

Базы Данных.

intitut.ru
citforum.ru

Литература: Кузнецов А.Д. "Основы БД"
 "БД модели и языки"
 Марков, Лисовский. БД. Введение в теорию и методологию
 Дейт. Введение в системы БД.

Информ. система - комплекс программно-аппаратный, его функции:
 1) хранение данных на внешних устр-вах
 2) выполнение специфич. задач (операций)
 3) вывод информации в спец. виде

Файл - именованная область памяти для хранения / записи информации.

Структуризация данных: на внешнем устр-ве информация логич. построена (переменной блок не дает возможности по ск-ти доступа, создает проблемы при дефрагментации).

Возможное представление файла:

- 1) послед-ть записей (переменной или пост. длины)
- 2) послед-ть байт

Именованное - иерархическая структура.

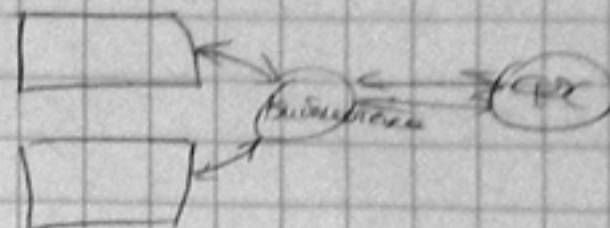
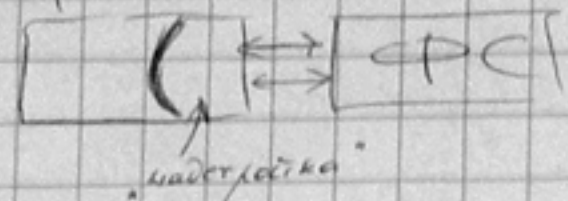
- 1) изолированная ФС (полное имя файла начинается с имени диска)
- 2) полностью централизованная ФС (OS Unix)
 (полное хранение данных о всех применяемых дисках)

Защита ФС и файлов: 1) дискретный (4 пользователя об-т идент-ром доступа ко 4 файлу)
 2) дискреционный (4 пользователя - 2 id
 4 файла - id владельца, содержатся права на R/W/E для пользователя)

ОС контролирует совместимость режимов открытия файлов при совместном использовании.

ФС - структуризация
 доступа
 поддержка многоп. режима
 представление файла на физ. устр-ве

Упор. система



Пример. Построим ИС

Служащие: Имя $\rightarrow N^{\circ}$
 $N^{\circ} \rightarrow$ Имя
 $\rightarrow$ Зарплата | Статус
 +
 Отдел: Рук-ль
 Числ-ть
 Сумм. зарп.
 +
 Список служащих по отделам

Уникальные ключи

Неуникальный ключ

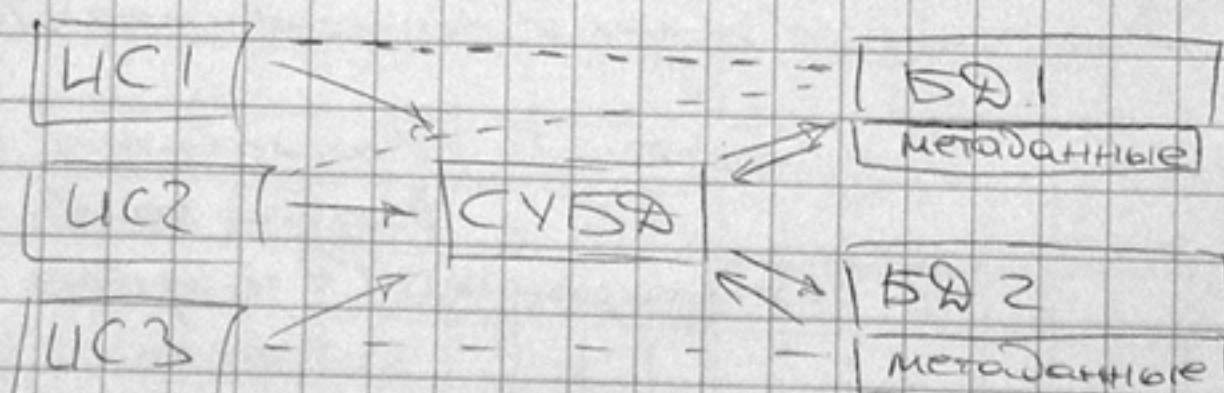
1 Файл: Записи: Имя | N° | Статус | Зарплата | N° отд. | Рук-отд.

2 Файл: Служащие: Имя | N° | статус | Зарплата | N° отд.

3 Отделы: N° | N° рук. | Σ зарплата | Размер

Ограничения

Если в I $\exists$ запись с N° отд. $\Rightarrow$ в II $\exists$ отдел с N°
 Если в II $\exists$ отдел с N° и $\Rightarrow$ в I $\exists$ служащий с N° отдела и
 Σ зарплата согласованы с данными файла I
 Размеры в II согласованы с данными файла I



ФС не обеспечивает
 избыточные
 восстановление данных
 параллельный доступ
 возможность запросов

СУБД - система для управл. лог. сог-тью данных во внешней памяти
 буферами ОП
 транзакциями
 функциями и восстановления данных
 языками БД

Св-ва транзакции (ACID (ACID)): атомарность
 согласованность
 изолированность
 долговечность

Сериализация транзакций - вып-е транзакций так, чтобы они в целом выполнялись так же, как и одна за другой.

Надежность СУБД - возм-ть восп-я после сбоя.

Типа | Источники: аппаратн / прог.
 Характер: мягкий / жесткий
 (несогл-ть) (потеря инфор-ции на носителе)

Журнал - недоступная пользователю часть БД, содержащая сведения о действиях над БД и о версионном БД.

Исп-е протокол управления записи WAL
 журнал исп-е, напр-е, для индивидуальной отмены транзакции.
 восп-е после сбоя

Языки → SDL (структурирование БД)
 → DML (манипулирование БД)

Организация БД.

- Ядро
- Компилятор (комплексная обработка в код)
- Система проверки времени выполнения (интерпретатор внутр. языка)
- Утилиты (загрузка БД, ...)

Модель данных:

- Структурная
- Математическая
- Целостная

Форелеционная БД: не поддерживают абстрактные модели данных
 производили доступ к данным на уровне языка
 доступ к БД осуществлялся конкретной системой
 нет оптимизации доступа ^{наблюдается} ~~оптимизация~~
 директивный доступ к ближайшей памяти

Допеляционные БД:

1. Иерархическая БД:

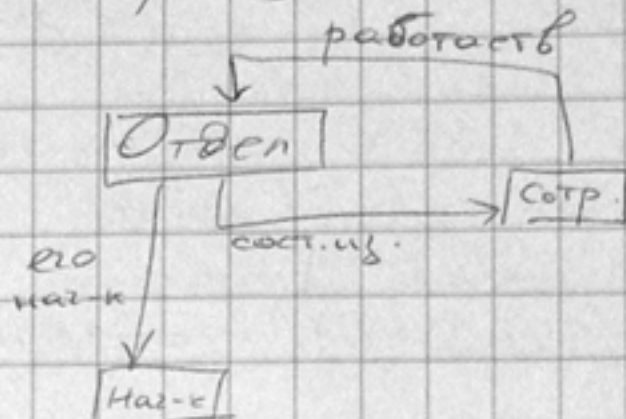
напр., "упорядоченный набор данных"

Отдел
 ↓
 Начальник Сотрудники

Связь базы - "нет записи без предка"

2. Сетевые БД:

"произвольный граф"

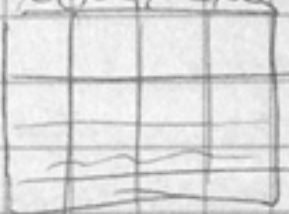


Связь базы - вообще говоря, не подвергается

3. Инвертированные списки:

"таблица записей"

записи



некоторые атрибуты - ключи

Общие правила целостности отсутствуют.

- ~~представление~~ структура

Эдгар

Реляционный подход - с 1969 г. / реализация - с начала 80-х.

основан на понятиях, которые можно легко определить формально

на теории мн-в и мат. логике
 обеспечивается работа без знания структуры и способа хранения данных

Типы данных:

- символьные
- числовые
- спец. числовые (деньги)
- битовые строки
- временные
- собств. типы (вводятся пользователем)

Домен - подмн-во данного типа данных

Атрибуты описания на доменах (домен определяет сравнимость атрибутов)

Схема отношения - конечное именованное мн-во пар вида $H_i = \{ \langle \text{имя атр-та, имя домена (типа)} \rangle \}$ ($\langle \text{зарплата, деньги} \rangle$)

Мощность схемы - мощность H_i .

Схема БД - кол-во хранящихся в БД отношений.

Кортеж - мн-во упор. пар вида $t_i(H_i) = \{ \langle \text{имя атр., значение} \rangle \}$

напр-р: $\{ \langle \text{служ-Н}, 1 \rangle, \langle \text{служ-Имя}, \text{Иванов} \rangle, \langle \text{служ-зарплата}, 100 \rangle, \langle \text{служ-отдел}, 1 \rangle \}$

Отношение — мн-во кортежей из одной схемы отн-и.

~~Отношение — мн-во отношений~~

БД подразумевает отсутствие пор-ка кортежей и атр в.

Нет кортежей дубликатов.

Первичный ключ отн-я — подмн-во S_1 атр-в: z_1 с перв. ключом во $\forall$ кортеже тела отн-я, ^{уникально} при этом ни одно другое S_2 этим сб-во не обладает (т.е. это мн-во атр-в, однозначно идентиф. кортеж)

Кандидатов на перв. ключ может быть несколько, но лишь один делается первичным, ост-е называют возможными.

Значение атрибута атомарно, т.е. значение не может быть мн-вом или же разбитым на составляющие.

Структурная часть: основана на нормализованном н-арном отношении.

Манипуляционная часть: реляционная алгебра (теория мн-в)

исчисление (теория предикатов) $\rightarrow$ ист-е доменов — и — кортежей

Целостная часть: 2 типа целостности: 1) Целостность сущности (все кортежи уникальны) $\{ \forall$ отн-я $\exists$ первичный ключ, и нет неопр. значений его атрибутов }

Неопр. зн-е: аном. оп-ция с ним — неопр. зн-е ($NULL \oplus \dots = NULL$)
лог. оп-ция с ним — UNKNOWN ($NULL \odot \dots = UNKNOWN$)

! UNKNOWN = UNKNOWN

$F \wedge U = F$

$T \vee U = T$

$F \vee U = U$

$T \wedge U = U$

2) ссылочная целостность

Атрибут может быть внешним ключом, т.е. первичным ключом в другом отношении $\Rightarrow$ возникает "ссылка", возможно составная (из неск-х атр-в)

Если в кортеже опр-н внешний ключ, то в теле кортежа, на который ссылается, он определен, либо в иск кортеже вторичный ключ NULL.

Потребуется удалить ссылку $\Rightarrow$

1) запретить удаление

2) NULL Default для вторичного ключа

3) Хранить все ссылающиеся объекты до удаления ссылки

Язык реляционно полон, если $\forall$ простая операция выражается 1-м оператором.

Большее число - больший приоритет

Алгебра Поле
 (издающая:
 $A \cap B = A \cap B$)

- ① UNION
- ② INTERSECT
- ① MINUS
- ② TIMES (рек. произв-е)
- ③ Where (огранич.)
- ③ Project (проектир.)
- ② Join (соединен.)
- ② Divide by (дел. дел.)
- Rename (переим.)
- ④ $\{ := \}$ (присв-е)

U
 $\cap$
 $\setminus$
 $\otimes$

{ гориз. выборка по условию }
 { по набору атрибутов }
 { агрегация }
 $A \cap B \setminus C = A$

$\{ := \}$ не может у-ть в отн-х

{ < имя-атр, имя-тип-домена > }

Rename - переименовывает имя-атрибута в случае конфликтов

Если мн-ва $A\{a\}$ и $B\{b\} \Rightarrow C\{c\} = A\{a\} \cup B\{b\} \Leftrightarrow \forall c \exists a: a=c \vee \exists b: b=c$
 и $\forall a \exists c: c=a \vee \forall b \exists c: c=b$

$C\{c\} = A\{a\} \cap B\{b\} \Leftrightarrow \forall c \exists a: a=c \wedge \exists b: b=c$
 $\forall a, b: a=b \Leftrightarrow \exists c: a=b$

$C\{c\} = A\{a\} \setminus B\{b\} \Leftrightarrow \forall c \exists a: a=c \wedge \nexists b: b=c$
 $\forall a: \nexists b: b=a, \exists c: a=c$

Пр-1

N	Имя	Зарп	Нотд
34	Ив-б	220	10
35	Па-б	220	10
36	Сид-б	180	13

Пр-2

N	Имя	Зарп	Нотд
34	Ив-б	220	10
35	Па-б	330	10
40	Сид-б	200	15

Проекты
 Пр-N
 1.
 2

Пр-рук
 Ив-б
 Па-б

UNION

N	Имя	Зарп	Нотд
34			
35			
36			
40			

INTERSECT

34			
35			

Пр-1 MINUS Пр-2

36			
----	--	--	--

Заголовок отн-х совп-ют

$A \{a\}, B \{b\} \Rightarrow C \{ \langle c_1, c_2 \rangle \} = A \{a\} \text{ Times } B \{b\}$
 $\forall a \in A, b \in B \exists \langle c_1, c_2 \rangle \in C$
 $c_1 = a, c_2 = b$
 $\forall \langle c_1, c_2 \rangle \in C \Rightarrow \exists a \in A, b \in B$
 $a = c_1, b = c_2$

пара $\langle c_1, c_2 \rangle$ преобр-ся в единств. кортеж, причем заголовки не пересекаются

Пр-1 Times Проекты

N	Line	Sapn	Notd	Пр-N	Пр-Рук
34	UB-6	~	~	1	UB-6
34	UB-6	~	~	2	Пр-6
35	Пер-6	~	~	1	UB-6
35	Пер-6	~	~	2	Пер-6
36	Сед-6	~	~	1	UB-6
36	Сед-6	~	~	2	Пер-6

Where.

2 арг-та: A Where Comp

↑
одноотн-е

↑
неот.хл-е: набор простых усл-ий объедин. логич. и простое усл-е: $a \text{ op } b$, где a, b арг-ты $a \text{ op } b \in \{=, <, >, \leq, \geq\}$

Рез-т - все кортежи из A, удовлетв. Comp

$A \text{ Where } (C1 \text{ AND } C2) = A \text{ Where } (C1) \text{ INTERSECT } A \text{ Where } (C2)$

$A \text{ Where } (C1 \text{ OR } C2) = A \text{ Where } C1 \text{ UNION } A \text{ Where } C2$

$A \text{ Where } (!C1) = A \text{ MINUS } A \text{ Where } C1$

Пр-1 Where Notd=PB

N	Line	Sapn	Notd
36	Сед-6	~	~

Project

Project $A \{a_1, \dots, a_n\}$
 $\uparrow$
 атрибуты - подмножество A

Project $Pr-1 \{k_1, \dots, k_n\}$
 $\uparrow$
 номера отсчетов, ведущих проект 1

Join (соединение на $\otimes$)

$A \text{ Join } B \text{ where Comp} (= (A \text{ Times } B) \text{ where Comp})$

Случае				Проекты		
Сл-№	Сл-имя	Сл-Зарп	Сл-Пр№	№	Пр-рук	Пр-Зарп
34	ИВ	220	1	1	ИВ	220
35	Пет	330	1	2	Пет	220
36	ИВ	180	1			
34	ИВ	220	2			
35	Пет	330	2			
40	Фед	200	2			

Случае Join Проекты where (Случ-Зарп == Пр-Зарп)

Сл-№	Сл-имя	Сл-Зарп	Сл-Пр№	№	Пр-рук	Пр-Зарп
34	ИВ	220	1	1	ИВ	220
34	ИВ	220	1	2	Пет	220
34	ИВ	220	2	1	ИВ	220
34	ИВ	220	2	2	Пет	220

Если $\text{Comp} = \text{Comp}' \text{ AND } (a \neq b) \Rightarrow$ действит. объединение

Если $\text{Comp} = \text{Comp}' \text{ AND } (a = b) \Rightarrow$ эквивалентное

Соединение через внешний ключ - естественное соединение

(Natural Join = Rename Вн-ключ $\rightarrow$ Join по вн. ключу $\rightarrow$ Project (вн. ключ удалить))

DIVIDE BY

(A Times B) Divide by B = A

Адреса А (Дет и Дарвин)

<AND> / <OR> релю. конъюнк. / дизъюнк.
 <NOT> - релю. доп. / инверсия
 Remove - удал. атриб.

