

# ЭВОЛЮЦИЯ СУБД

Олег Бартунов

ГАИШ МГУ, Postgres Professional



# When I started using Postgres

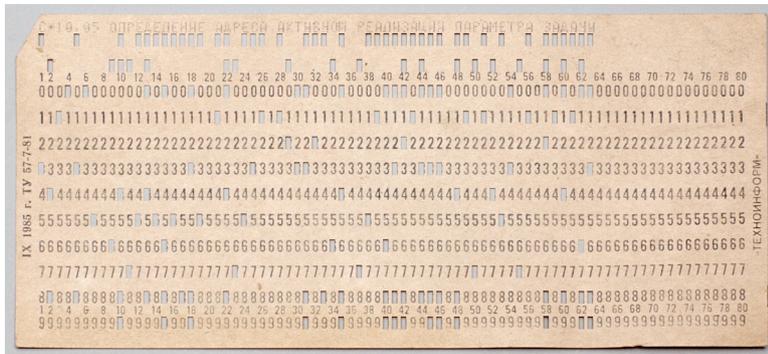
- No UTF-8, even no 8-bit
- No WAL
- No MVCC
- No replication
- No usable non-scalar data types
- No subselects, no window functions, no CTE
- It was Postgres95



- Файлы
- Иерархическая
- Сетевая
- Реляционная
- Объектно-ориентированная
- Объектно-реляционная
- NoSQL
- NewSQL
- Что дальше ?



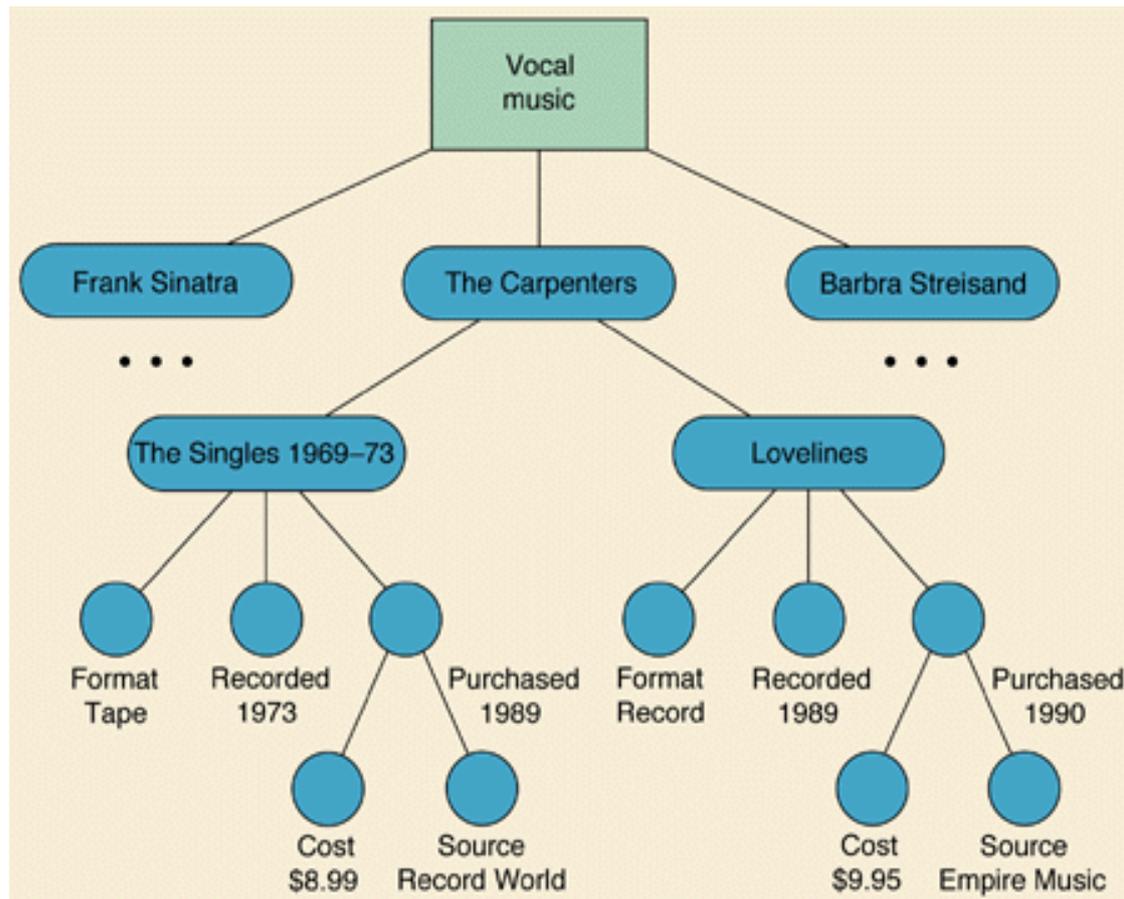
Продажа раба и здания, ~ 2600 BC  
Шумерская табличка из Шуруппака



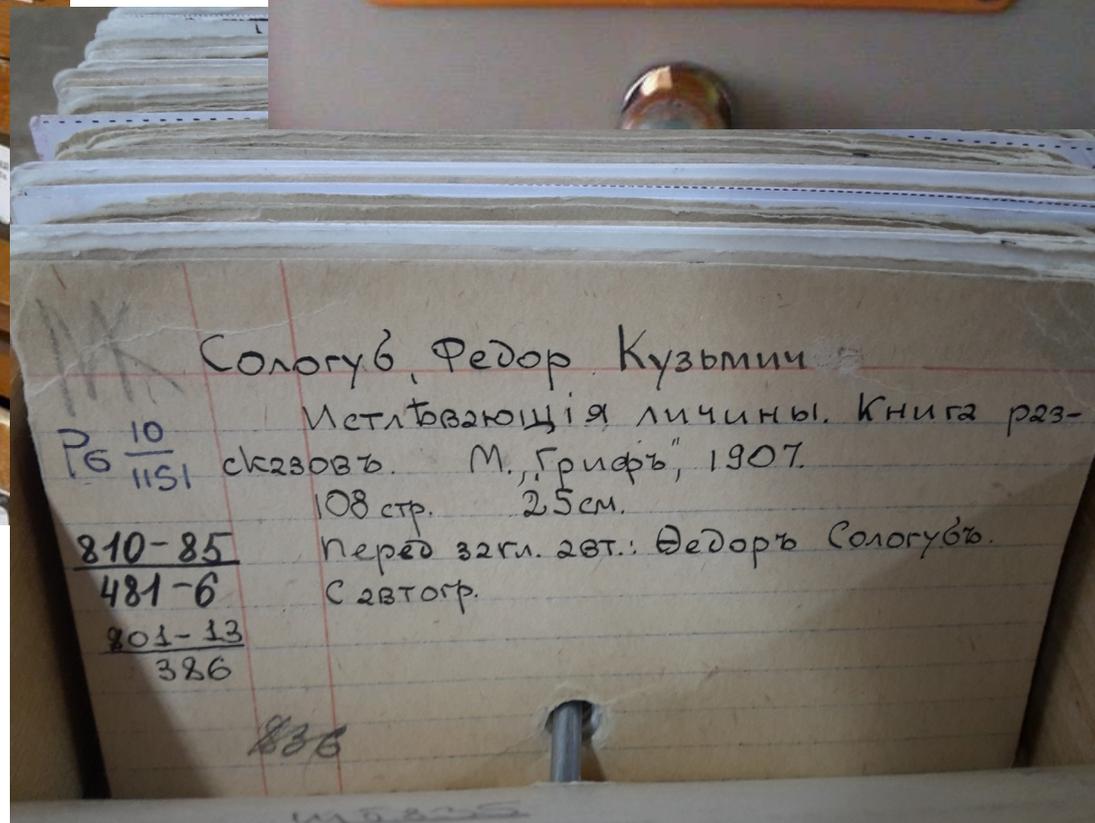
- Компьютеры для расчетов, программы и данные жили вместе
- Данные ушли в файлы, методы доступа (последовательный, индексный, случайный)
- Базы данных на «плоских» файлах
  - 1С до сих пор работает
  - Berkeley DB
  - Проблема многопользовательского доступа, нет транзакций, только программа знала формат записи
- Базы данных на инвертированных списках
  - ADABAS (1970), советские аналоги ДИСОН, ТРИАДА

## Иерархическая - 1950-1960

- IMS IBM, 196X — now! (ОКА), ИНЕС (ЕС), Windows Registry, fs
- Разделение схемы и данных
- Появились связи - «ОДИН-КО-МНОГИМ»

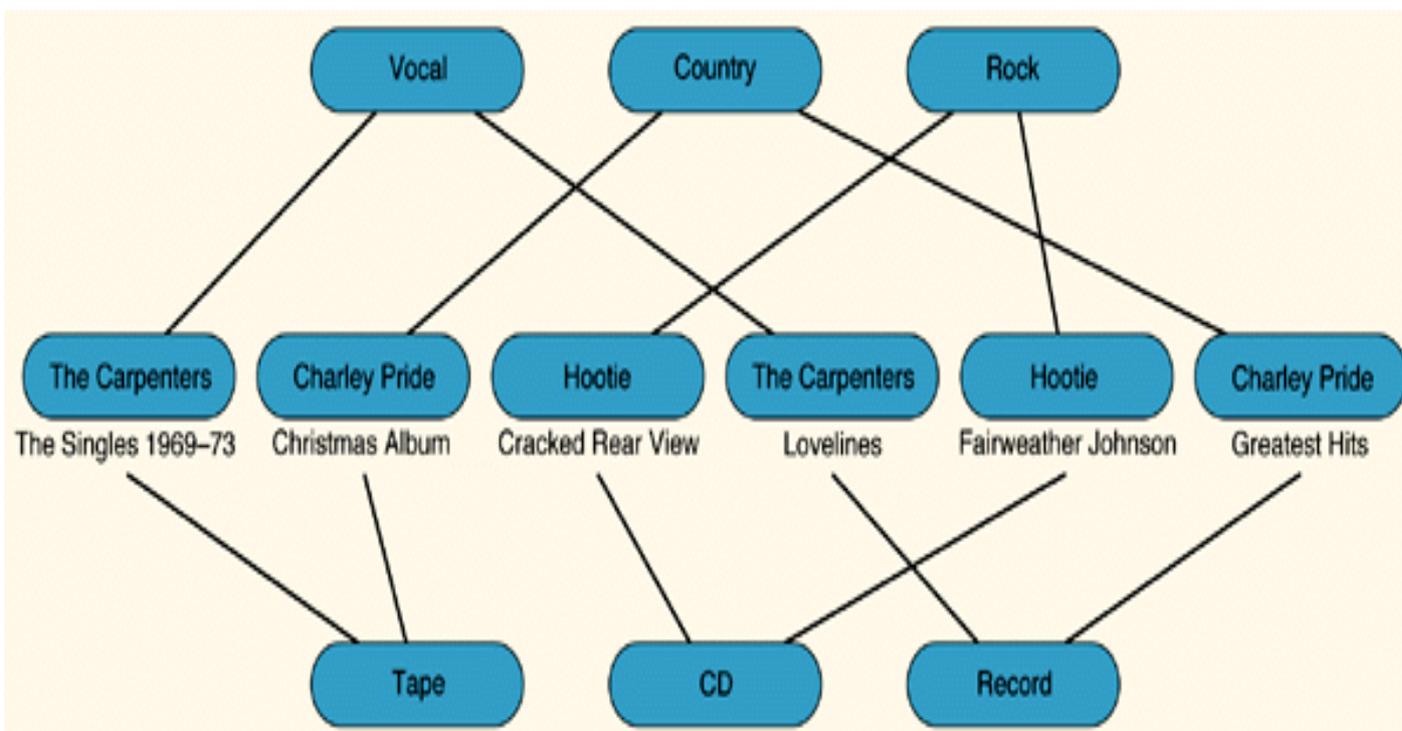


Иерархическая - 1950-1960

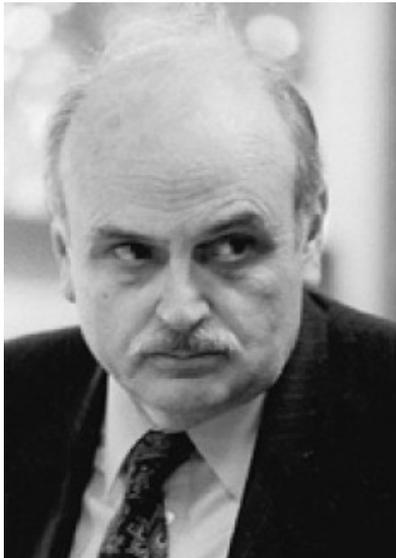


## Сетевая (произвольный граф) 1960-1970

- IDS (GE), IDMS (1973 — now !CA IDMS), куча советских субд
- Развитие иерархической
- Parent-child (parent — many children, children — any # **parents**)



Edgar F. Codd, Communications of the ACM, Vol. 13, No. 6, June 1970



From records to relations

Information Retrieval

P. BAXENDALE, Editor

## A Relational Model of Data for Large Shared Data Banks

E. F. Codd  
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on  $n$ -ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data base, data structure, data organization, hierarchies of data, networks of data, relations, derivability, redundancy, consistency, composition, join, retrieval language, predicate calculus, security, data integrity

CR CATEGORIES: 3.70, 3.73, 3.75, 4.20, 4.22, 4.29

### 1. Relational Model and Normal Form

#### 1.1. INTRODUCTION

This paper is concerned with the application of ele-

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for non-inferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the “connection trap”).

