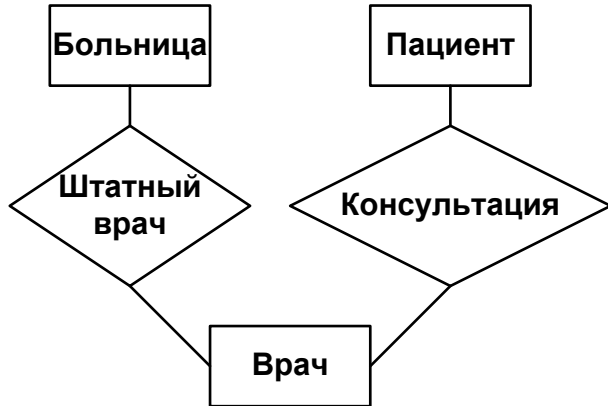
Этап 1. Построение подсхем для каждой функции в отдельности (продолжение)

1.Б. Определение множеств связей

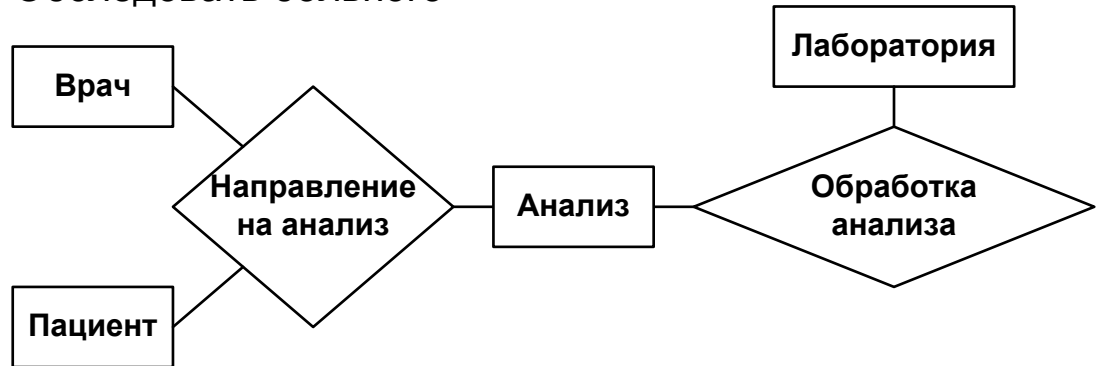
- 1) Какие множества связей между множествами сущностей ассоциируются с функцией?**
- 2) Каковы семантика, имя и степень каждого множества связей?**
- 3) Существуют ли у множеств связей собственные атрибуты, представляющие интерес с точки зрения функции?**
- 4) Если да, то каковы их семантика и имена?**

1.Б. Определение множеств связей (пример)

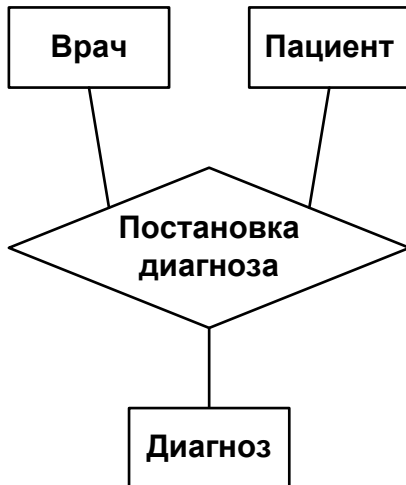
Провести осмотр



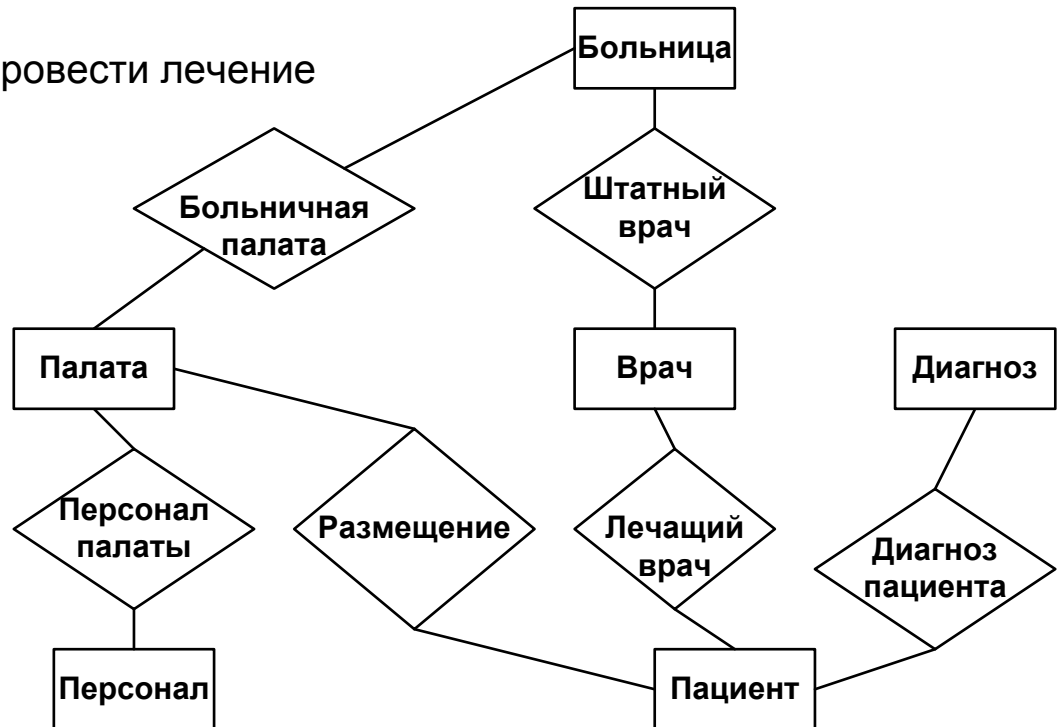
Обследовать больного



Поставить диагноз



Провести лечение



Семантическое моделирование данных

Этап 1. Построение подсхем для каждой функции в отдельности (продолжение)

1.В. Определение ограничений целостности

- 1) Каковы области значений каждого атрибута? Есть ли среди них многозначные атрибуты?**
- 2) Каковы ключи каждого множества сущностей и множества связей? Как можно еще идентифицировать сущности и связи каждого типа?**
- 3) Какие типы отображений соответствуют каждому множеству связей с точки зрения функции?**
- 4) Каковы другие ограничения целостности, которые напрямую не отражаются в ER-модели?
Сформулировать их на естественном языке или языке логики предикатов первого порядка.**

1.В. Определение ограничений целостности (пример)

Провести осмотр

ПАЦИЕНТ (Регистрационный номер (Р/Н), Фамилия, Адрес, Дата рождения (Д/Р), Пол, Номер медицинского полиса (НМП))

БОЛЬНИЦА (Название, Адрес, Телефон(М))

ВРАЧ (Фамилия, Специальность)

Обследовать больного

ПАЦИЕНТ (Регистрационный номер (Р/Н), Фамилия, Возраст)

ВРАЧ (Фамилия, Специальность)

ЛАБОРАТОРИЯ (Название, Адрес, Телефон(М))

АНАЛИЗ (Тип анализа (Т/А), Назначенная дата (Н/Д), Назначенное время (Н/В), Номер направления (Н/Н), Состояние)

Поставить диагноз

ПАЦИЕНТ (Регистрационный номер (Р/Н), Фамилия, Возраст)

ВРАЧ (Фамилия, Специальность)

ДИАГНОЗ (Тип диагноза (Т/Д), Осложнения)

Провести лечение

ПАЦИЕНТ (Регистрационный номер (Р/Н), Фамилия, Возраст, Пол)

БОЛЬНИЦА (Название, Адрес, Телефон(М), Число коек (Ч/К))

ВРАЧ (Фамилия, Специальность)

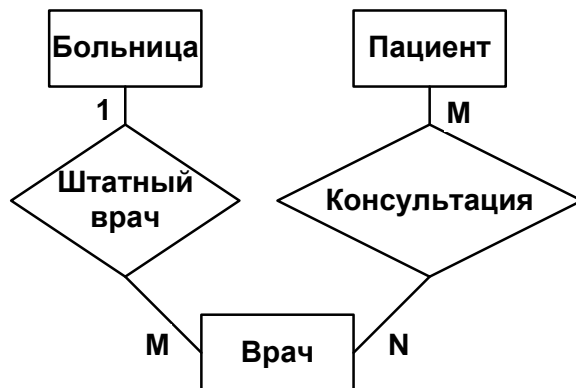
ДИАГНОЗ (Тип диагноза (Т/Д), Осложнения, Предупреждающая информация)

ПАЛАТА (Номер палаты (Н/П), Название, Число коек (Ч/К))

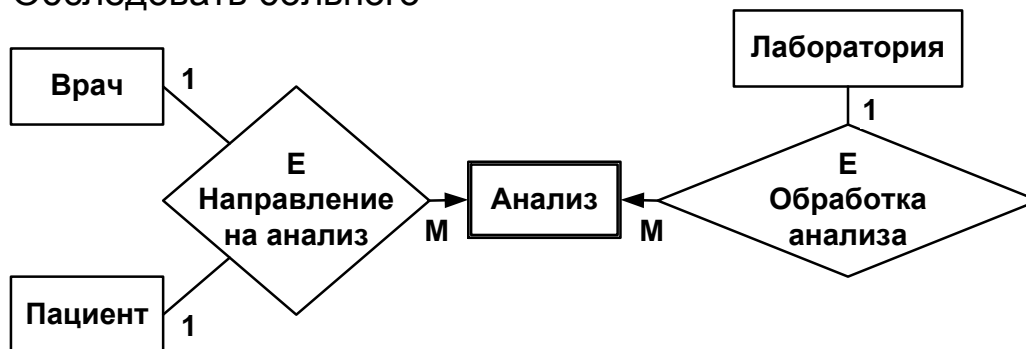
ПЕРСОНАЛ (Фамилия, Должность, Смена, Зарплата (З/П))

1.В. Определение ограничений целостности (пример)

