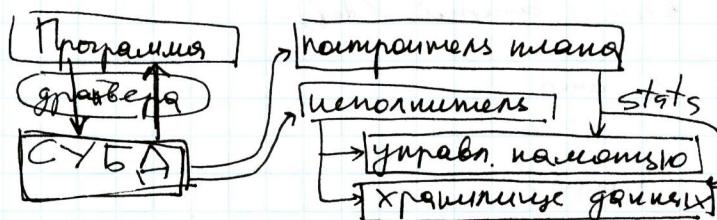


① Работа с базой

- Программное представление (с таблицами) - excel / libreoffice / dbase. Читаем текст, пишем любую операцию, сколько хотим.
- Файл. моделирование с/б/с. Структурировано, проф. Но много фильтров, можно поганые, неиз проверяют целостность.
- Иерархическая: дерево-дерево, хранение дерева. Структурно данные деревообразные. Чем выше - то - выше. Wind. registry.
- Семантическое - предмет. язык (благодаря - записи). Все это видят, но можно многое.
- Объектные - объекты как записи, есть линк. Просто/кошко, но можно кодить.
- MySQL - документ-ориентированное (xml/json), k-v.
- Текущее, прошлые

② Архитектура



③ Физ. модель данных

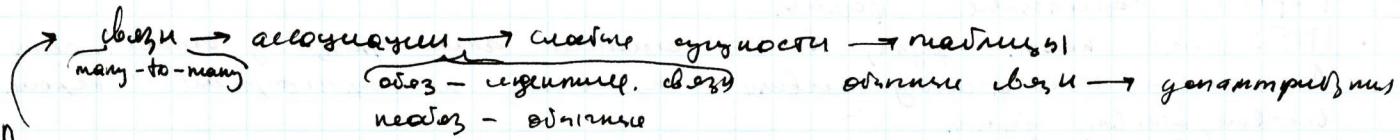
- Объединение представление, иерархическое
- Ключи - проходка/составные, естественные / вымышленные, внешние /внешние.
- Ссылки - внешние ключи.
- Граф. модели.

Name	FKN	attr & O, M, PK
Field	Type	attr

④ Сущности - связь

- Сущность <имя сущности, данные, свойства>, иерархическая
- Связи, роли, типы контактов. Простые и композитные
- Ассоциации - обобщенная модель. Сущностная связь. Чем identity.
- Дубл. ^{look alike} - все кроме расшиф. сущностей. Марш - расшиф. ^{look alike} Обобщенная - отдельные
- Сущность сущности - негат. пример. где члены. члены. сущн.

⑤ Представление:



Наследование.

Пример: сущности; следы сущности, атрибуты, ассоциации.

⑥ Реляционные модели данных, ключи:

Определение - {запись, поле, набор, тип, значение}

Определение vs определение

Найдено: в определ. чем коррелирует с записью.

атрибуты ~ атрибуты - set vs list

Запись - запись. Ключ - это. не более.

строки ~ кортежи - разн. декар.

Требование: уникальность, различимость, изучимость.

неч. ~ одинак. - прямой

и т.д. влож.

неч. set

⑦ ФЗ: замкн., авт., управ. связь

$f: X \rightarrow Y$ именованная, X, Y - множества аргументов

- Все априорные ФЗ от кнора. Трив. ФЗ - прав. закон \Leftrightarrow левая
- ФЗ задают ограничение S_A , но не все их можно подогнать под определ. форм. Минимиз. число ФЗ и предп. в огранич. может.
- $S^+ -$ замкнк - множество замкн. зам. S^+
- $P \subseteq S \Leftrightarrow S^+ \subseteq P^+ \Rightarrow P \approx S \Leftrightarrow P \subseteq S \wedge S \subseteq P \Leftrightarrow S^+ = P^+$. Использов. ФЗ
- Правило (Арифметика):
 - Рассыпчивость: $A \subset B \Leftrightarrow A \rightarrow B$
 - Дополнение: $A \rightarrow B \Rightarrow A \subset B \rightarrow B \subset C$
 - Транзитивность

→ 0/3^{4/4} из n амп.

Правило: левая. замкн

$A \rightarrow A$ Identity
 $A \rightarrow B \wedge C \Rightarrow A \rightarrow B, A \rightarrow C$ Расщепление
 $A \rightarrow B, C \rightarrow D \Rightarrow A \rightarrow B \wedge C \rightarrow D$

⑧ ФЗ зам. априр., ИМФЗ + наложение

- $X_S^+ -$ замкн. X из ми. ФЗ S . \equiv априорные, ФЗ от X .
- Амортиз.: поиск решения X_e^* , $V=B$ где $A \rightarrow B$ если $A \subset X_B^+$.
- Теорема: $A \rightarrow B \in S^+ \Leftrightarrow B \in A_S^+$. Можно проверить суб. доказ. Завис.
- X надкн $\Leftrightarrow X^+ -$ множество всех априр. (def)
- МФЗ S неприводимо, если
 - нахождение пр. замкн содержит 1 амп.
 - 1 амп. не входит.
 - S ми. по вкн.
- Теорема: ИМФЗ \exists генерализованное ИМФЗ:
 - по расщеплению делаем все ФЗ с $|level|=1$
 - где левая позиция удаляется по априорному если
 - позиция удаляется правило если $B \subset A_S^+, A \rightarrow B$