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

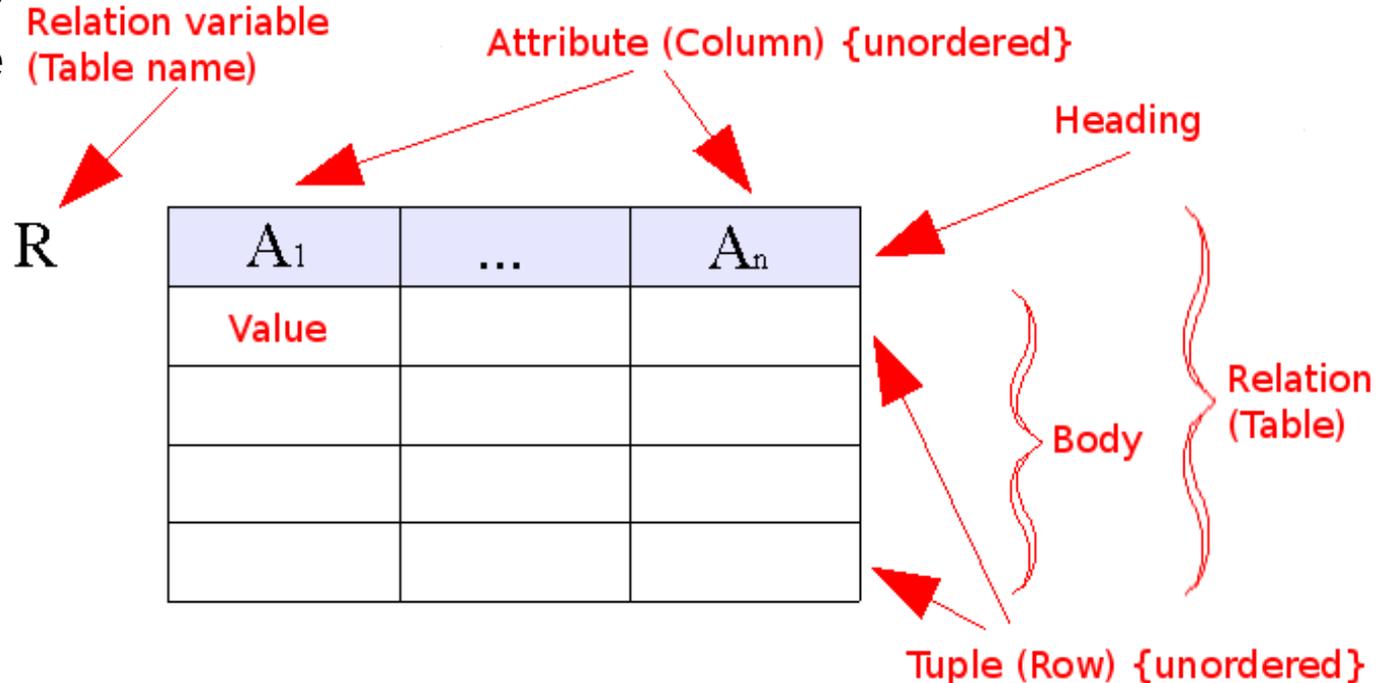
#### 1.2. DATA DEPENDENCIES IN PRESENT SYSTEMS

The provision of data description tables in recently developed information systems represents a major advance toward the goal of data independence [5, 6, 7]. Such tables facilitate changing certain characteristics of the data representation stored in a data bank. However, the variety of data representation characteristics which can be changed *without logically impairing some application programs* is still quite limited. Further, the model of data with which users interact is still cluttered with representational properties, particularly in regard to the representation of collections of data (as opposed to individual items). Three of the principal kinds of data dependencies which still need to be removed are: ordering dependence, indexing dependence, and access path dependence. In some systems these dependencies are not clearly separable from one another.

1.2.1. Ordering Dependence. Elements of data in a

## 1970 - now

- Оперирует логическими понятиями (отношениями), а не физическими структурами
- явное задание значений атрибутов в кортежах отношений (нет указателей)
- Реляционная алгебра
- Декларативное программирование
- Де **Relation variable**
- Де **(Table name)**



# Codd's 12 rules (1985)

- **Правило 0: Основное правило (Foundation Rule):**

Система, которая рекламируется или позиционируется как реляционная система управления базами данных, должна быть способна управлять базами данных, используя исключительно свои реляционные возможности.

- **Правило 1: Информационное правило (The Information Rule)**

Вся информация в реляционной базе данных на логическом уровне должна быть явно представлена единственным способом: значениями в таблицах.

- **Правило 2: Гарантированный доступ к данным (Guaranteed Access Rule):**

В реляционной базе данных каждое отдельное (атомарное) значение данных должно быть логически доступно с помощью комбинации имени таблицы, значения первичного ключа и имени столбца.

- **Правило 3: Систематическая поддержка отсутствующих значений (Systematic Treatment of Null Values):**

Неизвестные, или отсутствующие значения NULL, отличные от любого известного значения, должны поддерживаться для всех типов данных при выполнении любых операций. Например, для числовых данных неизвестные значения не должны рассматриваться как нули, а для символьных данных — как пустые строки.

- **Правило 4: Доступ к словарю данных в терминах реляционной модели (Active On-Line Catalog Based on the Relational Model):**

Словарь данных должен сохраняться в форме реляционных таблиц, и СУБД должна поддерживать доступ к нему при помощи стандартных языковых средств, тех же самых, которые используются для работы с реляционными таблицами, содержащими пользовательские данные.

- **Правило 5: Полнота подмножества языка (Comprehensive Data Sublanguage Rule):**

Система управления реляционными базами данных должна поддерживать хотя бы один реляционный язык, который

- (а) имеет линейный синтаксис,
- (б) может использоваться как интерактивно, так и в прикладных программах,
- (в) поддерживает операции определения данных, определения представлений, манипулирования данными (интерактивные и программные), ограничители целостности, управления доступом и операции управления транзакциями (begin, commit и rollback).

- **Правило 6: Возможность изменения представлений (View Updating Rule):**

Каждое представление должно поддерживать все операции манипулирования данными, которые поддерживают реляционные таблицы: операции выборки, вставки, изменения и удаления данных.

- **Правило 7: Наличие высокоуровневых операций управления данными (High-Level Insert, Update, and Delete):**

Операции вставки, изменения и удаления данных должны поддерживаться не только по отношению к одной строке реляционной таблицы, но и по отношению к любому множеству строк.

- **Правило 8: Физическая независимость данных (Physical Data Independence):**

Приложения не должны зависеть от используемых способов хранения данных на носителях, от аппаратного обеспечения компьютеров, на которых находится реляционная база данных.

- **Правило 9: Логическая независимость данных (Logical Data Independence):**

Представление данных в приложении не должно зависеть от структуры реляционных таблиц. Если в процессе нормализации одна реляционная таблица разделяется на две, представление должно обеспечить объединение этих данных, чтобы изменение структуры реляционных таблиц не сказывалось на работе приложений.

- **Правило 10: Независимость контроля целостности (Integrity Independence):**

Вся информация, необходимая для поддержания целостности, должна находиться в словаре данных. Язык для работы с данными должен выполнять проверку входных данных и автоматически поддерживать целостность данных.

- **Правило 11: Независимость от расположения (Distribution Independence):**

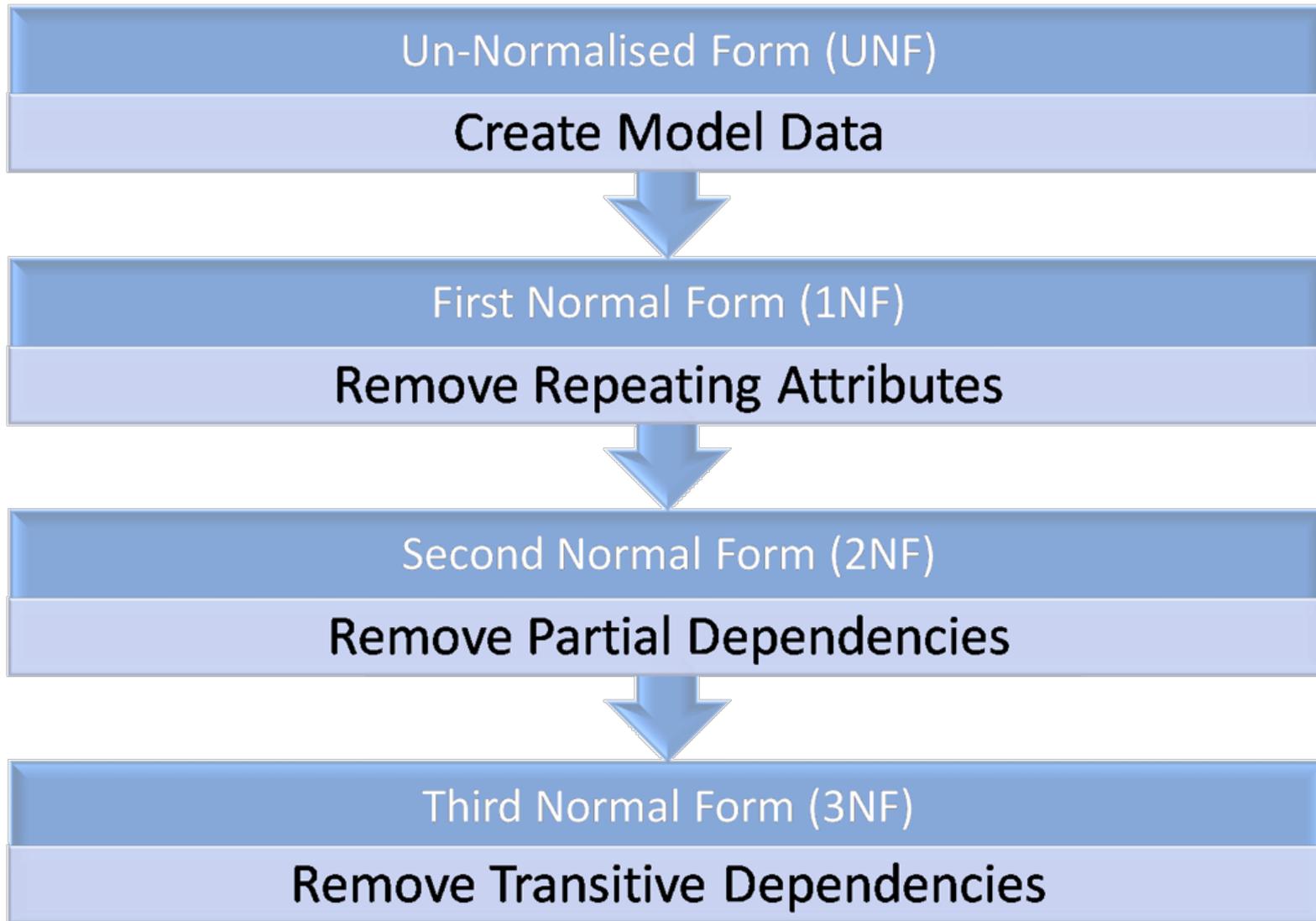
База данных может быть распределённой, может находиться на нескольких компьютерах, и это не должно оказывать влияния на приложения. Перенос базы данных на другой компьютер не должен оказывать влияния на приложения.

- **Правило 12: Согласование языковых уровней (The Nonsubversion Rule):**

Если используется низкоуровневый язык доступа к данным, он не должен игнорировать правила безопасности и правила целостности, которые поддерживаются языком более высокого уровня.

# Нормальная форма

НФ - критерий надежности, всего их 9



Donald Chamberlin and Raymond Boyce (IBM), 1974  
 Language “more accessible to users without formal training in mathematics or computer programming.”

Стандартный язык работы с данными в СУБД (ANSI SQL-86)

$$\pi_{e.name} ( \sigma_{e.salary > m.salary} ( \rho_e(employee) \bowtie_{e.manager = m.name} \rho_m(employee) ) )$$

(a) Relational Algebra version

```
RANGE employee e;
RANGE employee m;
GET w (e.name): ∃m((e.manager = m.name) ∧ (e.salary > m.salary))
```

(b) Relational Calculus version

```
select e.name
from employee e, employee m
where e.manager = m.name and e.salary > m.salary
```

(c) Sequel (SQL) version

**Figure 2.** Three versions of the query, “Find names of employees who earn more than their managers.”

## Объектная (8X-е годы)

- Данные сложной структуры, работа напрямую без проекции на строки и колонки
- Уникальный id, инкапсуляция (доступ только через API), типы и классы, наследование, полиморфизм
- Манипулирование данными на полноценном языке программирования
- Расширяемость типов данных (1<sup>st</sup> class citizen )
- ORDBMS захватили нишу ООБД, много данных уже были в RDBMS, устойчивый SQL стандарт, OQL (OMG) непопулярен

## Объектно-реляционные (гибрид) для преодоления проблем реляционной модели

- Мало типов данных
- Слабая расширяемость типов
- Сложная интеграция в новые системы, основанные на объектах
- SQL:1999 - добавили пользовательские типы
- PostgreSQL — яркий пример

# What is PostgreSQL

PostgreSQL — The world's most advanced open source object-relational database.

Extensible — data types, operators, functions, access methods (indexes).