Г - атриб.
 А - атриб.
 Т - тип
 V - значение атриб.
 H_r - заголовки {<A, T>}
 t_r - кортежи {<A, T, V>}
 B_r - тело отн-я {t_r}

- 1) <NOT>
- 2) <Remove>
- 3) <Rename>
- 4) <AND>
- 5) <OR>
- 6) :=

1) H_s = H_r (s-результатирующий, r-исходный)
 B_s = {t_s : ∃ t_r, t_r ∈ B AND t_s = t_r}

Рек-но		<Not> Руководство	
N	Имя	N	Имя
1	ИВ-Б	1	Петров
2	Петров	2	Иванов
		3	Иванов
		3	Петров

Домена {1, 2, 3} {Иванов Петров}

2) r <Remove> A ⇒ огранич.: ∃ T: <A, T> ∈ H_r
 H_s = H_r \ {<A, T>}
 B_s = {t_s : ∃ t_r, ∃ T: t_r ∈ B_r, ∃ T, <A, T, V> ∈ t_r, t_s = t_r \ {<A, T, V>}}

[~ Project]				[~ Remove]			
Случающие	N	Имя	3-п	NP	Случающие	NP	<Remove> 3-п
	1	ИВ-Б	220	1		1	ИВ-Б
	2	П-Б	330	1		2	П-Б
	3	ИВ-Б	180	1		3	ИВ-Б
	1	ИВ-Б	220	2		4	Петров
	2	П-Б	330	2			
	4	Петров	200	2			

3) $r \leftarrow \text{Rename}(A, B)$

$$\exists T: \langle A, T \rangle \in H_r$$

$$\exists T: \langle B, T \rangle \in H_r$$

$$H_s = \{ H_r \setminus \{ \langle A, T \rangle \} \cup \{ \langle B, T \rangle \} \}$$

$$B_s = \{ t_s : \exists t_r, \exists s : t_r \in B_r, s \in T, \langle A, T, s \rangle \in t_r \Rightarrow$$

$$t_s = t_r \setminus \{ \langle A, T, s \rangle \} \cup \{ \langle B, T, s \rangle \}$$

4) $r_1 \leftarrow \text{AND} \rightarrow r_2$

$$\forall A: \langle A, T_1 \rangle \in H_{r_1}, \langle A, T_2 \rangle \in H_{r_2} \Rightarrow T_1 = T_2$$

$$H_s = H_{r_1} \cup H_{r_2}$$

$$B_s = \{ t_s : \exists t_{r_1}, \exists t_{r_2} : t_{r_1} \in B_{r_1}, t_{r_2} \in B_{r_2} \Rightarrow t_s = t_{r_1} \cup t_{r_2} \}$$

$$1) \{ \langle A_1, T_1, s_1 \rangle \} \cup \{ \langle A_2, T_2, s_2 \rangle \} = \{ \langle A_1, T_1, s_1 \rangle, \langle A_2, T_2, s_2 \rangle \}$$

$$2) \{ \langle A, T, s_1 \rangle \} \cup \{ \langle A, T, s_2 \rangle \} = \emptyset$$

we $s_1 \neq s_2$

$$3) \{ \langle A, T, s_1 \rangle \} \cup \{ \langle A, T, s_1 \rangle \} = \{ \langle A, T, s_1 \rangle \}$$

$$\Rightarrow (i) \text{ } H_{r_1} \cap H_{r_2} = \emptyset \Rightarrow \text{Times} \Leftarrow \text{AND}$$

$$(ii) \text{ } H_{r_1} \neq H_{r_2} \Rightarrow \text{Intersect} \Leftarrow \text{AND}$$

$$(iii) \text{ } H_{r_1} \cap H_{r_2} \neq \emptyset \text{ AND } H_{r_1} \neq H_{r_2} \Rightarrow \text{Natural Join} \Leftarrow \text{AND}$$

Czymaque				Procesor		
N	Ch-klm	3-17	Npp	Npp	UmRyk	
1	UB-6	220		1	UB-6	
2	Per-6	330	1	2	17-6	
3	CU-6	180				
4	UB-6	220	2			
5	17-6	330	2			
6	Per-6	200	2			

$\Rightarrow$

Czymaque AND				Procesor		
N	Ch-klm	3-17	Npp	Npp	UmRyk	
1	UB-6	220	1	1	UB-6	
2	17-6	330	1	2	17-6	
3	CU-6	180	1		UB-6	
4	UB-6	220	2		17-6	
5	17-6	330	2		17-6	
6	Per-6	200	2		17-6	

Ch-17p1			
N	UmR	3-17	
1	UB-6	220	
2	17-6	330	
3	CU-6	180	

Ch-17p2			
N	UmR	3-17	
1	UB-6	220	
2	17-6	330	
4	Per-6	200	

Ch-17p1 AND Ch-17p2			
N	UmR	3-17	
1	UB-6	220	
2	17-6	330	

Ch-17p1			
N	UmR	3-17	
1	UB-6	220	
2	17-6	330	
3	CU-6	180	

Procesor			
Npp	UmRyk		
1	UB-6		
2	17-6		

Ch-17p1 AND Procesor			
N	UmR	3-17	Npp
1	UB-6	220	1
2	17-6	330	1
3	CU-6	180	1
4	UB-6	220	2
5	17-6	330	2
6	CU-6	180	2

10

5) $r_1 \text{ COR } r_2$

$\forall A: \langle A, T_1 \rangle \in H_1, \langle A, T_2 \rangle \in H_2 \Rightarrow T_1 - T_2$

$H_3 = H_1 \cup H_2$

$B_3 = \{t_3: \exists H_1, \exists H_2: t_1 \in B_1, t_2 \in B_2 \Rightarrow t_3 = t_1 \cup t_2\}$

(i) $H_1 = H_2 \Rightarrow \langle \text{COR} \rangle \Leftrightarrow \text{UNION}$

Проекты		№ Пр
Пр-Имя	Пр-Рис	N
A	UB-6	1
B	П-6	2
{A, B, B}	{UB-6, П-6}	{1, 2, 3}

$\Rightarrow$

Проекты $\langle \text{COR} \rangle$		№ Пр
Пр-Имя	Пр-Рис	N
A	UB-6	1
A	П-6	1
B	UB-6	1
B	П-6	1
B	UB-6	1
B	П-6	1
A	UB-6	2
A	П-6	2
B	UB-6	2
B	П-6	2
B	UB-6	2
B	П-6	2
A	UB-6	3
B	П-6	3

т.е. 6 стр-е $H_1 \neq H_2$ on-unique
 соединяется

Полнота антефронт.

- 1) Rename $\equiv \langle \text{Rename} \rangle$
- 2) $:= \equiv :=$
- 3) Project $\equiv \langle \text{Remove} \rangle$
- 4) Union $\equiv \langle \text{OR} \rangle$
- 5) Intersect $\leftarrow \text{AND}$
- 6) Times

- 7) Minus
- 8) Where
- 9) Join
- 10) Divide by

Ан-Кол
 Her

I) $r_1 \text{ Minus } r_2 = r_1 \langle \text{AND} \rangle \langle \text{NOT} \rangle r_2$

Схема Пр1

N	Имя	ЗП
1	UB-6	220
2	П-6	330
3	UB-6	180

Сх-Пр2

N	Имя	ЗП
1	UB-6	220
2	П-6	330
3	UB-6	200

N	Имя	ЗП
3	UB-6	180

$\langle \text{AND} \rangle \langle \text{NOT} \rangle$ Сх-Пр2

N	Имя	ЗП
2	UB-6	220
3	UB-6	220

WAŻNE

NOW

8) r where comp

Całkowicie Where (Całkowicie - 300 = 220)

300-220

300
220

Całkowicie <AND> 300-220

N	Umi	300
1	46-6	220

N	Umi	300
1	46-6	220
2	17-6	330
3	60-6	180

N	Umi	300	NP
1	46-6	220	1
2	17-6	330	2
3	60-6	180	1

Całkowicie Where $N = NP$

N	NP
1	1
2	2
3	3

<AND> Całkowicie =

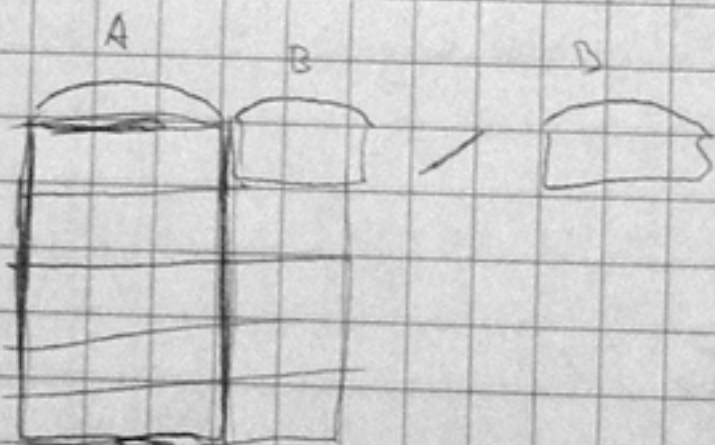
N	Umi	300	NP
1	46-6	220	1
2	17-6	330	2

9) Join $\rightarrow$ (A Times B) where Comp
 2 AND 2 AND

10) Divide by 1 $r1 = \{A, B\}$ $r2 = \{B\}$

$r1$ divide by $r2 = (r1 \text{ project } A) \text{ Minus } (r1 \text{ project } A)$
 (remove) (AND) (NOT) (remove) B

Times $r2$ Minus $r1$ project A
 (AND) (AND) (NOT) (remove) B



N	Umi	NP
1	46-6	1
2	17-6	2
3	60-6	1
1	46-6	2
2	17-6	2
4	60-6	2

N	NP
1	1
2	2

=

N	Umi
1	46-6
2	17-6

Можно избавиться от <Rename>

A			B = const		
N	имя	Np	N	np	PrN
1	LIB	1	1		1
2	П-б	1	2		2
3	Суд-б	1	3		3
4	Суд-б	2			

A <AND> B =>

N	имя	Np	PrN
1	LIB		
2	П-б		
3	Суд-б		
4	Суд-б	2	2

<remove>

кроме того справедливо:

$$A <AND> B = <NOT> A <OR> <NOT> B$$

$$A <OR> B = <NOT> A <AND> <NOT> B$$

$$S_H(A, B) = NOT A OR NOT B$$

$$P_i(A, B) = NOT A AND NOT B$$

$$NOT = S_H(A, A)$$

$$AND = \dots$$

$$OR = \dots$$

Реляционное исчисление.

Служ. = (CN, СШ, СВЗРП, СШПР) Пр = (ПН, ПШ, ПЗАРП)
 (Служ Join Пр where (CN = ПРШКН) where (ПЗАРП > 200)) Project(CN, СШ)
 { Вывести рук-лей с з-ом > 200 }

RANGE Служ is Служащие
 RANGE Пр is Прокеты

Служ. CN, Служ. имя Where Exist Пр (Служ. CN = Пр. ПРШКН AND Пр. ПЗАРП > 200)

Штисненне кортцей.

Кортцная пер-х: $RANGE \langle \text{имя пер.} \rangle \text{ IS } \langle \text{имя атт-а} \rangle$

$\langle \text{зн. атт.} \rangle = \langle \text{имя пер.} \rangle \cdot \langle \text{имя атт.} \rangle - \text{фр. к зн-ю атрибута переменной}$

WFF - таб. логич. формула

Простая формула $\langle \text{зн-е атт.} \rangle = \text{const}$
 $\langle \text{зн-е атт.} \rangle = \langle \text{зн. атт.} \rangle$

Рез-т WFF - набор кортцей, удовл-ляющих операционным усл-иям (критериям усл-ий).

	Сум-с	Сл-Н	Сум-Сл-к	Сум-Зарп	Пр-Н
Not	1	1	1	210	1
AND	2	1	1	330	1
OR	3	1	1	180	1
If Then (импликация)	1	1	1	220	2
(if A then B)	2	1	1	330	2
	4	1	1	200	2

$RANGE \text{ Сл} \text{ is } \text{Сл-к-с-е}$
 if $\text{Сл. Сл-к} = \text{"Иванов"}$ then $((\text{Сл. СЗарп} > 200) \text{ AND } (\text{Сл. СПр-Н} = 1))$

$RANGE \text{ Пр} \text{ is } \text{Проекты}$
 $(\text{Сл. Сл-к} = \text{Пр. Пр-рук})$

Сл-СН	Сл-Сл-к	Сл-СЗарп	Сл-Пр-Н	Пр-Н	Пр-рук
1	1	220	1	1	1
1	1	220	2	1	1
1	1	200	2	2	1

кванторы $\{ \text{Exist} \equiv \exists \text{ (существование)}$
 $\text{For All} \equiv \forall \text{ (всеобщность)}$

Пер-х после квантора - связанная, все ост-е - свободные.

$RANGE \text{ Сл} \text{ is } \text{Сл-к-с-е}$
 $RANGE \text{ Сл} \text{ is } \text{Сл-к-с-е}$

$\text{Exist } \text{Сл} \text{ (Сл} \text{1} \cdot \text{СЗарп} > \text{Сл} \text{2} \cdot \text{СЗарп}) \text{ (} \exists \text{ Сл} \text{2 : } \exists \text{ Сл} \text{1 : } \text{Сл} \text{1} \cdot \text{СЗарп} > \text{Сл} \text{2} \cdot \text{СЗарп)}$

Рез-т - на основе вкл пер-х.
 СН Сл-к СЗарп СПр-Н
 } все кроме Сл-к-с-е и СЗарп min

For All C_{u2} ($C_{u1}.C_{zapn} > C_{u2}.C_{zapn}$)
 / Для C_{u1} $\forall C_{u2}$: $C_{u2}.C_{zapn} < C_{u1}.C_{zapn}$, т.е.
 р-т - только так заплата $\Rightarrow$ только 2 строки с "Петров"

Target list - те переменные (вместо в виду ар-та) рез-та
 - var.attr
 - var
 - новое имя := var.attr

Опр. Выражение релец. на-т корректней:

$\langle \text{target list} \rangle$ Where $\langle WFF \rangle$

Range C_{u1} , C_{u2} is $C_{u1} \text{ same as } C_{u2}$
 Range Pr is $Пректор$

$C_{u1}.CN, C_{u1}.C_{имя}$ Where For all Pr EXIST C_{u2} ($C_{u1}.CN = C_{u2}.CN$ AND
 $C_{u2}.PrN = Пректор.PrN$)

$\Rightarrow$ $C_{u1} \text{ same as } C_{u2}$ Divide by $Пректоры$

Условие Доменов.

Формула гленства: $\{ \text{наименование} \text{ эл-та в ин-ве} \}$

$r, H_r \Rightarrow R(a_{i1}:V_{i1}, a_{i2}:V_{i2}, \dots)$

Сумма (CN:1, CИмя: Иванов, CЗарп:220, CНап:1)
 (Значение можно не писать, тогда ф-е с ним не прав-ся)

$CN, CИмя$ Where EXIST $CЗАРП1$ ($CИмя \text{ same as } CЗАРП1$) AND
 $CИмя \text{ same as } (CЗАРП, CN, CИмя)$ AND ($CЗАРП1 > CЗАРП$)

Нормальная форма - набор требований на отн-е.

Нормальные формы:

- 1-НФ (1NF)
- 2-НФ (2NF)
- 3-НФ (3NF)
- Бойс-кота (BCNF)
- 4НФ
- 5НФ (PJ/NF)

Опр. Функциональная зав-ть - если ботн-и $r \in H_r$ атр-ты $X, Y \in H_r$ (заголовок), то r, Y функц. зависит от r, X , если $\forall X$ соотв-т $!Y$. (Означ-е $r.X \rightarrow r.Y$)

CN	C _{лих}	C _{зари}	C _{прN}	C _{прN}
1	лих-в	220	1	лих-в
2	пр-в	330	1	лих-в
3	лих-в	180	1	лих-в
4	пр-в	240	2	пр-в

$CN \rightarrow C_{лих}$
 $CN \rightarrow C_{зари}$
 $CN \rightarrow C_{прN}$
 $CN \rightarrow C_{прN}$
 $(CN, C_{лих}) \rightarrow C_{зари}$
 $CN \rightarrow (C_{лих}, C_{зари})$
 $C_{прN} \rightarrow C_{прN}$
 $C_{лих} \rightarrow C_{зари}$

оф-ц-ност.
ности БД

$A_{клет} \Rightarrow \forall B \in H_r \exists A_{клет} \rightarrow B$ (клеточной - и первичности, и возможности)

Опр. Функц. зависимость транзитивна, если $A \rightarrow B$ и $B \rightarrow C$.

