

УДК: 004.65:004.43
JEL Classification: O 31, O32
doi: 10.31767/nasoa.3-4-2021.11

О. Л. ЄРШОВА,

кандидат економічних наук, доцент,
завідувач кафедри економіко-математичних
дисциплін та інформаційних технологій;
e-mail: midnight2003@nasoa.edu.ua,
ORCID: <https://orcid.org/0000-0002-3801-9730>;

О. В. СТАВИЦЬКИЙ,

кандидат економічних наук, доцент,
доцент кафедри економіко-математичних
дисциплін та інформаційних технологій;
Національна академія статистики, обліку та аудиту;
e-mail: OVStavitskiy@nasoa.edu.ua,
ORCID: <https://orcid.org/0000-0002-2114-0892>;

Дослідження сучасних тенденцій у технологіях баз і сховищ даних як технологічної та архітектурної основи для створення програмних і інтелектуальних систем засобами сучасних мов програмування.

Частина 1

Стаття містить аналітичний огляд тенденцій розвитку технології баз даних, зроблений на базі звітів, підготовлених за результатами восьми зустрічей фахівців у галузі баз даних, які відбулись у період 1988–2013 років. Об'єктами аналізу є найцікавіші прогнози, що містяться у звітах, а саме їх реалістичність, точність, прагматизм або, навпаки, утопічність чи кон'юнктурність.

Стаття складається з двох частин.

В першій частині розглядаються і оцінюються прогнози, висунуті у звітах про перші чотири зустрічі, що відбулися в 1988, 1990, 1995, 1996 роках. Ці прогнози стосуються створення, розвитку і застосування систем підтримки прийняття рішень, програмно-апаратних комплексів для управління базами даних, графічних процесорів, операційних систем, інтерфейсу мови структурованих запитів, додатків баз даних, розподілу інформації, універсальних систем управління базами даних, критеріїв оптимізації запитів, інтелектуальний аналіз даних усередині системи управління базами даних. Докладно розглянуто тематику досліджень у галузі баз даних, яка набула пріоритетності в той період: облік і розрахунки даних, безпека та конфіденційність даних, реплікація та узгодження даних, забезпечення структурованості даних, інтелектуальний аналіз даних, сховища даних.

Друга частина присвячена огляду прогнозів, що містяться у звітах про зустрічі, які відбулися у 1998, 2003, 2008 та 2013 роках. Прогнози стосуються самоналаштування систем баз даних, переосмислення традиційної архітектури систем баз даних завдяки новим апаратним можливостям. Окрема увага в них приділена можливості роботи зі структурованими та неструктурованими даними всередині архітектури системи управління базами даних, підтримці технології великих даних, окреслено проблематику досліджень для реалізації її потенціалу.

Ключові слова: бази даних, система управління базами даних, сховища даних, система підтримки прийняття рішень, аналітичне оброблення даних, мова структурованих запитів, інтелектуальний аналіз даних, транзакційні запити, аналітичні запити.

O. YERSHOVA,

*PhD (Economics), Associate Professor,
Head of Department for Econometric Disciplines
and Information Technologies;*

O. STAVYTSKYI,

*PhD (Economics), Associate Professor,
Associate Professor of Department
for Econometric Disciplines
and Information Technologies*

National Academy of Statistics, Accounting and Audit

A Study of Modern Trends in Database and Data Repository Technologies as the Technological and Architectural Basis for the Creation of Software and Intelligent Systems by Means of Modern Programming Languages.

Part 1

The article contains an analytical review of developments in database technologies, made on the basis of reports prepared by the results of eight meetings of database specialists held throughout 1988–2013. Objects of the analysis are most interesting predictions given in the reports: their realism, accuracy, pragmatism or, vice versa, utopianism or opportunism.

The article consists of two parts.

Part 1 is devoted to analysis and evaluation of predictions made in the reports of the four earlier meetings held in 1988, 1990, 1995, and 1996. These predictions are about creation, development and uses of decision support systems, database appliances, graphic processing units, operating systems, interface for structured query language, database applications, information distribution, universal database management systems, query optimization criteria, intellectual analysis of database within database management systems. A detailed description of research themes in the field of databases, which got the priority status in that time, is given: recording and computation of data, security and confidentiality of data, replication and harmonization of data, structuring of data, intellectual analysis of data, data warehouses.

Part 2 is devoted to an analytical review of the predictions contained in the reports on the meetings held in 1998, 2003, 2008, and 2013. The predictions are about self-adjustment of database systems, rethinking of the traditional database architecture as a result of new hardware capabilities. They make special emphasis on the feasibility of manipulations with structured and unstructured data within DSS architecture, support of Big Data technology, with outlining the themes of research aimed at implementation of its potential.

Keywords: *databases, database management system, data warehouses, decision support systems, analytical data processing, structural query language, data mining, transactional queries, analytic queries.*

Вступ. Раз на кілька років провідні дослідники в галузі баз даних проводять зустрічі для обговорення та оцінювання стану справ у галузі та формулювання тем досліджень, які, на думку учасників, будуть найбільш актуальними у найближчі роки. За результатами зустрічей зазвичай готуються і публікуються звіти, які мають надзвичайно високий авторитет серед дослідників галузі, а також серйозний вплив на розвиток досліджень і розробок, у тому числі при створенні програмних та інтелектуальних систем.