[ANSI SQL](#) (1992, 1999, 2003, 2008, 2011),  
NoSQL (key-value, JSON, JSONB)

Developed and supported by independent international community (users, developers, companies)

Pronunciation: post-gress-Q-L, post-gres, pgsql

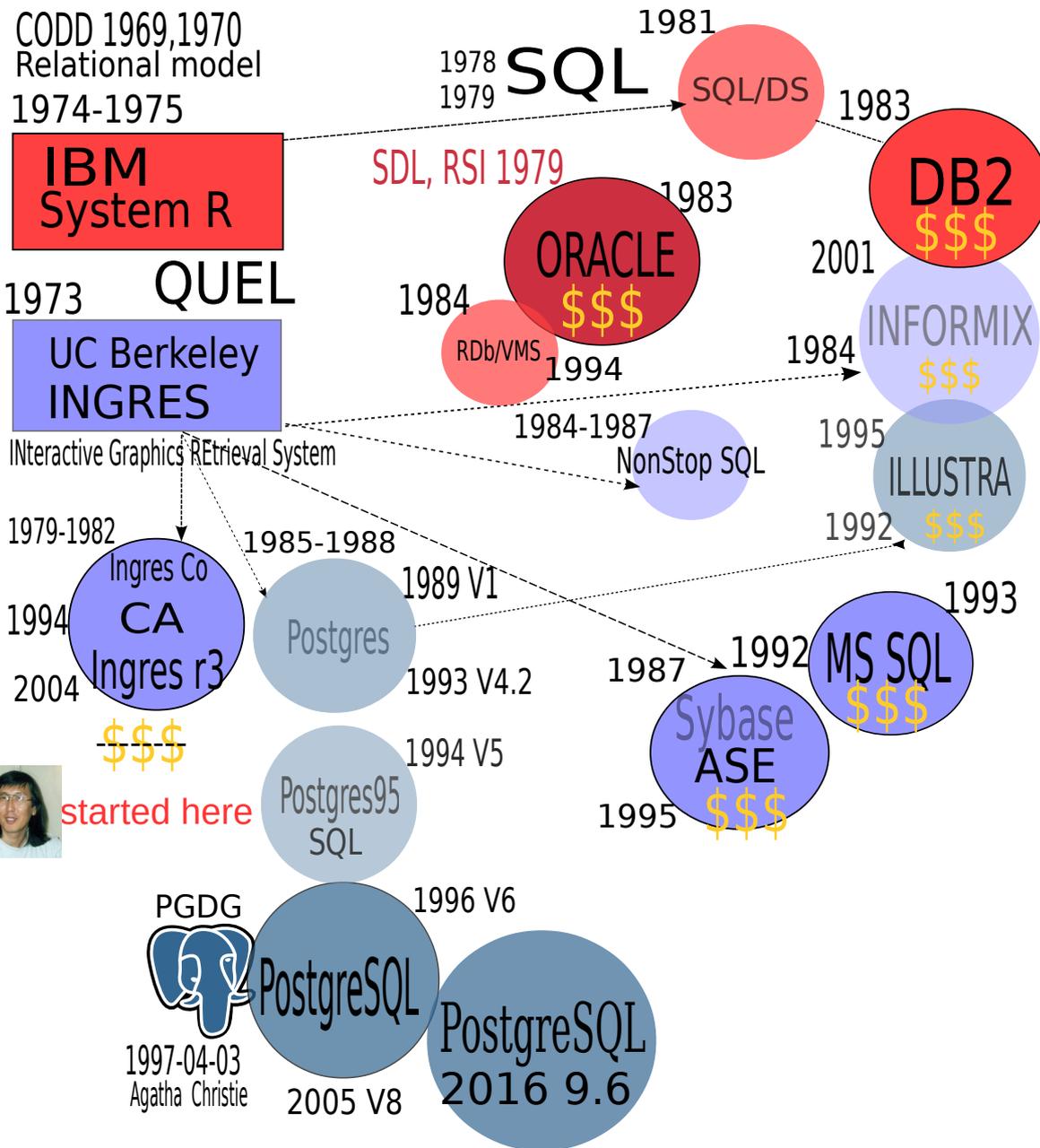
Web: <http://www.postgresql.org>, ( [BSD](#), [MIT](#)) - like



# PostgreSQL History >20 years



Michael Stonebreaker  
Turing Award, 2015



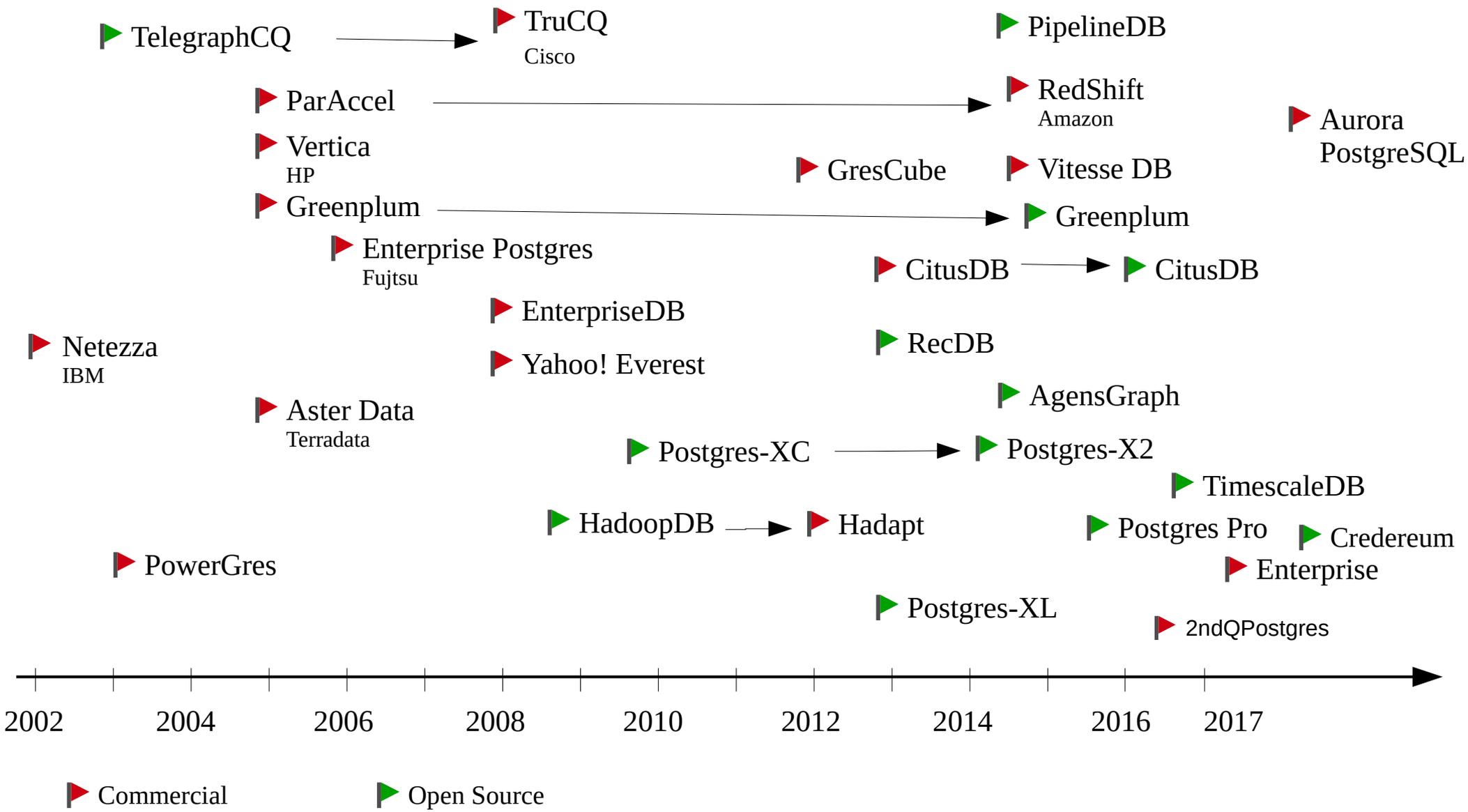
2017 10



- Any project could start with PostgreSQL
- PostgreSQL is a reliable and stable database with rich functionality and long history
- PostgreSQL has liberal BSD license, cross platform ( ~30)
- Developed by international community, no vendor lock
  
- PostgreSQL is **EXTENSIBLE**, this is the very important feature, which people miss ! It allow database to support
  - New workloads
  - New functionality
  - New environment
  - Often without restarting a server, no need core programmer.



# PostgreSQL: OLTP, MPP, OLAP, CLOUD, GIS, STREAM, TIMESERIES, GPU, BLOCKCHAIN



## Реляционные СУБД — интеграционные

- Все приложения общаются через СУБД
- SQL — универсальный язык работы с данными
- Все изменения в СУБД доступны всем
- Изменения схемы очень затратны, медл. релизы
- Рассчитаны на интерактивную работу
  - Интересны агрегаты, а не сами данные, нужен SQL
  - SQL отслеживает транзакционность, ограничения целостности... вместо человека

## Сервисная архитектура изменила подход к СУБД

- Приложение состоит из сервисов, SQL->HTTP
- Сервисам не нужна одна монокитная СУБД
- Часто достаточно простых СУБД — (key-value)
- Схема меняется «на ходу», быстрые релизы
- ACID → BASE
- Сервисы — это программы, которые могут сами заниматься агрегированием
- Сервисы могут сами следить за целостностью данных

# NoSQL (другие предпосылки)

Много данных, аналитика, большое кол-во  
одновременных запросов

- Распределенность - кластеры дешевых shared-nothing машин

NoSQL —горизонтальная масштабируемость и  
производительность

## Key-value databases

- Ordered k-v для поддержки диапазонов

## Column family (column-oriented) stores

- Big Table — value имеет структуру:
  - column families, columns, and timestamped versions (maps-of maps-of maps)

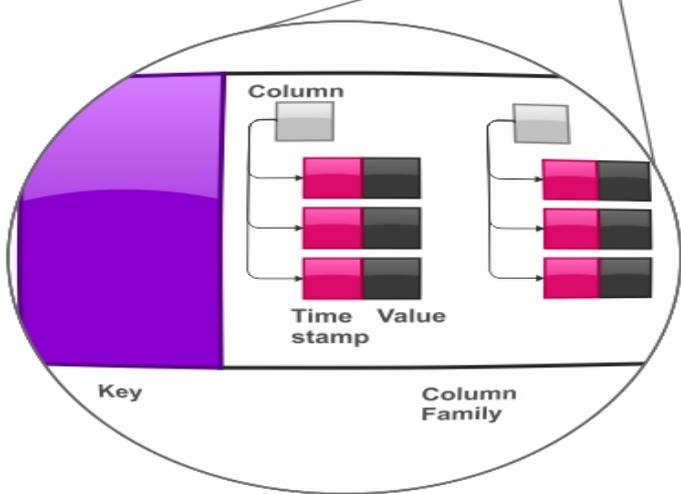
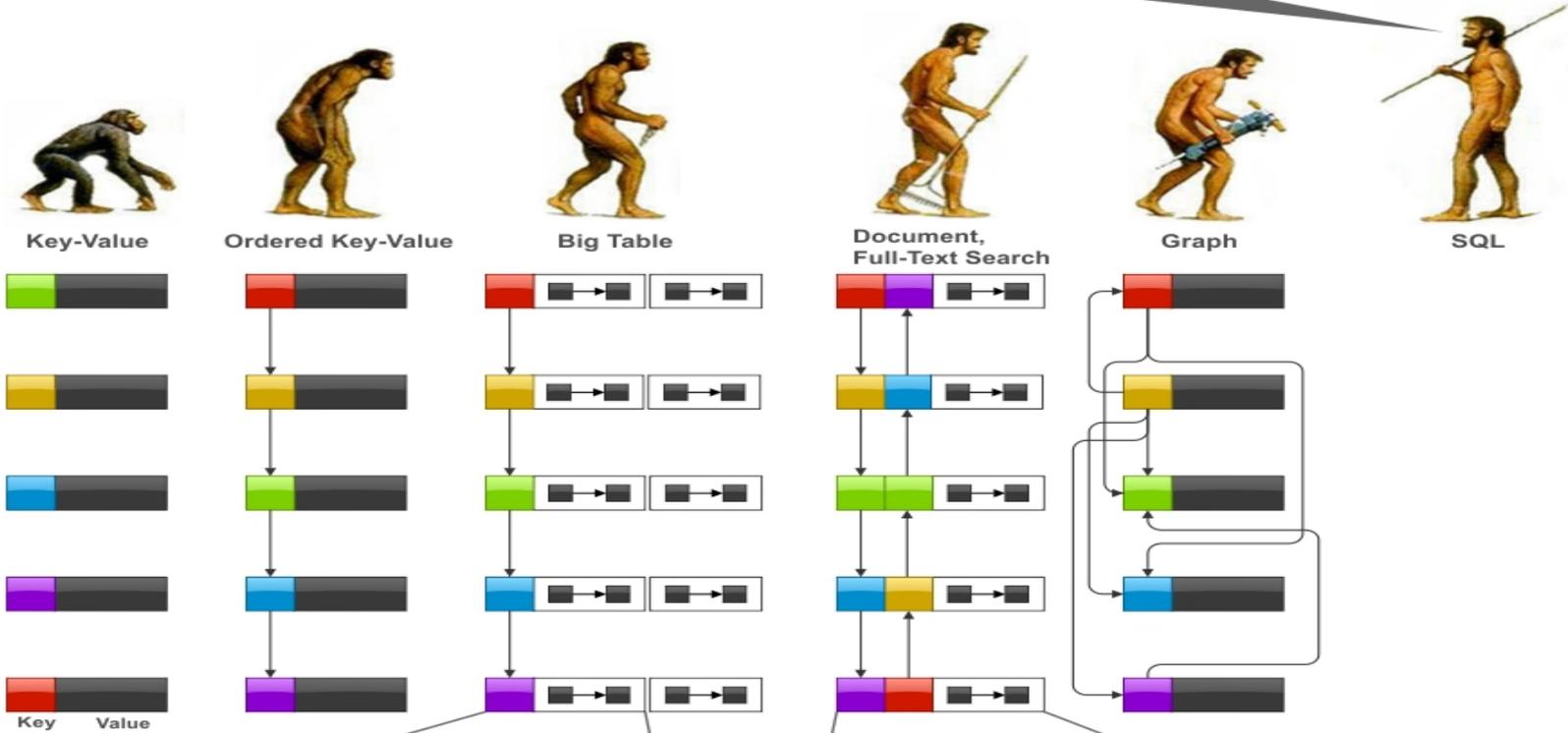
## Document databases