Опр. Если S - мн-во функц. зависимостей, то замыканием S наз-т мн-во S^+ функц. зав-тей, выводимых из S .

$$X \rightarrow (Y, Z) \Rightarrow X \rightarrow Y, X \rightarrow Z$$

$$X \rightarrow Y, Y \rightarrow Z \Rightarrow X \rightarrow Z$$

(Зависимость $A \rightarrow C$ транзитивная, если $\exists B: \exists A \rightarrow B, \exists B \rightarrow C, \nexists C \rightarrow A$)

Аксиомы Армстронга: $\{$ в строгом смысле - не аксиомы! $\}$
 Γ содержит атр. A, B, C

- 1) $B \in A \Rightarrow A \rightarrow B$ рефлексивность
- 2) $A \rightarrow B \Rightarrow AC \rightarrow BC$ пополнение
- 3) $A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C$ транзитивность

д-во:

1) очевидно

2) $\Gamma AC \nrightarrow BC \Rightarrow \exists t_1, t_2$ - интерпретации: $t_1 \models AC, t_2 \models AC$
 $t_1 \models BC, t_2 \models BC$
 $\exists AC \rightarrow A \Rightarrow t_1 \models A, t_2 \models A$
 $t_1 \models B, t_2 \models B \Rightarrow t_1 \models BC, t_2 \models BC$

3) $A \rightarrow C \Rightarrow t_1, \{A\} = t_2 \{A\} \quad t_2 \{C\} \neq t_1 \{C\}$

то $A \in t_1, \{B\} = t_2 \{B\}$
 $B \rightarrow C \quad t_1 \{C\} = t_2 \{C\}$

Система Аристотеля логика, т.е. все $\Rightarrow$ -ты S^+ выводятся по ним.
 Действительно к 1-3 следующие:

- 4) $A \rightarrow A$ самодетерминация
- 5) $A \rightarrow BC \Rightarrow A \rightarrow B, A \rightarrow C$ декомпозиция
 { по 1) $\exists BC \Rightarrow B, BC \Rightarrow C \Rightarrow$ т.к. $A \rightarrow BC$ тогда 3) $A \rightarrow B, A \rightarrow C$ }
- 6) $A \rightarrow B, A \rightarrow C \Rightarrow A \rightarrow BC$ объединение
 { по (2) $A \rightarrow AB, AB \rightarrow BC \Rightarrow A \rightarrow BC$ }
- 7) $A \rightarrow B, C \rightarrow D \Rightarrow AC \rightarrow BD$ композиция
 { по (2) $AC \rightarrow BC, BC \rightarrow BD \Rightarrow AC \rightarrow BD$ }
- 8) ~~накопление~~
 $A \rightarrow BC, B \rightarrow D \Rightarrow A \rightarrow BCD$ (накопление)
 { $BC \rightarrow BCD \Rightarrow A \rightarrow BCD$ }

Γ -отн-с, Z -мн-во атр, $z \in H_r, S$ -мн-во функ. Γ -тел
 Опр. Замыкание $Z^+ = \bigcup_{i=0}^{\infty} Z_i$ - $Z_0 = Z, Z_{i+1} = Z_i \cup \{y \mid \exists x \in Z_i, x \rightarrow y, y \in S^+\}$

Алг-м вывода Z^+
 $Z^+ = Z$
 Do
 $Z_{prev} = Z^+;$
 For Each $A \rightarrow B \in S$ do
 if $A \in Z^+$ then $Z^+ = Z^+ \cup B$
 end Do
 Until $Z^+ = Z_{prev}$

Пример: $H_r = \{A, B, C, D, E, F\}$
 $S = \{A \rightarrow D, AB \rightarrow E, BF \rightarrow E, CD \rightarrow E, E \rightarrow C\}$
 $\{AE\}^+ = ? \quad \{AED\}, \{AEDC\}, \{AEDCF\} = \{AE\}^+$ AE не м.б. кнопом

Заметим: $Z \rightarrow B \in S^+ \Leftrightarrow B \in Z^+$

Опр. Суперкноме - мн-во атр-б: в него входит все возможные или первичный кноме
 $\Leftrightarrow \forall A \in H_r \exists k \rightarrow A$
 $\Leftrightarrow k^+ = H_r$

Опр. S_2 - покрытие S_1 - мин-во ф.зав-тей, если $\forall y$ то $y \in S_1$ валиден
 из S_2 . ($S_1 \subseteq S_2$)
 S_1 эквивалентно S_2 , если S_1 покр-е S_2 и S_2 покр-е S_1 ($S_1 \subseteq S_2$)

FD S минимально, если
 1) правая часть $\forall$ зависимости - атрибут
 2) левая часть: $\forall$ попытка уменьшить ее приводит к утрате S .
 3) из S нельзя выдрать зависимость без утраты S .

Обозначение мин. набора: S^-
 Напр-р: CN - CN-мин
 CN - CN-свои
 CN - CN-прН
 CN-прН $\Rightarrow$ CN-прН $\subseteq$ - минимально

$\forall S \exists S^- : S \sim S^-$

- 1) в S разбиваем зависимости с составной правой частью.
- 2) $\forall$ левая часть $\Rightarrow$ делаем попытку удалить атрибут(ы) (уменьшение детерминанта)
- 3) делаем попытку уменьшить кол-во зависимостей.

$H = \{A, B, C, D\}$

FD $S = \{A \rightarrow B, A \rightarrow BC, AB \rightarrow C, AC \rightarrow D, B \rightarrow C\}$

1) $A \rightarrow B$ $A \rightarrow C$
 2) $A \rightarrow D$ ($\textcircled{A} A \rightarrow C \Rightarrow A \rightarrow AC \Rightarrow A \rightarrow D$)
 ($\textcircled{AC} AC \rightarrow A \Rightarrow AC \rightarrow D$)
 3) $(A \rightarrow C, B \rightarrow C \Rightarrow AB \rightarrow C)$
 $(A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C)$
 $S^- = \{A \rightarrow B, A \rightarrow D, B \rightarrow C\}$

Ситуация	Знач.	ПрН	ПрРНК	$\Rightarrow$	R1	Natural Join	R2
N	или	Знач.	ПрН		N	или Join	N
34	или	220	1	или	34	или 220	34
41	сред-б	220	2	сред-б	41	ред-б 220	41

R1	Знач.	ПрН	ПрРНК	П
220	1	или		
220	2	ред-б		

R1 Natural Join $\neq$ R2

Теорема Хита.

Иск. отн-е имеет атр-ты $r(A, B, C)$ и имеем $A \rightarrow B$, то

$$r = \left(\text{PROJECT}(AB) \right) \text{ Natural Join } \left(\text{PROJECT}(AC) \right)$$

$$\Rightarrow \{a, b, c\} \in r \Rightarrow \{a, b\} \in r_1, \{a, c\} \in r_2 \Rightarrow \{a, b, c\} \in r_3$$

$$\Leftarrow \{a, b, c\} \in r_3 \Rightarrow \{a, b\} \in r_1, \{a, c\} \in r_2 \Rightarrow \{a, b, c'\} \in r$$

$$A \rightarrow B \Rightarrow B' = B$$

N	N(от)	N(пр)
34	30	1
41	31	1
34	30	2
41	31	2

Т Хита

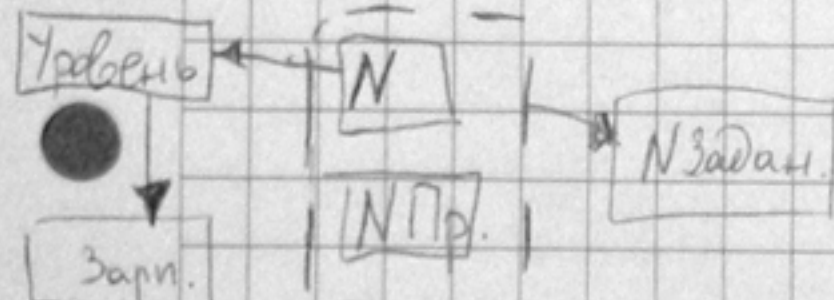
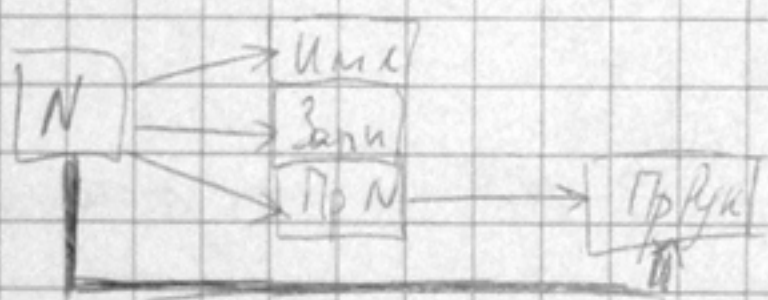
$\Rightarrow$

N	N(от)
34	30
41	31

N	N(пр)
34	1
41	1
34	2
41	2

$N \rightarrow N(от)$

Опр. Атр-т B имеет з-т от A, если вып-на мин. функц. зав-ть $A \rightarrow B$.

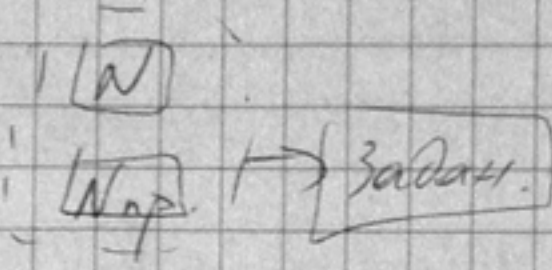
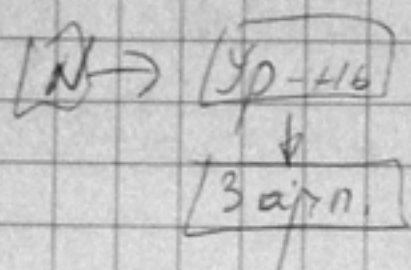


N	Ур-нь	З-т	N(пр)	N(задан.)
34	2	220	1	A
36	1	200	1	B
37	1	200	1	D
34	2	220	2	D
35	3	320	2	C

(*)

Опр. Аномалии обновления: 1) операция вставки Insert
 2) операция удаления Delete
 3) операция обновления Update

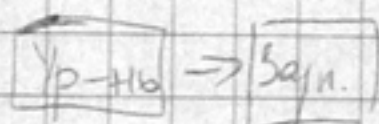
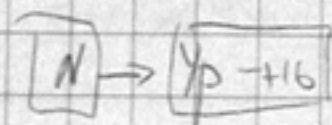
Вставка случается без N пр-та невозможна
 удалить 35 без потери св-й о з-те уровня 3 невозможно
 обновить уровень 34 можно в 2-х местах



N	Y_{p-116}	Z_{apn}
34	2	220
35	3	330
36	1	200
37	1	200

N	N_{pr}	Z_{apn}
34	1	A
35	2	D
36	2	C
37	1	B
37	1	D

Факт-ки превели к 2-НФ, т.е. иск. отн-е 1-НФ, а $\forall$ неавтономизар.
 зависит от него мин. образом.



N	Y_{p-116}	Y_{p-116}	Z_{apn}
34	2	1	200
35	3	2	220
36	1	3	200
37	1		

Факт-ки получили 3-НФ, т.е. иск. отн-е - 2-НФ и отсутствуют транзит.

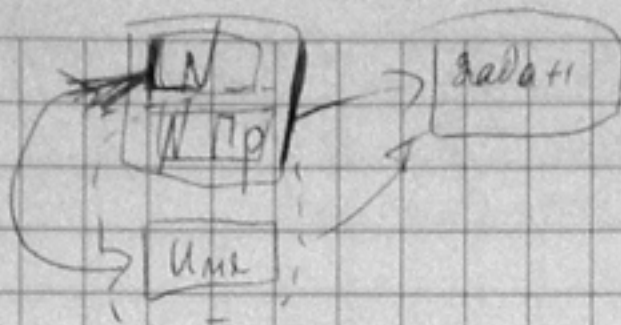
N	Y	Y3	V
YN	N3		- невозможно неавто. отн-е
Y3	3N		N3 даст потерю инф-ции

Теорема (Риссанена)

Проекции r_1 и r_2 иск. отн-я r неразделимы, если

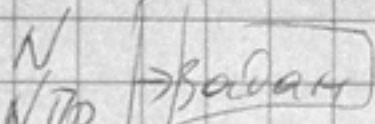
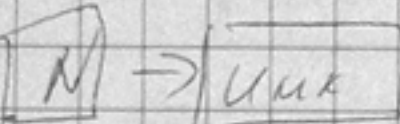
- 1) $\forall$ функц. z -то выводится из мин-ва функц. z -то r_1 и r_2
 $S_r \in (S_{r_1} \cup S_{r_2})^+$
- 2) H_{r_1} и H_{r_2} - классы или в H_{r_1} или в r_2 .

Вир. Отн-е отомарно, если его нельзя разбить на неавто. проекции.

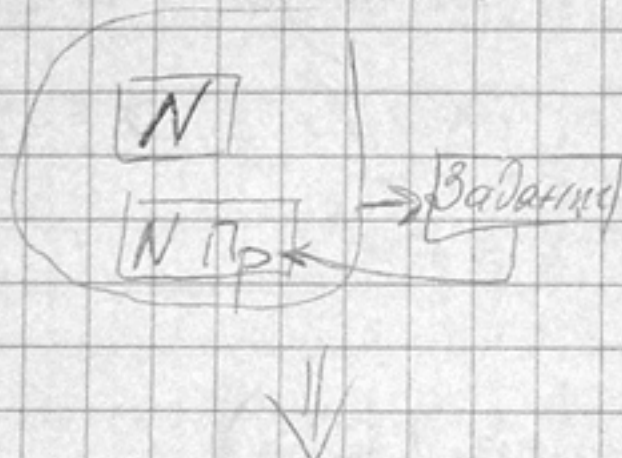


N	Лмс	N Пр	Задан.
34	уб-б	1	A
41	спед-б	2	B
34	уб-б	2	B
41	спед-б	1	A

BCNF (Босса - Корда) - нех-с отн-с 3-НФ
 и нетрив. мин. функц. зав-то в нел-бе
 детерминанта имеет функциональный имп.



Заданное связано с проектом.

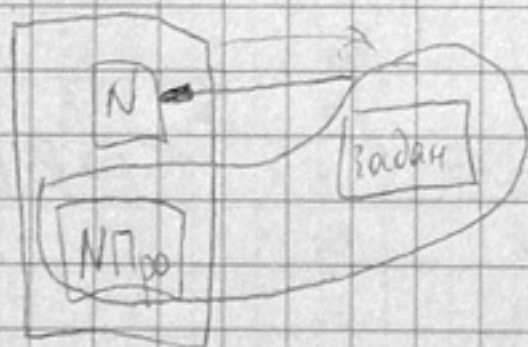


N	N Пр	Задан.
34	1	A
35	1	A
41	2	B
34	2	B
41	1	D

Невозможно
 ур-о данные
 в ситуации,
 но темк свед-
 ний о задании

N	Задан
Задан	N Пр

BCNF, но по Кассанена разбиение не обязательно



N	N Пр	Задан
34	1	A
34	1	B
34	2	A
34	2	B
41	1	A
41	1	D

(Во 4 проекте ситуация вын-т учитывать
 все наборы заданных)

if $(\langle N, N_{pr1}, s1 \rangle \in \Gamma) \text{ AND } (\langle N, N_{pr2}, s2 \rangle \in \Gamma)$
 Then $(\langle N, N_{pr}, s2 \rangle \in \Gamma) \text{ AND } (\langle N, N_{pr}, s1 \rangle \in \Gamma)$

Проекты	N	Задания	Случаи
N N пр		N Задан.	
34 1		34 A	
34 2		34 B	
41 1		41 A	
		41 D	

Роль в Фейнман: многозначная зависимость вида $N \rightarrow N_{pr}$
 $N \rightarrow \text{Задан}$
 $(N \rightarrow N_{pr} \mid \text{Задан})$

Опр. Многозн. зав-ть - если A опр. на опр. (A, B, C) , то A многозн. зав-ть $A \rightarrow B \Leftrightarrow B, \text{соотв.}$
 (A, C) зав-ть от A , но не зав-ть от C :
 if $(\langle a, b, c \rangle \in \Gamma \text{ AND } \langle a, b', c \rangle \in \Gamma)$
 Then $\langle a, b, c \rangle \in \Gamma$;
 $(A \rightarrow C) \langle a, b', c \rangle \in \Gamma$;

$\Gamma_i = (\Gamma \text{ Where } (\text{Сл} N = S \text{ AND } \Gamma N = H_i)) \text{ Project (Задан)}$

Заметим, $A \rightarrow B \Leftrightarrow A \rightarrow C$ ($\Rightarrow$ можно писать $A \rightarrow B/C$) - это
 утв-е есть Лемма Фейнмана

Если есть кружки, $\rightarrow$ -то $\Rightarrow$ есть многозначная

$A \rightarrow B \Rightarrow \langle a, b, c \rangle \in \Gamma, \langle a, b', c \rangle \in \Gamma \Rightarrow A \rightarrow B/C$
 $\uparrow$
 B
 Γ

Теорема: $A \rightarrow B \Leftrightarrow \Gamma \text{ PROJECT } (AB) \mid \Gamma \text{ PROJECT } (AC) = \Gamma$
 (Фейнмана)

а-во:

① $\leftarrow \langle a, b, c \rangle \text{ и } \langle a, b', c \rangle \in \Gamma \Rightarrow \langle a, b \rangle \in \Gamma_1, \langle a, c \rangle \in \Gamma_2 \Rightarrow \langle a, b, c \rangle \in \Gamma_3 \in \Gamma \Rightarrow$ по опр. имеем многозн. зав-ть

② $\rightarrow \langle a, b, c \rangle \in \Gamma \Rightarrow \langle a, b \rangle \in \Gamma_1, \langle a, c \rangle \in \Gamma_2 \Rightarrow \langle a, b, c \rangle \in \Gamma_3 \langle a, b, c \rangle \in \Gamma_3 \Rightarrow \langle a, b \rangle \in \Gamma_1, \langle a, c \rangle \in \Gamma_2 \Rightarrow \langle a, b, c' \rangle \in \Gamma \mid \text{многозн. зав-ть} \Rightarrow \langle a, b', c \rangle \in \Gamma$
 $\Rightarrow \langle a, b, c \rangle \in \Gamma$

★ (Перевернутая страница!)