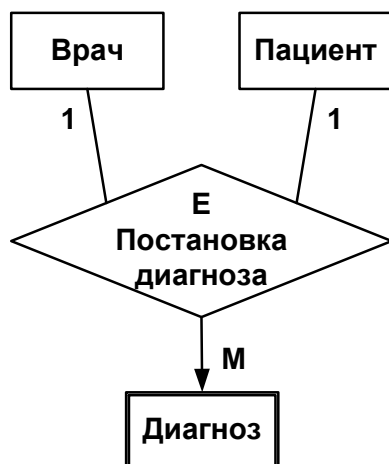
Провести осмотр



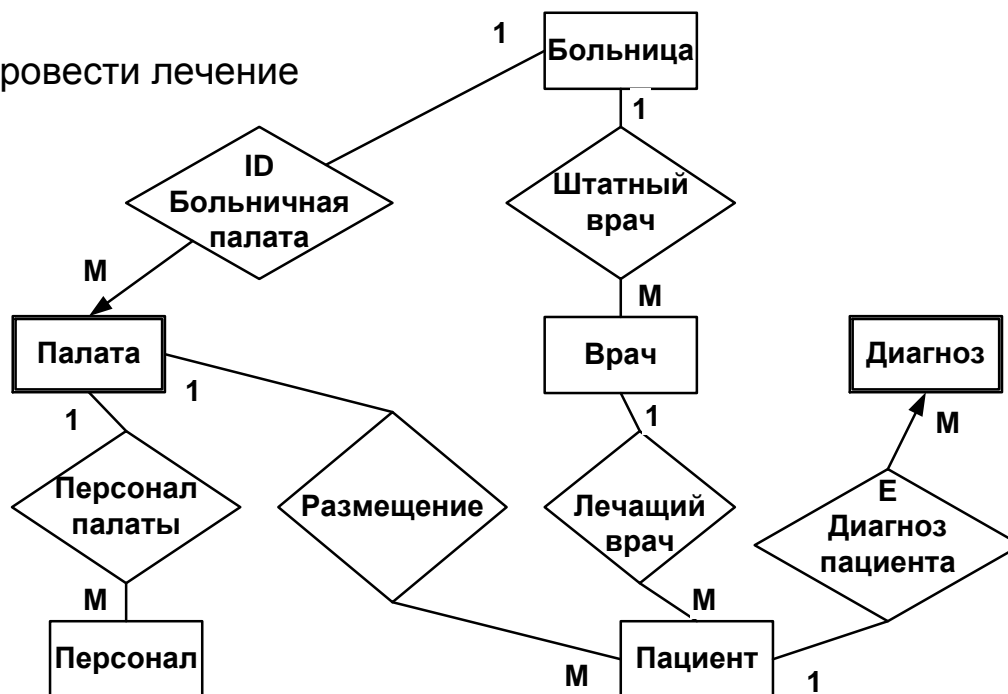
Обследовать больного



Поставить диагноз



Провести лечение

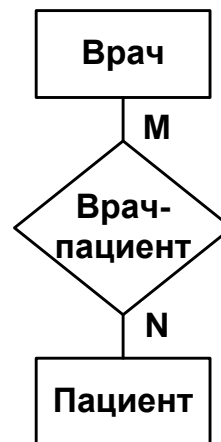
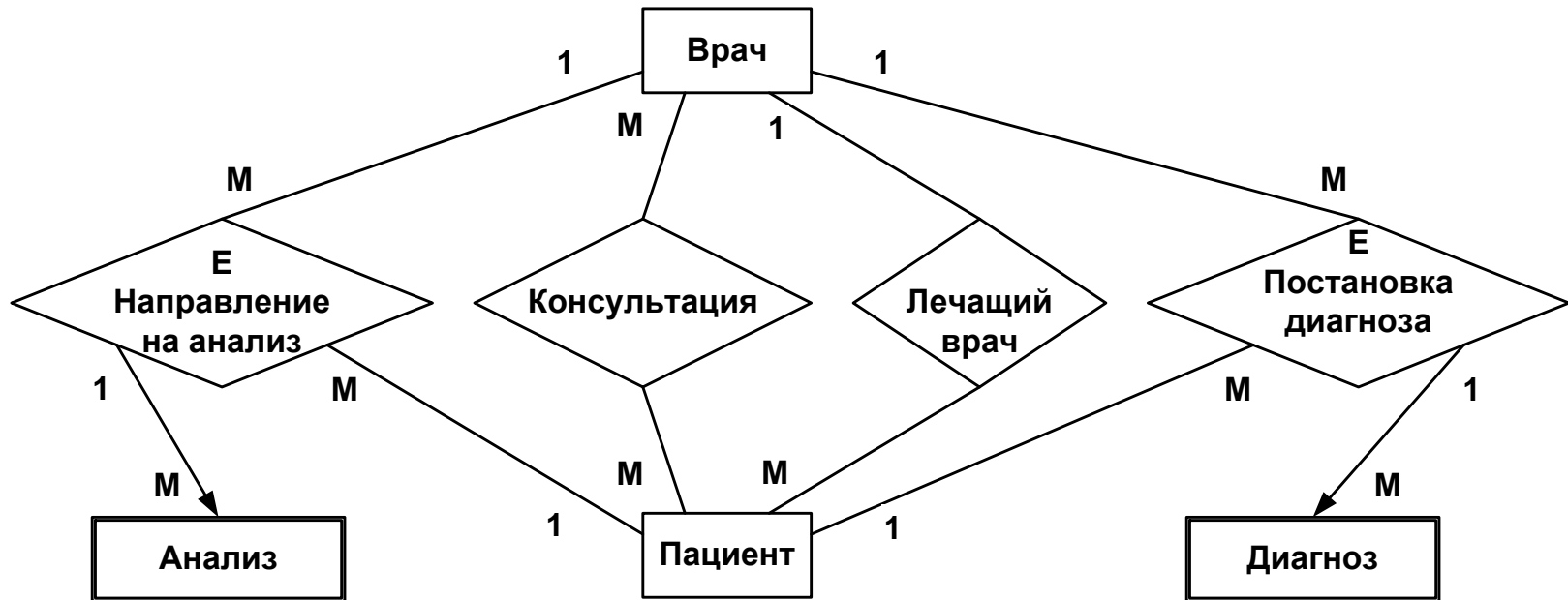


Семантическое моделирование данных

Этап 2. Интеграция подсхем в общую ER-схему предметной области

- 1) Интеграция множеств сущностей методом их семантического объединения**
- 2) Интеграция атрибутов каждого множества сущностей методом их семантического объединения**
- 3) Интеграция ограничений целостности, ассоциированных с каждым множеством сущностей методом их объединения**
- 4) Интеграция множеств связей методом их семантического объединения; возможна генерализация множеств связей**
- 5) Интеграция атрибутов каждого множества связей методом их семантического объединения**
- 6) Повторная тщательная проверка типов отображений, соответствующих каждому множеству связей**
- 7) Интеграция других ограничений целостности, которые напрямую не отражаются в ER-модели, методом их объединения; сформулировать их на естественном языке или языке логики предикатов первого порядка.**

2. Интеграция подсхем в общую ER-схему (генерализация множеств связей)



2. Интеграция подсхем в общую ER-схему (пример)

БОЛЬНИЦА (Название, Адрес, Телефон(М), Число коек (Ч/К))

ПАЛАТА (Номер палаты (Н/П), Название, Число коек (Ч/К))

ПЕРСОНАЛ (Фамилия, Должность, Смена, Зарплата (З/П))

ВРАЧ (Фамилия, Специальность)

ПАЦИЕНТ (Регистрационный номер (Р/Н), Фамилия, Адрес, Дата рождения (Д/Р), Пол, Номер медицинского полиса (НМП), Возраст(вирт.))

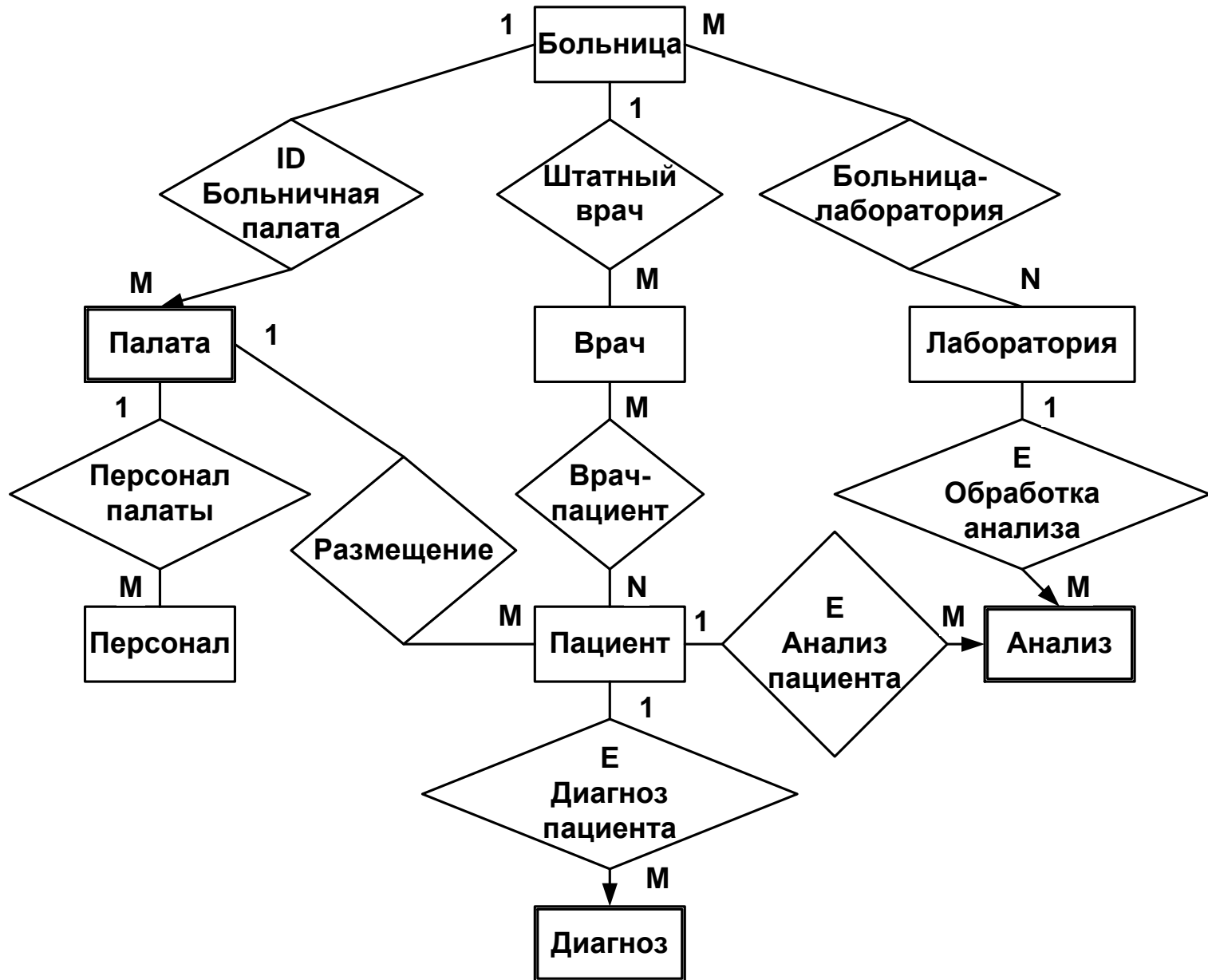
ДИАГНОЗ (Тип диагноза (Т/Д), Осложнения, Предупреждающая информация)

ЛАБОРАТОРИЯ (Название, Адрес, Телефон(М))

АНАЛИЗ (Тип анализа (Т/А), Назначенная дата (Н/Д), Назначенное время (Н/В), Номер направления (Н/Н), Состояние)

РАЗМЕЩЕНИЕ (Номер койки (Н/К))

2. Интеграция подсхем в общую ER-схему (пример)



5.3. Логическое проектирование данных

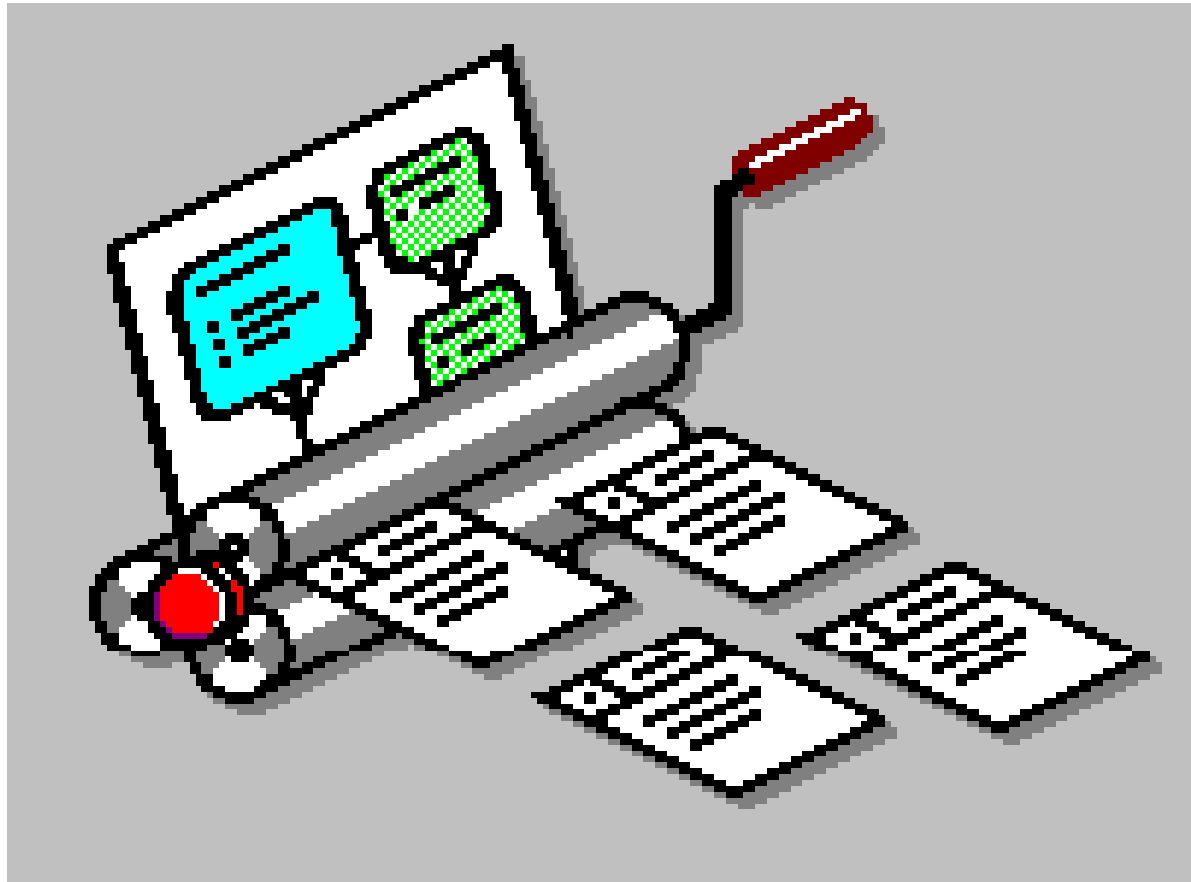
Логическое проектирование данных (реляционная модель)