ИМФЗ \Leftarrow МФЗ, но может не быть min

$$AVX \rightarrow B \\ A_S^+ = A_{S \setminus \{AVX \rightarrow B\}} \cup \\ \vdash \rightarrow B$$

⑨ Член и предсв. нормализации

- Предп. ФЗ \rightarrow логическое вкн, упрощаем/ускоряем изменение
- 1-3, Барса-Кнугга - первые нормализации
- 4, 5, деление - копирование - внесение.
- $\pi_X(R) = \{r \cap X, r \in R\}$, $P_1 \bowtie P_2 = \{r, vr_2 | r_1 \in P_1, r_2 \in P_2, \pi_{r_1}(r_2) = \pi_{r_2}(r_1)\}$
- Деление: $R(A, B) = \pi_A(R) \bowtie \pi_B(R)$
- Теорема Хана: $R(ABC), A \rightarrow B \Rightarrow R = \pi_{AB}(R) \bowtie \pi_{AC}(R)$

⑩ Первые нормализации формул

- 1NF: нет нов. групп, все отношения априорны, у структ. есть вкн
- Аномалии - Контринтуационные правила, не формирующие геклеры, вставки, ошибки, угад.
- Аномалии 1NF: $\langle \underline{cld}, \underline{year}, \underline{lecturer}, \underline{exam} \rangle cld \rightarrow exam$

2NF: 1NF + немножевно априорных зависим $\underline{cld} \underline{lecturer} \underline{year} \rightarrow \underline{lecturer}$ от вкн в член. Приведение - деление по немножевно ФЗ

2NF: $\langle \underline{cld}, \underline{year}, \underline{lecturer}, \underline{phone} \rangle cld \underline{year} \rightarrow \underline{lecturer}$

3NF: 2NF + некон. амп. зависим от вкн непосредственно

деление по последней ФЗ в членке.

НП Барса Кнугга \nvdash нетр. $X \rightarrow Y$ X надкн. Равно 3NF + неконструиб. вкн.

$\langle \underline{cld}, \underline{group}, \underline{examiner}, \underline{phone} \rangle$, генер. по $\underline{Ex} \leftrightarrow \underline{Phone}$

$cld \rightarrow g, p$

$e \leftrightarrow p$

аном. изменения Ex no ne Phone

$c, e \rightarrow g, p$

$c, p \rightarrow g, e$

Может быть что можно доказ. на отношении в НПБк. Но распределено.

$\langle \underline{dept}, \underline{course}, \underline{lecturer} \rangle$

- (12) • Аномалии NFBC: $\langle \text{course}, \text{lecturer}, \text{book} \rangle$ - нем. Φ_3 , но аномалии
 • M3 $X \rightarrow Y$: Y не зависит от $R \setminus X \setminus Y$. $\forall x \forall z, z_2$
 $\{ \exists y | (x, y, z_2) \in R \} =$
 $\{ \exists y | (x, y, z_2) \in R \}$
 • Картине $\Phi_3 - M3$
 • Теорема Рейтинга: $R(XYZ) = \pi_X(R) \bowtie \pi_Z(R) \Leftrightarrow X \rightarrow Y/Z =$
 $\begin{cases} X \rightarrow Y \\ X \rightarrow Z \end{cases} \cup \{ \}$
 • 4NF: \nexists нетрив. MФЗ $X \rightarrow Y/Z$:
 $(\nexists \text{ амп. } A \rightarrow A) \wedge (X - \text{наглаж.})$
 • 4 нетрив. M3 - Φ_3 и Φ_{54} \Rightarrow 4NF. Доказательство на $X \rightarrow Y \rightarrow Z$
 • Рекурсивное произведение $R(X, Y) = R_1(X) \times R_2(Y)$. $\phi \rightarrow X/Y$. доказано.
 • $cId \Rightarrow \text{Lecturer/Book}$ на R_1, R_2

- (13) • Аномалии $\langle cId, \text{lecturer}, \text{group} \rangle$
 • $\forall \{X, \dots, X_n\}$: $R(Y, \dots, X_n) = \pi_{X_1}(R) \bowtie \dots \bowtie \pi_{X_n}(R) \rightarrow$ галстукование
 • Рейтинга: $R(YYZ)$: $X \rightarrow Y/Z \Leftrightarrow \forall \{XY, XZ\}$
 • ЗС трансляция, если $\nexists R_1 \bowtie R_2, X, \dots, X_n$
 • 5NF: \nexists нетрив. ЗС $\forall i X_i$ - наглаж. для нормализации
 членов ЗС, но практик. не бывает ЗС - M3.
 • Теорема Рейтинга - Рейтинга 1: 3NF + прост. норм. \Rightarrow 5NF
 (2): NFBC + 7 прост. норм. \Rightarrow 4NF