Станом на сьогодні відбулося вісім таких зустрічей:

- ✓ Мабутні напрями досліджень систем управління базами даних (Future Directions in DBMS Research), 1988 р., Лагуна Біч, штат Каліфорнія, США [1];
- ✓ Вступний семінар з питань перспектив досліджень систем баз даних, організований Національним науковим фондом США (NSF Invitational Workshop on the Future of Database Systems Research), 1990 р., Пало Альто, штат Каліфорнія, США [2];
- ✓ Семінар з питань перспектив досліджень систем баз даних, організований Національним науковим фондом США (NSF Workshop on the Future of Database Systems Research), 1995 р., Пало Альто, штат Каліфорнія, США [3];

- ✓ Семінар зі стратегічних напрямів комп'ютерних досліджень (Workshop on Strategic Directions in Computing Research), 1996 р., Кембрідж, штат Массачусетс, США [4];
- ✓ Зустріч у Асіломарі (Asilomar Meeting), 1998 р., неподалік від м. Монтерей, штат Каліфорнія, США [5];
- ✓ Зустріч у м. Лоуелл (Lowell Meeting), 2003 р., штат Массачусетс, США [6];
- ✓ Зустріч у отелі "Claremont Resort" (Claremont Meeting), 2008 р., м. Берклі, штат Каліфорнія, США [7];
- ✓ Зустріч у Бекманському університетському центрі (Beckman Meeting), 2013 р., м. Ірвін, штат Каліфорнія, США [8].

Після кожної зустрічі публікувався звіт, що містить огляд сучасного стану галузі та програму досліджень на найближче майбутнє – своєрідний набір прогнозів розвитку дослідницької діяльності у галузі баз даних. Стаття присвячена аналізу найцікавіших прогнозів зі звітів про зустрічі дослідників з інтервалом у 25 років і обговоренню обґрунтованості наведених у них даних. Аналіз включає різноманітні питання, пов'язані з технологією баз даних: роль спеціалізованої апаратури при побудові ефективних систем управління базами даних (СУБД); мова структурованих запитів та додатки баз даних; перспективи об'єктно-реляційних розширень; розподілені неоднорідні системи баз даних; бази даних та Web; бази та сховища даних, методи аналітичного оброблення і інтелектуального аналізу даних; компонентна організація СУБД; критерії оптимізації запитів; самоналаштуваність та самоврядність СУБД; архітектура СУБД та нові апаратні можливості: SSD, енергонезалежна пам'ять, масивно-багатопотокові процесори; спеціалізовані СУБД; простору даних; проблема великих даних і реакція на неї з боку фахівців у галузі баз даних; зміни у архітектурі комп'ютерних систем.

Мета статті – надати читачам реальну картину розвитку технології баз даних протягом 1988–2013 років, яка допоможе зрозуміти потенціал і можливості їх застосування у прикладних задачах створення програмних та інтелектуальних систем.

У статті обговорюються найцікавіші, з погляду авторів, прогнози, що містяться у звітах про зустрічі минулих років [1–7]: їхня точність, реалістичність, прагматичність чи кон'юнктурність. Автори майже не торкаються звіту [8] про останню зустріч (2013 р.), оцінювати точність прогнозів якої ще не настав час.

Авторами поставлено трудові завдання – проаналізувати основні тенденції у розвитку технологій баз і сховищ даних. Воно виконуватиметься шляхом порівняння прогнозів різних років та фактичного стану через 25 років. Такий аналіз становить інтерес для адміністраторів і розробників додатків баз даних, оскільки експорт інформаційно-телекомунікаційних (ІТ) послуг в Україні неухильно зростає, перетворюючи їх на галузь національної економіки, що стабільно розвивається. Українські розробники ІТ засобів затребувані на світовому ринку праці та водночас є активними споживачами всіх світових досягнень у ІТ секторі, включаючи технології баз і сховищ даних. Українська ІТ галузь не пропонує власних рішень і продуктів, програмісти використовують актуальні у світі технології, підходи та продукти провідних розробників СУБД. Тому наданий у статті аналіз є цікавим і має практичну користь.

Результати дослідження

1. Зустріч у Лагуна Біч 1988 року. На зустрічі прогнозувалося широке впровадження систем підтримки прийняття рішень (decision support systems, DSS) водночас зі зниженням цін на апаратуру. Підхід до побудови DSS виник у 70-х рр. минулого століття на основі робіт, що виконувались ще у 1960-х рр. [9]. Термін "система підтримки прийняття рішень" ніколи не мав однозначного та чіткого визначення. Проте вже наприкінці 80-х років фахівцям у галузі баз даних було зрозуміло, що в багатьох випадках прийняття рішень у різних організаціях має ґрунтуватися на аналізі даних.

У 90-х роках ХХ століття було створено теоретичні основи і технологію сховищ даних (datawarehouse) та аналітичного оброблення даних (online analytical data processing, OLAP). У першому десятилітті ХХІ століття з'явилися горизонтально масштабовані масивно-паралельні аналітичні системи СУБД, виникла технологія map/reduce [10]. Відтоді набули активного розвитку методи інтелектуального аналізу даних (data mining, text mining). До аналізу активно залучаються дані, що генеруються різними користувачами Інтернету, зокрема із соціальних мереж.