Этапы логического проектирования данных

- 1. Трансформация ER-схемы (ERM-схемы) в реляционную схему с помощью соответствующих правил.**
- 2. Проверка отношений полученной реляционной схемы на выполнение условий требуемых нормальных форм и их дальнейшая нормализация (см. п. 4.2.5.5).**
- 3. В необходимых случаях (неэффективность выполнения запросов к БД) введение контролируемой избыточности данных (денормализация схемы), которой сопутствуют меры, исключющие возникновение аномалий.**

Логическое проектирование данных

Этап 1. Трансформация ER-схемы в реляционную схему с помощью соответствующих правил



Простейшие правила перехода от ER-схемы к реляционной схеме БД

- 1. Каждое множество сущностей представляется самостоятельным отношением, однозначные атрибуты множества сущностей становятся атрибутами отношения, ключи множества сущностей являются возможными ключами отношения; при необходимости в качестве первичного ключа отношения используется суррогатный атрибут.**
- 2. Бинарные множества связей типа 1:M без атрибутов представляются дублированием первичного ключа 1-отношения в M-отношение.**
- 3. Бинарные множества связей типа M:N без атрибутов представляются самостоятельными отношениями, куда дублируются первичные ключи отношений, построенных для множеств сущностей.**
- 4. Множества связей с атрибутами представляются самостоятельными отношениями, куда дублируются первичные ключи отношений, построенных для множеств сущностей. Однозначные атрибуты множества связей становятся атрибутами этого отношения.**

Простейшие правила перехода от ER-схемы к реляционной схеме БД (продолжение)

- 5. Множества связей степени больше 2-х представляются самостоятельными отношениями, куда дублируются первичные ключи отношений, построенных для множеств сущностей. Однозначные атрибуты множества связей становятся атрибутами этого отношения.**
- 6. Каждый многозначный атрибут множества сущностей представляется отдельным отношением, куда дублируется первичный ключ отношения, построенного для множества сущностей; второй атрибут этого отношения предназначен собственно для значения.**
- 7. Каждый многозначный атрибут множества связей представляется отдельным отношением, куда дублируется первичный ключ отношения, построенного для множества связей; второй атрибут этого отношения предназначен собственно для значения.**

Реляционная схема медицинской БД

БОЛЬНИЦА (Код больницы (К/Б), Название, Адрес, Число коек (Ч/К))

ПАЛАТА (Код больницы (К/Б), Номер палаты (Н/П), Название, Число коек (Ч/К))

ПЕРСОНАЛ (Код больницы (К/Б), Номер палаты (Н/П), Фамилия, Должность, Смена, Зарплата (З/П))

ВРАЧ (Код врача (К/В), Код больницы (К/Б), Фамилия, Специальность)

ПАЦИЕНТ (Регистрационный номер (Р/Н), Фамилия, Адрес, Дата рождения (Д/Р), Пол, Номер медицинского полиса (НМП))

ДИАГНОЗ (Регистрационный номер (Р/Н), Тип диагноза (Т/Д), Осложнения, Предупреждающая информация)

ЛАБОРАТОРИЯ (Код лаборатории (К/Л), Название, Адрес)

АНАЛИЗ (Регистрационный номер (Р/Н), Код Лаборатории (К/Л), Тип анализа (Т/А), Назначенная дата (Н/Д), Назначенное время (Н/В), Номер направления (Н/Н), Состояние)

БОЛЬНИЦА-ЛАБОРАТОРИЯ (Код больницы (К/Б), Код Лаборатории (К/Л))

ВРАЧ-ПАЦИЕНТ (Код врача (К/В), Регистрационный номер (Р/Н))

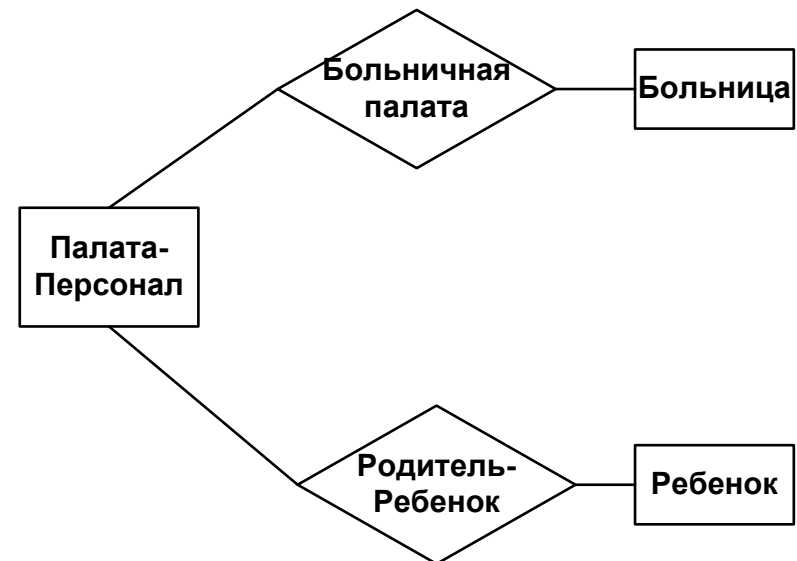
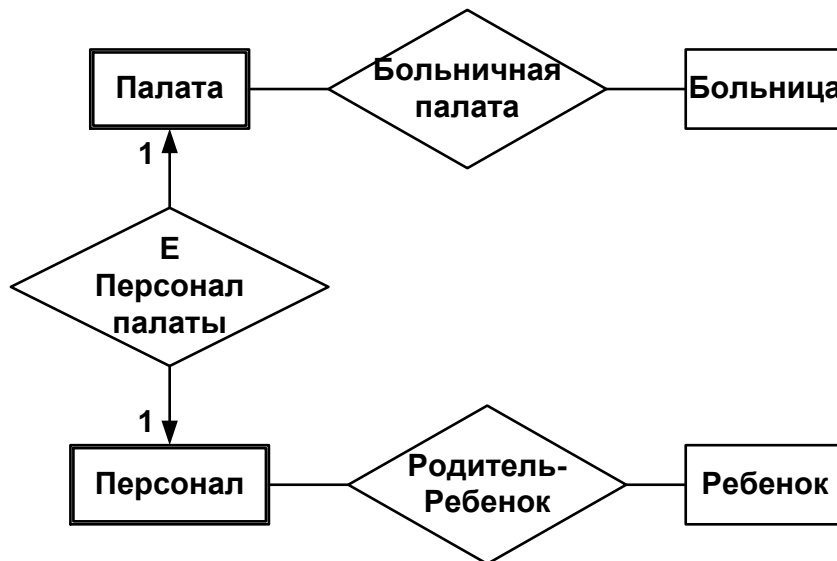
РАЗМЕЩЕНИЕ (Код больницы (К/Б), Номер палаты (Н/П), Регистрационный номер (Р/Н), Номер койки (Н/К))

ТЕЛЕФОН БОЛЬНИЦЫ (Код больницы (К/Б), Телефон)

ТЕЛЕФОН ЛАБОРАТОРИИ (Код лаборатории (К/Л), Телефон)

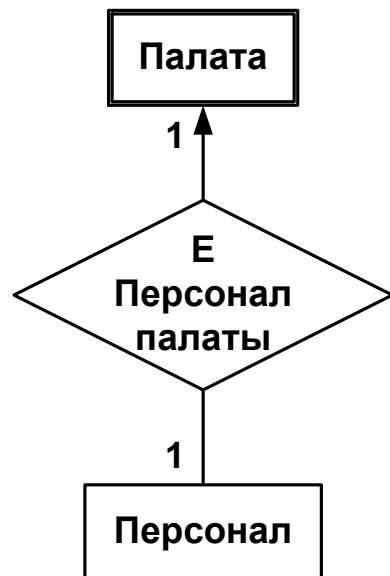
Усовершенствованные правила перехода от ER-схемы к реляционной схеме БД

1. Каждое бинарное множество связей (МСв) типа $R(S1(1,1) : S2(1,1))$ и множества сущностей (МСу) $S1$ и $S2$, в нем участвующие, заменяются в ER-схеме одним агрегированным МСу, с которым соединяются все другие МСв, имевшиеся у двух исходных МСу. Атрибуты нового МСу представляют собой объединение атрибутов обоих исходных МСу и МСв. Ключами полученного МСу являются ключи исходных МСу и МСв.



Усовершенствованные правила перехода от ER-схемы к реляционной схеме БД (продолжение)

2. Если МСу не вступает в связи типа $R(S1(1,1):S2(1,1))$, оно образует одно отношение, построенное на однозначных атрибутах этого МСу. Возможными ключами этого отношения являются ключи МСу.
3. Бинарное МСв типа $R(S1(1,1):S2(0,1))$ представляется дублированием первичного ключа отношения, построенного для МСу S1 (множества регулярных сущностей), в отношение, построенное для МСу S2 (множества слабых сущностей). Этот внешний ключ является возможным ключом второго отношения и имеет описатель NOT NULL.

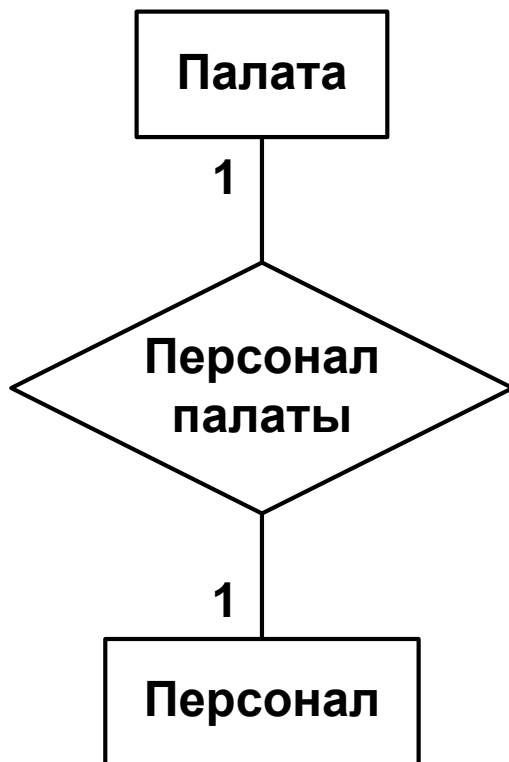


Палата_ID	...	Персонал_ID

Персонал_ID	...

Усовершенствованные правила перехода от ER-схемы к реляционной схеме БД (продолжение)

4. Бинарное МСв типа $R(S1(0,1):S2(0,1))$ представляется самостоятельным отношением, куда дублируются первичные ключи отношений, построенных для МСу $S1$ и $S2$. Каждый из этих внешних ключей является возможным ключом нового отношения и имеет описатель NOT NULL.



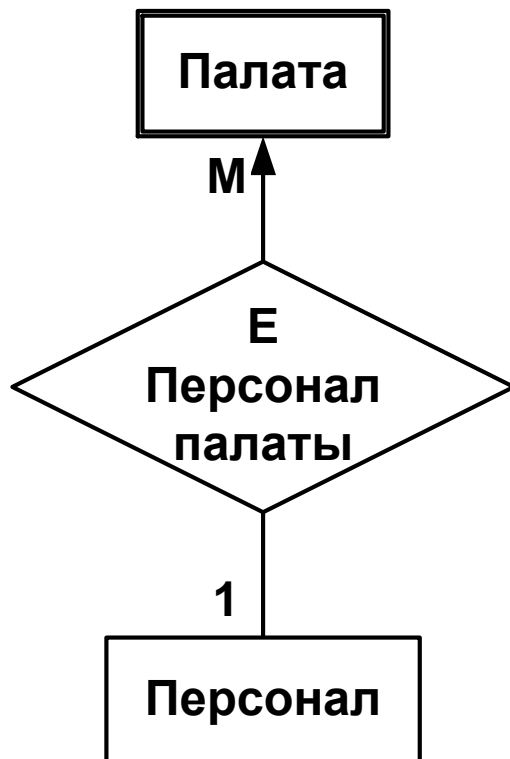
Палата_ID	...

Палата_ID	Персонал_ID

Персонал_ID	...