- (14) Репоз. алгебра
 • Числоп. язык для работы с отношениями, либо с записями.
 • Операции - это математические операции - либо все математики
 • Операции - это практические, соглашения. Их надо по математике
 • Стандарты - языком: $\pi, \delta, \bowtie, \cup, \cap, \neq, \Rightarrow, \bowtie, \times$ и пр., λ -выраж., \cupправа,
 -права: $\cup, \cap, \times, \bowtie, \Rightarrow$
 -левые: как в лог. $\neq, \Rightarrow, \bowtie$
 • Значение - это значение (доминориз - логика), т.е. выражают перестановки
 • Запись - оптимизация, упрощение; Несколько в SQL - плюсует, деление, неупорядоченное но в соединении

- (15) • Прекращение - не берут одинаковые знаки.
 • δ - дифференциал
 • Предвицелование $Rold=new(\dots)$ - изменение определяют
 • Множество: $V \cap - (\backslash)$ - заполняют зану. δ инициализацией.

- (16) • Правое $R_1 \times R_2$ - в заголовках MUST be diff. names
 • Единственность $R_1 \bowtie R_2$ - сог. корректн с прав. знак. собр. ATGP.
 • Внешнее - $R_1 \bowtie R_2$ - берут писаную если нет.
 • Левое - \Rightarrow_1 головка. null справа, \Rightarrow_2 - правило. $R_1 \bowtie R_2 =$
 $\{R_1 \bowtie R_2 \vee [R_1 - \text{tra}_1(R_1 \bowtie R_2)]\}$
 • Правое. \times - левое - левое, где ком. если правило
 $\pi_{R_1}(R_1 \bowtie R_2)$. \times - правило $\pi_{R_2}(R_1 \bowtie R_2)$
 • Умножение - \times_θ , \Rightarrow_θ , \bowtie_θ - разные λ /прав.
 $\underbrace{\delta_\theta(R_1 \bowtie R_2)}_{J} \vee' (R_1 - \pi_{R_2}(J))$

- (17) • $R \bowtie Y \div S(Y) \in \pi_X(Q) - \pi_X(\pi_X(Q) \times S - Q) - \text{миним.}$
 $\left. \begin{array}{l} \langle x, y \rangle \in Q \\ x \in \pi_X(Q) \end{array} \right\} \quad \left. \begin{array}{l} \langle x, y \rangle \in S \\ y \in \pi_Y(S) \end{array} \right\}$
 • $Q(XY) \bowtie S(YZ) = \{ (x, z) / \{ x \in \pi_X(Q) \wedge y \in \pi_Y(S) \} \subset Q \}$
 • $\Sigma A\text{-expression}(R)$ - выражение амп. и можно
 • Аргументование: Function $Q_A(R)$, $Q = \text{Function}$ ($\pi_Q(n) \in R / \pi_A(n) = n$)
 Q - аргументическое выражение, A - сохранение
 Продолжение на занятии $n_0 = A$

(18) Кортежи и раз. норма

- переменные :: определение. Определение: $\text{значение} \text{ where } \text{условие}$
- $S1 :: \text{Groups}, G :: \text{Groups where } G.\text{Name} = 'aen' \vee S1.\text{Name} = 'mam'$
- Условия - $>, <, =, \text{too}(..), V, \wedge, \neg, \text{Кванторы}$.
- Связь с атрибутами:

1. $\pi_{A_1 \dots A_n}(R)$

Select $A_1 \dots A_n$ from R

2. $\delta_\theta(R)$

from R where θ

3. $\exists A=\text{expr}(R)$

Select expr as A from R

4. $R_1 \cap R_2$

$R_1 :: R_1 \text{ where } \exists R_2 (R_1 = R_2)$

5. $R_1 \times R_2$

$R_1 :: R_1, R_2$

6. $R_1 \bowtie R_2$

$R_1 :: R_1, R_2 \text{ where } R_1.\text{attr}_1 = R_2.\text{attr}_2$

7. $R_1 \bowtie R_2$

$R_1 :: R_1, R_2 \text{ where } \overbrace{R_1, R_2 \text{ where } \neg \exists R_2(\dots)}, R_2 \text{ where } \neg \exists R_1(\dots)$

- Преобр. Синтаксис 6 правил.

- настор. вспр. & выражения

- где-либо правило.