Навіть якщо не торкатися варіантів DSS, безпосередньо не основаних на аналізі даних (наприклад різноманітних експертних систем), то можна сказати, що прогноз про широке впровадження DSS повністю виправдався. Використання аналітики з метою підтримки ухвалення рішень сьогодні є повсюдним.

Далі, у звіті за результатами зустрічі у Лагуна Біч зазначалося, що “у сфері управління базами даних комп’ютери загального призначення перспективніші, ніж спеціалізована апаратура” [1]. Зауважимо, що до 1988 р. стала очевидною безрезультатність японського проекту комп’ютерів п’ятого покоління [11] (принаймні у зв’язку з розробленням спеціалізованої апаратури для побудови особливо ефективних і потужних СУБД).

У японців передбачалась розробка двох видів спеціалізованих апаратних засобів для підтримки СУБД: 1) процесори, в яких операції реляційної алгебри підтримувалися на рівні системи команд, та 2) магнітні диски з фіксованими головками, в головки яких мали вбудовуватися мікропроцесори для фільтрації даних “на льоту” у міру їх читання з дисків (розумні диски).

Перший підхід (процесори, в яких операції реляційної алгебри підтримувалися на рівні системи команд) не вдався з різних причин, головні з яких наведено нижче:

- ✓ оскільки бази даних зберігаються у зовнішній дисковій пам’яті, використовувати структуру даних оптимізовані для цього; для безпосереднього застосування команд процесора недостатньо перемістити дані з диска на основну пам’ять;
- ✓ таблиці баз даних, орієнтованих на мову структурованих запитів (structured query language, SQL), або SQL-орієнтованих баз даних (а до початку проекту п’ятого покоління вже було зрозуміло, що майбутнє за SQL-орієнтованими СУБД), часто є настільки великими, що не можуть повністю поміститися в основній пам’яті комп’ютера; ітераційне застосування команд “реляційної алгебри” далеко не завжди є можливим та/або ефективним;
- ✓ наявність алгоритмів виконання реляційних операцій, “захитих” у машинні команди, суперечить ідеї гнучкої оптимізації запитів, коли оптимізатор обирає найбільш вдалий алгоритм залежно від поточного стану бази даних;
- ✓ при збереженні баз даних на магнітних дисках основний час виконання запиту витрачається на виконання обмінів із зовнішньою пам’яттю; економія на часі оброблення даних в основній пам’яті часто буває несуттєвою.

Другий підхід (магнітні диски з фіксованими головками) не знайшов широкого застосування передусім з економічних причин. Легко бачити, що об’єм необхідної апаратури в дисковому пристрої з фіксованими головками набагато більше, ніж у традиційних жорстких дисках. Тому і вартість таких пристроїв значно вища (а надійність нижча). Крім того, з різних причин (зокрема через проблеми розподілу основної пам’яті) у традиційній архітектурі SQL-орієнтованих СУБД передбачається блоковий обмін даними із зовнішнім сховищем баз даних, і тому не дуже зрозуміло, як СУБД може отримати від “розумного” диска відфільтровані на льоту дані.

Цікаво, що Девітт і Грей [12] ще 1992 р. авторитетно вказували на безуспішність другого підходу і пророкували, що майбутні високопродуктивні паралельні СУБД, як і раніше, спиратимуться на використання дисків із рухливими головками. Але вже через десять років Джим Грей у своєму інтерв’ю [13] казав, що через стрімке здешевлення апаратних засобів (зокрема потужних мікропроцесорів) тепер можливе перенесення СУБД цілком на головку магнітного диска. Але ця ідея не знайшла поширення, тим більше що Джим вочевидь фантазував, бо така архітектура навіть на перший погляд видається дуже проблематичною.

Слід зазначити, що список рішень XXI століття для управління базами даних, які зорієнтовані на використання спеціалізованої апаратури, виглядає дуже скромно.

Що стосується *програмно-апаратних комплексів для управління базами даних* (database appliance), то, за визначенням компанії “Gartner” [14], це заздалегідь зібраний та/або конфігурований набір обладнання (сервери, пам’ять, накопичувачі та канали введення/виведення даних), програмне забезпечення (операційна система, СУБД та ін.), засоби обслуговування та підтримки. Таке визначення є досить розпливчастим, оскільки відповідно до нього database appliance є швидше маркетинговим, а не технологічним поняттям, і це відображає дійсність. У деяких продуктах цієї категорії (найякращим

прикладом є Oracle Exadata Database Machine [15]) використовуються спеціалізовані програмно-апаратні компоненти, але самі продукти є універсальними, однаково придатними для оброблення і транзакційних, і аналітичних робочих навантажень. Інший підхід сповідує Майкл Стоунбрейкер [16]. На його думку, до складу database appliance не повинна входити спеціалізована апаратура, втім спеціалізованим має бути програмне забезпечення самої СУБД. У будь-якому разі database appliance ніяк не схожі на машини баз даних 1980-х років.