Усовершенствованные правила перехода от ER-схемы к реляционной схеме БД (продолжение)

5. Бинарное МСв типа $R(S1(1,1):S2(_,\infty))$ ($_$ означает, что минимальное кардинальное число отображения не имеет значения) представляется дублированием первичного ключа отношения, построенного для МСу $S1$, в отношение, построенное для МСу $S2$. Этот внешний ключ имеет описатель NOT NULL.

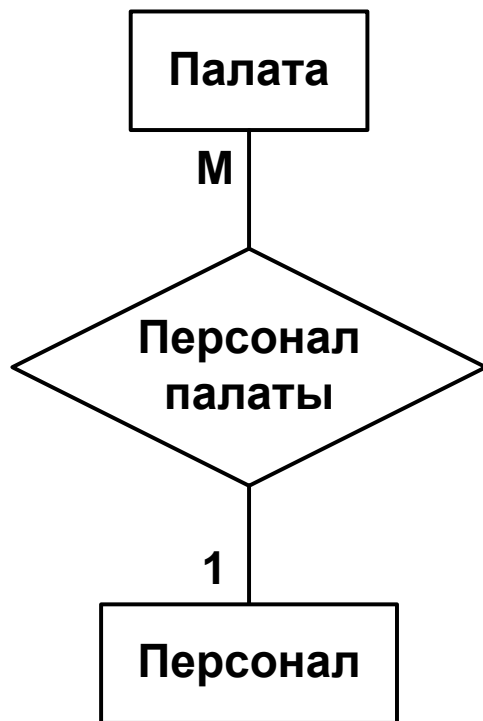


Палата_ID	...	Персонал_ID

Персонал_ID	...

Усовершенствованные правила перехода от ER-схемы к реляционной схеме БД (продолжение)

6. Бинарное МСв типа $R(S1(0,1) : S2(_, \infty))$ представляется самостоятельным отношением, куда дублируются первичные ключи отношений, построенных для МСу $S1$ и $S2$. Внешний ключ, являющийся дубликатом первичного ключа второго отношения (для $S2$), представляет собой возможный ключ нового отношения. Оба атрибута этого отношения имеют описатель NOT NULL.



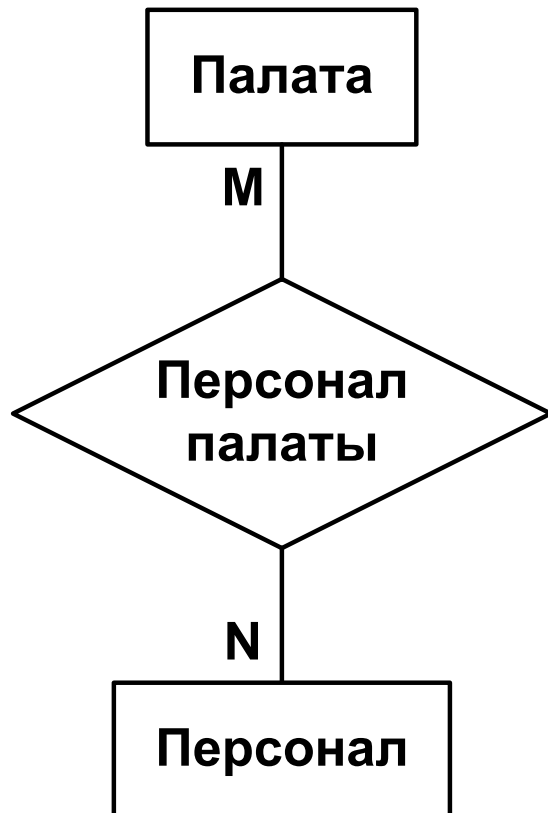
Палата_ID	...

Палата_ID	Персонал_ID

Персонал_ID	...

Усовершенствованные правила перехода от ER-схемы к реляционной схеме БД (продолжение)

7. Бинарное МСв типа M:N представляется самостоятельным отношением, куда дублируются первичные ключи отношений, построенных для МСу S1 и S2. Возможным ключом нового отношения является группа, состоящая из обоих внешних ключей, и оба они имеют описатель NOT NULL.



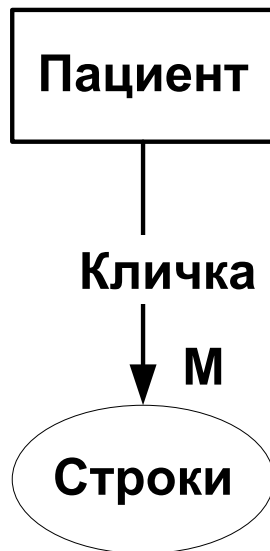
Палата_ID	...

Палата_ID	Персонал_ID

Персонал_ID	...

Усовершенствованные правила перехода от ER-схемы к реляционной схеме БД (продолжение)

8. МСв степени больше двух представляется самостоятельным отношением, куда дублируются первичные ключи отношений, построенных для МСу. Анализ ограничений целостности в этом случае требует более детального рассмотрения.
9. Если МСв имеет свои собственные однозначные атрибуты, они добавляются в то отношение, куда осуществляется дублирование первичного(-ых) ключа(-ей) для представления этого множества связей.
10. Каждый многозначный атрибут МСу представляется отдельным отношением, куда дублируется первичный ключ отношения, построенного для МСу; второй атрибут этого отношения предназначен собственно для значения.

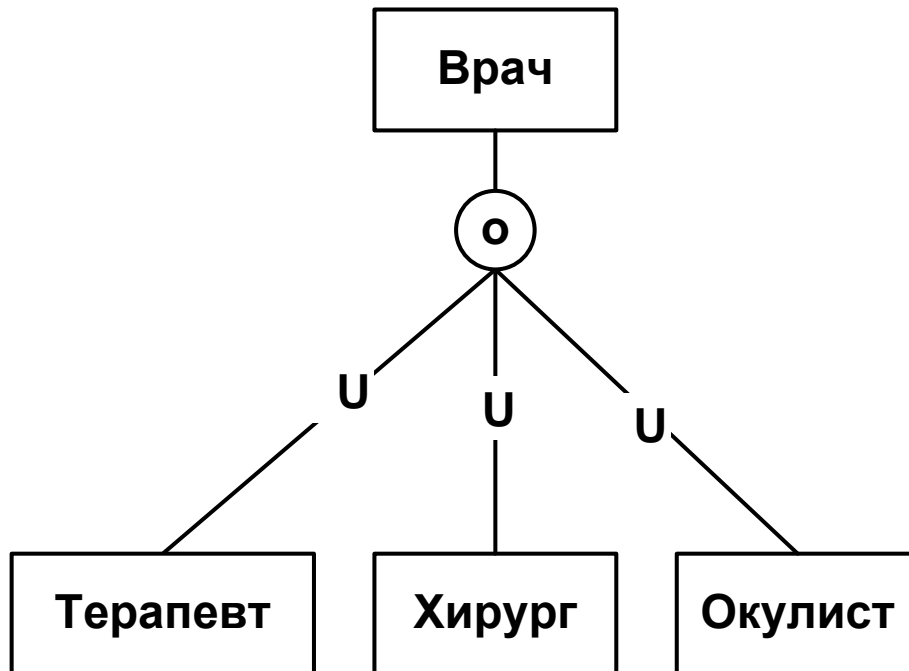


Пациент_ID	...

Пациент_ID	Кличка

Усовершенствованные правила перехода от ER-схемы к реляционной схеме БД (продолжение)

11. Каждый многозначный атрибут МСв представляется отдельным отношением, куда дублируется первичный ключ отношения, построенного для МСв; второй атрибут этого отношения предназначен собственно для значения.
12. Каждая ветвь специализаций и категоризаций трактуется как обычное бинарное МСв типа 1:1, которое преобразуется согласно указанным ранее правилам в зависимости от минимальных кардинальных чисел отображений.



Врач_ID	...

Терапевт_ID	Врач_ID	...

Хирург_ID	Врач_ID	...

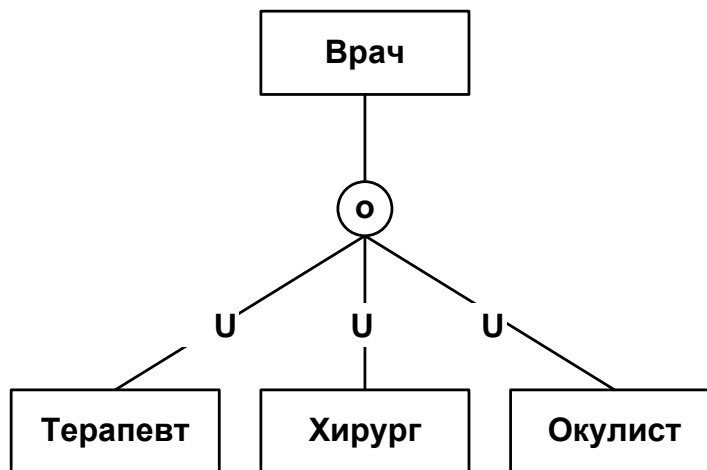
Окулист_ID	Врач_ID	...

Усовершенствованные правила перехода от ER-схемы к реляционной схеме БД (продолжение)

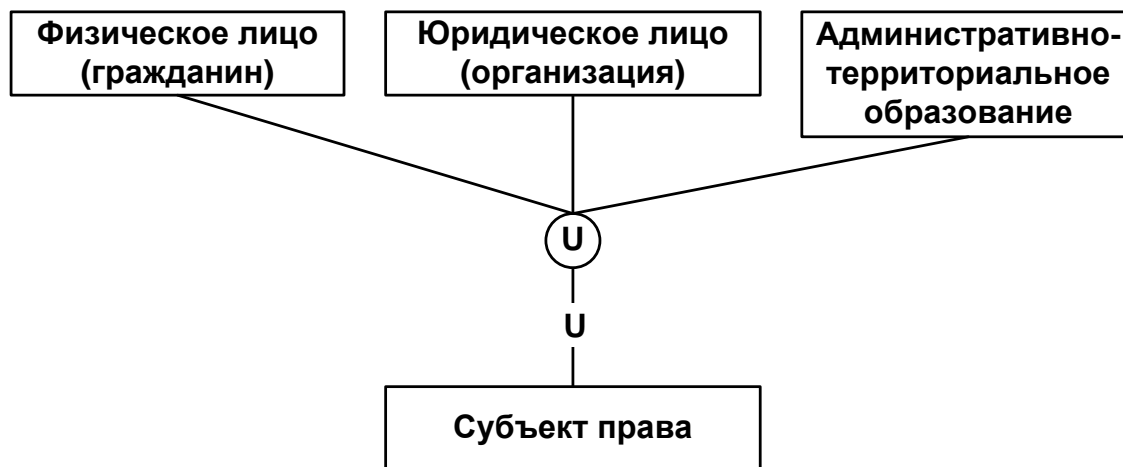
- 13. Перенос в реляционную схему ограничений целостности на значения атрибутов, как правило, весьма прост и определяется выразительными возможностями конкретного языка определения данных.**

Трансформация схемы БД из EER-модели в реляционную модель

Пример специализации



Пример категоризации



Трансформация схемы БД из EER-модели в реляционную модель

Методы представления специализаций в реляционной схеме

(n – количество подклассов).

Метод 1: 1 отношение. Создается одно отношение, которое включает однозначные атрибуты суперкласса и всех подклассов, т.е. каждый объект в рамках специализации описывается кортежем только одного отношения.

ВРАЧ (Врач_ID, ..., Терапевт_ID, ..., Хирург_ID, ..., Окулист_ID, ...)

Метод 2: $n+1$ отношение. Для суперкласса и каждого из подклассов создается по одному отдельному отношению. Один объект может описываться кортежами нескольких отношений: от 1 до $n+1$.

- 1 ВРАЧ (Врач_ID (PK), ...)
ТЕРАПЕВТ (Терапевт_ID (PK), ..., Врач_ID (FK))
ХИРУРГ (Хирург_ID (PK), ..., Врач_ID (FK))
ОКУЛИСТ (Окулист_ID (PK), ..., Врач_ID (FK))
- 2 ВРАЧ (Врач_ID (PK), ..., Терапевт_ID (FK), Хирург_ID (FK), Окулист_ID (FK))
ТЕРАПЕВТ (Терапевт_ID (PK), ...)
ХИРУРГ (Хирург_ID (PK), ...)
ОКУЛИСТ (Окулист_ID (PK), ...)
- 3 ВРАЧ (Врач_ID (PK), ...)
ТЕРАПЕВТ (Терапевт_ID (PK, FK), ...)
ХИРУРГ (Хирург_ID (PK, FK), ...)
ОКУЛИСТ (Окулист_ID (PK, FK), ...)