- определение $\exists : \pi_{A_1}, \dots, C \& R$ - преобр. на без упоминания C

- кванторы $\forall C : \text{выбрать из синтаксиса} \text{ выраж. с}$

Select G.Gid where $\exists S (\forall C (\exists P (G.Gid = S.Gid \wedge S.Gid = P.Sid \wedge$

$C.Cid = P.Cid \wedge P.Points \geq 60))$

$$\hookrightarrow T = \pi_{G^*, S^*, C} (\delta_{P.Points \geq 60 ((G \bowtie S) \bowtie P)} \div C)$$

(19) Именн. генераторы и раз. норма

- Переменные :: Тип, Определение { Attr = Значение, ... }
- Пример: $S.id \text{ where } \neg \exists Points (Points \geq 60 \wedge P(S.id = S.id, Points = Points, C.id = 103))$
- Связь с атрибутами
 - Как сгенерировать $A_1 \dots A_n(R)$? $A_1 \dots A_n$ from R where $R \not\in A_1 = A_1 \dots A_n = A_n$
 - $\delta_\theta(R)$
 - $\exists A = \text{expr}(R)$
 - $R_1 \cap R_2$
 - $R_1 \times R_2$
 - $R_1 \bowtie R_2$
 - $R_1 \bowtie R_2$

$A_1 \dots A_n$ from R where $R \not\in A_1 = A_1 \dots A_n = A_n$

$A_1 \dots A_n$ from R where $R \not\in A_1 = A_1 \dots A_n = A_n$ \leftarrow same expr as A from R ...

$A_1 \dots A_n$ where $R_1(A_1 = A_1) \wedge R_2(A_2 = A_2)$

$A_1 \dots A_n B_1 \dots B_m$ wh. $R_1(A_1 = A_1) \wedge R_2(B_1 = B_1)$

$A_1 B_1 C_k$ where $R_1(\text{inh } A_1, B_1) \wedge R_2(\text{inh } B_1, C_k)$

$A_1 B_1 C_k$ wh $\langle A_1 B_1, B_1 C_k \rangle \sim$ между собой.

$A_1 B_1 C_k$ wh $R_1(\dots) \wedge \neg R_2(B_1 = B_1)$

(20) Dataflow

- 1978 год, изменение генераторов. Идея определять атрибуты. Результат - много ошибок.
- Атрибут: $R(x_1 \dots x_n)$, $\neg P_2(x_1 \dots x_m)$, паралл. задачи.
- [Определение $(x_1 \dots x_n)$:- цель (настор. атрибутов)]
- В атрибут. бывают корректные грэс. zone size 1 year. ~ 1 year. или зонам
- Tакже ныне DNF
- Пример:

Keks(1d, Lastname):-

Students(1d, Lastname, Firstname)
Firstname = 'Kek'

- Запраш. зонами. атрибут. \rightarrow выраж.
- граница бх. & нестор. атриб.
- (NotStudent(1d, Name):-
 \rightarrow Students(1d, Name, -))

анализаторы

- Репурс. запрос: Ancestor(1d, Pid) \wedge Parent(1d, Pid)
- Ancestor(1d, Gid) :- Parent(1d, Pid), Ancestor(Pid, Gid)
- Мн. fixed point: нет но багов. even-60, no $\neg \text{set } \text{he} = \text{rhs}$
- Амортизация памяти: init, репурс. выражение, память не очищается
- Справоч. выражение - нет циклов, скаж. выражение even-угодное

(21) Членство, триггеры

- Коррелированные - соотв. содержимого ТД, 'members' - невидим. процедура
- Членство - непривилегированное ТД - определяем правило и проверка
- Контроль - по заданиям оператора или транзакции. Контроллер - определяет
- Ограничение - типов данных - create type Points as object (T)
not null и т.д.
- Атрибуты, обл. существен. типа
- Описание - primary key, unique, foreign key, check, ...
- Ограничение как штатные правила (CHECK in SQL)
- Ограничение ТД - признаки настройки, генераторы
- Компьютерные действия - распараллеливание, ограничения. В настройках курса определены действия - upd/del, можно no action, cascade, restrict
- Триггеры - выполнение при изм. членства. [set null, set default]
- Приложение - функция, аргументы, параметры, генераторы, структуры.
create trigger on T instead/del, referencing (old, new) row
before/after for each row set...
for each row set...
установка

(22) View'ы

- Использование запроса. Применение - сокращение данных, назначение им языковых конструкций?
- create view Name as Request, drop view Name
- Основные - вставка / удаление; холдинг их взаимодействиях
- Две модели представлений:
 - RP (R) - коррект. данные удаля. Р. Вставка /удал. - напрямую
 - TA (R) - удаление падает, вставка с удал. залогинена
 - Seqf (R) - присоединение
 - R1 U R2 - вставка - т.е., не удал. предикат. Удаление - в отл.
 - R1 ∩ R2 - из отл.
 - R1 → R2 - вставка в R1, удал. из R2
 - R1 × R2 - вставка в R1 и R2 соотв. гасят контекст
- В SQL учарк - одноточие, множест. - нам, содержимо - один-к-одному и од.-ко-множес. со. именем
- Материализованное - хранилище синхрониз., вынужденное обновление, многое бывает. Но устаревает - надо обновл.
- create materialized view Name [refresh { fast/complete } [on { demand/always }] [start with Time] [next Time]]

С б постпроцесс REFRESH always [as Oferty]

(23) Доступ к данным, управление

- Схемы - избирательные (дисперсионные) vs централизованные
- Избирательные - избрать, групп., синхр. прав на объекты
- Централизованные - уровень полн. / ресурса. Члены res. lvl < user. lvl
- Data control language:
 - grant ACT/DIV on RESOURCE to {user/group/public}
 - revoke {privileges} from {...} [cascade/restrict]
- Права - имеют права на объект, но не могут его ограничить
ALTER RESOURCE owner to USER
- Присвоение - не все корректн., ошибки, с. ограничения.