Зараз дуже модно (і, напевно, у більшості випадків перспективно) намагатися використовувати *графічні процесори* (graphic processing unit, GPU) в інших сферах окрім оброблення графіки. Зокрема, зростає популярність застосування GPU для оброблення запитів у СУБД, орієнтованих на зберігання даних в основній пам'яті [17]. Але з низки причин немає впевненості в перспективності цього підходу (принаймні тому, що є дуже різні графічні процесори, а розробникам СУБД часто для отримання ефективності потрібно використовувати унікальні можливості GPU). Крім того, тут аж ніяк не йдеться про спеціалізовану апаратуру, розроблену саме для підтримки СУБД.

Можна вважати, що прогноз стосовно недоцільності розробки спеціальної апаратури саме для потреб СУБД та специфічного використання графічних процесорів для цього виявився досить точним.

Ще однією інновацією 1980-х років були *операційні системи* (ОС) із мікроядерною архітектурою. Тоді багатьом фахівцям із баз даних здавалося, що майбутнє за мікроядерними ОС із більш розвиненими і сучасними функціональними можливостями, ніж у тодішнього (та й теперішнього) лідера – сімейства ОС UNIX. Виконувалися численні проекти, серед яких Chorus [18], CLOS [19] та Mach [20]. З різних причин більшість цих проєктів виявилися безрезультатними (принаймні, ті результати, на які сподівалися розробники, отримати не вдалося).

Фактично обидві операційні системи – Linux і Windows – мікроядерними не є. Як і двадцять років тому, єдиною комерційною мікроядерною системою є QNX [21], але ця ОС (родина ОС) є нішевою і не може бути підтвердженням значного поширення мікроядерного підходу.

Цікаву роботу у XXI столітті провів Ендрю Таненбаум у рамках проєкту Minix 3 [22]. Це сучасний, добре опрацьований проєкт, але навряд чи можна очікувати його широкого практичного використання (принаймні тому, що в ньому реалізується тільки API Posix). Цікаво, що основна розробка Minix 3 завершилася в 2014 році одночасно з виходом Таненбаума на пенсію.

Отже, прогноз учасників зустрічі в Лагуна Біч стосовно неминучості бурхливого розвитку ОС з мікроядерною архітектурою під потреби архітектури СУБД не виправдався.

Що стосується “жахливого” *інтерфейсу SQL*, то учасники зустрічі зазначали, що бажано покращити існуючий “жахливий” інтерфейс SQL для вбудованих і динамічних запитів. Зауважимо, що це відбувалося за чотири роки до ухвалення стандарту SQL-92 (SQL:1992), у якому відповідні можливості було повністю та остаточно визначено. З ранньої історією стандартів мови SQL можна ознайомитися, наприклад, у [23].

Ідеї вбудовування операторів SQL у мови програмування та динамічної компіляції SQL з'явилися ще на зорі SQL у 1970-х рр. під час виконання компанією “IBM” проєкту System R (історію цього проєкту див. [24]). Причини появи засобів вбудовування та динамічної компіляції SQL часів проєкту System R коротко і чітко описані в інтерв'ю одного з основних розробників початкового SQL Дона Чемблена [25].

Звичайно, засоби вбудованого та динамічного SQL важко назвати красивими та/або елегантними (що можна сказати і про SQL в цілому). Але за час існування (понад 40 років) нічого принципово нового винайти не вдалося. Набуло поширення використання інтерфейсів рівня виклику функцій ODBC та JDBC. Але ці інтерфейси не позбавляють необхідності зберігати в програмі або динамічно формувати рядки, що містять тексти необхідних операторів SQL, а саме наявність у програмі таких рядків робить “жахливими” традиційні засоби SQL для підтримки операторів SQL, які вбудовуються і динамічно формуються.

Тож покращити інтерфейс SQL так і не вдалося.

2. Зустріч у Пало Альто 1990 року. На зустрічі прогнозувалось, що *додатки баз даних* наступного покоління матимуть мало спільного з додатками баз даних, які вико-

ристовувались у той час для оброблення даних про бізнес-сектор. Вони потребуватимуть набагато більше даних; вимагатимуть підтримки розширень системи типів даних, мультимедійних даних, складних об'єктів, оброблення правил та архівної системи зберігання даних, переосмислення алгоритмів виконання більшості операцій в СУБД.

Для правильної оцінки цього прогнозу необхідно відновити загальний контекст заходу, на якому його було висунуто. Щойно було опубліковано Маніфест про системи об'єктно-орієнтованих баз даних (Перший маніфест) [26], автори якого фактично заперечували подальшу перспективність технологій реляційних та SQL-орієнтованих баз даних, дорікаючи їм за обмеженість і нездатність задовольняти вимоги майбутніх додатків. Готувався у відповідь Маніфест про системи баз даних третього покоління (Другий маніфест) [27], в якому не заперечувалися конструктивні аспекти [26] (зокрема потреба в наявності у майбутніх СУБД розвиненої та розширюваної системи типів даних), але стверджувалося, що необхідні можливості СУБД можна отримати шляхом еволюції існуючої технології баз даних. Крім того, у Майкла Стоунбрейкера була комерційна СУБД *Illustra*, основана на дослідницькому прототипі *Postgres*, в якій вже підтримувалася більшість можливостей, згаданих у прогнозі (історія *Postgres-Illustra* добре описана Стоунбрейкером у [28]).