Трансформация схемы БД из EER-модели в реляционную модель

Метод 3: n отношений. Создается n отношений – по одному для каждого подкласса, однозначные атрибуты суперкласса включаются в каждое из этих отношений. Один объект может быть представлен кортежами нескольких отношений: от 1 до n .

ТЕРАПЕВТ (Врач_ID,..., Терапевт_ID,...)

ХИРУРГ (Врач_ID,..., Хирург_ID,...)

ОКУЛИСТ (Врач_ID,..., Окулист_ID,...)

Метод 4 (гибрид 2 и 3 методов): $n+1$ отношение. Для суперкласса и каждого из подклассов создается по одному отдельному отношению. Кроме этого, однозначные атрибуты суперкласса добавляются в каждое отношение, построенное для подклассов. Один объект может быть представлен кортежами нескольких отношений: от 1 до n .

ВРАЧ (Врач_ID,...)

ТЕРАПЕВТ (Врач_ID,..., Терапевт_ID,...)

ХИРУРГ (Врач_ID,..., Хирург_ID,...)

ОКУЛИСТ (Врач_ID,..., Окулист_ID,...)

Трансформация схемы БД из EER-модели в реляционную модель

Метод 5: $n+2$ отношения. Помимо n отношений для подклассов и одного для суперкласса создается связующее отношение, которое имеет следующие атрибуты: имя родительского отношения; первичный ключ объекта в родительском отношении; имя дочернего отношения; первичный ключ объекта в дочернем отношении. Таким образом, один объект может описываться кортежами нескольких отношений: от 1 до $n+2$.

ВРАЧ (Врач_ID (PK),...)

ТЕРАПЕВТ (Терапевт_ID (PK),...)

ХИРУРГ (Хирург_ID (PK),...)

ОКУЛИСТ (Окулист_ID (PK),...)

РОД_РЕБ (Имя_Род_Отн, Род_Отн_ID, Имя_Реб_Отн, Реб_Отн_ID)

Трансформация схемы БД из EER-модели в реляционную модель

Метод 6: *2ⁿ отношений.* Для любого возможного поддерева в иерархии специализаций, включающего корневой суперкласс, создается одно отношение, которое содержит в качестве атрибутов все однозначные атрибуты множеств сущностей, принадлежащих поддереву. Таким образом, каждый объект описывается кортежем одного и только одного отношения.

ВРАЧ (Врач_ID,...)

ТЕРАПЕВТ (Врач_ID,..., Терапевт_ID,...)

ХИРУРГ (Врач_ID,..., Хирург_ID,...)

ОКУЛИСТ (Врач_ID,..., Окулист_ID,...)

ТЕРАПЕВТ-ХИРУРГ (Врач_ID,..., Терапевт_ID,..., Хирург_ID,...)

ТЕРАПЕВТ-ОКУЛИСТ (Врач_ID,..., Терапевт_ID,..., Окулист_ID,...)

ХИРУРГ-ОКУЛИСТ (Врач_ID,..., Хирург_ID,..., Окулист_ID,...)

ТЕРАПЕВТ-ХИРУРГ-ОКУЛИСТ (Врач_ID,..., Терапевт_ID,...,
Хирург_ID,..., Окулист_ID,...)

Трансформация схемы БД из EER-модели в реляционную модель

Методы представления категоризаций в реляционной схеме
(n – количество суперклассов).

Метод 1: n отношений. Данный метод предполагает, что все однозначные атрибуты категории становятся атрибутами в отношениях, представляющих суперклассы. Отношение для категории не создается.

ФЛ (ФЛ_ID, ..., СП_ID, ...)

ЮЛ (ЮЛ_ID, ..., СП_ID, ...)

АТО (АТО_ID, ..., СП_ID, ...)

Метод 2: $n+1$ отношение. Для подкласса (категории) и каждого из суперклассов создается отдельное отношение. Варианты способов организации связей между кортежами отношений суперклассов и кортежами отношения категории аналогичны предлагавшимся для соответствующего метода специализаций.

СП (СП_ID, ...)

ФЛ (ФЛ_ID, ..., СП_ID)

ЮЛ (ЮЛ_ID, ..., СП_ID)

АТО (АТО_ID, ..., СП_ID)

Метод 3: $n+2$ отношения. Аналогичен методу 5 для специализаций.

Трансформация схемы БД из EER-модели в реляционную модель

Предлагается при выборе метода представления специализаций и категоризаций руководствоваться совокупностью следующих **критериев**:

1. Простота реализации. Помимо собственно времени, затрачиваемого на реализацию метода, критерий может включать и специфические требования, такие как учет потенциальной необходимости вносить изменения в реляционную схему (в частности, возможность появления дополнительных подклассов).

2. Эффективность хранения. Имеется возможность за счет выбора метода снизить влияние некоторых факторов, отрицательно сказывающихся на эффективности хранения:

а) дублирование значений атрибутов:

– дублирование значений ключей;

– дублирование значений остальных атрибутов;

б) представление отсутствующих значений (NULL).

3. Эффективность обработки:

а) обновление;

б) проверка ограничений целостности (совокупность ограничений целостности, заданных как декларативно, так и процедурно, неизменна, однако, сложность проверки одного и того же ограничения будет различной в зависимости от выбранного метода);

в) выборка.

Трансформация схемы БД из EER-модели в реляционную модель

Для специализаций необходимо учесть следующие **факторы**:

1. Количество атрибутов и множеств связей, ассоциированных с суперклассом и подклассами.
2. Свойства специализации (полное или частичное участие членов суперкласса в подклассах, пересечение или непересечение подклассов).
3. Степень пересечения подклассов в случае пересекающейся специализации.
4. Степень неполноты в случае специализации с частичным участием.
5. Количество подклассов специализации, их динамизм.

Трансформация схемы БД из EER-модели в реляционную модель

6. Наличие множеств связей или многозначных атрибутов, для представления которых правила преобразования в реляционную схему создают для некоторого отношения внешние ключи, ссылающиеся на ключ суперкласса, но не входящие в состав первичного ключа этого отношения.

7. Выполнение условия, аналогичного условию пункта 6, для одного или нескольких подклассов.

Трансформация схемы БД из EER-модели в реляционную модель

8. Наличие множеств связей или многозначных атрибутов, порождающих в процессе трансформации в реляционную схему отношения, внешние ключи которых ссылаются на ключ суперкласса, а также входят в состав первичного ключа данного отношения.
9. Выполнение условия, аналогичного указанному в пункте 8, для одного или нескольких подклассов.
10. Использование несуррогатных (естественных) ключей для суперкласса или подклассов.

Трансформация схемы БД из EER-модели в реляционную модель

Для категоризаций необходимо учесть следующие **факторы**:

1. Количество атрибутов и множеств связей, ассоциированных с категорией и суперклассами.
2. Свойство полноты/неполноты участия для категоризации.
3. Степень неполноты в случае категоризации с неполным участием.
4. Количество суперклассов, их динамизм.
5. Наличие множеств связей или многозначных атрибутов, порождающих в процессе трансформации в реляционную схему отношения, внешние ключи которых ссылаются на ключ категории, но не входят в состав первичного ключа данного отношения.
6. Наличие множеств связей или многозначных атрибутов, порождающих в процессе трансформации в реляционную схему отношения, внешние ключи которых ссылаются на ключ категории, а также входят в состав первичного ключа данного отношения.
7. Использование естественных ключей для суперклассов и подкласса.

Методика трансформации схемы базы данных из ERM-модели в реляционную модель

Критерии качества проекта БД

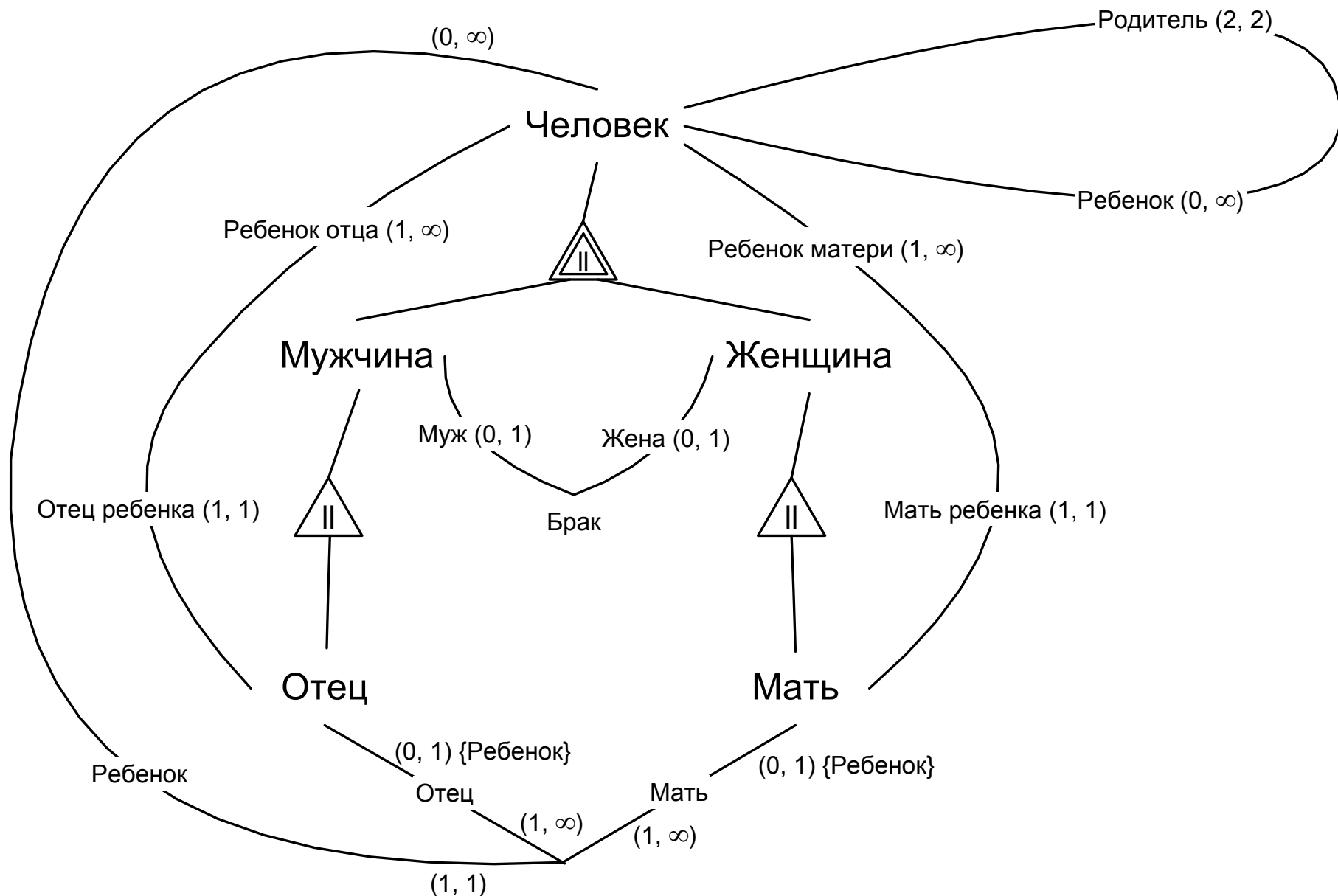
1. **Информационная полнота:** схема БД удовлетворяет этому критерию, если она полностью обеспечивает информационные потребности всех будущих пользователей системы.

2. **Информационная корректность:** схема БД удовлетворяет этому критерию, если в ней нашли отражение все выявленные на этапе анализа ПрО типичные свойства информационных объектов – ограничения целостности.

3. **Информационная избыточность:** схема БД удовлетворяет этому критерию, если в ней отсутствуют избыточные структуры данных и ограничения целостности. Избыточными будут структуры, предназначенные для представления высказываний, которые могут быть получены из данных другой более оптимальной структуры с помощью операций манипулирования данными. Ограничение целостности избыточно, если оно является следствием других ограничений целостности.

Исходная ERM-схема

Человек (ФИО, Умер, Пол, Прозвище(M))



Методика трансформации «ERM-модель – реляционная модель» (шаг 1)

Каждому множеству сущностей соответствует свое отношение, включающее следующие атрибуты:

- суррогатный первичный ключ;
- однозначные атрибуты этого множества сущностей.