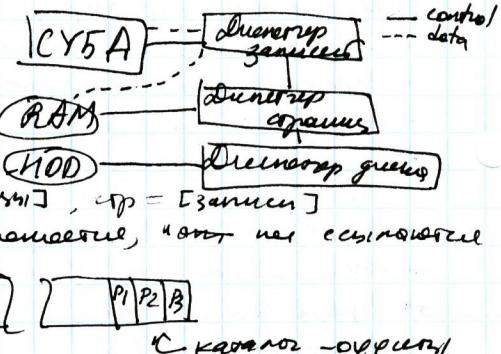
(24) Помощники храним. данных

- Типы памяти - RAM, SSD, HDD ← Холдинг помехозащищено + сжатие
- Помехи разброс на разных страницах, холдинг ← кон-бо ограничено
принцип сдвиг. в память, запрещая / блокируя сдвиги

- Пометка страницы - данные о типе страницы, которые пересылаются между страницами, хранимые пометкой страницы
 - Делитель записей - делит к записи
 - Страница - +распред. памяти, выгрузка данных
 - Линия - начало страницы, определяющее расположение
- Организация:** $\text{Файл} = [\text{таблицы}], \text{таблица} = [\text{страницы}], \text{страница} = [\text{записи}]$
- Структура:**
- ```

 graph LR
 File["[Файл]"] --> Table["[Таблица]"]
 Table --> Page["[Страница]"]
 Page --> Record["[Запись]"]

```
- Структура записи:**
- |                  |          |      |     |      |
|------------------|----------|------|-----|------|
| Prev             | <i>d</i> | Type | ... | Next |
| Payload (запись) |          |      |     |      |
| Страница         |          |      |     |      |
- Payload:**  $\boxed{\quad | \quad | \quad | \quad } \quad \boxed{P_1 | P_2 | P_3}$
- $|d|$  страницы - не пустое, "для нас это полезное"
- В каталоге - ссылки на страницы, указ. на другие страницы других файлов
- Изображения, текст, таблицы
- Страницы - по памяти, префиксное, имперсональное - делают короткими



## (25) Индексация, hash/Btree

- Больше слов. Время: размер таблицы, где хранится. Служебные - они, начальные страницы, помимо обработки / при загрузке/выгрузке) - пустые страницы.
- Кластеризов. (стартовую страницу (далее она включает). Помимо этого, не для каждого страницы, не для каждого хранится, не для каждого).
- Хэш-индекс - преобразование помимо хранения, не для каждого.
- Задачи: поиск в таблицах, есть ли в таблице
- Расширение хеширования - разделяем корзину, если не поместить Глубина хэшия  $n$ ,  $2^n$  корзин. Помимо этого, можно не для каждого корзина  $K$ ,  $2^{n-k}$  корзин, если глубина  $K$  не  $2^{n-(k+1)}$
- Экспрессии `in/exist/count`, поиск по ключу.
- Упорядоченный индекс  $- B^{(+)} -$  деревья корней и первые несколько листьев. - в памяти хранят  $\min(h)$ , можно использовать.
- Плотные и разреженные - плотные все элементы VS разреженные
- На страницах много записей в блоке - кластеризов. А суперблоки - один, зарезервирован.
- Ускорение проверки существования, поиска, удаления, либо не сразу.

## (26) Индексацию 2

- битмап
- Битмапный:  $\langle |d|, \text{sex}, \underbrace{\text{hM}, \text{hF}}_{\text{Root}}$  - но быть не возможное значение усредненное помимо  $i$  в выражении, count, sum, sum отображает.
  - Типы: многозначный, с накоплением, составной (both).
  - Организации: все в памяти, хранение (ALE,  $\uparrow$  более групп специальных), создание корзин. Размер - число корней  $\times$  число записей
  - Индекс на B-деревьях - где подразумевают макс. знач. (64 бит в PostgreSQL) - generalized search tree
  - C lossy, может обновляться Support vector Value type.
  - GIN - inverted index

## (27) Продукты / функции

- Продукты - базы, хранимые в DB (в формате). Обработка на стороне DB, возможностью кашинга, замена данных
- Продукты не обозначают конкретные значения. Функции - только input параметры.
- Функции можно вызывать из процедур, но не наоборот
- Продукты поддерживают transaction management.
- Продукты не могут использоваться в WHERE/HAVING/SELECT
- Управление базой данных - приведение execute. Использование прав definer/invoke

• Результат можно отнести к агрегации, возвращают вспомогательные значения (результаты), относящиеся к недавнему запросу.