На початку XXI століття після подій 1990-х років, коли провідні вендори¹ SQL-орієнтованих СУБД прийняли концепцію об'єктно-реляційних баз даних і було випущено стандарт SQL:1999, яким затверджено цю концепцію, технологія об'єктно-орієнтованих СУБД стала стрімко втрачати популярність. Сьогодні, незважаючи на неодноразові спроби відродити привабливість ООСУБД для широкої аудиторії розробників баз даних та їхніх додатків, ці системи є нішевіми. З іншого боку, в сучасних провідних SQL-орієнтованих СУБД реалізовано практично всі можливості, відсутність яких мотивувала авторів [27, 28]. Можна сміливо сказати, що Другий маніфест переміг, але при цьому не можна стверджувати, що прогноз стосовно того, що додатки баз даних наступного покоління матимуть мало спільного з додатками баз даних, які використовувались у той час, повністю здійснився.

Безумовно, сьогоденні програми баз даних у середньому працюють із набагато більшими обсягами даних, ніж на початку 1990-х років. Розробники баз даних та їхніх додатків мають змогу визначати власні типи даних з довільно складною внутрішньою структурою та поведінкою. Але сьогодні, як і 10 років тому [29], користувачі не надто прагнуть визначати власні типи даних усередині баз даних. Скоріше можна говорити про те, що типи даних, які визначаються користувачами, застосовують самі вендори SQL-орієнтованих СУБД для розширення їх функціональних можливостей. Отже, оцінка потреби розробників баз даних та їхніх додатків у розширюваній системі типів даних і складних об'єктах виявилася перебільшеною.

Складна система правил залишилася прерогативою *Postgres* (тепер ще й *PostgreSQL*), а в переважній більшості сучасних СУБД правила використовуються в обсязі визначеного стандартом SQL механізму тригерів. Підтримка мультимедійних даних реалізується вендорами СУБД завдяки наявності стандартного типу *BLOB* та додаткових типів даних, що реалізуються за допомогою стандартного механізму визначення типів даних. Наявність архівної системи зберігання передбачалася в *Postgres*, але в сучасних SQL-орієнтованих СУБД така система явно відсутня. Зрештою, в джерелах не зафіксовано, щоб будь-який виробник SQL-орієнтованих СУБД докорінно змінив алгоритми виконання операцій.

Можна вважати, що прогноз стосовно переосмислення з часом алгоритмів виконання більшості операцій в СУБД справдився частково.

Учасники зустрічі вважали, що для кооперації різних організацій при розв'язанні наукових, технічних і комерційних проблем знадобляться *масштабовані, неоднорідні, розподілені бази даних*. Складними проблемами технології розподілених СУБД є неузгодженість баз даних, безпека та масштабованість.

Тематиці інтеграції неоднорідних баз даних починаючи з 1980-х рр. присвячено безліч досліджень, результати яких описувалися в багатьох публікаціях. Йдеться про так звану віртуальну інтеграцію різнорідних баз даних. На відміну від фізичної інтеграції

¹ Вендор – це фізична або юридична особа, яка просуває та постачає товари під власним брендом.

кількох баз даних з метою побудови сховища даних, ґрунтованого на ретельно розроблених процедурах ETL (Extract, Transform, Load), при віртуальній інтеграції додаткове сховище даних не створюється, дані вилучаються з різних джерел та інтегруються на льоту при обробленні інтеграційним сервером запитів до “інтегрованої” бази даних. Зрозуміло, що джерела даних, що інтегруються, зазвичай розподілені в локальній або глобальній мережі, можуть підтримуватися різними СУБД і навіть ґрунтуватися на різних моделях даних.

Багаторічні дослідження призвели до розроблення численних підходів і методів віртуальної інтеграції баз даних (нещодавній та короткий їх огляд див. [30]), але жодна з робіт не привела до створення працездатної технології. Незважаючи на наявність на ринку багатьох інтеграційних продуктів, широким попитом вони не користуються, і для дослідників у галузі даних тематика інтеграції різнорідних даних сьогодні не є магістральним напрямом, хоча потреба у віртуальній інтеграції неоднорідних джерел даних періодично виникає у різних галузях людської діяльності. Тож не можна сказати, що прогноз стосовно розвитку працездатних методів і технологій віртуальної інтеграції баз даних справдився, але й не можна вважати його невиправданим.

Разом із тим, у XXI столітті технологія розподілених систем змикається з технологією баз даних у кількох напрямках, відмінних від інтеграції даних. Перший напрям – спеціалізовані масивно-паралельні SQL-орієнтовані СУБД. У першому десятилітті XXI століття було розроблено дослідницькі прототипи аналітичної масивно-паралельної СУБД C-Store зі зберіганням таблиць по стовпцях [31] та транзакційної масивно-паралельної СУБД H-Store зі зберіганням даних в основній пам’яті [32]. Обидві ці системи ґрунтувалися на підході shared-nothing (без використання загальних ресурсів) з метою забезпечення горизонтальної масштабованості при зростанні числа вузлів у системі, обидві показали добрі результати на відповідних стандартних тестових наборах.