GODZINA TIME 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
DZIEŃ DAY 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
MIESIĄC MONTH 1 2 3 4 5 6 7 8 9 10 11 12

Опр. Отн-е отн-ск к 4NF, когда иск-е отн-е 3NF, а все многозн. зав-ти стали функц. зав-тими с детер-минантом - возможным ключом.

Зависимости проекций соединений.
(Project Join Dependency) {PJD}

Опр. N-декомпозируемое отн-е - отн-е, которое нельзя декомпозировать менее чем на N частей

N	PrN	Задан	N	PrN	PrN	Задан	N	Задан
34	1	A	34	1	1	A	34	A
34	1	B	34	2	1	B	34	B
34	2	A	41	1	2	A	41	A
41	1	A						

N	J	
34	1	A
34	1	B
34	2	A
41	1	A
41	1	B

NJ

34	1	A
34	1	B
34	2	A
41	1	A

IF ($\langle H1P3 \rangle \in r$ $\langle H1P3 \rangle \in r$ $\langle H1P3 \rangle \in r$) Then $\langle NP3 \rangle \in r$

r задано на $(A, B, \dots, Z)$, то $\ast(A, B, \dots, Z) \Leftrightarrow r = r \{A\} \cap r \{B\} \cap \dots \cap r \{Z\}$

34	1	B	1	B	34	B	34	1
34	2	A	2	A	34	A	34	2
			1	A	41	A	41	1

? +41 1 A > нельзя просто удалить и добавить

Заметим, $A \Rightarrow B \Leftrightarrow \ast((AB), (AC))$

Опр. Зав-ть проекций соед. $\ast(A, \dots, Z)$ на-т подразум. возможн. ключами, если $\forall$ атр-т в ней - суперключ.

Опр. Зав-ть с полной системой г. тривиальной, если хотя бы 1 атр-т совп.

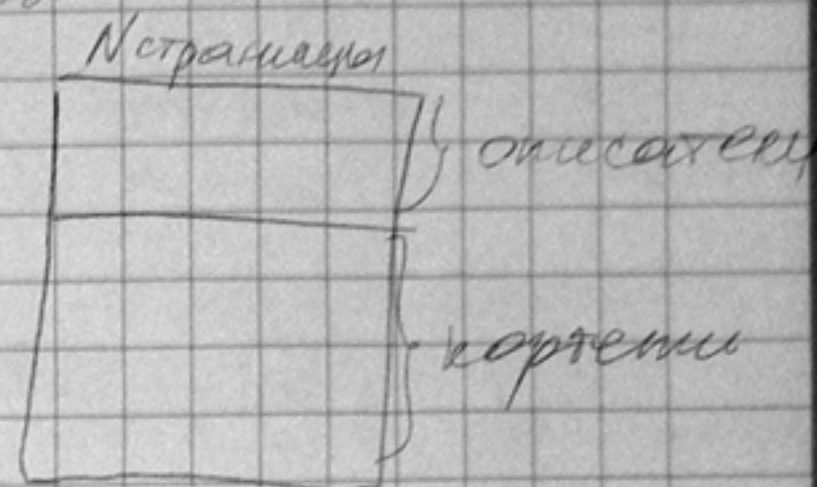
Опр. 5NF (P5NF). Отн-е - 5NF $\Leftrightarrow \forall$ нефик. з-ть проекций соед. подразум-се возм. ключами.

Важно! Проблемы аномалий обновления разрешаются до 5NF включ-но, 23 атр-е аномалии декомпозицией не разрешаются.

OLTP Online Transaction Processing
 OLAP Online Analytical Processing

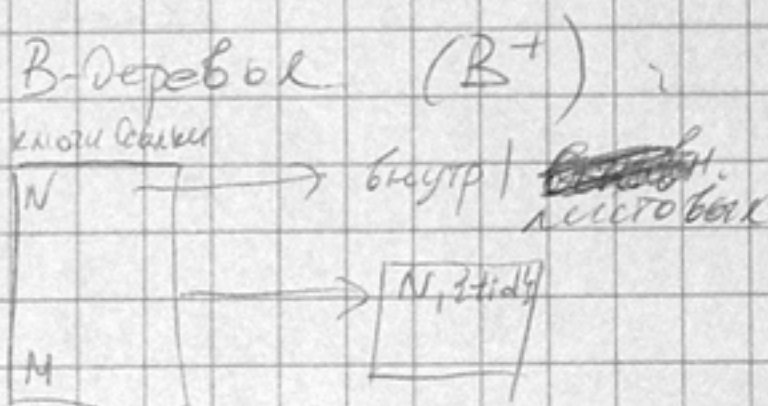
Структура внешней памяти.
 Принципы организации данных
 ≠ покортежная
 → по столбцовой

- 1) Отношения.
- 2) Связанные отношения
- 3) Индексы
- 4) Журнальная информация
- 5) Связанная информация.



tid - идентификатор кортежа ($\langle N_{стр}, N_{кортежа} \rangle$ - напр. р)

Индексы:



страницы → внутр.
 → ~~внешн. листовых~~

Алг-мы добавл-я
 удаление

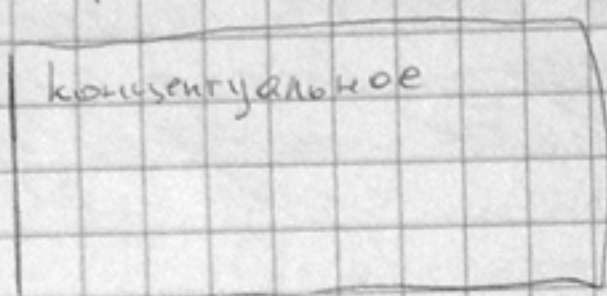
ссылки 362
 перебивание
 упоряд. разбиение

Можно покортежкам предв. упорядок-м данных
 Хэширование по ключу

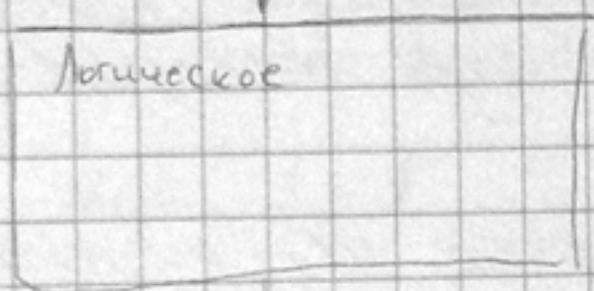
концептуальное проектирование БД.

Опр. Предметная область - часть реального мира, рассматриваемая в определенном смысле как целое, сведения о которой - информация, ресурсы в аспектах созд-я, управ-я БД.
 (Финансы, управ-е предпр-ем)

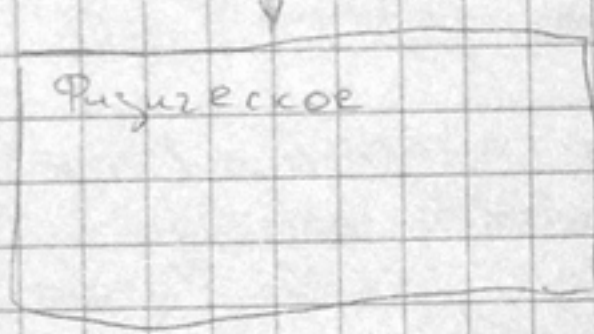
Проектирование БД.



Процесс созд-я модели использующей на предпр-и инфор-и, не зависящей от физич-ских асп-к ее представления



Процесс созд-я модели иссл-мой на предпр-и инфор-и с учетом выбранной модели организации данных, но независимой от типа целевой СУБД и других физич-ских асп-к реализации



Процесс описания реализации БД с учетом выбора целевой СУБД и указ-м структур хр-я и методов доступа, использ-х для организации форм-т обработки данных.

При исп-ии в проектир-ии ограни-то реал-е модели проявл-ся в следующих аспектах:

1. реал-е модель не предл-т какого-либо механ-ма для раздел-я существ-х объектов предм. области и связей между ними
2. модель не предл-т жест. средств для опис-я семантики (смысла данных)
3. во многих отл-х трудно осуществ-ть работу с плоскими табл-ми
4. не предост-т формализ-е средств для предст-я организацион-х и других зав-тей, на основе которых осущ-ся процесс проектирования
5. на ранних стадиях проект-я, как правило, треб-ся участие спец-ов, хорошо знающих предметную область, но не владеющих теорией БД.

Информационная модель - способ представления понятий или объектов
проблем. области, описываемых существ. Для данного
предст-я совокуп-ти объектов, их внутр-ей, внутр-ей и
вн-шней между ними.

как при инф. модели предст-я графическим - диаграммами.

Достоинства инф. модели:

1). простота и наглядность концепт. схематизм
более полно сценить специфику моделир-я проблем.
области и судить о возм-х ошибках на стадии проект-я
схем ВД.

2). инф. модель как P-виде широко напр-е. Диаграмм
с комментариями - важная документация, полезная
не только на стадии проект-я но и при эксплуатации,
совершенствовании и развитии.

3). на рынке представлены CASE системы, авт-ки проектир-я
диаграмм концепт. схем, вкл-е, спецификаци-
руемые как правило на языке SQL.
(Computer-Aided Software Engineering)

CASE-системы - системы поддержки технол-й автоматизации
проект-я, реализации, совершенствования сложных программ.
систем на всех этапах их жизн. цикла. Характерно
исп-е формальных средств для автоматизации генерации.

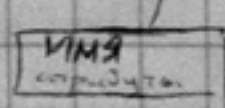
Информационная модель, сущность - связь
(Entity-Relationship, ER-модель, диаграмма Чена)

1976г. - П. Чен

Модель описывается на исп-и граф-х диаграмм.
Модель не стандартизована.

Основные понятия:

- сущность
- атрибут
- связь

Сущность - реальная или предст-я объект, инф-к о котором должна
сохран-ся и быть доступной. () ИМЯ

Атрибут - свойство, которое существует для уточнения и идентификации,
классификации, словесной харак-ки или формальной
сорт-к сущности.

(атрибуты)

Пример:

ЧЕЛОВЕК
ф.и.о.
дата рожд-я
пол

Связь - граф-ки изобр-е ассоциация, ут-е между двумя сущност-трами.

(в ER связи всегда 1:n или 1:1)

на каждом из концов связи опис-е или конца связи,

степень, обязатель-ть связи. (иногда вместо концы говорят роль)

Связь-множ-е. Мн-е, сущ-ти. 2 сущ-ности (или от сущ-ности к +гейсалин)

Обязат-е. конец-сильной, неопред-е. и неметир.

В месте стыковки связи с сущ-ностью или-е. или-е. одного

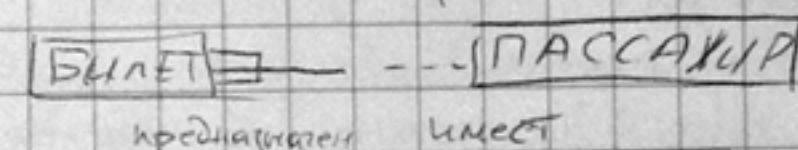
вход, если в связи должен ут-е (или может) один из-р сущ-ности,

мн-е трехот., если или-е. много из-р сущ-ности.

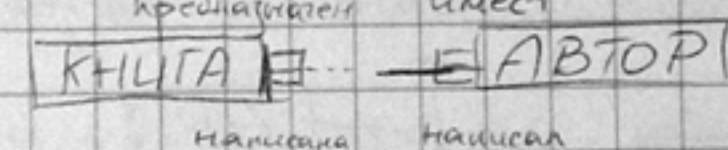
Связь 1 к 1



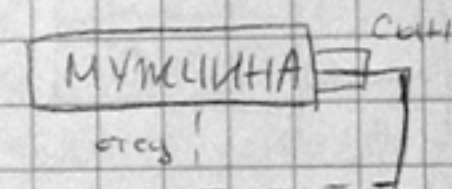
Связь 1 ко многим



Связь многие ко многим



Рекурсивная связь



Наследование сущностей.

Тип сущности может быть расширен на подтипы с общими атр-ми и связями. Тип сущ-ти на основе кот-го опис-е подтипы с сущ-ти. У подтипов могут б-е собств. атр. и связи. На практике дост-но 2-х типов подтипизации.

Простым типом жаз-е тип, чев-е. подтипом и не именован подтипов.

Особ-ти насл-е в ER:

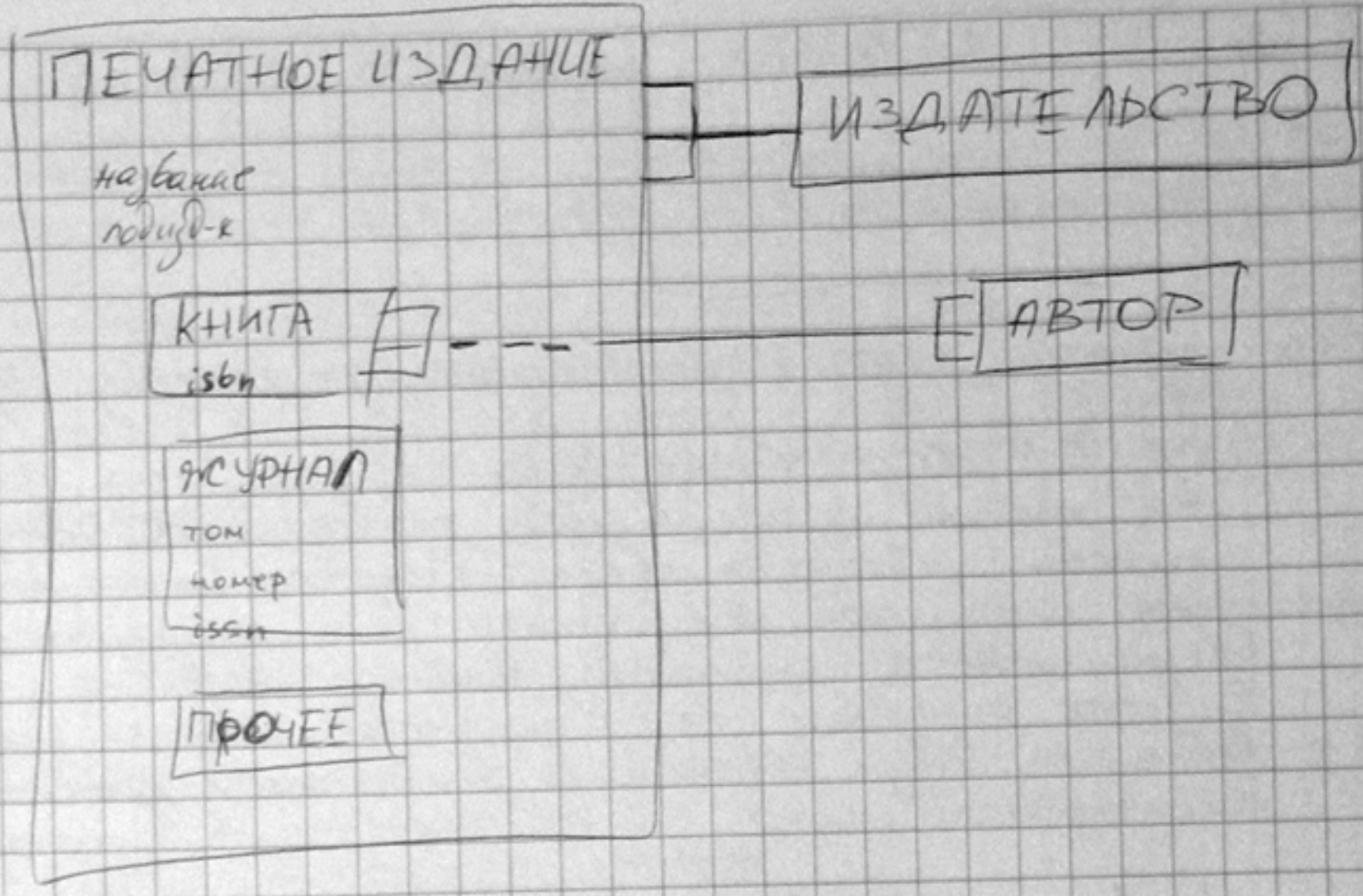
тип подтипы $A, B_1, \dots, B_n$ $\Rightarrow$ вын-ны 3 правила:

1. $\forall b \in B_i, i \in \overline{1, n} \Rightarrow b \in A$

2. отсутствие собств. жаз-ров у сущ-ти: $\forall a \in A \exists i: a \in B_i$

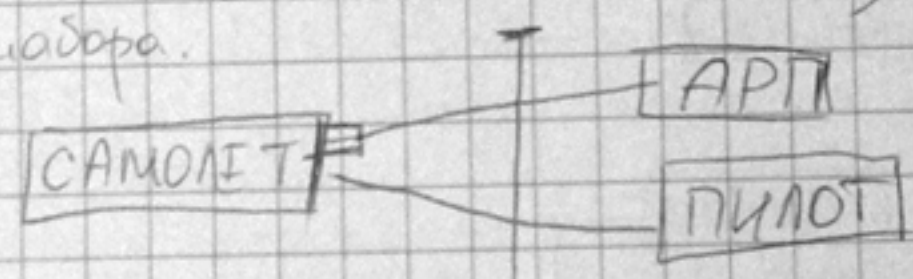
3. разб-ность подтипов: $\forall b \in B_i \Rightarrow b \notin B_j, j \neq i$

3) $\Rightarrow$ множественное насл-е невозможно



Взаимоскл. связи.
 Для каждой сущности можно описать набор связей: для экз. сущности может или должен 1 экземпляр только одной связи из данного набора.