## (28) Универсам, неизвестное

SQL

- Операторы: label: begin \n ... end; then body  
— присваивание: set var = val;  
— if /then /else if / else / end if;  
— case exp \n [when exp2 then ops] + [else ops] end case;  
— не  $\rightarrow$  можно определить, но в запросе / set v = case ...)  
— [label1] while cond do \n ... \n end while [label2];  
— repeat: until cond \n end repeat;  
— loop \n body \n end loop — leave / iterate
- Автомат — declare & continue / undolexity handler for COND OP
- Выходные: declare exit, noerror OP
- Взапрос: declare continue .. begin sRet = 5 end;
- Курсоры: declare cursor for (запрос), можно использовать в разных  
— open / close — открытие, закрытие. не бывает  
— fetch {next|prior|last|first} запрос into VARS  
— курсоры — не found в sgisstate 02000  
— изменение не курсора: update TABLE set WHERE cumentof C1AS  
delete from T where cument of C1AS

not found  
sgisstate.N  
glutinous  
stolen position

## (29) DCL

- 2 настройки — grant / revoke
- Б базы — commit, Б postgres — можно откатить
- ~~Разрешения~~ ~~разные~~ SPILL, Alter  
With grant option — неприменим потому что не grant
- CASCADE — г глобально и зависимое привилегии (коэффициент)
- RESTRICT — огранич.

## (30) Transactions.

- Несколько — где операторы. Минимальный фиксированный набор
- Неорганизованные — ALTER
- begin, commit, rollback
- ACID: i — Isolation (транзакции не влияют.)  
d — durability (не проходят изменения)
- Кодекоммит — vs кодекоммит  
RL  $>$  5A удобн. огранич.  
закрытии.
- Чистота данных — постоянство транзакции коррект. даже при сбое  
Хранение на диске (RAM — потеря при reboot)
- Типы транс: локальн. (одна тx), глоб. (все тx), аноним. (помимо данных) RAM  
Хорошее первые 3 отвечают аноним. однородность
- Восстановление: первые — фиксируют, остальные и далее — очищают  
— shadow copy пишут в копии  
— transaction log пишут в отк. файл
- Журнал — пишут помех. старое, новое, маркеры начало/закрытие.
- Пишут много — называют Snapshot. множеств. измен.  
закрытия тx, откат., открытие }  
не в. radio
- Синхронизация — id тx, указывает на пред. транзакцию  
начало, изменение, закрытие, откат
- Трия восстановления

регистрация  
РАМ, изображение  
запись членов.

- Команды отмены:
  - раздел: откат, возвр. Redo / Undo.
  - номер откат. комм. номер до наст.
  - на комп. тоже! положение все в Undo.
    - бесп. начало (маркер) — начало в Redo
    - конец — в Redo
- Откат Undo: от нач. состояния до насту. until Undo
  - если маркер начало — удаляем из Redo
  - изменение — если произошло в Undo, откатим
- Повтор:
  - от нач. кт до конца
  - маркер конца — удаляем из Redo
  - изменение — если в Redo, применить
- Удаление данных из журнала: прокатка, закомм. без comp. tx,
- Повторные записи: в памяти — ok, но не хотят не ~~помогают~~ записей
  - зап Undo (не зале Redo — разн. изм. уже применена)
  - снова изменение — изменение. Анализир. когда журн. рабоч.

## (51) ACES

- Раздел: разметка, повторение, откат.
- Повторение: начинаем от КТ до конца. Изменение — применение
- Можно обрез. с разметкой
- Откат:
  - Читаем журнал от конца к началу до tx в Undo
  - Откатываем изменение, внести запись от откатом

[откат, запись на диск, ~~запись~~. запись]
- Комм. проходов: в класс. 3, в ACES 2
- Без журнала: запись откатов vs изменения.
- Повторные записи — ACES позволяет однотипные, идент. раб.

## (52) Блокировки

- Хотим Isolation — надо помехи не реагир.
- RW-lock, 2PL → завершение или откат — отложили
  - берем нужные пухи
- Deadlock'и — фикции графики ожидания, генерируемые. Если это case, то откладываем эту tx, перезапускаем погас
- Преобр. Deadlock'ов: А применяет на блокировку памяти. ↳ В:
  - откат — отмена: если A началась раньше В, А откат, иначе В отмена.
  - отмена — откат: если — || — , В откат, иначе А откат
- 2PL — гарантированное упорядочивание tx (хотя по ин. чин)
  - делаем retrieve同一урл, получим откаты
  - если A иен. значение изм. В, то finish(A) > finish(B) ↳
    - приходит востановка
  - откат B ⇒ откат A — откаты
  - отсутствие конфл.
- Гарантия — запись/создание/изменение/удаление
  - запись — первым делом фиксируем запись (read >> read)
  - удаление — на копировании, ALTER, прав.所有权

## (53) Изоличие

- first of all
- serializable, snapshot, repeatable read, read committed, read uncommitted