- Value - произвольная сложность, индексы
  - Имена полей, FTS — значение полей

## Graph databases — эволюция ordered-kv

# NoSQL

Stop following me, you fucking freaks!



```

"employee" :
{
  "name" : "Mohana Pillai"
  "position" : "Delivery"
  "projects" : [
    {
      "name" : "Easy Signu
    }
  ], Semi-Structured Data
}
Plain Text
is a confidential word or number
combination used as a code to
identity when accessing
en 8 and 15 characters
number and may ne
spaces
  
```

Неудачное имя, есть реляционная СУБД NoSQL,  
1998

([http://www.strozzi.it/cgi-bin/CSA/tw7//en\\_US/nosql/Home%20Page](http://www.strozzi.it/cgi-bin/CSA/tw7//en_US/nosql/Home%20Page))

– Not-only-SQL

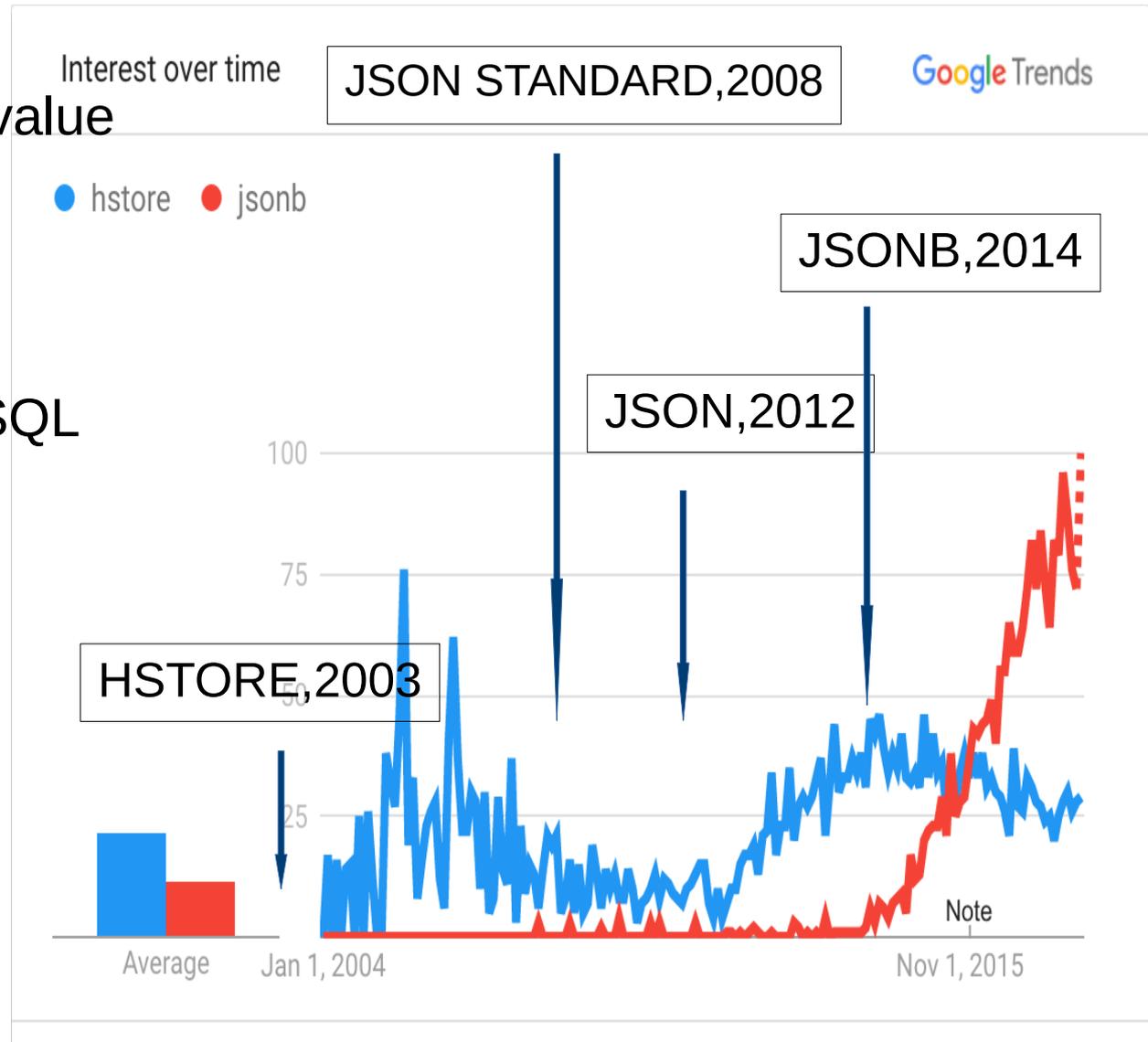
На самом деле **“SQL is intergalactic data-speak”**  
(некоторые NoSQL СУБД имеют поддержку SQL)

В основном используются для Web, для  
социальных сетей, OLAP. непригодно для OLTP с  
ACID

Незрелые, нет поддержки

# NoSQL PostgreSQL

- HSTORE — binary key-value storage, index support
- 2003 — initial release
- 2006 — part of PostgreSQL



# SQL/Foundation recognizes JSON after 8 years

4.46	<b>JSON</b> data handling in SQL. ....	174
4.46.1	Introduction. ....	174
4.46.2	Implied JSON data model. ....	175
4.46.3	SQL/JSON data model. ....	176
4.46.4	SQL/JSON functions. ....	177
4.46.5	Overview of SQL/JSON path language. ....	178
<b>5</b>	<b>Lexical elements. ....</b>	<b>181</b>
5.1	<SQL terminal character>. ....	181
5.2	<token> and <separator>. ....	185

PostgreSQL implementation ( 1 year of development)

- Uses native data types JSON, JSONB
- JSONPATH data type for SQL/JSON path language
- Nine functions SQL/JSON functions for constructing:
  - JSON\_OBJECT, JSON\_ARRAY,  
JSON\_OBJECTAGG, JSON\_ARRAYAGG
- and retrieving
  - JSON\_VALUE, JSON\_QUERY, JSON\_TABLE,  
IS [NOT] JSON, JSON\_EXISTS
- Extensions: more methods, JSONB op JSONPATH

# SQL/JSON in PostgreSQL

SQL-2016 path language specifies the parts (the projection) of JSON data to be retrieved by path engine for SQL/JSON functions.

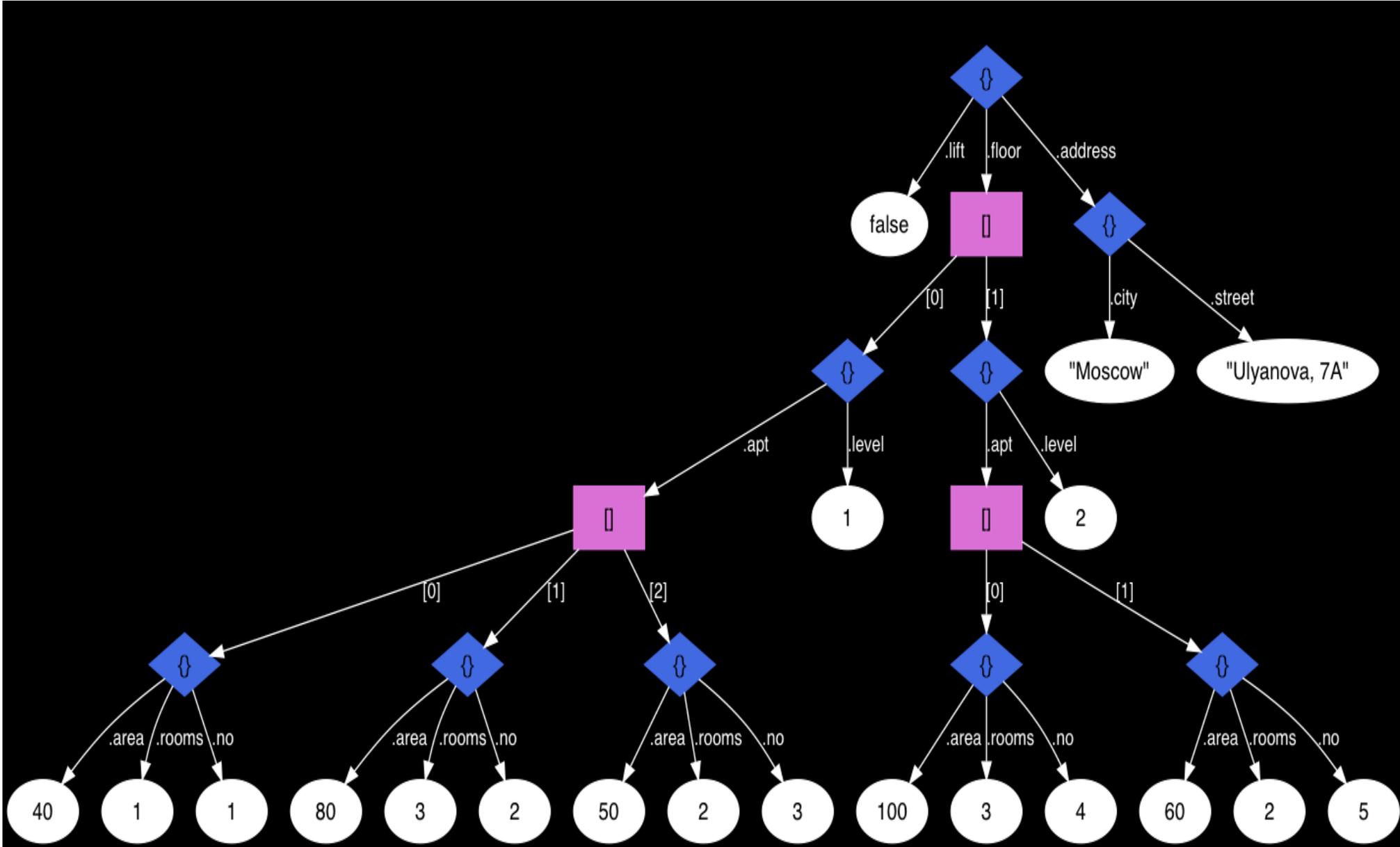
**Jsonpath** — the binary data type for SQL/JSON path expression to effective query JSON data.

```
SELECT JSON_QUERY(js,  
    '$.floor[*] ? (@.level >1).apt[*] ? (@.area>$min && @.area < $max).no'  
    PASSING 40 AS min, 90 AS max )  
FROM house;
```