Человек (Человек_ID, ФИО, Умер, Пол);

Мужчина (Мужчина_ID);

Женщина (Женщина_ID);

Отец (Отец_ID);

Мать (Мать_ID).

Методика трансформации «ERM-модель – реляционная модель» (шаг 2)

Для каждого множества связей степени n (представленного в графе классов ребром с n концами) строим отношение реляционной модели с n атрибутами, каждый из которых представляет собой внешний ключ – дубликат первичного ключа отношения, построенного для соответствующего множества сущностей.

Однозначные атрибуты этого множества связей становятся атрибутами отношения, построенного для его представления.

Если множество связей имеет многозначные атрибуты или участвует в специализации множеств связей, в отношении, построенном для него, понадобится выделить первичный ключ.

Родитель-Ребенок (Родитель_Человек_ID, Ребенок_Человек_ID);

Брак (Мужчина_ID, Женщина_ID);

Отец-Ребенок (Отец_ID, Человек_ID);

Мать-Ребенок (Мать_ID, Человек_ID);

Отец-Мать-Ребенок (Отец_ID, Мать_ID, Человек_ID).

Методика трансформации «ERM-модель – реляционная модель» (шаг 3)

Каждой многозначной характеристике каждого класса (множества сущностей или множества связей) соответствует свое бинарное отношение с атрибутами:

- внешний ключ, представляющий собой дубликат первичного ключа класса;
- атрибут для значения характеристики.

Прозвище человека (Человек_ID, Прозвище)

Методика трансформации «ERM-модель – реляционная модель» (шаг 4)

Если группа классов (множеств сущностей или множеств связей) образует независимую иерархию специализаций, для идентификации объектов всех классов этой группы используется один суррогатный ключ. Он определяется сначала на уровне корневого класса в качестве первичного ключа его отношения. В классах, являющихся ближайшими потомками корневого класса, создаются соответствующие ему внешние ключи, выступающие в то же время в качестве первичных ключей. На них в свою очередь ссылаются внешние ключи потомков 2-го рода и т.д.

Пример

В ПрО родственных отношений имеется одна иерархия множеств сущностей, состоящая из трех специализаций.

Применение этого правила не изменяет структуры спроектированных ранее отношений с точки зрения состава атрибутов.

Следует отметить лишь, что:

- областью значений внешнего ключа *Мужчина_ID* является первичный ключ отношения *Человек – Человек_ID*;
- областью значений внешнего ключа *Женщина_ID* является первичный ключ отношения *Человек – Человек_ID*;
- областью значений внешнего ключа *Отец_ID* является первичный ключ отношения *Мужчина – Мужчина_ID*;
- областью значений внешнего ключа *Мать_ID* является первичный ключ отношения *Женщина – Женщина_ID*.

Методика трансформации «ERM-модель – реляционная модель» (шаг 5)

Для представления фактов множественного наследования (класс принадлежит одновременно специализациям различных родительских классов) создается одно дополнительное отношение с четырьмя атрибутами:

- имя отношения суперкласса (*SUPERCLASS*);
- значение первичного ключа объекта в отношении суперкласса (*SUPER_ID*);
- имя отношения подкласса (*SUBCLASS*);
- значение первичного ключа объекта в отношении подкласса (*SUB_ID*).

Методика трансформации «ERM-модель – реляционная модель» (шаг 6)

Ограничения целостности на области значений отображений-характеристик. В реляционной модели их иногда называют ограничениями целостности на значения атрибутов. К ним относятся:

- обязательное указание типа данных для каждого атрибута отношения (с максимальной длиной – для строк и разрядностью – для чисел);
- конструкция *CHECK*.

ФИО CHARACTER(50);

Умер CHARACTER(6);

Пол CHARACTER(7);

Прозвище CHARACTER(20);

CHECK (Умер IN {'Истина', 'Ложь'});

CHECK (Пол IN {'Мужской', 'Женский'}).

Методика трансформации «ERM-модель – реляционная модель» (шаг 7)

Ограничения целостности на отображения-характеристики и обратные им отображения. В реляционной модели их иногда называют ограничениями уникальности и определенности.

Если отображение-характеристика – полное функциональное (МинКЧ = 1, МаксКЧ = 1), соответствующий ей атрибут принадлежит отношению класса и снабжается описателем *NOT NULL* (в случае частичного функционального отображения (МинКЧ = 0, МаксКЧ = 1) – описателем *NULL*).

Если отображение-характеристика многозначно (МаксКЧ \neq 1), соответствующий ей атрибут расположен в специальном дополнительном отношении. Оба атрибута этого отношения всегда объявляются *NOT NULL*.

Если отображение, обратное однозначному отображению-характеристике, функционально, соответствующий атрибут отношения класса объявляется возможным ключом (*UNIQUE*).

Если отображение, обратное многозначному отображению-характеристике, функционально, первичным ключом (*PRIMARY KEY*) дополнительного отношения объявляется атрибут со значениями характеристики, в противном случае – оба атрибута этого отношения.

Методика трансформации «ERM-модель – реляционная модель» (шаг 8)

Ограничения целостности на отображения между множествами сущностей. Эти ограничения целостности воплощаются в реляционной модели в виде так называемых ограничений ссылочной целостности, связанных с определением внешних ключей отношений, образованных из множеств связей.

Каждое определение внешнего ключа неявно подразумевает функциональность отображения между отношением, в котором он определен, и отношением, первичный или возможный ключ которого задает область допустимых значений внешнего ключа. Если это отображение к тому же полностью определено, внешний ключ снабжается описателем *NOT NULL*.

Если отображение, определяемое ролью (группой ролей), функционально, внешний ключ (группа внешних ключей), соответствующий этой роли (группе ролей), является возможным ключом (*UNIQUE*) отношения, которое задает экстенционал этого отображения.

Методика трансформации «ERM-модель – реляционная модель» (шаг 9)

Ограничения целостности на специализации. Возможности представления ограничений целостности этого типа в реляционной модели весьма ограничены.

В случае независимых иерархий специализаций полная функциональность отображения «подкласс-суперкласс» представляется ограничением ссылочной целостности и описателем *NOT NULL* для внешнего ключа отношения, представляющего подкласс.

Функциональность обратного отображения гарантируется объявлением одновременно этого внешнего ключа первичным ключом отношения для подкласса.

В случае же реализации множественных наследований одним универсальным дополнительным отношением невозможно определить вообще никаких ограничений целостности, даже ссылочных.

Пример

Человек (Человек_ID, ФИО, Умер, Пол):

Человек_ID	NUMBER(6,0)	NOT NULL,
ФИО	CHARACTER(50)	NOT NULL,
Умер	CHARACTER(6)	NOT NULL,
Пол	CHARACTER(7)	NOT NULL,

PRIMARY KEY (Человек_ID),
CHECK (Умер IN {'Истина', 'Ложь'}),
CHECK (Пол IN {'Мужской', 'Женский'}).

Прозвище человека (Человек_ID, Прозвище):

Человек_ID	NUMBER(6,0)	NOT NULL,
Прозвище	CHARACTER(20)	NOT NULL,

PRIMARY KEY (Человек_ID, Прозвище).
FOREIGN KEY (Человек_ID) REFERENCES Человек (Человек_ID).

Мужчина (Мужчина_ID):

Мужчина_ID	NUMBER(6,0)	NOT NULL,
------------	-------------	-----------

PRIMARY KEY (Мужчина_ID),
FOREIGN KEY (Мужчина_ID) REFERENCES Человек (Человек_ID).

Женщина (Женщина_ID):

Женщина_ID	NUMBER(6,0)	NOT NULL,
------------	-------------	-----------

PRIMARY KEY (Женщина_ID),
FOREIGN KEY (Женщина_ID) REFERENCES Человек (Человек_ID).

Пример (продолжение)

Отец (Отец_ID):

Отец_ID NUMBER(6,0) NOT NULL,
PRIMARY KEY (Отец_ID),
FOREIGN KEY (Отец_ID) REFERENCES Мужчина (Мужчина_ID).

Мать (Мать_ID):

Мать_ID NUMBER(6,0) NOT NULL,
PRIMARY KEY (Мать_ID),
FOREIGN KEY (Мать_ID) REFERENCES Женщина (Женщина_ID).

Родитель-Ребенок (Родитель_Человек_ID, Ребенок_Человек_ID):

Родитель_Человек_ID NUMBER(6,0) NOT NULL,
Ребенок_Человек_ID NUMBER(6,0) NOT NULL,
UNIQUE (Родитель_Человек_ID, Ребенок_Человек_ID),
FOREIGN KEY (Родитель_Человек_ID) REFERENCES Человек
 (Человек_ID),
FOREIGN KEY (Ребенок_Человек_ID) REFERENCES Человек
 (Человек_ID).

Пример (продолжение)

Брак (Мужчина_ID, Женщина_ID):

Мужчина_ID NUMBER(6,0) NOT NULL,
Женщина_ID NUMBER(6,0) NOT NULL,
UNIQUE (Мужчина_ID),
UNIQUE (Женщина_ID),
FOREIGN KEY (Мужчина_ID) REFERENCES Мужчина (Мужчина_ID),
FOREIGN KEY (Женщина_ID) REFERENCES Женщина
 (Женщина_ID).

Отец-Ребенок (Отец_ID, Человек_ID):

Отец_ID NUMBER(6,0) NOT NULL,
Человек_ID NUMBER(6,0) NOT NULL,
UNIQUE (Человек_ID),
FOREIGN KEY (Отец_ID) REFERENCES Отец (Отец_ID),
FOREIGN KEY (Человек_ID) REFERENCES Человек (Человек_ID).

Мать-Ребенок (Мать_ID, Человек_ID):

Мать_ID NUMBER(6,0) NOT NULL,
Человек_ID NUMBER(6,0) NOT NULL,
UNIQUE (Человек_ID),
FOREIGN KEY (Мать_ID) REFERENCES Мать (Мать_ID),
FOREIGN KEY (Человек_ID) REFERENCES Человек (Человек_ID).

Пример (продолжение)

Отец-Мать-Ребенок (Отец_ID, Мать_ID, Человек_ID):

*Отец_ID NUMBER(6,0) NOT NULL,
Мать_ID NUMBER(6,0) NOT NULL,
Человек_ID NUMBER(6,0) NOT NULL,
UNIQUE (Человек_ID),
FOREIGN KEY (Отец_ID) REFERENCES Отец (Отец_ID),
FOREIGN KEY (Мать_ID) REFERENCES Мать (Мать_ID),
FOREIGN KEY (Человек_ID) REFERENCES Человек (Человек_ID).*

Предварительная реляционная схема БД (структурный компонент)

Человек (Человек_ID, ФИО, Умер, Пол);

Прозвище человека (Человек_ID, Прозвище);

Мужчина (Мужчина_ID);

Женщина (Женщина_ID);

Отец (Отец_ID);

Мать (Мать_ID);

Родитель-Ребенок (Родитель_Человек_ID, Ребенок_Человек_ID);

Брак (Мужчина_ID, Женщина_ID);

Отец-Ребенок (Отец_ID, Человек_ID);

Мать-Ребенок (Мать_ID, Человек_ID);

Отец-Мать-Ребенок (Отец_ID, Мать_ID, Человек_ID).