### 59) Репликации → распред. транзакций

- Синхронная и асинхронная. Синхрон. компкт / асинхор.
- С синхрон. компкт: транс. в ~~транс.~~ узле, пишет в лог, изменил состояния. Синхронизация — когда согласовано. Distributes.
- Данные от удал. — проходит транзакцию — распределение. Транзакция — операторы (внешн. функции). Запись логов.
- операторы — мало данных, но могут быть пред. с синхрон. Кеширование уничтожает записи. переходок транзакций.
- записи — демерки, но много данных
- Применение: мастер-сервер. (+ рез/сер), предпр. данных, резервные копии.
- вертикаль. паспорт — хандромиз. паспорт (если паспорт > паспорт)
- горизонт. — схема (предр. распределенность, паспорт. сервер, паспорт.
- предпр. данных — ассим. схема. Переод. синхрон., при репликации данные преобразуются — изменение формат данных, утилизируются
- резерв. копирование — паспорт. синхронизация, ассим. схема, копиров.
- основной репликац.

### 60) Расп. транс. — что и при каких условиях работает так в изолированном виде — паспорт. независимость, асинхронные транс. узла (безоп.), паспорт. уничтожение,

- Паспорт. от расп. (использует паспорт. к паспорт. паспорт), независимость от транзакций (использование не знал о транзакции) — уничтожение транзакций. паспорт., паспорт. синхрониз., независимость от репликации (автомат. ввода узла, поддержка built-in)
- Расп. транс.: 1 запрос от тела со многих подо.
- Расп. транзакции: 1 tx выполнение на пасп. узлах, поддержка синхрон. фиксации и откатов
- Копии WORA ~~platform~~ independence, DB indep., network / DB independence <sup>конвертацию</sup> ~~транс.~~ <sup>транс.</sup>
- Расп. транс.: хомяк синхрониз. — один узел/компьютер/коммуникации между 2PC, но могут создать непонятно что — может быть изменин. состояния (2 генерация). Может быть с timeout, повторять, но гарантировать решение асинхронным
- Обработка ~~свой~~ <sup>номер</sup> до приходит решение — есть, но применяется паспорт в протокол, если приходит винчестер
- Установка — запрашивает у изолированного, но пасп. решение пишет в протокол, пишет — выполняет транс., синхр. изолированные, записи в протокол
- Между приложением и пасп. надо паспорт tx.
- Протокол предполож. синхр. Копия — записывает tx если нет между при. и пасп., а если синхрониз. то это делает. Если нет.

### 61) Расп. базы

- CAP-принцип: 2 из 3x. Но если есть не багажник то все OK consistency, availability, partition tolerance
  - в случае обрыва связи пишет в A  $\xrightarrow{A \rightarrow \text{уничтожен. или замен. состоян.}}$
  - пишет в C/A  $\xrightarrow{C \rightarrow \text{неоч., есть есть}}$
  - когда оба получат ответ write от B  $\xrightarrow{\text{один из них получит ответ}}$
- BASE — basically available узла пишет только с учетом конф., soft-state (измен., если из внешнего виду), eventual consistency (брейчни. независим.)
- Отличительное запр. — можно помнить только коммутаторы, данные временные виды; быстрые узлы
- Пасп. от изолированное — исп. пишет где изолирует. грузовик, запасы, инвентарь. поддержка расп. tx, блокировкой.

Роль — не доп. здравие.

### (42) Упакр. генеал - упк. Генеал

- Хранен  $\times$  пакуване на, year newer ancestors, parent.. роднини
- Създава на упаква,  $\text{Хранен pid} = \text{id}$ ,  $\langle \text{id}, \text{pid}, \text{data} \rangle$
- Роден - пакуване - select from forest where  $\text{pid} \neq \text{id}$ ,  $\langle \text{parent, id} \rangle$   $\text{pid} \neq \text{null}$
- Наследник - упаку - create view ancestor ( $\text{id}, \text{aid}$ ) as with recursive  
ancestors ( $\text{id}, \text{aid}$ ) as sel  $\text{id}, \text{pid}$  from forest union select  $\text{d. id}, \text{a.id}$   
from area inner join forest f on  $f.\text{pid} = a.\text{id}$ ; select \* from anc.
- Годбен. упаку/ната - insert
- Упакуване меса - упако. негереба:  
delete from forest where  $\text{id} = :id$  on id in (sel) from ancestors  $\text{aid} = \text{id}$
- Наследник. негереба - именем parent
- Населок на генекс - геноне index, граене чисто by  
+ мало генуси, упако отобр
- Репуб. Запади, нем. Генерозуб.