# Visual guide on jsonpath

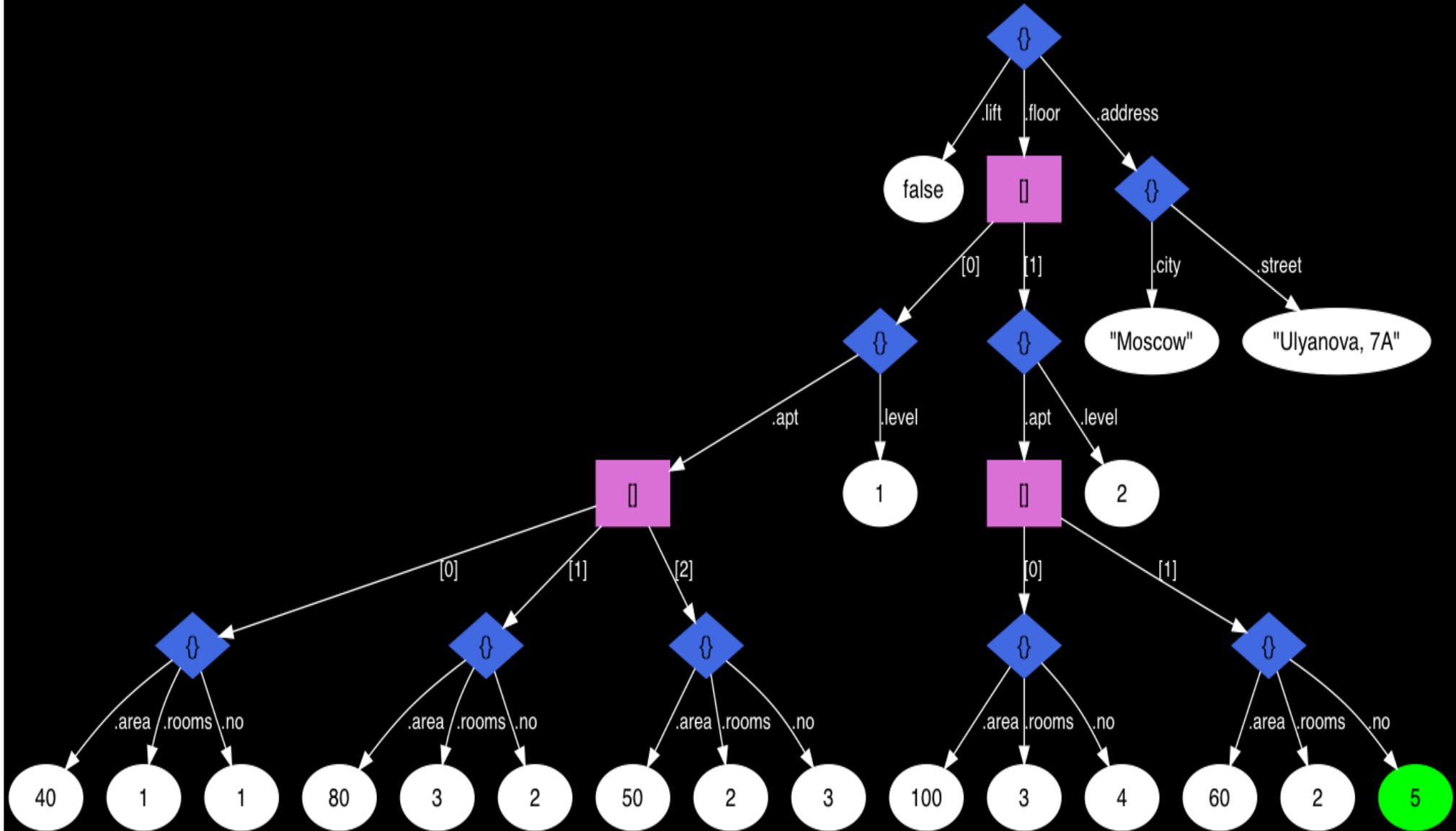
```
{
  "address": {
    "city": "Moscow",
    "street": "Ulyanova, 7A"
  },
  "lift": false,
  "floor": [
    {
      "level": 1,
      "apt": [
        {"no": 1, "area": 40, "rooms": 1},
        {"no": 2, "area": 80, "rooms": 3},
        {"no": 3, "area": 50, "rooms": 2}
      ]
    },
    {
      "level": 2,
      "apt": [
        {"no": 4, "area": 100, "rooms": 3},
        {"no": 5, "area": 60, "rooms": 2}
      ]
    }
  ]
}
```

# 2-floors house

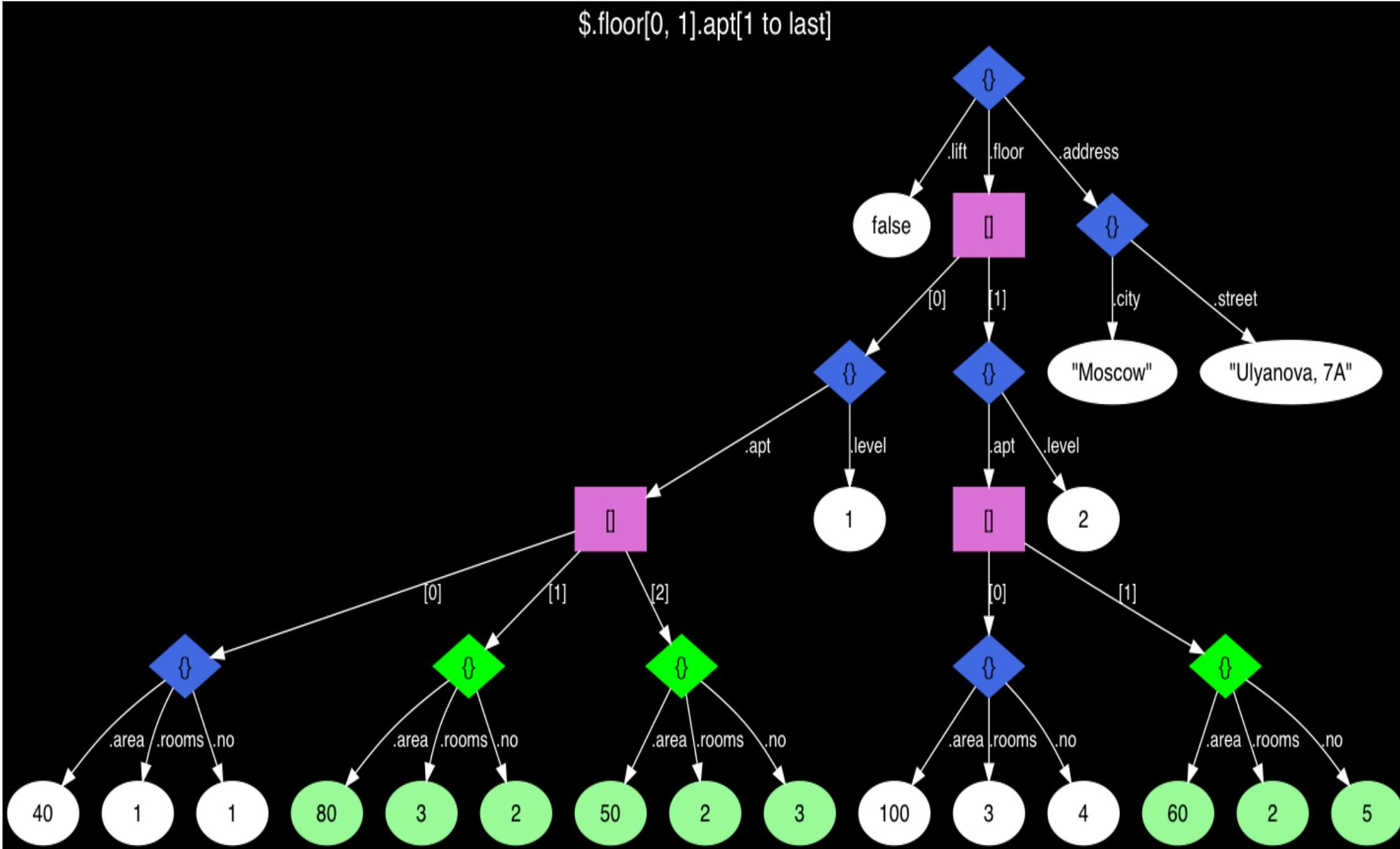


**`$.floor[*]?(@.level >1).apt[*]?  
 (@.area >40 && @.area < 90).no`**

`$.floor[*]?(@.level >1).apt[*]?(@.area > 40 && @.area < 90).no`



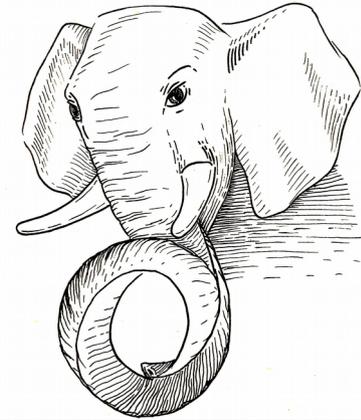
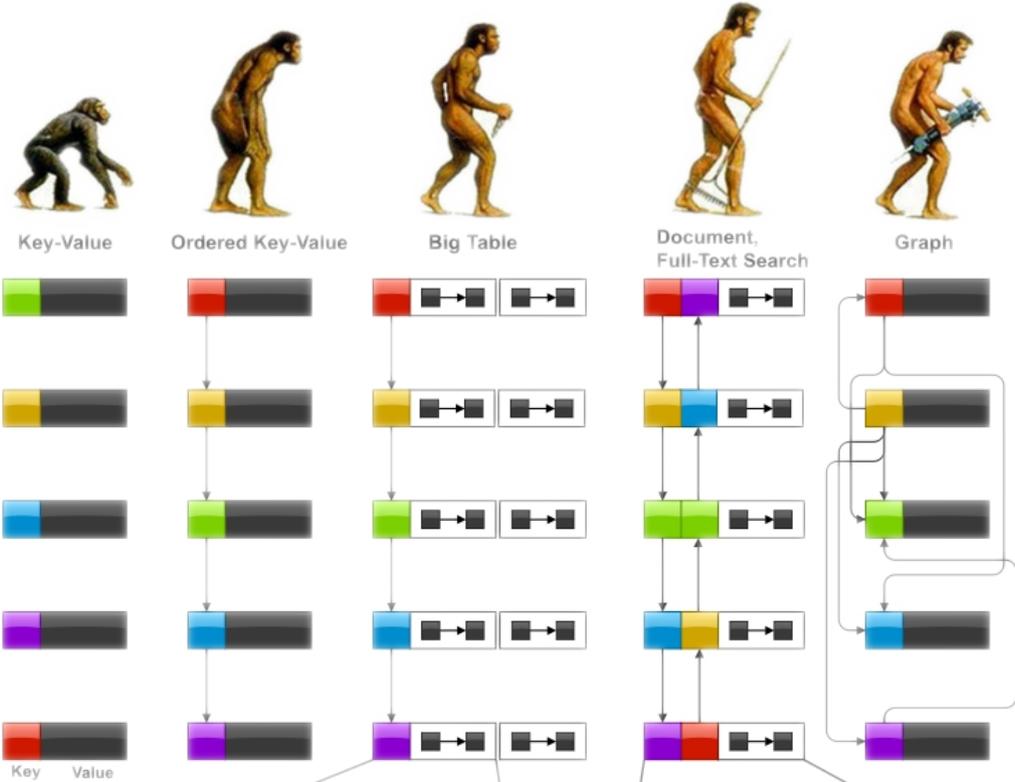
# \$.floor[0, 1].apt[1 to last]



# SQL/JSON availability

- Currently under review for PG 12
- Github Postgres Professional repository  
<https://github.com/postgrespro/sqljson>
- **WEB-interface to play with SQL/JSON**  
<http://sqlfiddle.postgrespro.ru/#!21/0/1819>
- **Technical Report (SQL/JSON)** - available for free  
[http://standards.iso.org/ittf/PubliclyAvailableStandards/c067367\\_ISO\\_IEC\\_TR\\_19075-6\\_2017.zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c067367_ISO_IEC_TR_19075-6_2017.zip)

# RDBMS >< NoSQL



SQL/JSON - 2019

- SQL-2016 standard
- PostgreSQL 12

JSONB - 2014

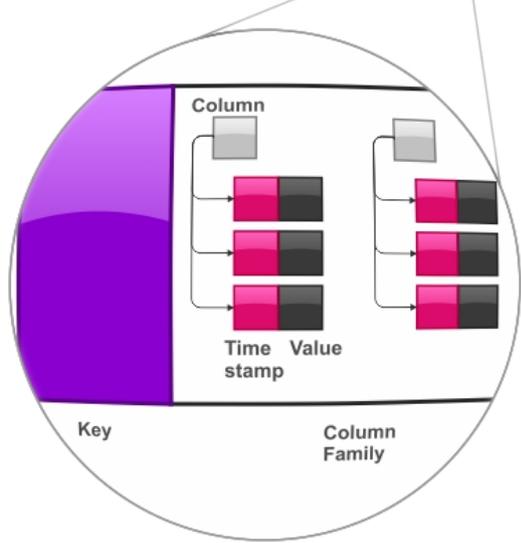
- Binary storage
- Nesting objects & arrays
- Indexing