Обидва проекти було комерціалізовано. На основі результатів проекту C-Store було засновано компанію “Vertica”, поглинуту в 2011 р. компанією “Hewlett-Packard” [33]. Результати проекту H-Store використовуються в комерційному продукті VoltDB, що розробляється та підтримується однойменною компанією [34]. Є низка інших SQL-орієнтованих СУБД, спочатку призначених для використання у масивно-паралельному середовищі. Звичайно, під масивно-паралельним середовищем тут маються на увазі кластери, але що таке кластер, як не однорідна спеціалізована локальна мережа? Тому в масивно-паралельних архітектурах СУБД активно використовуються методи розподілених систем: реплікація даних, розподілені транзакції та ін.

Другий напрям – глобально розподілені СУБД категорії NoSQL. Цей напрям бурхливо розвивається і постійно змінюється, тому важко охарактеризувати його поточний стан. Проте, гадаємо, низка характеристик, зазначених у [35], наразі залишається актуальною. Напевно, найбільш загальними рисами цього напрямку є прагнення забезпечення горизонтальної масштабованості систем та високого рівня доступності даних за рахунок реплікації. Доступність забезпечується за рахунок суворої транзакційності (у глобально розподіленому середовищі суворозго узгодженість реплік коштує надто дорого). Незважаючи на загальну назву напрямку, в системах дедалі частіше підтримуються обмежені діалекти SQL.

3. Зустріч у Пало Альто 1995 року. Що стосується *розподілу інформації*, то в 1996 р. вже всім було зрозуміло, що людство вступило в нову еру Всесвітньої павутини. Учасники зустрічі зазначали, що WWW – це розподілене середовище, що складається з автономних систем, вузли якого дедалі частіше формуються як реляційні бази даних. Наявність цього середовища змушує переосмислити багато концепцій існуючої технології розподілених баз даних. Малася на увазі гіпотетична система, що підтримує оброблення запитів до Мережі в цілому і автоматично визначає, до яких веб-сайтів слід звернутися для виконання запиту. Наголошувалися такі напрями необхідних досліджень:

1) Облік і розрахунки.

Доступ до інформації, що зберігається на веб-сайтах, може бути платним. Гіпотетична система при обробленні запиту користувача повинна враховувати вартість цієї процедури і відтак обирати (можливо, на шкоду якості відповіді на запит) той набір

web-ресурсів, за які користувач буде здатен сплатити. При цьому, враховуючи, що отримання відповіді на запит не повинно обходитися користувачеві занадто дорого, ця служба обліку та розрахунків має бути гранично економічною.

Ця проблема на сьогоднішній день не вирішена у масштабах глобального Інтернету. Можливо (хоча й малоймовірно), що колись за неї візьметься якась велика Інтернет-компанія, наприклад “Google”. Однак є рішення подібних проблем у контексті сервіс-орієнтованих архітектур (Data as a Service [36]).

2) Безпека та конфіденційність.

Як зазначено у звіті про цю зустріч, є потреба в розробленні лише гнучких систем аутентифікації та авторизації, які підтримують доступ на основі різноманітних “ролей”, виконуваних користувачами (наприклад, один і той самий індивід може виступати в ролі лікаря деякого пацієнта, в ролі “лікаря взагалі” або в ролі приватної особи). Тематика досліджень і розробок у сфері безпеки та конфіденційності в Інтернеті є надзвичайно широкою. Важливість підтримки конфіденційності даних багаторазово зросла у зв’язку з появою соціальних мереж. Однак немає відомостей про захист конфіденційності даних у масштабі Інтернету на основі ролей.

3) Реплікація та узгодження даних.

З міркувань ефективності дані часто реплікуються на кількох вузлах. Коли всі ці вузли пов’язані мережею, можна підтримувати ідентичність копій. Однак у ситуаціях, коли зв’язок порушується, у копіях можуть виникнути відмінності. Після відновлення зв’язку повинен включитися механізм узгодження, який має узгодити всі копії та сформувати одну нову копію, яка відображатиме всі зроблені зміни. У новому інформаційному середовищі подібні ситуації стають не винятком, а нормою. Особливості розподілених систем категорії NoSQL показують, що достатньо ефективні механізми узгодження копій наразі не створено. Наслідуючи теорему CAP Еріка Брюера [37], розробники цих систем забезпечують високий рівень доступності на шкоду забезпечення ідентичності реплік.

4) Не повністю структуровані та неструктуровані дані.

Більшість даних в Інтернеті формуються за нечіткою та мінливою схемою, а іноді є взагалі неструктурованими. Тому фахівцям у галузі баз даних дуже важливо розширити механізми індексації та інші засоби підтримки пошуку для високоструктурованих даних та адаптувати їх до неструктурованого світу Інтернету. Така робота була потрібна, але виконується вона не фахівцями з баз даних. Пошук інформації в Інтернеті підтримують універсальні та спеціалізовані пошукові системи на базі результатів сегментного аналізу текстів і семантичного пошуку, які активно розвиваються. Із засобів управління базами даних у найкращому випадку використовуються СУБД категорії NoSQL (зокрема системи “ключ-значення”).