Трансформация ограничений целостности

Класс ERM-ограничений	ERM-ограничение	Ограничение реляционной модели
Ограничения целостности на области значений отображений-характеристик	Синтаксис значений	Тип данных атрибута
	Логическое выражение	Конструкция CHECK
Ограничения целостности на отображения-характеристики и обратные им отображения	<p>Отображение-характеристика:</p> <p>МаксКЧ = 1 МинКЧ = 0 МинКЧ = 1 МаксКЧ = ∞ МинКЧ = 1</p>	<p>Атрибут в отношении-классе NULL NOT NULL Атрибут в доп. отношении – NOT NULL ---</p>
	<p>Обратное отображение с простой или сложной ООО:</p> <p>при однозначной характеристике</p> <p>МаксКЧ = 1 МинКЧ = 0 МинКЧ = 1 МаксКЧ = ∞ МинКЧ = 1</p> <p>при многозначной характеристике</p> <p>МаксКЧ = 1 МинКЧ = 0 МинКЧ = 1 МаксКЧ = ∞ МинКЧ = 1</p>	<p>Атрибут отношения класса UNIQUE --- --- Атрибут отношения класса не UNIQUE --- Атрибут доп. отношения PRIMARY KEY --- --- Оба атрибута доп. отношения составляют PRIMARY KEY ---</p>

Трансформация ограничений целостности (продолжение)

Ограничения целостности на отображения между множествами сущностей	<p>Отображение, определяемое ролью или группой ролей: $\text{МаксКЧ} = 1$</p> <p>$\text{МинКЧ} = 0$ $\text{МинКЧ} = 1$ $\text{МаксКЧ} = \infty$</p> <p>$\text{МинКЧ} = 1$</p>	<p>Соответствующий внешний ключ или группа внешних ключей объявляется UNIQUE</p> <p>---</p> <p>---</p> <p>Соответствующий внешний ключ не объявляется UNIQUE</p> <p>---</p>
Ограничения целостности на специализации	Специализация полная, пересекающаяся $\cup \text{ПК} \rightarrow \text{СК} (1, 1), \text{СК} \rightarrow \cup \text{ПК} (1, \infty)$	<p>Декларативными средствами реляционной модели удастся представить лишь полную функциональную определенность отображений $\text{ПК} \rightarrow \text{СК}$, а вместе с ними – и отображения $\cup \text{ПК} \rightarrow \text{СК}$.</p>
	Специализация полная, непересекающаяся $\cup \text{ПК} \rightarrow \text{СК} (1, 1), \text{СК} \rightarrow \cup \text{ПК} (1, 1)$	
	Специализация частичная, пересекающаяся $\cup \text{ПК} \rightarrow \text{СК} (1, 1), \text{СК} \rightarrow \cup \text{ПК} (0, \infty)$	
	Специализация частичная, непересекающаяся $\cup \text{ПК} \rightarrow \text{СК} (1, 1), \text{СК} \rightarrow \cup \text{ПК} (0, 1)$	
Ограничения целостности на взаимоотношения отображений	<p>Следствие</p> <p>Эквивалентность</p>	<p>---</p> <p>---</p>

Методика трансформации «ERM-модель – реляционная модель» (шаг 10)

Полностью избыточные отношения для множеств связей, удаление которых не приводит к изменению остальных отношений. Отношение, построенное для множеств связей степени большей двух, избыточно, если для него и соответствующих бинарных отношений выполняются условия эквивалентности схем. Эти условия обеспечивают теоремы ТСЗО.

В частности, если в отношении есть функциональное отображение, определяемое ролью, и его проекции эквивалентны соответствующим родственным отображениям отношений, построенных для бинарных множеств связей, можно смело избавляться от первого отношения.

В нашем примере таковым является отношение *Отец-Мать-Ребенок*.

Отображение *Родители*, определяемое ролью *Ребенок*, – функционально. Для его проекций справедливо *Родители [Отец] <-> Отец ребенка* и *Родители [Мать] <-> Мать ребенка*.

Исключение этого отношения из схемы не приведет к потере функциональной зависимости *Ребенок -> Отец, Мать* поскольку она является следствием функциональных зависимостей *Ребенок -> Отец* и *Ребенок -> Мать*.

Методика трансформации «ERM-модель – реляционная модель» (шаг 10) (продолжение)

Полностью избыточными являются также отношения, построенные для множеств связей, все отображения которых эквивалентны отображениям (возможно полученным с помощью операций над отображениями), определяемым другими отношениями проекта.

Подобная ситуация наблюдается с отношением *Родитель-Ребенок*. Оба определяемые им отображения *Родитель* и *Ребенок* таковы, что

Отец ребенка \cup *Мать ребенка* \leftrightarrow *Родитель* и
Ребенок отца \cup *Ребенок матери* \leftrightarrow *Ребенок*.

В переводе на «реляционный» язык это означает, что отношение *Родитель-Ребенок* может быть получено объединением отношений *Отец-Ребенок* и *Мать-Ребенок*. В таком случае отношение *Родитель-Ребенок* также избыточно.

Методика трансформации «ERM-модель – реляционная модель» (шаг 11)

Избыточные отношения для множеств связей, удаление которых приводит к изменению других отношений. Если хотя бы одно из определяемых бинарным отношением отображений функционально, это отношение может быть исключено с одновременным добавлением дополнительного внешнего ключа в отношение для множества сущностей, являющегося множеством прообразов функционального отображения. Областью значений этого внешнего ключа является первичный ключ отношения для другого множества сущностей.

Это правило следует безусловно применять в случае полной функциональности отображения. При этом добавленный внешний ключ имеет описатель *NOT NULL*.

В случае частичной функциональности применение этого правила оправдано, если справедливо хотя бы одно из следующих условий:

- СУБД эффективно хранит пропущенные значения на диске;
- вероятность наличия у объектов-прообразов соответствующих образов близка к единице;
- операция соединения с участием рассматриваемого отношения часто выполняется и требует немалых затрат памяти и времени.

Если все-таки принимается решение об исключении отношения, добавленный внешний ключ в таком случае имеет описатель *NULL*.

Пример

В нашем примере применение этого правила к отношениям *Отец-Ребенок* и *Мать-Ребенок* приведет к тому, что эти отношения будут исключены из схемы, а в отношение *Человек* будут добавлены два дополнительных атрибута и ограничения внешних ключей:

Человек (Человек_ID, Умер, Пол, Отец_ID, Мать_ID):

<i>Человек_ID</i>	<i>NUMBER(6,0)</i>	<i>NOT NULL,</i>
<i>ФИО</i>	<i>CHARACTER(50)</i>	<i>NOT NULL,</i>
<i>Умер</i>	<i>CHARACTER(6)</i>	<i>NOT NULL,</i>
<i>Пол</i>	<i>CHARACTER(7)</i>	<i>NOT NULL,</i>
<i>Отец_ID</i>	<i>NUMBER(6,0)</i>	<i>NOT NULL,</i>
<i>Мать_ID</i>	<i>NUMBER(6,0)</i>	<i>NOT NULL,</i>

PRIMARY KEY (Человек_ID),
CHECK (Умер IN {'Истина', 'Ложь'}),
CHECK (Пол IN {'Мужской', 'Женский'}),
FOREIGN KEY (Отец_ID) REFERENCES Отец (Отец_ID),
FOREIGN KEY (Мать_ID) REFERENCES Мать (Мать_ID).

Описатель *NOT NULL* у обоих дополнительных атрибутов отражает полную функциональность отображений *Отец ребенка* и *Мать ребенка*.

Пример

Как правило, в случае отношения типа 1:1 осуществляется дублирование только одного атрибута. В нашем случае при исключении отношения *Брак* мы осуществим перекрестное дублирование: *Мужчина_ID* в отношение *Женщина* и *Женщина_ID* в отношение *Мужчина*.

Мужчина (Мужчина_ID, Женщина_ID);

Мужчина_ID NUMBER(6,0) NOT NULL,

Женщина_ID NUMBER(6,0) NULL,

PRIMARY KEY (Мужчина_ID),

UNIQUE (Женщина_ID),

FOREIGN KEY (Мужчина_ID) REFERENCES Человек (Человек_ID),

FOREIGN KEY (Женщина_ID) REFERENCES Женщина (Женщина_ID).

Женщина (Женщина_ID, Мужчина_ID);

Женщина_ID NUMBER(6,0) NOT NULL,

Мужчина_ID NUMBER(6,0) NULL,

PRIMARY KEY (Женщина_ID),

UNIQUE (Мужчина_ID),

FOREIGN KEY (Женщина_ID) REFERENCES Человек (Человек_ID),

FOREIGN KEY (Мужчина_ID) REFERENCES Мужчина (Мужчина_ID).

Методика трансформации «ERM-модель – реляционная модель» (шаг 12)

Полностью избыточные отношения для множеств сущностей, удаление которых не приводит к изменению остальных отношений. Если отношение, построенное для множества сущностей, можно получить в результате выполнения операций над другими отношениями, и это множество сущностей не участвует ни в каких отображениях, его можно удалить из схемы. Другими словами, в таком отношении нет атрибутов кроме первичного суррогатного ключа, и последний не участвует в определениях внешних ключей. К тому же, текущее множество значений этого ключа может быть автоматически получено по запросу с участием других отношений.

Логической основой таких запросов являются формальные определения понятий ПрО, полученные на шаге 1 методики ERM-моделирования.

В схеме нашего примера подобных отношений нет.

Методика трансформации «ERM-модель – реляционная модель» (шаг 13)

Избыточные отношения для множеств сущностей, удаление которых приводит к изменению других отношений. Если отношение, построенное для множества сущностей, можно получить в результате выполнения операций над другими отношениями, и это множество сущностей участвует в отображениях между множествами сущностей, его можно удалить из схемы, переопределив внешние ключи, ссылающиеся на первичный ключ этого отношения. Вместо него необходимо указать первичный ключ отношения, построенного для множества сущностей, которое является суперклассом по отношению к исходному.

Если кандидатом на исключение является отношение, множество сущностей которого имеет однозначные характеристики, решение во многом зависит от эффективности представления в СУБД пропущенных значений. Если вас устраивает, что в кортежах отношения суперкласса будут пропущенные значения в этих характеристиках, перенесите их в отношение суперкласса. С многозначными характеристиками проблем не будет. Надо лишь поменять ссылки на первичные ключи во внешних ключах соответствующих отношений для многозначных характеристик.

Пример

В нашем случае с родственными связями на нижнем уровне иерархии специализаций имеются отношения для множеств сущностей *Отец* и *Мать*. Приведем соответствующие определения понятий:

$$\begin{aligned} x \text{ Отец}(x) &=_{Df} x (x \text{ есть } a \text{ Мужчина}(a) \ \& \ \exists y \text{ Родитель-Ребенок}(x, y)) \\ x \text{ Мать}(x) &=_{Df} x (x \text{ есть } a \text{ Женщина}(a) \ \& \ \exists y \text{ Родитель-Ребенок}(x, y)). \end{aligned}$$

Экстенсионалы этих отношений можно получить из других отношений схемы, выполнив следующие запросы SQL:

```
SELECT Мужчина.Мужчина_ID FROM Мужчина, Человек
WHERE Мужчина.Мужчина_ID = Человек.Отец_ID и
SELECT Женщина.Женщина_ID FROM Женщина, Человек
WHERE Женщина.Женщина_ID = Человек.Мать_ID
```

Удаляем отношения для множеств сущностей *Отец* и *Мать*, заменив определения внешних ключей отношения *Человек* на следующие:

```
FOREIGN KEY (Отец_ID) REFERENCES Мужчина (Мужчина_ID),
FOREIGN KEY (Мать_ID) REFERENCES Женщина (Женщина_ID).
```

Пример (продолжение)

В более сложном случае отношение для множества сущностей может иметь дополнительные атрибуты – внешние ключи, представляющие функциональные отображения, в которых рассматриваемое множество сущностей играет роль множества прообразов. Такое отношение можно исключить из схемы, только в случае, когда эти внешние ключи удастся переместить в отношение-суперкласс, не нарушив критериев полноты и корректности. К тому же следует оценить такие изменения с точки зрения повышения эффективности с учетом уже упоминавшихся особенностей физической модели СУБД.

Пример (продолжение)

К настоящему моменту в нашем примере на нижнем уровне иерархии остались множества сущностей *Мужчина* и *Женщина*, соответствующие понятиям:

$x \text{ Мужчина}(x) =_{Df} x (x \text{ есть а } \text{Человек}(a) \ \& \ x = \text{Человек пола (Мужской)})$ и
 $x \text{ Женщина}(x) =_{Df} x (x \text{ есть а } \text{Человек}(a) \ \& \ x = \text{Человек пола (Женский)})$.

Экстенсионалы их отношений можно получить из отношения *Человек*, выполнив следующие запросы SQL:

```
SELECT Человек.Человек_ID FROM Человек  
WHERE Человек.Пол = 'Мужской' и  
SELECT Человек.Человек_ID FROM Человек  
WHERE Человек.Пол = 'Женский'
```

Обобщая понятия *Муж* и *Жена*, можно прийти к понятию *Супруг*, объем которого является объединением объемов исходных понятий. Поскольку специализация множества сущностей *Человек* на множества сущностей *Мужчина* и *Женщина* – полная и непересекающаяся, можно из внешних ключей отношений *Мужчина* и *Женщина* образовать один внешний ключ отношения *Человек* – *Супруг_ID*. Областью определения значений этого ключа является первичный ключ отношения *Человек* – *Человек_ID*.