JSON - 2012

- Textual storage
- JSON verification

HSTORE - 2003

- Perl-like hash storage
- No nesting
- Indexing

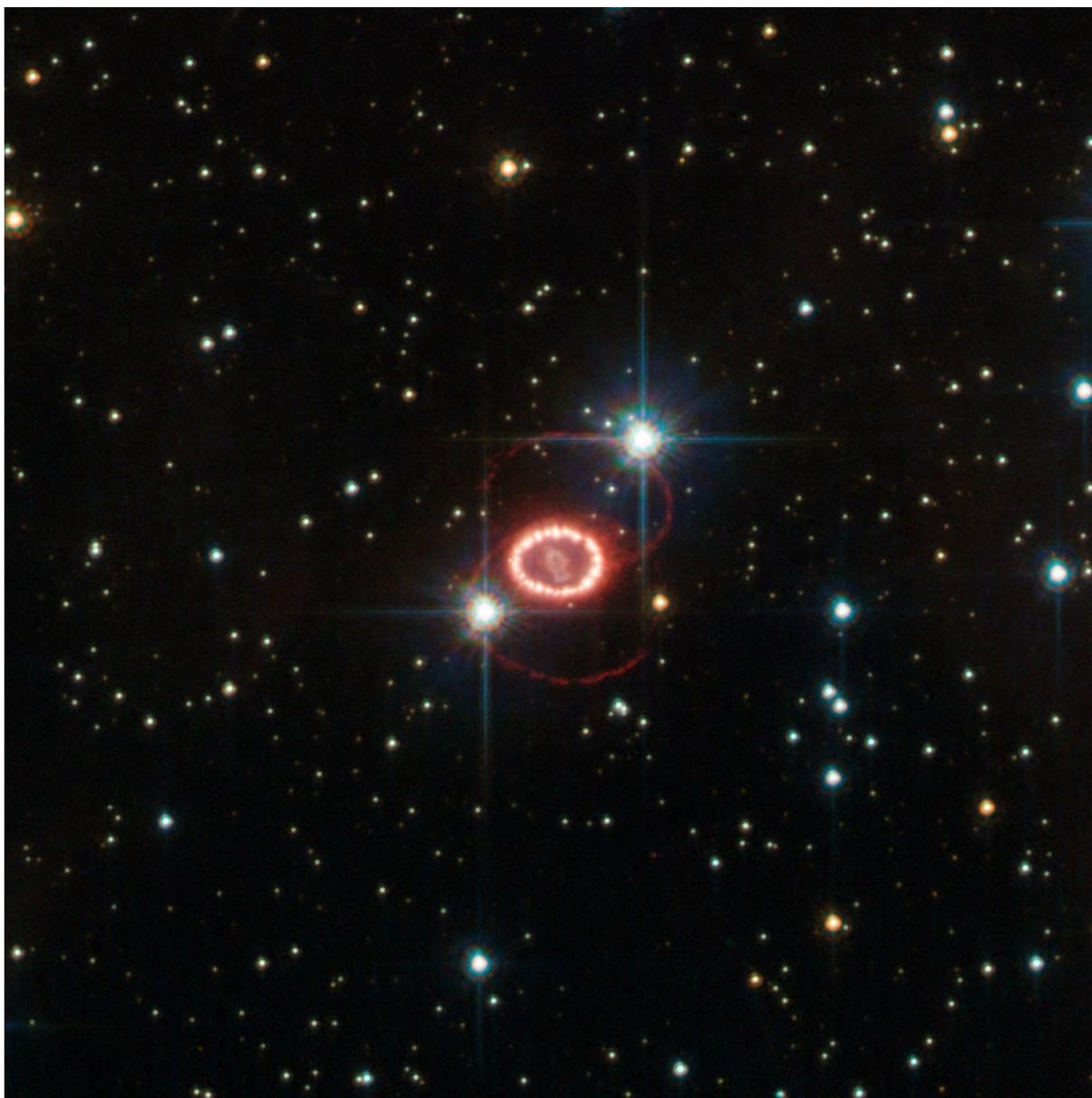


```

"employee" :
{
  "name" : "Mohana Pillai",
  "position" : "Delivery Manager",
  "projects" : [
    {
      "name" : "Easy Signatures"
    }
  ]
}, Semi-Structured Data
Plain Text
...
is a confidential word or number
combination used as a code to
identify when accessing
...
en 8 and 15 characters
number and may ne
spaces
  
```

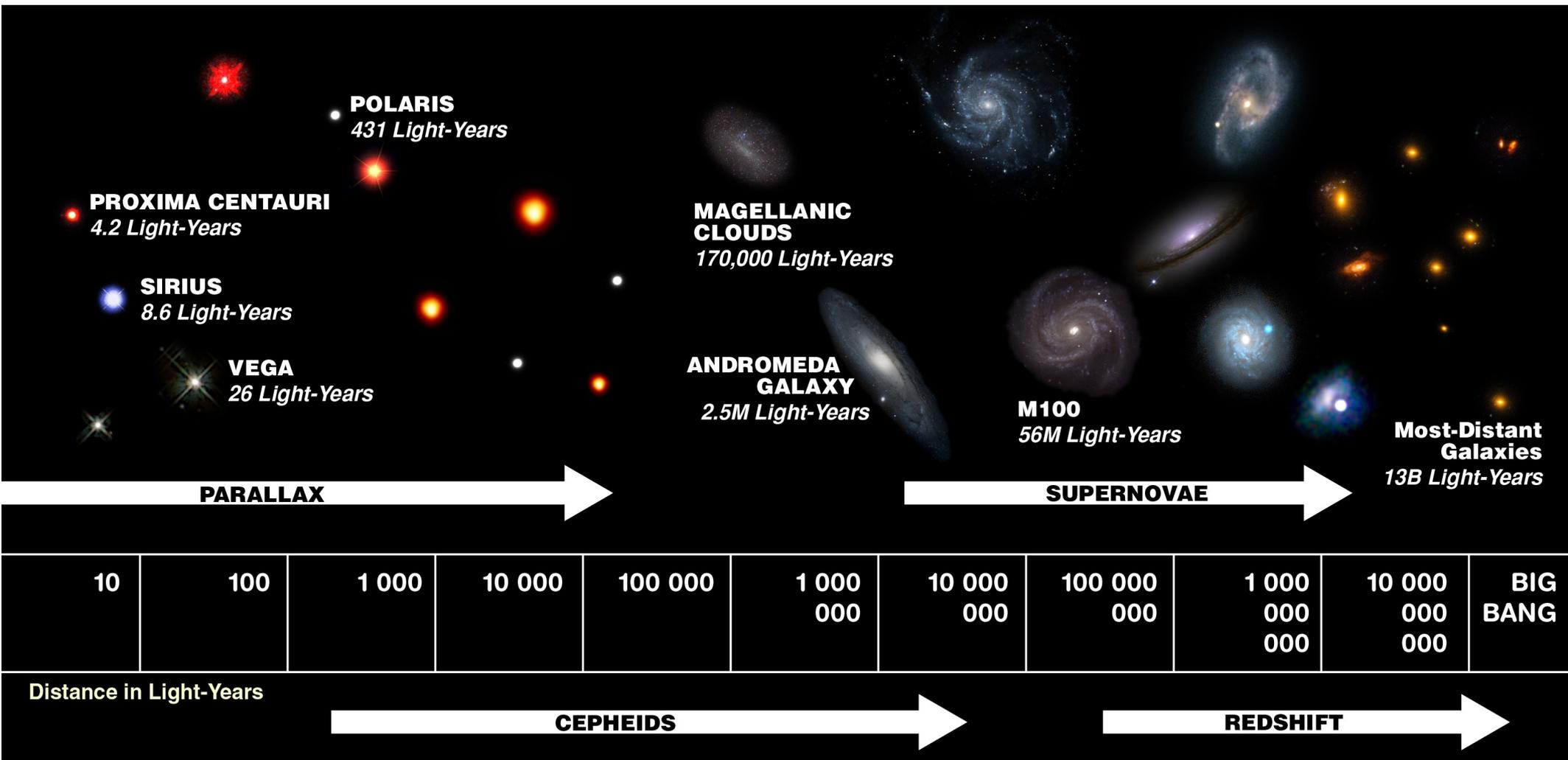
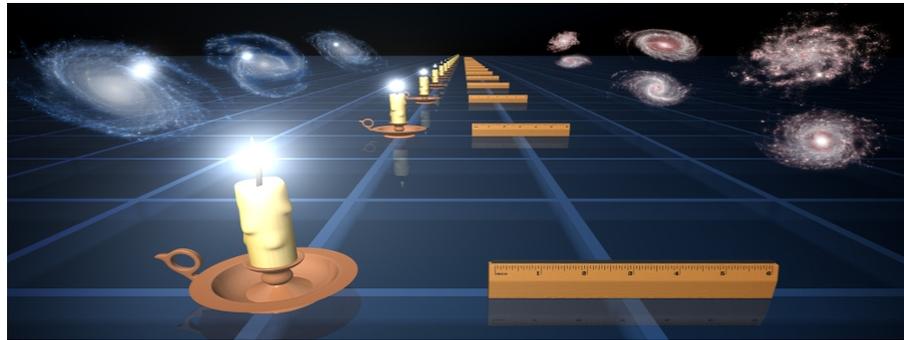
# PostgreSQL and Supernovae

SN 1987a, Type II, +2.9, Tarantula Nebulae in LMC,  
168 000 ly, progenitor: Sanduleak -69° 202,



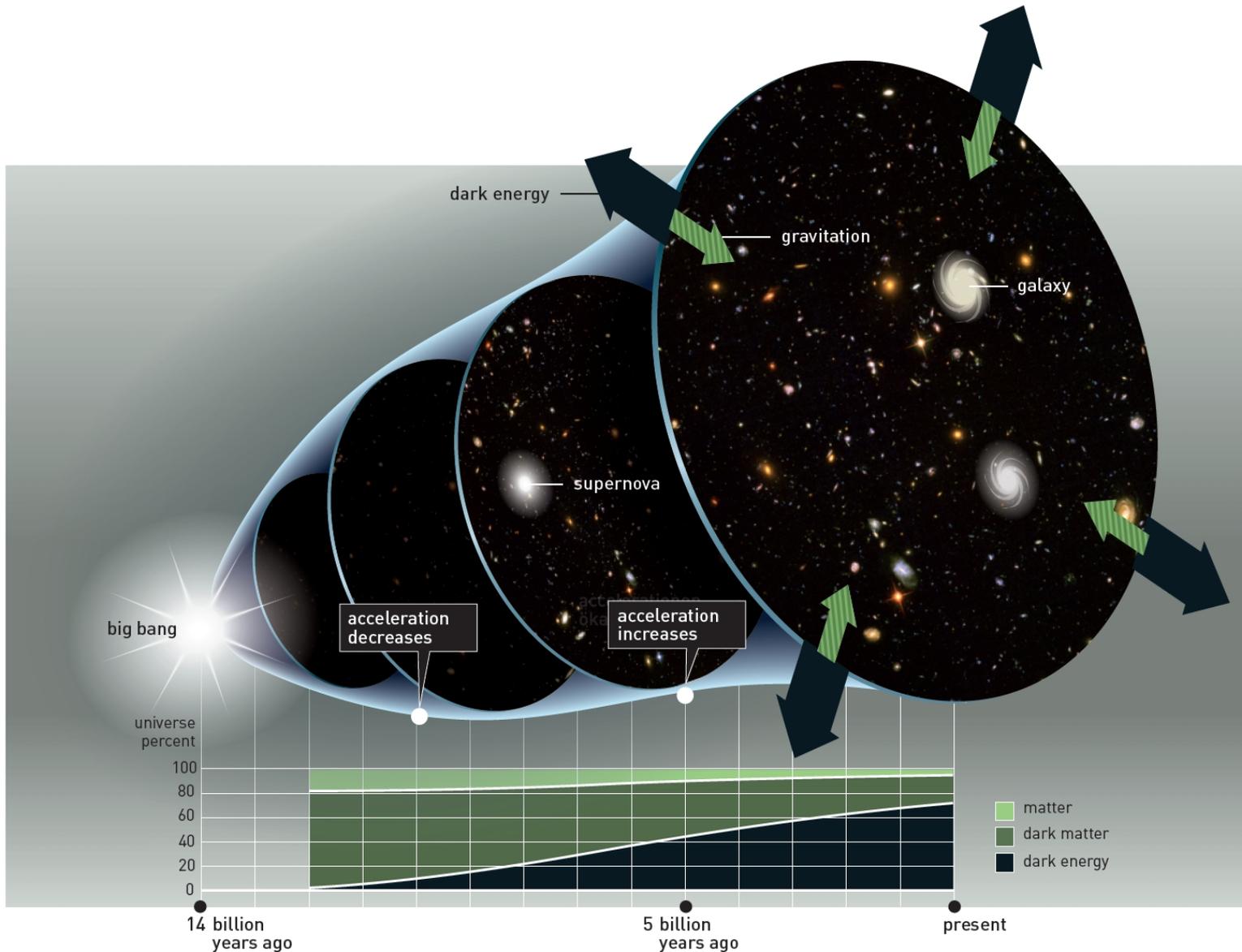
# The Scale of the Universe

- Supernovae(Ia) - «standard candles»
- Used to measure the distance to the host galaxy