В останні десятиліття розширилися сфери застосування баз даних, і кожна з них являє собою принципово нове середовище, до якого необхідно адаптувати технології СУБД: інтелектуальний аналіз даних (data mining), сховища даних (data warehousing), репозитарії даних (data repository). Це обумовило пріоритетність досліджень таких проблем:

1) Інтелектуальний аналіз даних за напрямками: методи оптимізації складних запитів, що включають агрегацію та групування; підтримка “багатомірних” запитів, що містять дані, складені у вигляді “куба”, в осередках якого знаходяться дані, які представляють інтерес; мови запитів дуже високого рівня, а також інтерфейси підтримки користувачів, що не є експертами, яким потрібні відповіді на нерегламентовані запити.

Тут, перш за все, потрібно відзначити, що на зустрічі виникла плутанина між поняттями “інтелектуальний аналіз даних” (data mining) і “аналітичне оброблення даних” (on-line analytical processing, OLAP). Агрегація та групування – це дії, необхідні формування багатомірного куба, а аналіз даних з урахуванням їх у вигляді куба визначаються як OLAP. Тож у цьому пункті йдеться про аналітичне оброблення даних, а не про інтелектуальний аналіз даних.

Майже в той самий час, коли проходила друга зустріч у Пало Альто, Джим Грей із колегами винайшли метод і відповідні розширення SQL, які дозволяють ефективно будувати багатомірний куб на основі табличного представлення даних [38]. Ця робота справила вирішальний вплив на розвиток технології ROLAP (Relational OLAP) і

вважається одним із найвищих досягнень Джима Грея. Вона також дозволила створити нові методи оптимізації запитів із групуванням та агрегацією.

Роком пізніше компанією “Microsoft” було створено мову запитів у багатомірному кубі MDX (MultiDimensional eXpressions) [39], основним розробником якого був Моша Пасуманський. Більшості фахівців із баз даних ця мова здається ще більш “жахливою”, ніж SQL, але існує поширена думка, що вона близька аналітикам. Принаймні, як пропріетарна мова, MDX реалізований у всіх системах, що підтримують OLAP.

Близькі аналітикам інтерфейси, що дозволяють на напівінтуїтивному рівні досліджувати багатомірні куби, підтримуються у всіх OLAP-системах. Проте будь-якої стандартизації у цьому напрямі не видно.

2) Сховища даних. Для ефективно побудови сховищ даних потрібно створити додаткові інструменти: засоби створення “насосів даних” (data pump), тобто модулів, які розташовуються поза середовищем джерел даних і забезпечують ті зміни у сховищі, які є для нього істотними; методи “очищення даних” (data scrubbing), які забезпечують узгодження даних, видалення дублікатів даних, а також неправдоподібних значень; засоби для створення та підтримки метасловника, що інформує користувачів про засоби отримання даних.

Оскільки інтегровані дані з різних джерел та сховища “історичних” даних є незамінним постачальником даних для OLAP-аналітиків, їх якісній побудові в останні 20 років приділяли серйозну увагу як вендори SQL-орієнтованих СУБД, так і незалежні виробники. Загалом усі засоби, що забезпечують отримання даних із зовнішніх джерел, їх перенесення до схеми сховища даних та очищення і завантаження у сховище даних, зазвичай називаються засобами ETL (Extract, Transform, Load). Опис сучасних уявлень про ETL надано в [40] (наукові публікації на цю тему відсутні).

Засоби ETL включають і засіб очищення даних. Однак очищення в основному передбачає приведення даних до єдиного формату та видалення надлишкових даних. Менш формальне очищення вимагає виконання семантичного аналізу завантажених даних і виявляється надто трудомістким через дуже великі обсяги даних.

Відстеження походження даних (data lineage) і зараз є важливим засобом. Дослідження на цю тему тривають, але практичних результатів не отримано.

3) Репозитарії даних. Додатки, які стосуються категорії репозитаріїв, характеризуються тим, що вони призначаються для зберігання та управління як даними, і метаданими, тобто інформації про структуру даних. Приклади репозитаріїв – бази даних підтримки комп’ютерного проектування, включаючи CASE (системи проектування програмного забезпечення), і навіть системи управління документами. Відмінною рисою цих систем є часті зміни метаданих, характерні для будь-якого середовища проектування.

У 1990-х роках технологія репозитаріїв вважалася перспективною [41]. Однак згодом фахівцям у галузі баз даних не вдалося домовитися з фахівцями з автоматизації проектування, для якого призначалася ця техноЛОГІЯ.

4. Зустріч у Кембриджі 1996 року.

Розширюваність і компонентна організація. Серед стратегічно важливих дослідницьких проблем того періоду в звіті згадується про необхідність створення *універсальних СУБД, що дозволяють розробнику легко вводити нові типи даних*, розроблені поза цими СУБД, якими можна маніпулювати всередині бази даних нарівні з її власними типами даних, – так звану *розширюваність*. Потрібно знайти способи зробити архітектуру СУБД відкритою в такий спосіб, щоб забезпечити можливість а) підключення до неї нових функціональних компонентів і б) використання більш гнучких методів конфігурації бази даних відповідно до потреб додатків.