Методика трансформации «ERM-модель – реляционная модель» (результат)

Человек (Человек_ID, Умер, Пол, Отец_ID, Мать_ID, Супруг_ID):

Человек_ID	NUMBER(6,0)	NOT NULL,
ФИО	CHARACTER(50)	NOT NULL,
Умер	CHARACTER(6)	NOT NULL,
Пол	CHARACTER(7)	NOT NULL,
Отец_ID	NUMBER(6,0)	NOT NULL,
Мать_ID	NUMBER(6,0)	NOT NULL,
Супруг_ID	NUMBER(6,0)	NULL,

PRIMARY KEY (Человек_ID),
UNIQUE (Супруг_ID),
CHECK (Умер IN {'Истина', 'Ложь'}),
CHECK (Пол IN {'Мужской', 'Женский'}),
FOREIGN KEY (Отец_ID) REFERENCES Человек (Человек_ID),
FOREIGN KEY (Мать_ID) REFERENCES Человек (Человек_ID),
FOREIGN KEY (Супруг_ID) REFERENCES Человек (Человек_ID).

Прозвище человека (Человек_ID, Прозвище):

Человек_ID	NUMBER(6,0)	NOT NULL,
Прозвище	CHARACTER(20)	NOT NULL,

PRIMARY KEY (Человек_ID, Прозвище).

Методика трансформации «ERM-модель – реляционная модель» (заключительные проверки)

На завершающей фазе методики следует проверить оставшиеся отношения на удовлетворение критериев полноты и корректности – все ли формы высказываний и ограничения целостности поддерживаются схемой.

Логическое проектирование данных

Этап 3. Введение контролируемой избыточности данных

- 1. Объединение таблиц со связями типа «один к одному» (1:1).**
- 2. Дублирование неключевых атрибутов родительской таблицы в связях «один ко многим» (1:M).**
- 3. Создание агрегирующих столбцов родительской таблицы в связях «один ко многим» (1:M).**
- 4. Дублирование атрибутов внешних ключей в иерархии связей «один ко многим» (1:M).**
- 5. Дублирование атрибутов в связях «многие ко многим» (M:N).**
- 6. Введение повторяющихся групп полей.**
- 7. Объединение справочных таблиц с базовыми таблицами.**
- 8. Создание таблиц из данных, содержащихся в других таблицах.**

5.4. Физическое проектирование данных

Физическое проектирование данных

- 1. Анализ транзакций.**
- 2. Выбор файловой структуры.**
- 3. Определение индексов.**
- 4. Определение требований к дисковой памяти.**

Физическое проектирование данных.

Анализ транзакций

Цель. Определение характеристик транзакций, которые будут выполняться в проектируемой базе данных, и выделение наиболее важных из этих транзакций.

- Отношения и атрибуты, к которым осуществляется доступ в процессе выполнения транзакции, а также тип доступа.**
- Атрибуты, которые используются в предикатах.**
- Атрибуты, участвующие в соединении двух или нескольких отношений, атрибуты, применяемые для упорядочения результатов, атрибуты, применяемые для группирования данных.**
- Ожидаемая частота выполнения транзакции.**
- Установленные показатели производительности для транзакции.**

Физическое проектирование данных.

Выбор файловой структуры

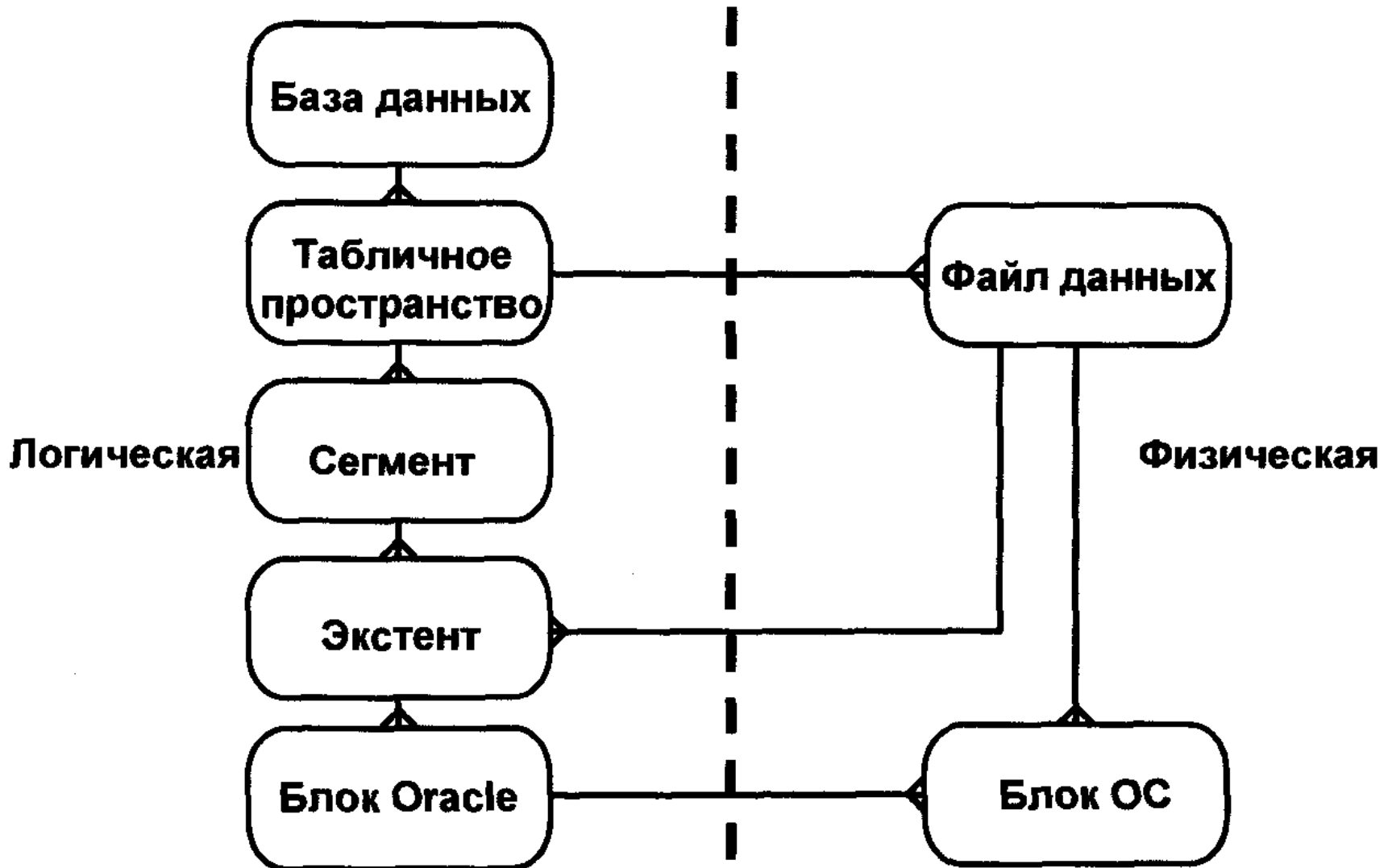
Цель. Определение наиболее эффективного файлового представления для каждого из базовых отношений.

Варианты представления базовых отношений в СУБД Oracle:

- Обычная таблица.**
- Секционированная таблица.**
- Индекс-таблица.**
- Кластеризованная таблица.**

Элементы физической модели Oracle

Структура базы данных

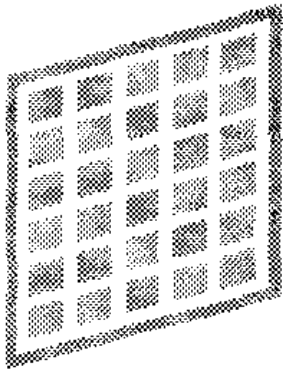


Элементы физической модели Oracle (продолжение)

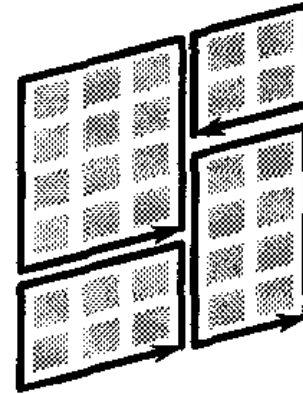
Database																PROD															
TABLESPACES SYSTEM				USER_DATA								RBS				TEMP															
DATAFILES DISK1/SYS1.dbf				DISK2/ USER1.dbf				DISK3/ USER2.dbf				DISK1/ ROLL1.dbf				DISK1/ TEMP.dbf															
SEGMENTS D.D. Table Data Seg		D.D. Index Index Seg		RB Seg	S_DEPT Data Seg		S_EMP Data Seg		S_DEPT (cont'd) Data Seg		S_EMP FIRST_N AME Index Index Seg		RBS1 RB Seg		RBS2 RB Seg		RBS1 (cont'd) RB Seg		RBS2 (cont'd) RB Seg		Temp Temp Seg										
EXTENTS				1 2 1 2 1 2 1 1 2 2 1				FREE 1 1 2 2 1																							
Oracle DATA BLOCKS																															

Элементы физической модели Oracle (продолжение)

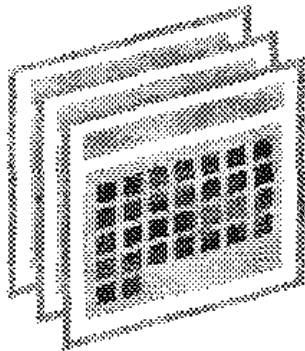
Виды сегментов



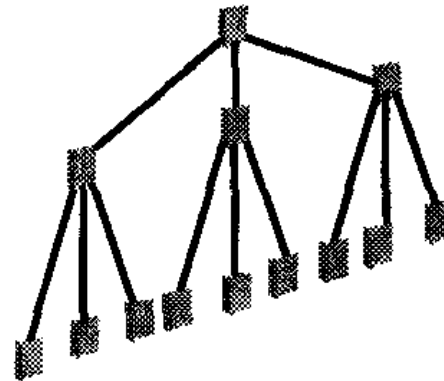
Таблица



Секция таблицы



Кластер



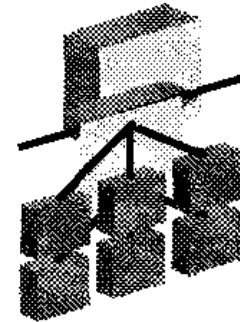
Индекс

Элементы физической модели Oracle (продолжение)

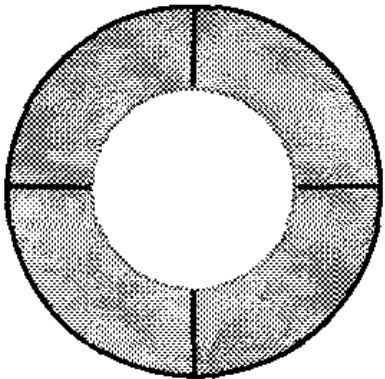
Виды сегментов



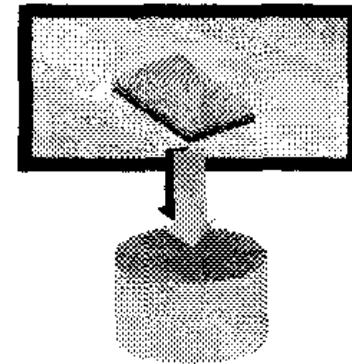
Индекс-таблица



**Индексная
секция**



Сегмент отмены



**Временный
сегмент**

Физическое проектирование данных.

Определение индексов

Цель. Определение того, будет ли добавление индексов способствовать повышению производительности системы.

- 1. Не создавать индекс на небольших отношениях.**
- 2. Следует создавать индекс на первичном ключе отношения**
- 3. Ввести дополнительный индекс на внешнем ключе.**
- 4. Ввести дополнительные индексы на всех атрибутах, которые часто применяются в качестве дополнительного поискового ключа.**
- 5. Ввести дополнительные индексы на атрибутах, которые часто применяются в конструкциях *WHERE*, *ORDER BY*, *GROUP BY*;**
- 6. Не индексировать атрибут или отношение, которые часто обновляются.**
- 7. Не индексировать атрибут, если в запросах с использованием этого атрибута обычно происходит выборка значительной части (например, 25%) кортежей в отношении.**
- 8. Не индексировать атрибуты, которые состоят из длинных символьных строк.**

Физическое проектирование данных. Определение требований к дисковой памяти

Цель. Определить объем дискового пространства, который требуется для базы данных.

```
CREATE TABLE [<имя схемы/владельца>.]  
<имя таблицы>  
(<имя столбца> <тип данных столбца>  
[,<имя столбца> <тип данных столбца>]...)  
[TABLESPACE <имя ТП>]  
[PCTFREE <целое>]  
[PCTUSED <целое>]  
[INITRANS <целое>]  
[MAXTRANS <целое>]  
[<фраза STORAGE>]  
[LOGGING | NOLOGGING]
```