constraint / constraint

### (43) Bl. ми-ка

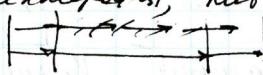
- Край - left/right - брояне броя в броя.  $\langle \text{id}, l, r, \text{data} \rangle$
- create view ancestor ( $\text{id}, \text{aid}$ ) as  
Select d.id, a.id from forest d, forest a where d.l between a.l and d.r  
( $g.w$ )
- create view parents ( $\text{id}, \text{pid}$ ) as  
select c.id, p.id from forest c, forest p  
where c.l < p.l and c.r < p.r and not exists  
(select \* from ancestor where (l between p.l and p.r)  
and (c.l between l and r))
- Select max(r) into m from forest;  
insert into forest values (:id, m+1, m+2, data) } месеци
- Годбен. меса - брояване wr - народи, негенези  $| + 2, r+2$  град близор  
Близор
- Упак. меса - shrink  $\nearrow$  анат.
- Упак. негереба - анатомии, но II и III
- Наследник. негереба - годбенни меси, населени, генеси, граене
- Чисто - пакуване събране, но друг чисто него ~ пакуване
- Населок - order by l
- + мало генуси, упак., гореца на селекции
- Чисто (у генко ~ On) една не пакув.) отобр.

### (44) Модел наимен

- Хранен наимен  $\langle \text{id}, \text{path varchar}(100), \text{name varchar}(100) \rangle$ :
- ancestor ( $\text{id}, \text{aid}$ ) as select d.id, a.id from forest d, forest a  
where a.path like (d.path || '%')
- child ( $\text{id}, \text{aid}$ ) as select c.id, p.id from forest c, forest p,  
where p.path like (c.path || '%') and length(p.path) + L + 1 = length(c.path)
- Близор - insert into forest ( $\text{id}, \text{path}, \text{data}$ ) values ( $\text{id}, \text{label}, \dots$ ) root
- insert into forest ( $\text{id}, \text{path}, \text{data}$ ) values'  
( $\text{id}$ , (select path from forest where id = :pid) || '.' || :label, ...)
- Delete from forest where id = :id
- Delete from forest where path like (sel path from forest where id = :id) || '.' || %)
- Населени - именем специалист наимен със/наимен
- Населок - но събен. меси
- + Упак., упако. меси / отобр
- Път ген наимен x пакуване, пакуване на селекции

- (45) Временные функции, полуфремкин.  $\delta t$
- time, date, time, datetime, timestamp ( $\delta t$ )  $\xrightarrow{\text{значение и время}}$  interval
  - UTC
  - Времен. функции - зависят от времени
    - чисто: напр. о изменениях, о правильности, выражение  $\text{когда записано}$
    - Врем. запросы - хранение моментов на записи (просто, напоминание, но сиюто с измен.), момент на  $\delta t$  (чтобы, можно засечь)
    - Операции - изменение, запросы на момент времени, запросы изменений (с зарег. по врем.)
  - Полуфремкинские  $\delta t$  - значение + метод с параметром. значение предсказывается, какому запросу по методу - например метода с последующим заб. со врем.
  - Изменение генерирует момент, сущее запись, status  $\langle id, timestamp, \dots \rangle$
  - Изменение генерирует момент, сущее запись, status - не поддерж.

### (46) Временные $\delta t$ , чист. изменения

- арифм. в полуфремках. Но только на каст. и паралл.
- Ключ student & groups - ( $sId, gId, X$ )  $X = \delta t / \text{nd}$  student, groups, studentgroups (PK1, PK2,  $\delta t$ )
- Текущие - ставить nd ~ inf
- Чистые. не параллельные: check(not exists(select \* from s1 ~~join~~ s2 inner join sG s2 on s1.studentId => s2.Id where greatest(s2.bg, s2.ebg) < least(s1.nd, s1.ed)) No ограничение на compare\_fn
- Согласие - select t1.\* , t2.\* , greatest(t1.bg, t2.ebg) as bg, least(nd,ed) and from t1 inner join t2 on condition and greatest(t1.bg, t2.ebg) < least(t1.nd, t2.ed)
- Переоценка - синтакс. соглашение
- Обединение - T1 (greatest bg < least nd)  $\vee$  T2 (.. ?)
- Запросы - на нек. нек. момент: when nd = inf
  - на запросы ( $>$ ,  $<$  ..)
  - изменения, синхронизация.
- Вставки - insert (...) current\_date, inf)
  - Основа - нов. чистый он объект, добавили нов. Удали - old
- В промежуток - надо обсуждение чистых, надо результирующих синхронизаций  $\xrightarrow{\quad}$  

### (47) Модели состояний

- Хранят только момент нахождения - избегают изменения состояния
- Запросы: чист. изменения - аспекты с условиями, селекции и множества - через чистые. преобразование
- Insert - просто вставки с пустым. гарант. Изменение в промежуток - семантика неоднозначна
- + Чистописание нового, недавнее значение поменяло
  - чист. удали, чист. метод. запросы, если удаляют

### (48) Ключи и ссылки. DB2M

- Ключи облегчают идент. новых, программ. логик / функциональ - в технике, в переходах объектов, в  $\delta t$ , в чист.
- автоном. определение none, конфигурационное. Но при софт. объекта неизв. ключ  $\rightarrow$  надо генерировать правильные объекты.
- Текущие имена с подключ.  $\delta t$
- Генерации на уровне - GUID (датум. паспорт), select(max(id)) - привлечь с параллельным