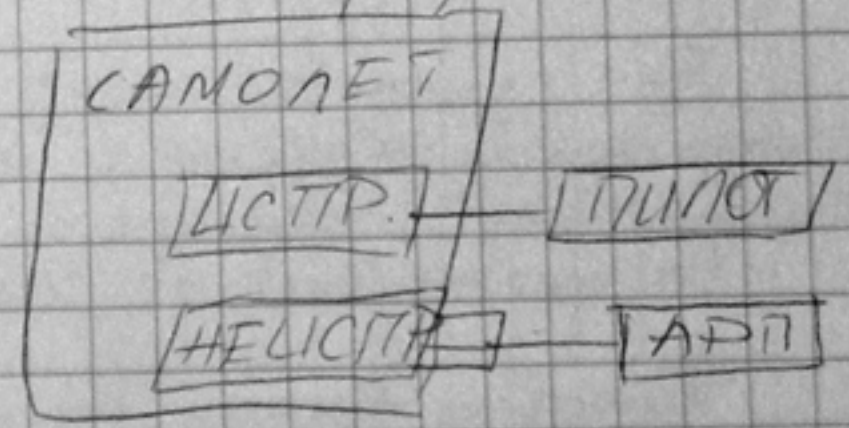
Пример:



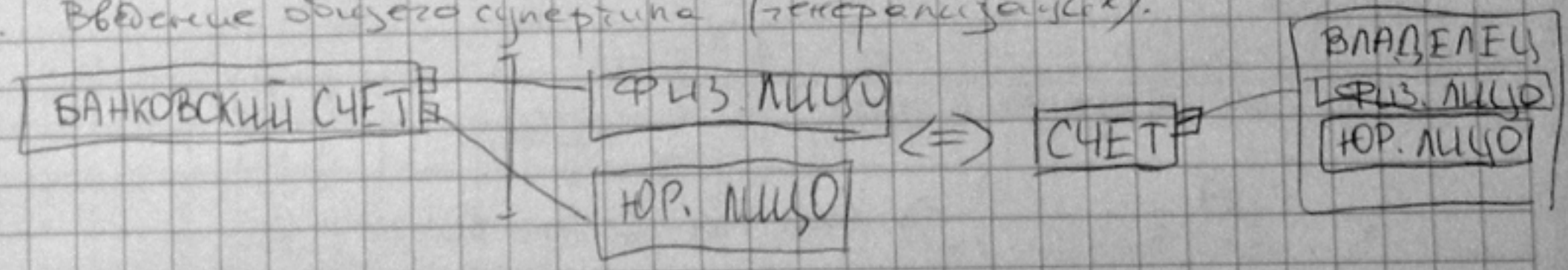
↑ либо пилотируется, либо на ремонте

В диаграмма со взаимоскл. связями может быть встроена таблица.

1) Введение подтипов (специализация)



2) Введение общего супертипа (генерализация)



Panta Plast

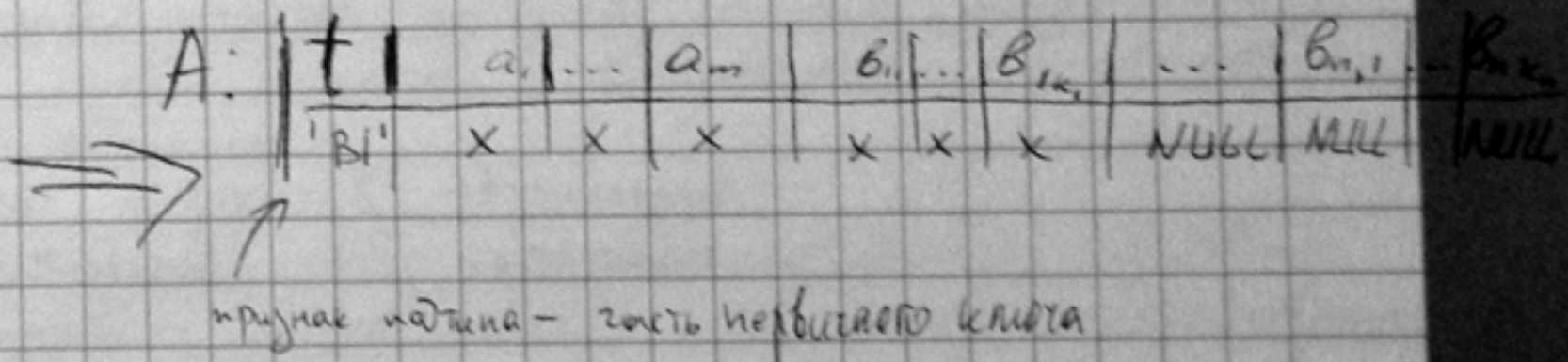
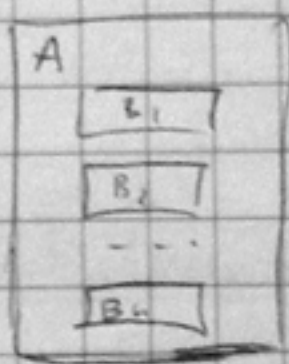
Уникалон. идентиф-ры сущн-ти.

При опред-ти сущн-ти необх-но гарант-ть, чтобы $\forall$ ее экз-р был-ся отличимым от $\forall$ другого экз-ра той же сущности.

В ER модели UID аб-ки не назван системой проек-к, а назн. с проек-ком на основе инд-х харак-к сущности $\Rightarrow$ UID может быть атрибут, комп-т атрибутов, связь, комп. связей, комп-т рекурсивн-х, уникально 1-й экз-р от других в пределах типа.

Получение реляц. схемы из ER-диаграмм.

- 1). $\forall$ простой тип превр-ся в таблицу: Имя сущности $\rightarrow$ Имя таблицы
Экземпляр $\rightarrow$ Строки
- 2). $\forall$ атрибут-столбец соотв. таблицы
- 3). компоненты UID становятся первичным ключом (если в его состав входят связи, то к нему добав-ся первичн. ключа добав-ся концы UID нахв-ся-ся - добавлен концы связи)
- 4). Связи 1 к 1 и многие к 1 стан-ся внешними ключами для связей многие к 1 в таблице соотв. сущности на концы многие - об-ся концы связи на концы "1" Неодн-е, связи допускают неопред-з-е. Обрат - нет
- 5). Для 1 к 1 вч. ключ добав-ся в 1-й у таблицы но не в проек-та.
- 6). Связь многие ко многим: сор-ся доп. табл. с как мн 2-мя столбцами в каждой из кот-х - UID A и B - сгрупп-ты на концах.
- 7). Прост. с сгрупп-ты подтипов: а) общая таблица всех подтипов



Цепочка подтипа B_i : PROJECT A $\{a_1, \dots, a_m, b_1, \dots, b_k\}$ where $t = 'B_i'$

5). По табл-це на $\forall$ подтип.

B_i : $\{a_1, \dots, a_m, b_1, \dots, b_k\}$

WAZNE

NOW

Собрать весь тип A: Project B, $\{a_1, \dots, a_m\}$ UNION ... UNION B, $\{a_1, \dots, a_m\}$

Project.

достоинства а)

- связь-е логики супертипов и подтипов
- связь-е простое доступа к экзemplу супертипа и не смешивать сложный к подтипам
- минимизация кол-во таблиц

недостатки а).

- усл-е профаниного кода (таблица 1, набор столбцов много)
- одна таблица - узкое место при многопольз. доступе
- расход памяти на хранение большого числа NULL-знач-ий

достоинства б).

- понятные правила работы с подтипами
- упрощение логики приложений

недостатки б).

- большое кол-во отдельных таблиц
- усл-е доступа к экзemplу супертипа

в). предст-е взаимосвязей

а). преобр-то диагр. к диагр. в наглядном виде, далее 7.

б). если взаимосвязь - "1 ко многим" и "многие ко многим" сущности, то 7 еще 2 способа.

I). объединение внешних ключей

(возможно прим-то, если внешние ключи все потенциально связ-ы таблиц от-ся к 1 типу или до-типу: созд-м 1 столбец - идент-типа связи 2 столбец(ы) - внешний ключ

II). разделение х-е внешних ключей

(созд-ся 2 столбца (возм-но сост-к), соотв-х соотв. связи и внешним ключам).

достоинства I): мин. число столбцов

недостатки I): усл-е вын-е оп-ции соедин-я

(A where A = B) JOIN B; where A.t = B.t. PK

↑
тип связи

↑
взаимосвязь

↑
первичный

достоинства II): упр-е соедин-я (Оно просто естественное)

недостатки II): большое число столбцов с NULL-значениями

Диаграммы классов в UML. (Unified Modeling Language)

Авторы UML: Р. Бух, Д. Румбах, А. Якобсон

1997г. - одобрение UML в качестве стандарта UML 1.0. | спецификация доступна на

2002г. - UML 2.0.

www.omg.org;

2009г. - UML 2.2.

UML позв-т модел-ть прогр-е, апп-е, прогр-ани. системы, системы, экв-ва вывол. действий пользов-ля.

Текущий станд-т - 12 типов диаграм, одно-х как статических, так и динам-е св-ва моделируемых систем. Для проекти-я реал-у, БД примени-м только диагр. классов (ран-то не в полном объеме).

IBM: Rational Rose (Rational)

Lapsody (Telelogic)

Portland Together

Objecteering

Umbrella

Основные понятия диаграмм классов.

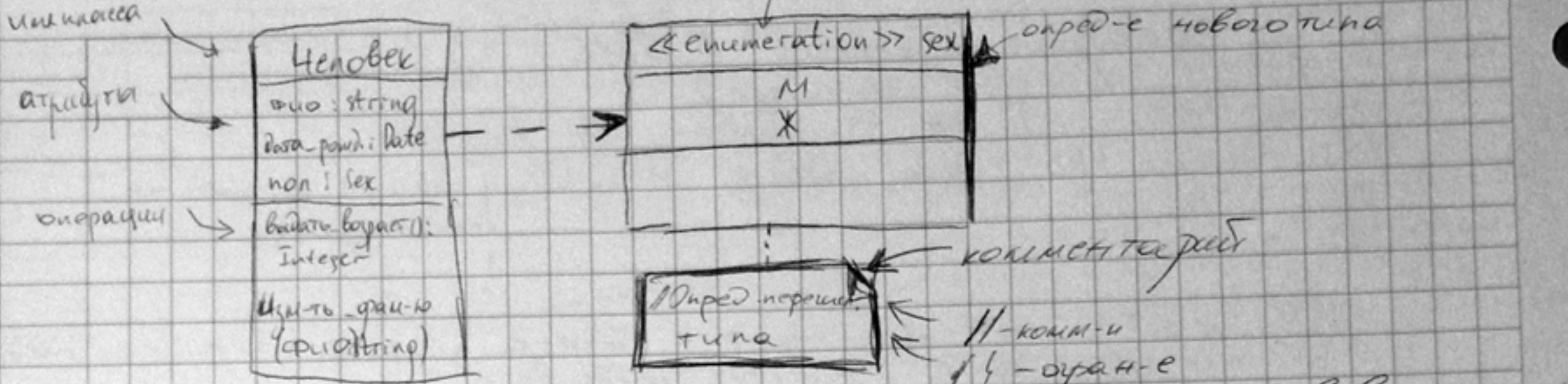
Диагр. классов UML опис-т типы объектов системы и разл. рода статические отно-я между ними. Кроме того там возможны коммент-ии и ограничения, неформальные (на естеств. языке) или формальные (на языке OCL {Object Constraint Language} - часть спецификации UML).

класс - именован-е опис-е совокуп-ти объектов с общ-ми атр-ми, опер-ми, связями и семантикой.

Атрибут - именованное св-во класса, опис-е мн-возмож-ей, которые могут принимать экземпляры этого св-ва. Атрибут - абстракция состояния объекта.

Операция - именованная услуга, которую можно запросить у объекта этого класса. Операция - абстракция поведения объекта.

Стереотип - механизм расширения семантики UML, позв-ет созд-ть новые тип-ы UML на основе существующих с учетом особ-тей решаемой задачи. В станд-те UML предум-н набор предопред-ленных стереотипов.



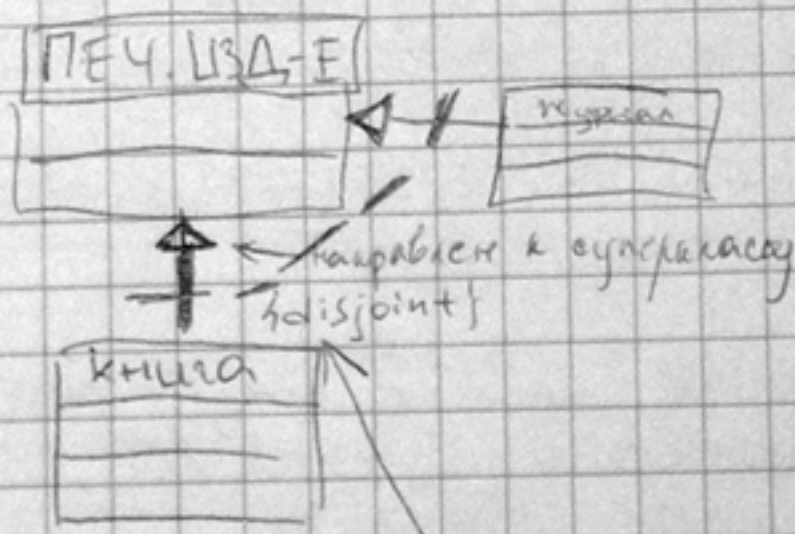
Чтобы чл-к, по описанию не менял сост-е объекта, указ-е ее св-во query (присутствует → не меняет)

Категории связей:

зависимость
 обобщение
 ассоциация

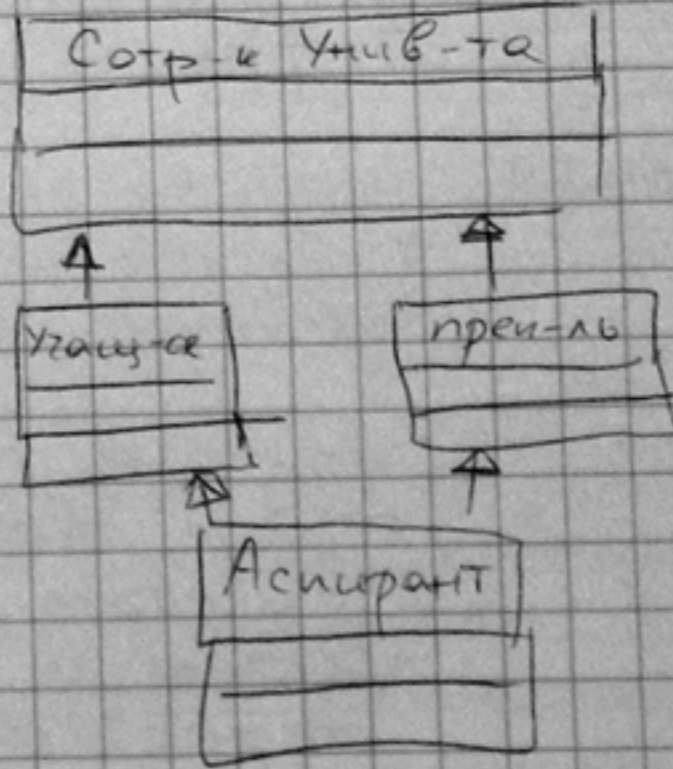
зависимость - связь по применению: если в специфич. экземпляре может повл-ть на повед-е класса, указ-е **нельзя** (обозначение —→) (Время, БД не прим-ся)

обобщение - связь между суперклассом (родителем, общей сущностью) и более специфич. разновидностями - подклассами или потомками.