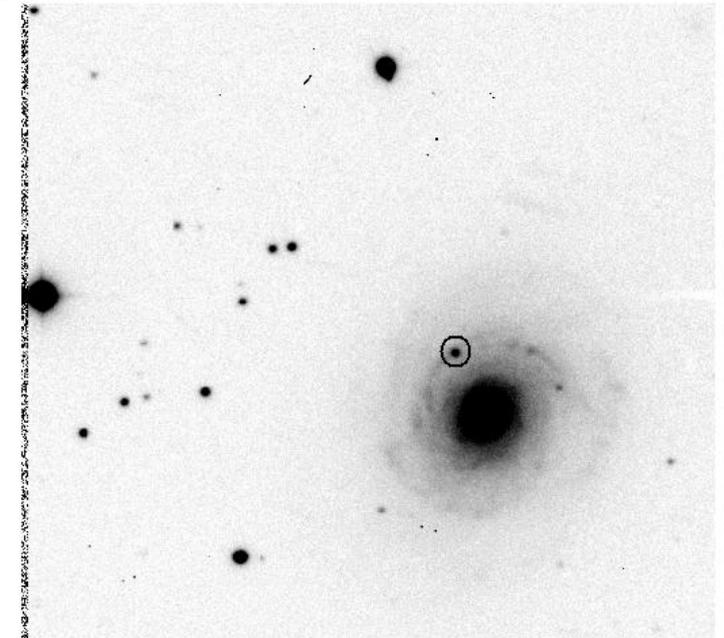
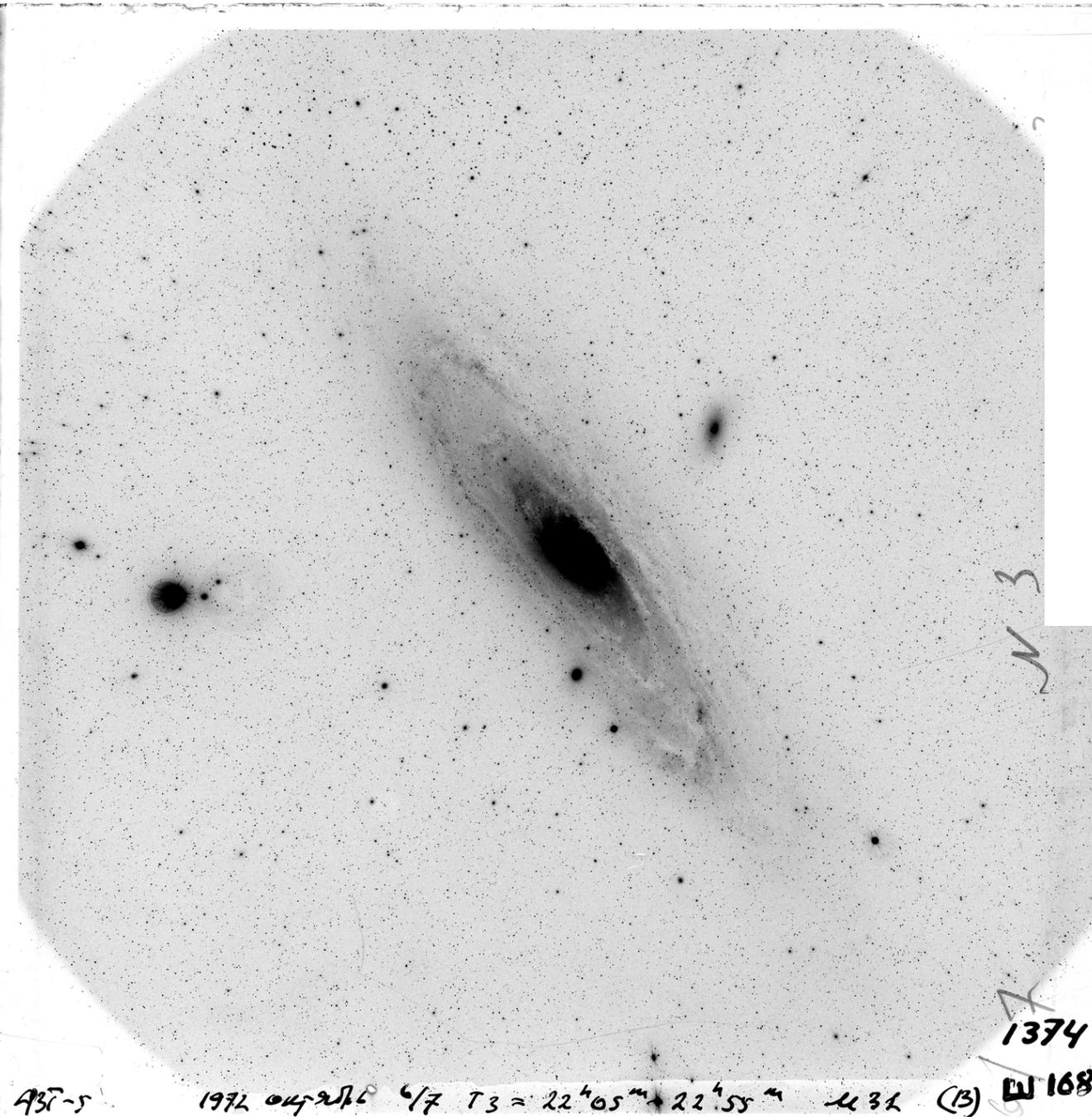
# What is the fate of the Universe ?

## WRITTEN IN THE STARS



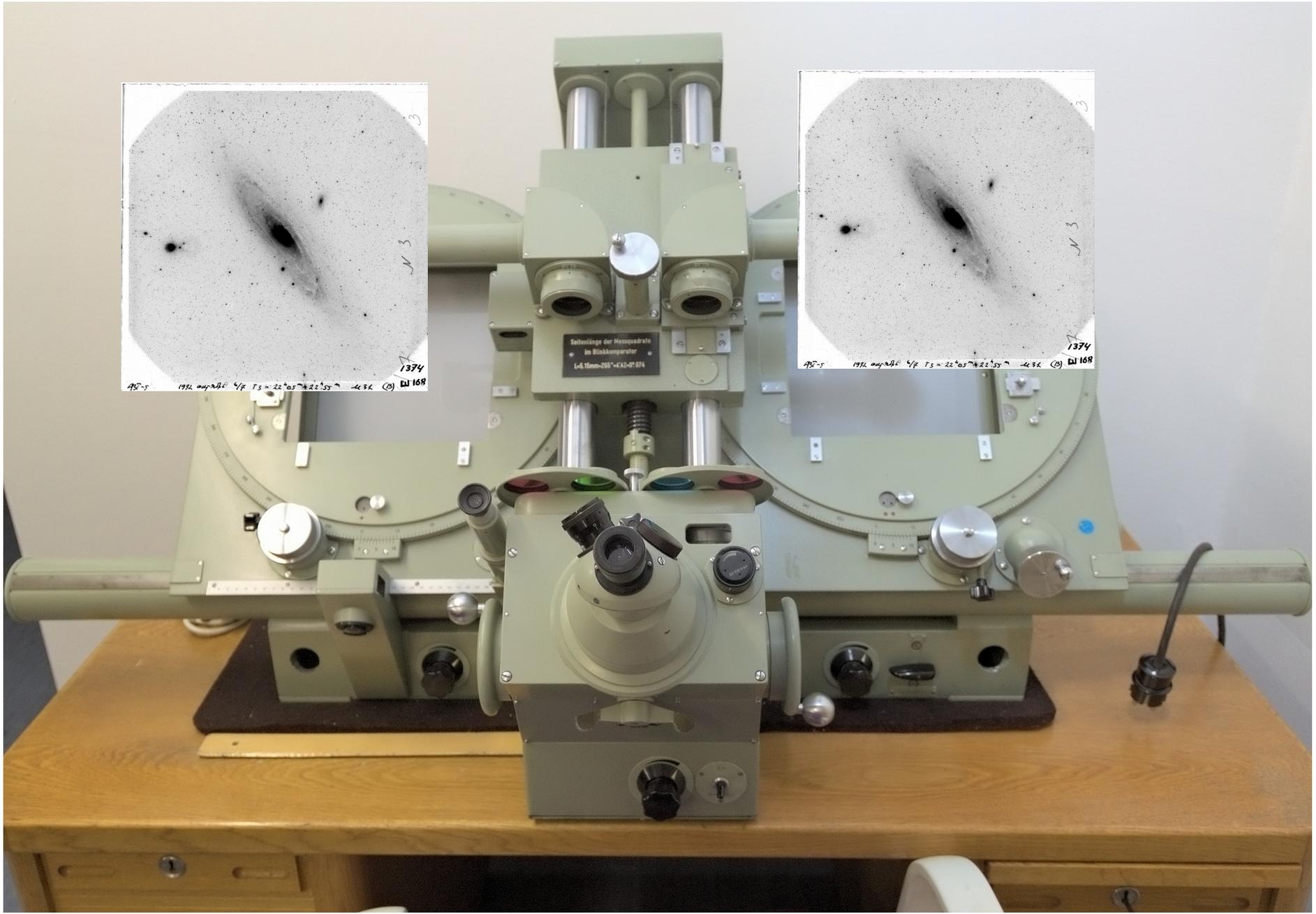
**Figure 1. The world is growing.** The expansion of the Universe began with the Big Bang 14 billion years ago, but slowed down during the first several billion years. Eventually it started to accelerate. The acceleration is believed to be driven by dark energy, which in the beginning constituted only a small part of the Universe. But as matter got diluted by the expansion, the dark energy became more dominant.

# M31 ( Andromeda), AZT-5 telescope

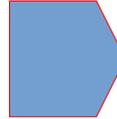
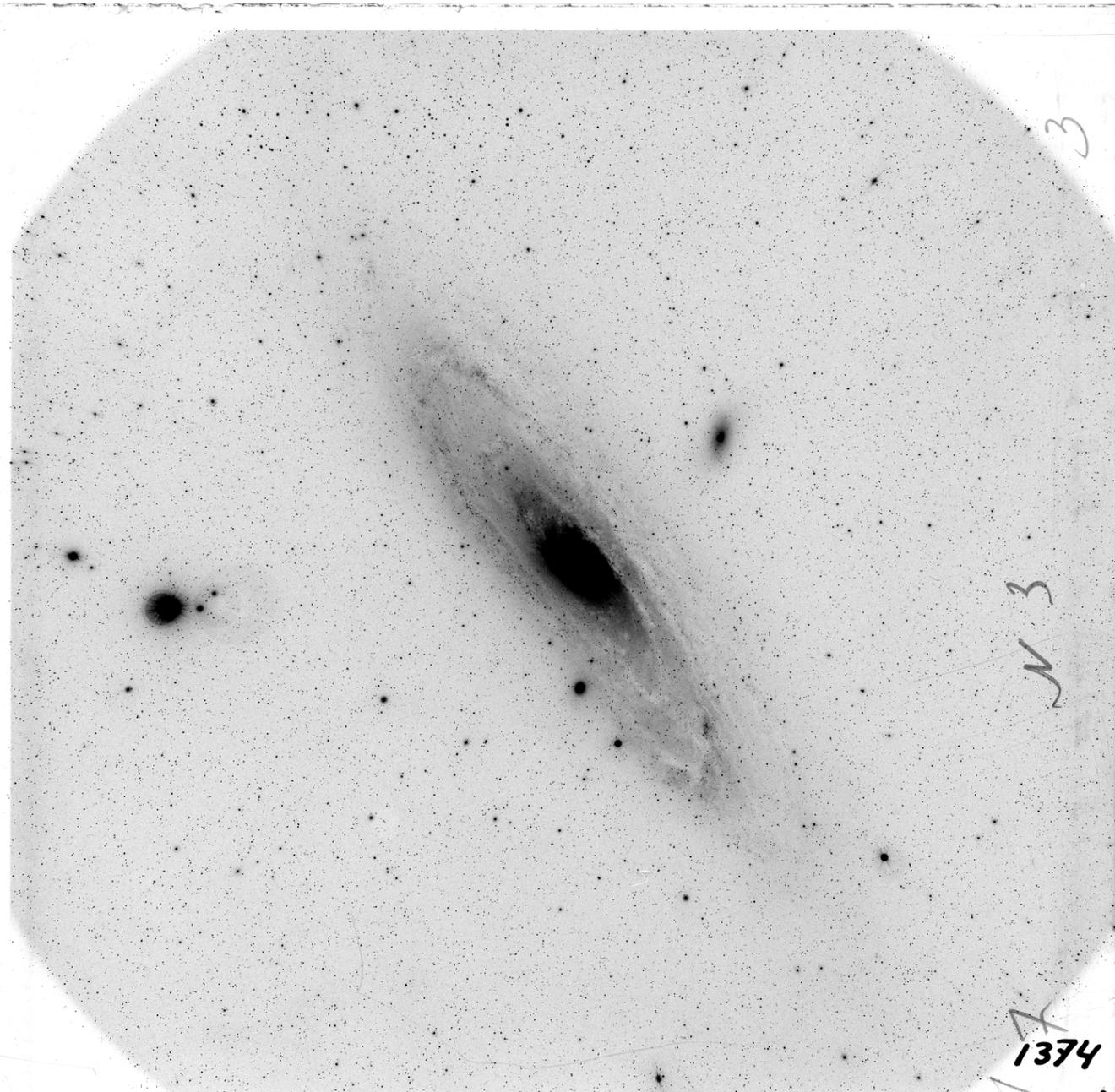


SN 2008fv in NGC 3147, Draco  
Dmitry Tsvetkov, SAI MSU

# Blink Comparator (manual discovery) Many hours of hard work !



# M31 ( Andromeda), AZT-5 telescope



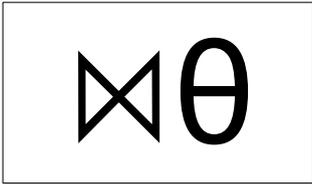
X-00001	1.1458447	-89.9186147
X-00002	1.3300139	-89.9332336
X-00003	3.2556022	-89.9641031
X-00004	3.6464625	-89.9060142
X-00005	6.3110253	-89.9523947
X-00006	6.6275517	-89.9279197
X-00007	7.8266025	-89.9129272
X-00008	9.0694378	-89.9714031
X-00009	9.6627953	-89.9244314
X-00010	10.0494292	-89.9705058
X-00011	10.4863922	-89.9699058
X-00012	11.0953692	-89.9016031
X-00013	11.3240233	-89.9344336
X-00014	11.7906064	-89.9070308
X-00015	12.0416581	-89.9300586
X-00016	12.0522308	-89.9002281
X-00017	12.2808536	-89.9107669
X-00018	13.0316142	-89.9214558
X-00019	13.8727033	-89.9577031
X-00020	14.6546639	-89.9191919
X-00021	18.3035981	-89.9447475
X-00022	18.5185631	-89.9446836
X-00023	19.8675597	-89.9836308
X-00024	20.9699533	-89.9226864
X-00025	21.6777744	-89.9256808
X-00026	23.3660669	-89.9036558
X-00027	24.2841308	-89.9516475
X-00028	24.3273161	-89.9202392
X-00029	24.5540458	-89.9246003
X-00030	24.5655172	-89.9122336
X-00031	26.3487519	-89.9460336
X-00032	26.5268008	-89.9311503
X-00033	26.6070808	-89.9271808
X-00034	27.4104919	-89.9768558
X-00035	27.8290442	-89.9304622
X-00036	28.5552036	-89.9199117
X-00037	29.4407347	-89.9762836
X-00038	30.5729608	-89.9377753
X-00039	30.7101131	-89.9105642
X-00040	33.2918250	-89.9106614
X-00041	33.4843678	-89.9442058