Тип проче не нужен, т.к. возм-ны экземпляры "Печ. Узд-е" это можно запретить вставив `isabstract = true`

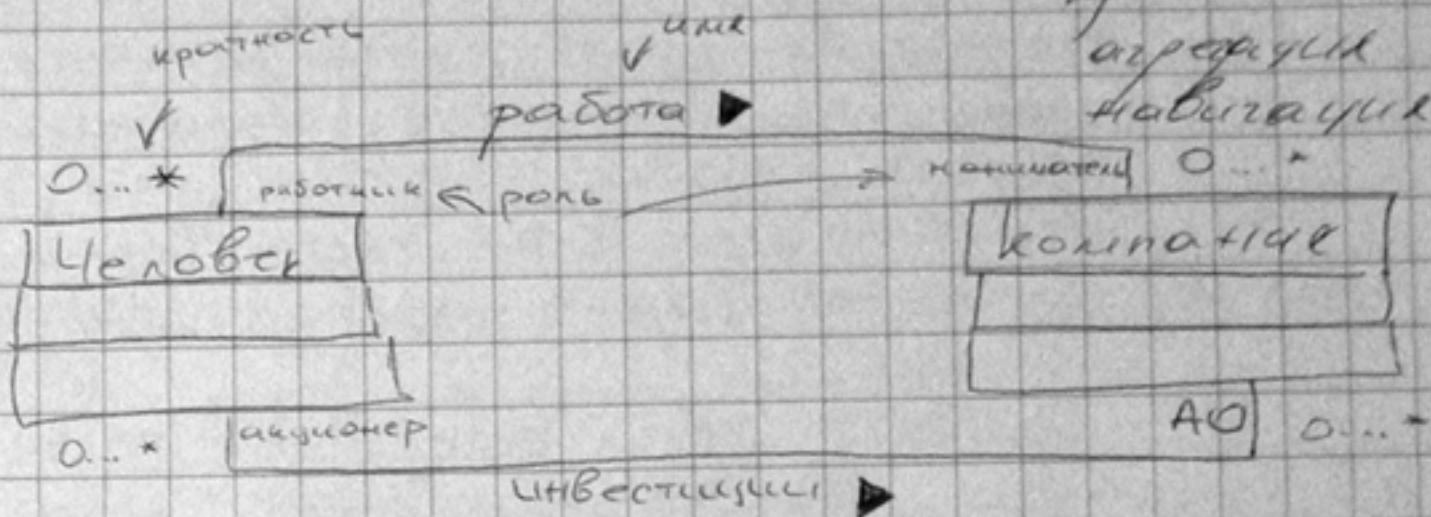
Кроме того Допустимо множественное наследование!



Но такое можно запретить ограничением.

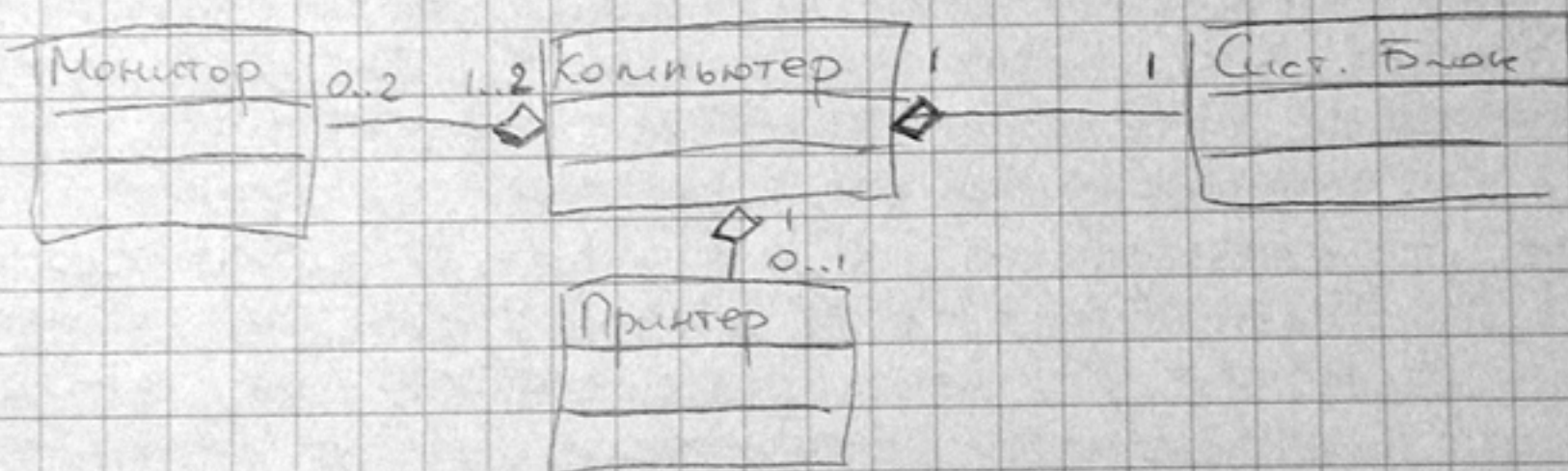
ассоциация - связь между об-ми одного класса с об-ми другого или того же самого класса
 В отличие от ER в UML допускаются и-орные ассоциации - связь сразу и-классов.
 С п-м ассоциацией связывают: имя, роль, кратность, агрегация, навигация

Пример:

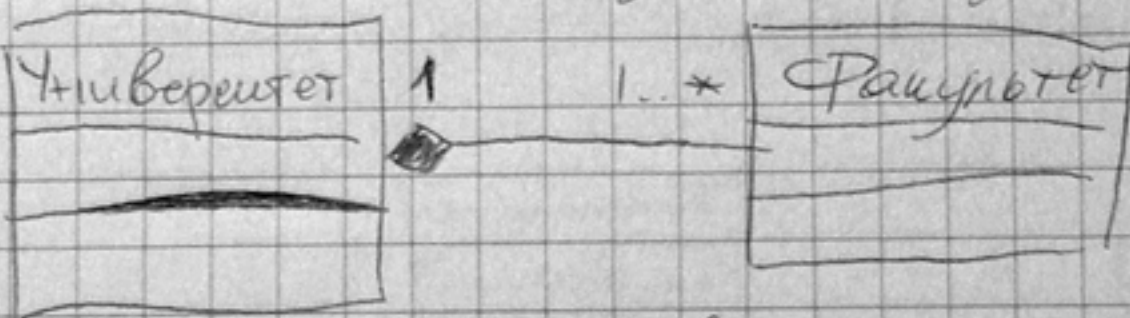


кратность - число, инт-л, список инт-л в терезе, где $\infty = \infty$
 (наличие 0 - обязательность связи)

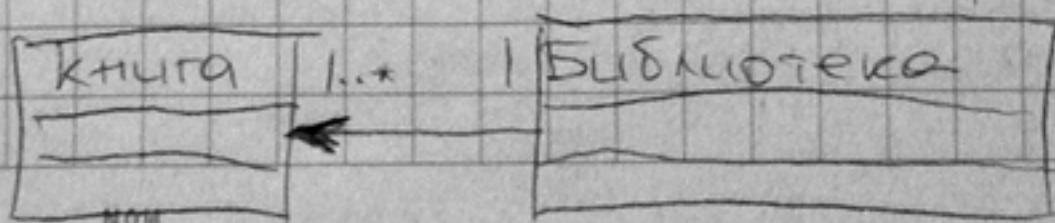
агрегация (слабая ассоциация) и-сх. "мотор" и-и "часть - целое". класс "целое" имеет более высокий концептуальный уровень, чем "часть".



В случае, когда связь "часть - целое" имеет только 1 часть, а целое состоит из многих частей, то связь наз-ся композицией, а связь - композицией.



навигация: в UML, если не оговорено иное, навигация по ассоциации осуществляется в обоих напр-х, но бывают случаи, когда необход-но от-ть напр-е навигации $\Rightarrow$ на линии ас-ции стрелка напр-е навигации



Получение схемы реалю. БД и ИМБ - декларативный классиф.
 Алг-м ~ преобр-ю из FR-декларативной. Но 7 особенностей:

- 1). не увлекайтесь оптимизацией в классах (они не всегда реализуемы в СУБД)
- 2). избегайте множества наследования и осторожно используйте обобщение.
- 3). Реализацию асс-ий с точки зрения матрицы булевых значений (с помощью триггеров и встраиваний) - это усиление
- 4). агрегатные ассоциации неестественны в СУБД. композиция влияет только на поддержку связной целостности
- 5). Однонапр. связи естественны только для объектно-ориент. СУБД. Для реалю. БД это скрывает порог и возм. дополнит. расходы.
- 6). не злоупотребляйте ор-ми, большое их число скрывает порог

OSL (Язык объективных оцр-ий).

Впервые введен в ИМБ 1.3 (1999 год)

OSL - декларативный язык → стандарт не описывает правил отобр-я в исполн. языке и правил выг-я вар-ий.

Поддержка OSL декл-на только в Portland Together.

OSL описывает пред и постусловия классов инварианты классов.

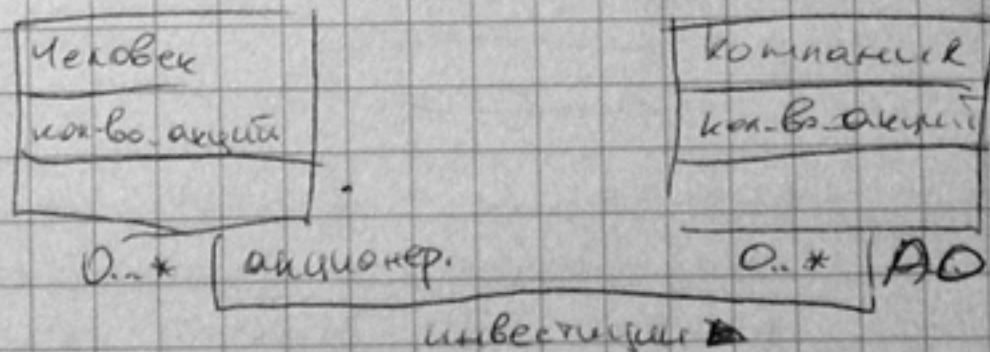
1-й вариант - логическое выраж., при выполнении которого для каждого объекта данного класса в течение всего времени существования этого объекта выполняется следующее:

Пример: В компании обяз-но 1 акционер с контр. пакетом акций

context компания inv:

self. Акционер → exists

($\exists x: \text{Человек} \mid x.\text{кол-во акций} \geq \text{self. кол-во акций} \cdot 2$)



OSL описывает 4 склп. типа данных: Integer, Real, String, Boolean

4 типа коллекций:

Множество

Дубликаты

Тип	Множество	Дубликаты
Set	—	—
Bag	—	+
Sequence	+	+
Ordered Set	+	—

Выражение OCL.

Mayer Surtz LHA-461

1. предопред. типы и операции над ними
2. типы данных, опред. ИМБ-моделью (классы, ...)
3. атрибуты классов, роли ассоциаций
4. методы классов, не меняющие ~~структуру~~ ~~состояние~~ ~~системы~~

- Tim.
- Boolean
- Integer
- Redd
- String
- collection →

Операции

and, or, xor, implies (unnaekwage), if - then - else

 $+, -, *, /, \text{div}, \text{mod}, \text{min}, \text{max}, <, >, =, <=, >=$

+, -, *, /, abs, floor, round, min, max, <, >, =, <=, >=, <=

select (iterat | nor. bap.)
reject (iterat | nor. bap.).

) нов. коллекции: баш.-ск. лог. вар., итерат-ан-т
коллекции,

collect (iterat | bapam)

нов. колл. : соответ- вероятно (вероятно по вариан.)

exists (iterat. method)

true $\Leftrightarrow$ если $\exists$ м-т корректны лог. вып. true

for Alt (interact | no. boys old)

$\text{tr}(\rho) \leq 1$ сума ρ — не-отрицательная матрица

size()

разм-р

at (index)

тср. множеств

union, intersect, -

тсор. множк
гиско в элем

count + (elem)

гуско в свет
туско $\Rightarrow$ свет в волленици

includes (elem)

$\neg \text{true} \Leftrightarrow$ свет в коллекции
 $\text{true} \Leftrightarrow$ свет не в коллекции

объект: к-ль атр. | к-ль роли | к-ль оп-ции (методы) (факт. погран.)

self принт з-е экземпляра класса (this) в момент выз-а м-та

self. auctioneer $\rightarrow$ exists (a: человек / a. к-волящий $\exists$ = self. кор-волящий $\downarrow \exists \vee 2 + 1$)

Преимущества ИМЛ:

исае двар. классов при прох-и и релю. бау
данных пох-т остав-е в ббшней контасте и мл
и прим-ть другие виды двар. при прох-и и про-
потенций БД.

SQL.

Специализация SQL по своей сути - спецификация отдельной модели данных - (суперкласса реляц. модели $\Rightarrow$ более прав. термин "модель данных SQL").

История SQL:

возник при вып-и проекта реляц. СУБД System R в IBM.

Первый вариант - 1974г.

Исх. название языка SEQUEL (Structured English Query Language)

Язык SQL - полный язык БД, обеспечивающий:

- 1). средства формул-к запросов
- 2). средства манипул-к данными
- 3). средства опред-к и манипул-к схемой БД (метаданные)
- 4). средства опред-к огранич-й целостности
- 5). средства опред-к представлений (View)
- 6). средства опред-к структур орг-з. уровней для транз-го языка запросов
- 7). средства авторизации доступа
- 8). средства управл-к транзакциями

Oracle - первая коммерческая СУБД 1979г.

SQL/PS - IBM СУБД 1981г.

DB2 - IBM СУБД 1983г.

Informix SQL - Informix СУБД 1985г.

Chase SQL Server - Chase СУБД 1987г.

MS SQL Server - Microsoft 1989

Модели:	IBM DB2
(коммерц. системы)	Oracle Microsoft
(свободн.)	MySQL PostgreSQL

Стандартизация SQL:

1982г. - ANSI (начинает разработку стандарта)

1986г. - SQL/86

1989г. - SQL/89

(сокращенная синтаксис и семантика операторов выборки данных, манипул-к данными, факты орг-к огранич-й целостности)

SQL/92 (SQL2) (станд-ция средств манипул-к схемами, управл-к транзакциями, сессиями, стандарт-ция служебных таблиц (каталогов), стандарт-ция динамического SQL)

SQL:1999 (5 томов)

SQL:2003 (9 томов):

- 1). контент. структура стандарта
- 2). типы данных; синтаксис и семантика осн-х операторов; правила свертывания SQL для процедурных ЯП
- 3). SQL/CLI (надзор специф-и интерфейсов процедур-ко-м-х пов-т вып-т-б данных; созд-е оп-ции SQL)
- 4). SQL/PSM (способы опред-к и испол-к хранимых процедур (специал-н. образом оформл-ны прог-мы с опер-ми SQL, сохраняемые в БД, могут вы-ск приложениями а вып-ск в СУБД (на стороне сервера))
- 5). языковые средства бизнес-к-с бизнесными данными
- 6). связь SQL с ОО ЯП
- 7). Описание информ-ц. схем

13) SQL версии Java

14) языковые средства работы с XML в SQL

SQL: 2006 / SQL: 2003 с переработанным синтаксисом

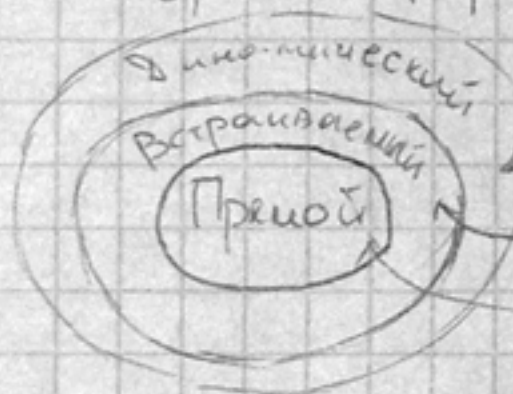
SQL: 2008 (устранены недостатки SQL: 2003, улучшена поддержка OLAP (Online Analytical Processing))

Структура SQL.