AZT-5 1972 04 27 6/7 T3 = 22<sup>h</sup> 05<sup>m</sup> - 22<sup>h</sup> 55<sup>m</sup> M31 (B) 168

# Spatial Join (Machine Discovery, < 1s)

Observations:  $10^5$

```
x-00001|1.1458447|-89.9186147
x-00002|1.3300139|-89.9332336
x-00003|3.2556022|-89.9641031
x-00004|3.6464625|-89.9060142
x-00005|6.3110253|-89.9523947
x-00006|6.6275517|-89.9279197
x-00007|7.8266025|-89.9129272
x-00008|9.0694378|-89.9714031
x-00009|9.6627953|-89.9244314
x-00010|10.0494292|-89.9705058
x-00011|10.4863922|-89.9699058
x-00012|11.0953692|-89.9016031
x-00013|11.3240233|-89.9344336
x-00014|11.7906064|-89.9070308
x-00015|12.0416581|-89.9300586
x-00016|12.0522308|-89.9002281
x-00017|12.2808536|-89.9107669
x-00018|13.0316142|-89.9214558
x-00019|13.8727033|-89.9577031
x-00020|14.6546639|-89.9191919
x-00021|18.3035981|-89.9447475
x-00022|18.5185631|-89.9446836
x-00023|19.8675597|-89.9836308
x-00024|20.9699533|-89.9226864
x-00025|21.6777744|-89.9256808
x-00026|23.3660669|-89.9036558
x-00027|24.2841308|-89.9516475
x-00028|24.3273161|-89.9202392
x-00029|24.5540458|-89.9246003
x-00030|24.5655172|-89.9122336
x-00031|26.3487519|-89.9460336
x-00032|26.5268008|-89.9311503
x-00033|26.6070808|-89.9271808
x-00034|27.4104919|-89.9768558
x-00035|27.8290442|-89.9304622
x-00036|28.5552036|-89.9199117
x-00037|29.4407347|-89.9762836
x-00038|30.5729608|-89.9377753
```

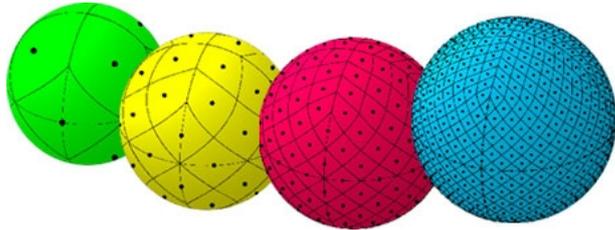


Catalog(s):  $10^9$

```
t-0000001|1.1458447|-89.9186147|0.015|0.028
t-0000002|1.3300139|-89.9332336|0.050|0.110
t-0000003|3.2556022|-89.9641031|0.050|0.050
t-0000004|3.6464625|-89.9060142|0.204|0.224
t-0000005|6.3110253|-89.9523947|0.114|0.050
t-0000006|6.6275517|-89.9279197|0.098|0.150
t-0000007|7.8266025|-89.9129272|0.025|0.021
t-0000008|9.0694378|-89.9714031|0.200|0.200
t-0000009|9.6627953|-89.9244314|0.000|0.000
t-0000010|10.0494292|-89.9705058|0.050|0.228
t-0000011|10.4863922|-89.9699058|0.200|0.200
t-0000012|11.0953692|-89.9016031|0.050|0.259
t-0000013|11.3240233|-89.9344336|0.050|0.050
t-0000014|11.7906064|-89.9070308|0.159|0.131
t-0000015|12.0416581|-89.9300586|0.216|0.050
t-0000016|12.0522308|-89.9002281|0.050|0.050
t-0000017|12.2808536|-89.9107669|0.050|0.050
t-0000018|13.0316142|-89.9214558|0.152|0.120
t-0000019|13.8727033|-89.9577031|0.050|0.121
t-0000020|14.6546639|-89.9191919|0.050|0.069
t-0000021|18.3035981|-89.9447475|0.139|0.440
t-0000022|18.5185631|-89.9446836|0.057|0.268
t-0000023|19.8675597|-89.9836308|0.050|0.120
t-0000024|20.9699533|-89.9226864|0.050|0.050
t-0000025|21.6777744|-89.9256808|0.055|0.105
t-0000026|23.3660669|-89.9036558|0.050|0.135
t-0000027|24.2841308|-89.9516475|0.213|0.050
t-0000028|24.3273161|-89.9202392|0.550|0.999
t-0000029|24.5540458|-89.9246003|0.160|0.086
t-0000030|24.5655172|-89.9122336|0.205|0.050
t-0000031|26.3487519|-89.9460336|0.050|0.095
t-0000032|26.5268008|-89.9311503|0.335|0.245
t-0000033|26.6070808|-89.9271808|0.050|0.075
t-0000034|27.4104919|-89.9768558|0.094|0.090
t-0000035|27.8290442|-89.9304622|0.017|0.019
t-0000036|28.5552036|-89.9199117|0.050|0.115
t-0000037|29.4407347|-89.9762836|0.635|0.265
t-0000038|30.5729608|-89.9377753|0.314|0.170
```

## Indexing the SKY with PostgreSQL

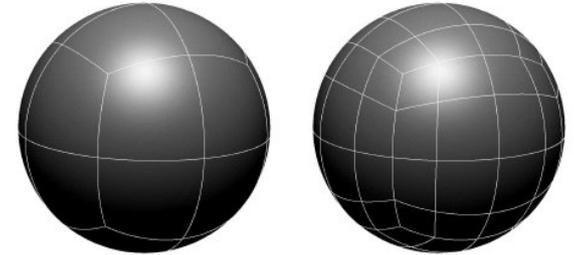
HEALPIX



HTM

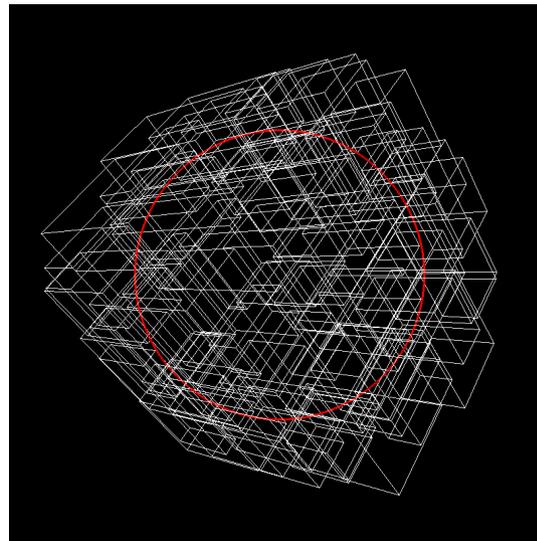


Q3C (PG)



The sphere segmentation in Q3C

PGSphere (PG)



Database of astronomic catalogues in  
Cambridge University

~5 dbs, ~ 40 users, up to ~  $10^7$  queries per  
day, size 40Tb

pg 9.4 + q3c + hstore

Example of research: Kuposov, S. E., Belokurov, V.,  
Torrealba, G., & Evans, N. W. (2015). Beasts of the Southern  
Wild: Discovery of nine Ultra Faint satellites in the vicinity of  
the Magellanic Clouds. *The Astrophysical Journal*, 805(2),  
130.

Real time Detection of alerts in the Gaia

~10 dbs, 10 users, up to ~  $10^6$  queries per day, size 30Tb

pg 9.3 + synchronous replication + q3c

Example of research: Campbell, H. C., Marsh, T. R., Fraser, M., Hodgkin, S. T., de Miguel, E., Gänsicke, B. T., ... & Kozlov, S. E. (2015). Total eclipse of the heart: the AM CVn Gaia14aae/ASSASN-14cn. Monthly Notices of the Royal Astronomical Society, 452(1), 1060-1067.

Robotic network telescopes by SAI MSU

8 observatories (5 in Russia, 3 outside)

total size ~100TB

pg 9.0-9.4 + pgsphere + replication

See:

- Lipunov, Vladimir, et al. "Master robotic net." *Advances in Astronomy* 2010 (2010).
- Kornilov, Victor G., et al. "Robotic optical telescopes global network MASTER II. Equipment, structure, algorithms." *Experimental Astronomy* 33.1 (2012): 173-196.

**BIG DATA OLTP**, аналитика реального времени  
вместо ETL.

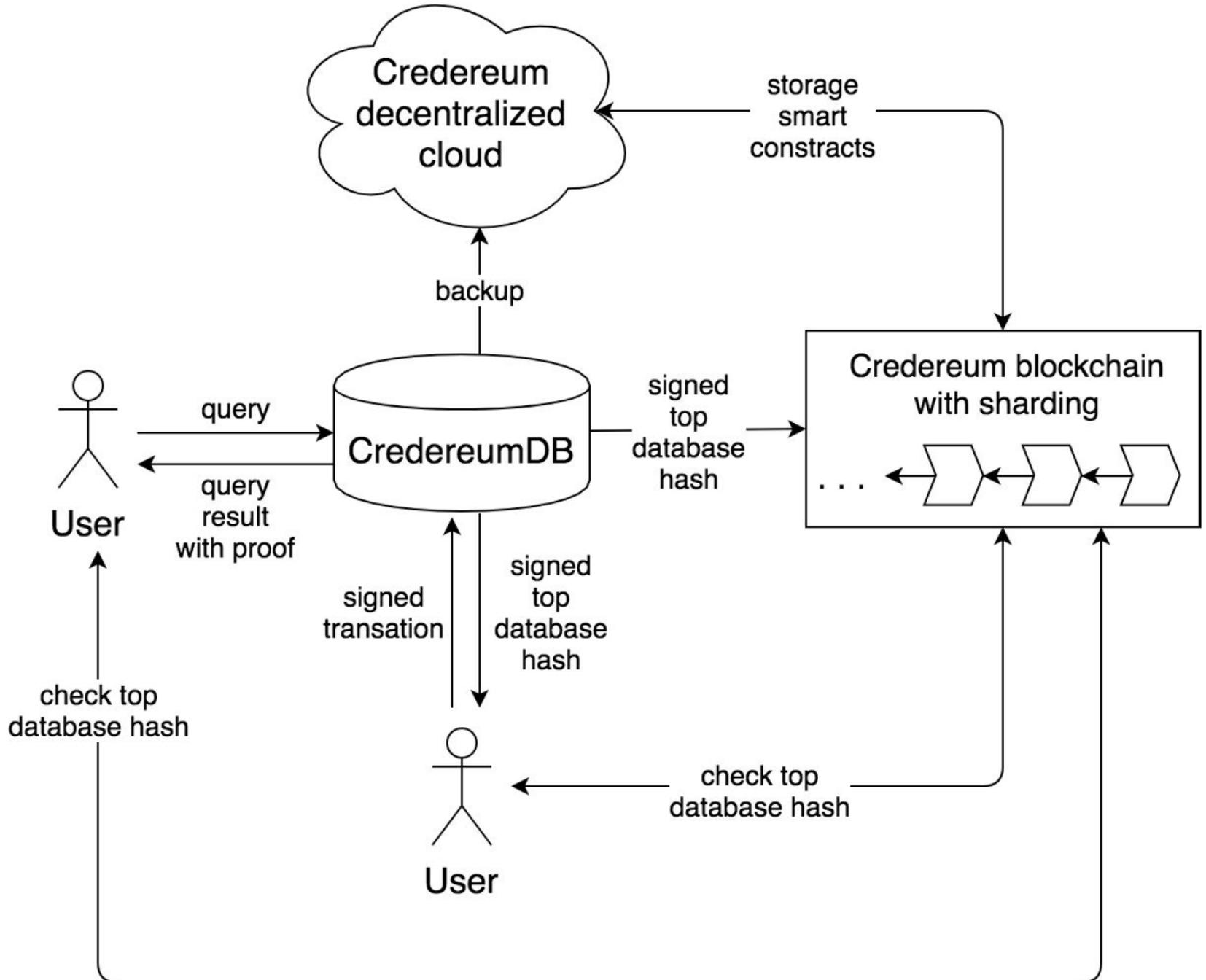
полный **ACID**, производительность и  
масштабируемость NoSQL. Три вида:

- Новая архитектура:
  - Общего назначения — распределенные «с нуля»  
Google Spanner, CockroachDB, Apache Ignite, VoldDB...
- Сильно-оптимизированные хранилища для SQL — ScaleDB, TokuDB, MemSQL, Maria DB Columnstore
- «Прозрачный» шардинг — middleware для автоматического распределения данных — dbShards, ScaleArc, ScaleBase

# How to know your database is not compromised ?

**Credereum** uses digital signature to sign transactions and Blockchain to store signatures to solve this problem.

# Credereum platform (Postgres Pro)



SQL — основное средство общения приложений

ACID транзакции

Неблокирующий механизм конкурентности, так что чтение не конфликтует с записью

Производительность узла выше традиционных СУБД

Масштабируемость, shared-nothing архитектура, большое количество узлов

# Передний край и ближайшее будущее

- СУБД в виртуальных машинах (Amazon RDS)
- Горячая тема:
  - Облачные СУБД (Amazon Aurora, AliCloud PolarDB)
- Горячая тема:
  - Автономные, адаптивные СУБД
- World Wide Cloud — World Wide Database ?
- DBA не нужны ?



**СПАСИБО ЗА ВНИМАНИЕ !**