1) С точки зрения разработчика СУБД:

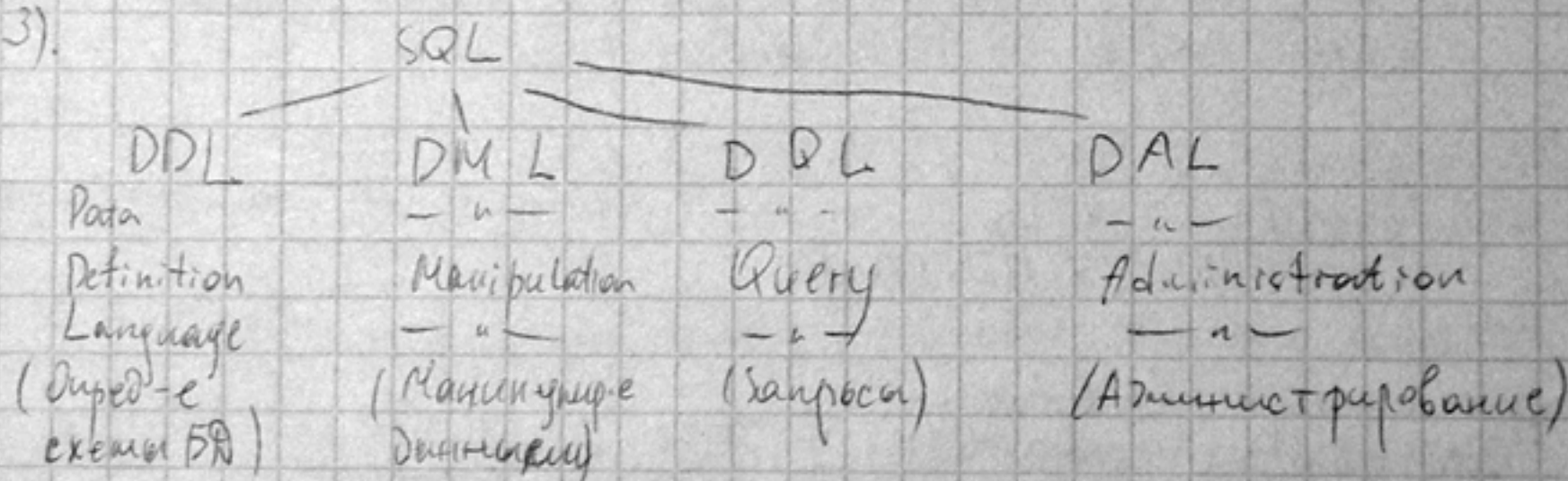


2) С точки зрения программиста приложения БД



конструкции, построенные в момент исполнения программы
программы для исполнения в СУБД
непоср. команды пользователя

3)



Типы данных SQL.

Данные таблицы типизированы. При описании таблицы в столбце указывается тип или домен. СУБД следит за присутствием в таблице лишь допустимых значений.

NULL - не установленное значение.

Операции с NULL завершаются значением NULL (если операция приоритетна и UNKNOWN - если сравнение).

Логический Boolean (TRUE | FALSE | UNKNOWN)

Точные числовые:

Целые типы:

SMALLINT
INTEGER
BIGINT

С дробн. частью:

NUMERIC (p,s)
DECIMAL (p,s)

NUMERIC - точность p - строка DECIMAL - точность s - строка, а потом строка.

p - точность (количество десятичных цифр)
s - число цифр дробной части
s default = 0
p default - max в базе

NUMERIC(4,1)
DECIMAL(4,1)

| 1234,5 |

- недопустимо
- допустимо

Приблизительные числовые : REAL (1-точность)
DOUBLE PRECISION (2-точность)
FLOAT(p) p-точность (используются цифровые мониторы)

Символьные строки : CHAR(x) - фикс. длина
VARCHAR(x) - пер. длина
CLOB(z) - пер. длина, но очень большого размера
(параметр - в Кбайт, Мбайт, Гбайт)
Строка всегда зап-ся пробелами до полной длины
строка зап-ся либо в ' ', либо в " "

Датовые строки : BIT(x) - "-
BIT VARYING(x) - "-
BLOB(z) - "-

состоит из произв. байтов
запись B'0101' или в 16-ричной форме X'...'

Дата и время :

DATE

DATE 'YYYY-MM-DD'

TIME(p)

TIME 'HH:MM:SS.F...F'

TIMESTAMP(p)

TIMESTAMP 'YYYY-MM-DD HH:MM:SS.F...F' default=0

max p - цифр реализации.
precision

TIME WITH TIMEZONE

TIMESTAMP WITH TIMEZONE

} добавл-я смещение от Гринвича.

CURRENT_DATE

CURRENT_TIME

CURRENT_TIMESTAMP

} данные на момент выполнения

Временные интервалы - продолжительность во времени :

INTERVAL start(p) [TO end(q)]

stat+(end); YEAR

MONTH

DAY

HOURL

MINUTE

SECOND

p-точность, default=2

q-ис-я, если есть second и обозн-я - число цифр.
с точн. частн.

Год - месяц

День - время суток

} - длина точности интервала

Преобр-е типов данных.

Поддерж-ся явное и неявное преобр-е значений из 1-типа в другой
Преобр-е соответствует правилам ВЯП.

Явное преобр-е CAST (expression AS type)

В подробные правила возможных преобразований.

Средства опред-л доменов

Домен - тип данных + ограничения на допустимые значения
Домен - допустимый минимальный объект схемы БД. Или
домена можно задать при соз-и таблиц в табл-це. Домены
можно созд-ть, измен-ть, модифицировать.

Определение: `CREATE DOMAIN SALARY AS NUMERIC(10,2)`
`DEFAULT 10000.00`
`CHECK (VALUE BETWEEN 1000.00 AND 10000000.00)`
`CONSTRAINT SAL_NOT_NULL CHECK (VALUE IS NOT NULL)`

`VALUE BETWEEN min AND max`
`VALUE IN (список допустим-х)`

Изменение опред-л Домена: `ALTER DOMAIN` и 4 возможных действия:

- `SET DEFAULT` (новый DEFAULT)
- `DROP DEFAULT` (отмена DEFAULT)
- `ADD CONSTRAINT` (новый CONSTRAINT)
- `DROP CONSTRAINT` (отмена ограничений, которое имеет имя)

Если существ. данные противостоят новому ограничению, то его добавление отвергается

`ALTER DOMAIN SALARY SET DEFAULT 15000.00;`
`ALTER DOMAIN SALARY ADD CONSTRAINT NEW_MIN_SAL`
`CHECK (VALUE >= 15000.00)`

Отмена опред-л Домена: `DROP DOMAIN name RESTRICT;`
`DROP DOMAIN name CASCADE;`

В SQL есть 3 типа таблиц: 1) базовые (реально хранятся)
2) порожденные (т.е. момент созд-я запросов)
3) представленные (сущест-е не физически)

`RESTRICT` удалит только не задействованный в БД
`CASCADE` если задействован в в) то таблицы удалятся
в в) то в столбцы такого домена отн-ся
к опред. типу, в столбцы наследует DEFAULT и ограничения.

Средства опред-л, изм-л и удаления базовых таблиц.

- 1) `CREATE TABLE`
- 2) `ALTER TABLE`
- 3) `DROP TABLE`

1) `CREATE TABLE` создает пустую базовую таблицу
указывается имя таблицы и список ат-б опред-л таблицы
Элемент - либо определение столбца, либо опред. табл. ограничения
в элементе опред-л столбца указ-ся имя и тип (тип данных/наим домена)
для столбца можно указать значение по умолчанию и ограничение с табл-цей.
Действующий DEFAULT: если при опред. `DEFAULT` → DEFAULT берется
значение, если в домене `DEFAULT` → DEFAULT этот
значение, NULL, если он не задан.

иначе, значения по умолчанию нет и при вставке значение в этом столбце должно быть явно задано.

Ограничения целостности столбцов и таблиц:
 # опр-е в опр-д. столбца может быть выражено в виде табл. ограничения (обратное неверно)

виды ограничений:

- CHECK (проверочное)
- PRIMARY KEY (первичного ключа)
- UNIQUE (возможного ключа)
- FOREIGN KEY (внешнего ключа {только для таблиц})
- REFERENCE (внешнего ключа {только для столбцов})

CHECK - провер. лог. выражение. Указ-е лог. выр-е будет включаться при вставке, обновлении таблицы. Считается, что таблица упр-та огранич., если # строки рез-т выр-х CHECK не является FALSE
 NOT NULL = CHECK (C IS NOT NULL) (С-име столбца)

ключевые ограничения - требование уникальности элементов указ. столбца (комбинаций столбцов).

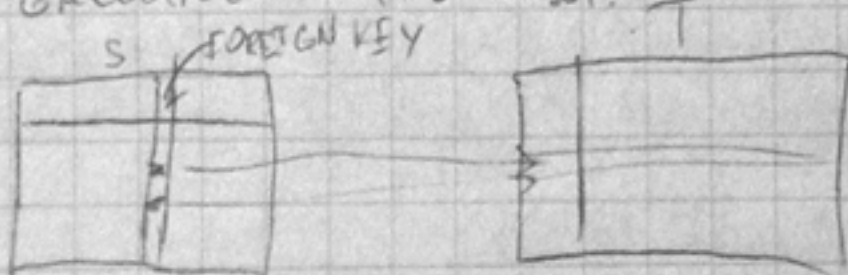
Кроме того, PRIMARY KEY несет ограничение NOT NULL
 Возникает произв. кол-во возможных ключей $\Rightarrow$ NULL-возможное указание (не более одного первичного ключа $\Rightarrow$ отличие SQL-модели от реляц.: возможны дубликаты)

унар. мн-во в отличие от реляц. модели

кроме того атрибуты -

При опр-д. FOREIGN задается имя таблицы T и список столбцов, которые должны совпасть точно до порядка имен столбцов с опр-д. первичного или возможного ключа таблицы T. Если список столбцов не задан, то считается, что подразумевался первичный ключ.

Задаются действия, определяющие ссылочную целостность при удалении (ON DELETE) или обновлении (ON UPDATE) строк внешней таблицы.



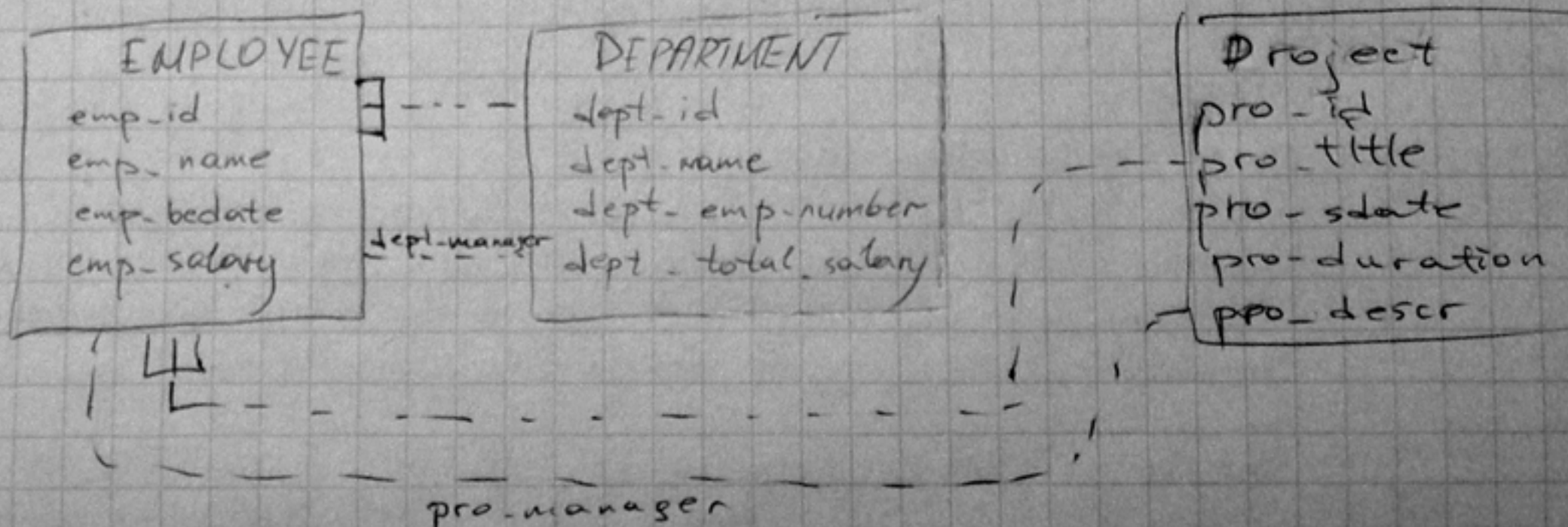
RESTRICT = NO ACTION - если ссылка есть, то запрещается отб-ск

CASCADE - удалить эту T и ссылки на нее в S

SET DEFAULT - удалить, а ссылки в DEFAULT

SET NULL - удалить, а ссылки в NULL

Пример: Определить таблицы для хранения данных о служащих, отделах и проектах компании. Работ-ли с заданной ER-диаграммой с учетом опр-й: В отделе 1, ..., 100, служащих в проекте не более 50 служащих, объем фонда заработной платы отдела не меньше 100000р.




```

CREATE TABLE EMPLOYEE (
    EMP_ID INTEGER PRIMARY KEY,
    EMP_NAME VARCHAR(200) NOT NULL,
    EMP_HIREDATE DATE,
    EMP_SALARY SALARY
    DEPT_ID INTEGER DEFAULT NULL
    REFERENCES DEPARTMENT
    ON DELETE SET NULL
    PROJ_ID INTEGER DEFAULT NULL
    FOREIGN KEY PROJ_ID REFERENCES
    PROJECT (PROJ_ID) ON DELETE SET NULL,
    CONSTRAINT PROJ_EMP_NULL
    CHECK ((SELECT COUNT(*) FROM EMPLOYEE E WHERE E.PROJ_ID = PROJ_ID
    <= 50));

```

```

CREATE TABLE DEPARTMENT (
    DEPT_ID INTEGER PRIMARY KEY,
    DEPT_EMP_NUMBER INTEGER NOT NULL
    CHECK (VALUE BETWEEN 1 AND 100)
    DEPT_NAME VARCHAR(200) DEFAULT "Nameless" NOT NULL,
    TOTAL_SALARY SALARY DEFAULT 1 000 000.00
    NOT NULL
    CHECK (VALUE >= 100 000.00)
    DEPT_MANAGER INTEGER DEFAULT NULL
    REFERENCES EMPLOYEE ON DELETE SET NULL
    CHECK (IF (VALUE IS NOT NULL) THEN
    ((SELECT COUNT(*) FROM DEPARTMENT D
    WHERE D.DEPT_MANAGER = VALUE) = 1))
    CHECK (DEPT_EMP_NUMBER = (SELECT COUNT(*) FROM
    EMPLOYEE WHERE DEPT_ID = EMPLOYEE.DEPT_ID))
    CHECK (DEPT_TOTAL_SALARY >= (SELECT SUM EMP_SALARY
    FROM EMPLOYEE WHERE DEPT_ID = EMPLOYEE.DEPT_ID));

```

```

CREATE TABLE PROJECT
    PROJ_ID INTEGER PRIMARY KEY
    PROJ_TITLE VARCHAR(200) DEFAULT "No title" NOT NULL;
    PROJ_START DATE DEFAULT CURRENT_DATE NOT NULL,
    PROJ_DURATION INTERVAL YEAR DEFAULT
    INTERVAL '1' YEAR NOT NULL,
    PROJ_MANAGER INTEGER UNIQUE NOT NULL
    REFERENCES EMPLOYEE ON DELETE NO ACTION
    PROJ_DESCRIPTION CLOB (10 M));

```

NO UPDATE ON UPDATE NO ACTION

Целем модификации таблиц: ALTER TABLE

- 6 Действия:
- a) ADD COLUMN
 - b) ALTER COLUMN SET DEFAULT
 - c) — " — DROP DEFAULT
 - d) DROP COLUMN
 - e) ADD CONSTRAINT
 - f) DROP CONSTRAINT

2). RESTRICT, если участвует во внешних ограничениях $\Rightarrow$ отвернется CASCADE удалит столбцы ограничения, связанные с ним

c). если удален первичного или внешнего ключа то, RESTRICT - отб-ся если есть внешние ссылки CASCADE - всегда вып-ся ~~иначе не вып-ся~~

Далее:

DROP TABLE RESTRICT - отб-ся, если участв. участв. в внешних ограничениях целостности CASCADE - вып-ся и отмен-ся все внешн. ограничения с ней

Базовые средства манипулирования данными

- 1) INSERT - добавлять
- 2) UPDATE - изменять
- 3) DELETE - удалять

1) INSERT INTO

далее имя табл. для вставки

далее в кругл. скобках через ',' список столбцов

(если пропущен, то в произвольном порядкеorig. список столбцов)

далее конструкторы строк (в соотв-ии со списком столбцов)

элемент конструктора - литерал, DEFAULT, NULL, скалярный запрос, т.е. ответ которого - единств. строка

INSERT INTO EMPLOYEE VALUES

ROW (319, 'Иванов И.И.', DATE '1980-07-19', DEFAULT, NULL, NULL)

ROW (320, 'Петров П.П.', DATE '1983-05-09', SELECT EMP_SAL FROM EMPLOYEE WHERE EMP_ID=200), 12, NULL);

INSERT INTO EMPLOYEE

(EMP_ID, EMP_NAME, EMP_BDATE)

VALUES ROW (321, 'Сидоров С.С.', DATE '1962-02-01');

Оператор UPDATE.

UPDATE tab_name SET список-модифиц.-столбцов WHERE лог.-усл.;
↓
"имя-столбца = выражение" через запятую

- 1) В строки с указ. именем выг-ся лог.-с
Если true в подпункте модификации, обозн-м их во всех строк Sm.
- 2) В строка из Sm модифицируется та, что из-с условия из списка модиф-и выг-ся из-м из списка, остальные не меняются
Подз-м-м столбцов в выражении помечаются из-м до модификации.

Прр: всех суммарных проекта 5 перебросит в отдел 12 и повысит зарплату

```
UPDATE EMPLOYEE SET DEPT-ID=12 EMP-SAL=EMP-SAL+10000  
WHERE PROJ-ID=5;
```

Оператор DELETE.

DELETE FROM имя-табл. WHERE лог.-усл.;

- 1) В строки выг-ся усл.-с, если оно true ⇒ подложат удаление, их из-м обозн-м Sd
- 2) В строка из Sd удаляется из табл.

Пример: удалить всех суммарных с зарплатой больше, чем у м-в-х в-х в

```
DELETE FROM EMPLOYEE WHERE EMP-SALARY > (SELECT EMP.EMP-SALARY  
FROM EMPLOYEE EMP, DEPARTMENT DEPT WHERE EMPLOYEE.DEPT-ID  
= DEPT.DEPT-ID AND DEPT.DEPT-MANAGER=EMP.ID);
```

Триггеры.

Опр. Триггер - процедура, хранящаяся в БД и автоматически выполняемая с СУБД при возникновении соотв. условий.
Условиями являются вставка, модификация, удаление данных или нескольких строк.

Таблица с которой связан триггер - предметная таблица.

Оператор SQL, выг-ся которого приводит к срабатыванию триггера - инициализирующий оп-р.

Триггер может быть ассоциирован только с одной базовой таблицей.

Триггеры стандартизованы с SQL 99.

Переносимость триггеров между СУБД ограничена.

Назн-е триггеров:

- 1) соглас-е данных и целостка данных
- 2) журналирование и аудит
- 3) операции не связанные с изм-м БД (исход, электр. рассылка).

Опр-е Триггера: CREATE TRIGGER

Отмена оп-р: DROP TRIGGER

CREATE TRIGGER ... ON имя-табл [WHEN...]

Указ-ся имя триггера, момент срабатывания (BEFORE, AFTER), тип инициз. оператора (INSERT, UPDATE, DELETE: если UPDATE, то указать те столбцы, которые инициз. триггер), также

3. В рез-те GROUP BY получаем T2 (сгруппированную таблицу), в которой строки T1 расставлены в том числе и так, что во всех строках одной группы 3-х столбцов в списке GROUP BY (столбцов группировки) одинаково
4. Таблица T3 (сгруппированная), содержащая те и только те строки T2, где cond-expr = true
 cond-expr в HAVING применяется только для групп, т.е. predicates работают с группой целиком; можно использовать имена ст-в группировки и агрегатные функции.
 Если GROUP BY отсутствует, то $T3 = T1$ - табл. из 1-й группы
 Если HAVING отсутствует, то $T3 = T2$

5. а). Если нет GROUP BY и HAVING
 б). Один из них присутствует

- 1) а). в списке квалифиц. имена столбцов таблицы из FROM
 $Z_i.c$ или $Z_i.*$ - все столбцы Z;
 б). $Z_i.*$ допустимо только если группировка ведется по всем столбцам или только GROUP BY ст-ца

- 2). value-expr [AS column-name]

↑
 может сод-ть литеральные константы, функции, для а) - имена столбцов T₁. Вар-е вст-а в строки T₁. Имена столбцов соотв. 3-х ст-в в констр. строке. Для б) - входит только те столбцы T3, кот-е входят в GROUP BY - список, имена других ст-в T3, если-ся только внутри агрегатных функций; вар-е воис-к для 4-й группы.

Таким образом, получаем T4 -

На закл. этапе, если есть DISTINCT, дубликаты строк удалит

6. Произв-ся сортировка T4 по списку эл-в сортировки:
 value-expression [ASC / DESC]

Рожден FROM.

В нем могут встречаться не только имена таблиц, но и другие вар-ты.

В каз-ве ссылок на табл. можно использовать

или левобок табл.

представления

порождаемой табл.

явное присоед-к датному, с помощью WITH

Представленные таблицы (предст-л | вирт. таблица) (View)
 при их созд-и указ-ли [имена столбцов], а также вар-е заголовка и построения представления

Пр-р: создать предст-е с имен-ей о наз-х от-в.

```
CREATE VIEW DEPARTMENT_MANAGER AS
SELECT E.EMP_ID, E.EMP_NAME, E.EMP_BDATE,
       E.EMP_SALARY, D.DEPT_NAME
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.EMP_ID = D.DEP_MANAGER
```

Име предст-л не только сод-ть с алиасами предст-л и базовых таблиц. В общем случае предст. таблица не модифицируется, но

Допускается создание представлений с WITH CHECK OPTION.
Отмена представлений: DROP VIEW.

Порождаемые таблицы и присоединяемые запросы.
Семантически порождаемая таблица соответствует логическому представлению в момент выполнения запроса. Запрос с порождаемой таблицей и запрос SELECT с WITH.

Пр-р: найти общ. число суммарных с одним max разн-м з.и.

SELECT SUM (TOTAL-EMP), MAX-SAL FROM
порожд. табл. [SELECT MAX(EMP-SALARY), COUNT(*) FROM EMPLOYEE
WHERE DEPT-ID IS NOT NULL GROUP BY DEPT-ID)
AS DEPT-MAX-SAL (MAX-SAL, TOTAL-EMP)
GROUP BY MAX-SAL.

Контрольные : 18.12, 12.15 - 13.15

Зачет : 21.12, 25.12

прис. запрос
WITH DEPT-MAX-SAL (MAX-SAL, TOTAL-EMP) AS
(SELECT MAX(EMP-SAL), COUNT(*) FROM EMPLOYEE
WHERE DEPT-ID IS NOT NULL GROUP BY DEPT-ID)
SELECT SUM (TOTAL-EMP, MAX-SAL FROM DEPT-MAX-SAL
GROUP BY MAX-SAL.

Агрегатные и кванторные функции.

Агрегаты: COUNT
MAX
MIN
AVG
SUM

аргумент
[DISTINCT] имя_столбца

кванторы: EVERY
SOME {ANY}

арг-т
лог. выраж.

Агр: Работают над мультимножествами строк: группой или всего таблицы.
На осн-и вар-та аргумента и мультимножества строки происходит мультимножества
зн-я; в случае DISTINCT из него удаляются дубликаты и некорректные,
затем производится вычисление.

кван: var-та лог. выраж-е и строки из мультимножества:
EVERY = TRUE $\Leftrightarrow$ TRUE во всех строках
SOME (ANY) = TRUE $\Leftrightarrow$ $\exists$ строка с TRUE
Если мн-во пустое, то COUNT = 0
EVERY = TRUE
SOME (ANY) = FALSE

Для COUNT есть точный аналог COUNT(*): считает,
что все строки различны независимо от того, что может на
самом деле

Логические выражения.

Лог. выраж. - булевые выраж., состоящие на основе предикатов с операциями AND, OR, NOT и ().

S - скалярное зн-е

R - строка таблицы

$|R|$ - кол-во зн-ий в строке R (степень R)

T - таблица как рез-т выполнения запроса.

Для n арг-тов предикатов выполняется: 1. совм-ть типов
2. рав-во степеней строк операндов

$\neg$ предиката $\exists$ обратный (NOT).

1. Предикат сравнения

R_x сравн-ор R_y

сравн-ор: $>, <, =, <>$ (иногда $\neq$), $>=, <=$
NULL сравн-ор S (и подобные) = UNKNOWN

2. Предикат BETWEEN

R_x BETWEEN R_y AND $R_z \sim (R_x \geq R_y) \text{ AND } (R_x \leq R_z)$

3. IS NULL - позволяет проверить зн-е арг-та на неопред-ть

	R_x IS NULL	R_x IS NOT NULL	NOT R_x IS NULL	NOT R_x IS NOT NULL
$ R_x =1, \text{ NULL}$	T	F	F	T
$ R_x =1, \text{ NOT NULL}$	F	T	T	F
$ R_x >1, \text{ все NULL}$	T	F	F	T
$ R_x >1, \text{ не все NULL}$	F	T	T	F
$ R_x >1, \text{ все NOT NULL}$	F	T	T	F

4. IN - усл-е вхождения в указ. мн-во

скаляр IN мн-во скаляров
строка IN таблица

R_x IN $T = \begin{cases} \text{TRUE} & \exists R_t \in T : R_x = R_t \\ \text{FALSE} & \forall R_t \in T \Rightarrow R_x \neq R_t \\ \text{UNKNOWN} & \end{cases}$

$T = \emptyset \Rightarrow R_x \text{ IN } T = \text{FALSE}$.

Прпр: Найти номера сот-ков - не руковод-щих отделов, получ. зарплату совм. с зарп. одного из рук. отд-в

```
SELECT EMP_ID FROM EMPLOYEE BETWEEN
WHERE EMP_ID NOT IN (SELECT DEPT_MANAGER
FROM DEPARTMENT) AND
EMP_SALARY IN (SELECT EMP_SALARY
DEPARTMENT_MANAGER)
```


5). OVERLAPS - проб-т перес-с по времени 2-х событий

$R_x \text{ OVERLAPS } R_y$, $|R_x| = |R_y| = 2$

$R_x = (t_s, t_f)$ - дата, время начала и конца

$(t_s, interval) = (t_s, t_s + interval)$

Пр-р: найти номера проб, вошедших в 01.01.2000 по 31.12.2002

SELECT PRO-ID FROM PROJECT WHERE
(PRO-SDATE, PRO-DURATION) OVERLAPS (DATE '2000-01-15',
DATE '2002-12-31')

6). Сравнение с квантором

$R_x \text{ comp-op ALL } T$

$R_x \text{ comp-op SOME } T$

ALL :

TRUE $\Leftrightarrow T = \emptyset$ или $\forall R_T \in T \Rightarrow R_x \text{ comp-op } R_T = \text{TRUE}$
FALSE $\Leftrightarrow \exists R_T \in T: R_x \text{ comp-op } R_T = \text{FALSE}$
UNKNOWN иначе

SOME:

FALSE $\Leftrightarrow T = \emptyset$ или $\forall R_T \in T \Rightarrow R_x \text{ comp-op } R_T = \text{FALSE}$
TRUE $\Leftrightarrow \exists R_T \in T: R_x \text{ comp-op } R_T = \text{TRUE}$
UNKNOWN иначе

7). LIKE

string LIKE pattern ESCAPE symbol
↑
либо символьная, либо байтовая

Pattern: $_$ = 1 сим. символ
% = произв. подстрока
symbol: $X_$ = X $X\%$ = X

8). SIMILAR $\approx$ LIKE, но в pattern можно применить рег. выраж.

9). EXISTS (таблица)

TRUE $\Leftrightarrow$ мощность таблицы > 0
FALSE иначе

10). UNIQUE (таблица)

TRUE $\Leftrightarrow$ таблица рез-т не содержит строк-дубликатов
FALSE - иначе

11). DISTINCT

$R_x \text{ IS DISTINCT FROM } R_y \Rightarrow \text{TRUE} \Leftrightarrow R_x \text{ не дублирует } R_y$

12). MATCH - уст-т соотв-с строки и запроса
(не обобщ-ся)

Управление транзакциями в SQL.

В SQL транзакция образуется либо явно оператором `START TRANSACTION`, либо неявно, когда выис-ся оператор, для которого требуется контекст транзакции, а такого контекста не существует.

Контекст транзакции требуется не для всех оп-ов.

Зав-е транзакции: `COMMIT` - завершить с описанным рез-том

`ROLLBACK` - " - с возвратом к состоянию на момент `START`

`START TRANSACTION` mode - commit list ← список режимов

режимы доступа `READ ONLY`, `READ WRITE`

уровни изоляции

`DIAGNOSTIC SIZE` число (опред-и число диагональ. тп-в

кот-е можно разбить в области диагностики, default зависит от реализации)

`SET TRANSACTION` mode - commit list

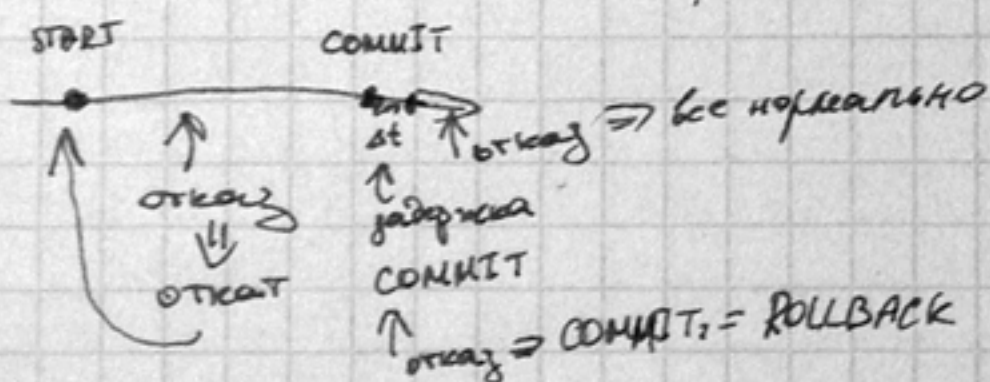
позв-т установить характеристики транзакции по умолчанию (менять режим на лету или нельзя!)

`COMMIT [AND [NO] CHAIN]`

`ROLLBACK [AND [NO] CHAIN]`

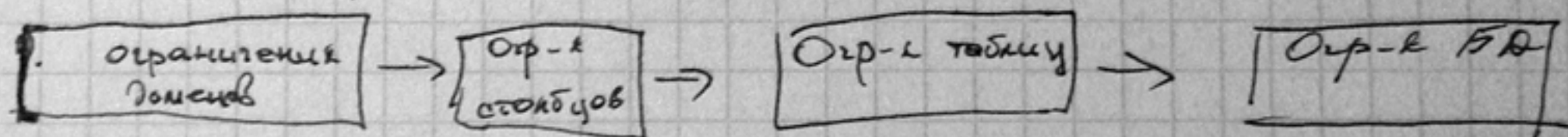
default или `NO` ⇒ кодах транзакция не начнется сразу по опис-и в транзакционной

Согласование с принципами ACID



Durability

Церфрия ограничений БД:



Опр-е БД созд-ся оп-ром

`CREATE ASSERTION` или `CHECK(лог.выр.)`
`DROP ASSERTION` или

В контексте исполняемой транзакции + оп-с целостности может нах-ся в 1-м из 2-х режимов: немедленная проверка
 или отложенная проверка
 Для этого во + опр. целостности можно добавить специф-ко

`INITIALLY { IMMEDIATE | DEFERRED } [ [NOT] DEFERRABLE ]`

`NOT DEFERRABLE` - немедл.

`DEFERRABLE` - отлож.

`INITIALLY` - настройка на мом-т старта транзакции

Если сост-е базы данных меняется, то запускается проверка опр-т.
Если немедл. проверяемое опр-е = FALSE $\Rightarrow$ вы-дана отв-ка
Отложенное опр-е проб-ся когда перейдет в сост. немедл., т.е.

неб-но (выход из транзакции по COMMIT)
вно (вызов SET CONSTRAINTS {ALL, constraint_name_list} {DEFERRED, IMMEDIATE}) -

т.им опр-ом меняется режим работы DEFERRABLE опр-т.

При отлож. проверке и выводе COMMIT, если произошли ошибки целостности, то COMMIT = ROLLBACK.

Чтобы этого не произошло нужно использовать точки сохранения.

Опр. Точка сохр-я - пометка в лог-те операций транзакции, которую в дальнейшем можно использовать для полного отката транзакции с сохранением ее целостности и рез-в операции, выв-д до т. сохранения.