Зустріч відбулася в середині 1996 р., а на початку року компанія “Informix” придбала компанію “Illustra” разом із її технологією побудови об’єктно-реляційних СУБД [28]. На момент зустрічі в Кембриджі в компанії “Informix” велись активні роботи з інтеграції її тодішнього основного продукту Dynamic Server і СУБД компанії “Illustra”, щоб до кінця року побудувати об’єктний керівництвом компанії “Informix” Universal Server. Стандарт SQL, що включає об’єктно-реляційні можливості, тоді ще не був створений, і розробники спиралися на недопрацьовані драфти SQL3 [23]. У тому ж 1996 р. компанія “Sun Microsystems” реалізувала мову та віртуальну машину

Java [44], компонентну модель розробки програмного забезпечення Java Beans [45] та інтерфейс Java із SQL-орієнтованими базами даних JDBC [44].

У цьому контексті слід розглядати першу частину прогнозу. Звичайно, немає і не було можливості експортувати в базу даних розроблені поза нею типи даних користувача, щоб можна було використовувати їх нарівні з типами даних, визначеними засобами SQL (наприклад, визначати стовпці таблиць). Але серверні програми (збережені процедури, функції та методи нових типів даних) можна програмувати різними мовами, використовуючи різні бібліотеки, у тому числі технологію Java Beans.

Що стосується *компонентної організації* самих СУБД, то ця ідея приваблювала багатьох дослідників і до, і після зустрічі в Кембриджі. Найбільш переконливою публікацією на цю тему є [45]. Зовні ідея виглядає дуже зрозумілою та привабливою. Універсальні СУБД розраховані на загальне використання. У будь-якому конкретному середовищі і за будь-якого конкретного робочого навантаження багато складових СУБД не працюють, а їх наявність тільки уповільнює оброблення запитів користувача. Було б логічно і правильно враховувати (бажано автоматично) особливості поточного середовища використання та конфігурувати СУБД в такий спосіб, щоб у ній не залишалися непотрібні компоненти.

Побудувати компонентну СУБД з автоматичною конфігурацією технічно можливо. Але складність у тому, що чим більш структурованою є будь-яка складна система, тим більше кількість взаємодій між її компонентами, і ці взаємодії знижують продуктивність системи. Багато програмних систем спочатку проектувалися як набір багатьох відносно автономних компонентів, що взаємодіють, а потім з метою підвищення ефективності приводилися до практично монолітної архітектури.

Цікаво, що приблизно ті ж міркування про недоліки універсальних СУБД привели на початку ХХІ століття до ідеї про потребу переходу до спеціалізованих СУБД [46].

Що стосується *критеріїв оптимізації запитів*, то користувачі, наприклад, можуть віддати перевагу відповідям на свої запити з меншою швидкістю, точністю і повнотою, якщо це буде коштувати помітно дешевше.

Оптимізація запитів залишається однією з найскладніших функцій СУБД. Але сьогодні загальне уявлення про сутність оптимізації запитів краще отримувати не з академічних публікацій, а з документації вендорів [47].

Робота оптимізатора запитів складається з трьох фаз: перетворення запиту, генерація планів виконання запиту, оцінка планів і вибір найдешевшого плану. На вхід оптимізатора надходить внутрішнє подання запиту, отримане синтаксичним аналізатором. На першій фазі це подання перетворюється на семантично еквівалентне (непроцедурне) з імовірно кращою якістю (наприклад, запити із вкладеними підзапитами перетворюються на запити зі з'єднаннями). На другій фазі, керуючись набором закладених у СУБД стратегій та методів виконання окремих операцій низького рівня, оптимізатор будує набір процедурних планів виконання запиту. Для складних запитів цей набір потенційно може бути дуже великим, і завданням оптимізатора є збереження його осмисленого розміру з використанням евристики з метою вилучення імовірно найгірших планів. Нарешті, на третій фазі проводиться оцінка кожного залишеного у наборі плану і обирається план із найменшою вартістю.

У роботі оптимізатора використовуються два фундаментальні припущення: запит потрібно виконати повністю та точно; база даних зберігається у традиційній дисковій пам'яті на пристроях з рухомими головками. Відмова від цих припущень вимагатиме внесення принципових змін у кожен етап роботи оптимізатора. Тому зміна критеріїв оптимізації запитів є складним завданням, якого виробники СУБД намагаються уникнути.

Прогноз стосовно доцільності застосування оптимізатора запитів можна вважати нездійсненим.

Інтелектуальний аналіз даних усередині СУБД. Класифікація та кластеризація можуть розглядатися як випадкові запити, для яких потрібні нові сімейства мов запитів. Одним із дослідницьких завдань у цій галузі є розроблення адекватного набору простих примітивів запитів і нового покоління методів оптимізації запитів.

У 1990-х роках тема навантаження SQL-орієнтованих СУБД функціями інтелектуального аналізу даних була популярною серед фахівців у галузі баз да-

них [48]. Така популярність пояснювалось тим, що технологія баз даних вже переросла традиційні потреби бізнес-додатків і була здатна підтримувати нові класи додатків, у тому числі аналітичних. Одна з проблем, якій присвячувались тодішні дослідження, полягала у визначенні ступіня впливу виконання інтелектуального аналізу даних усередині СУБД на ефективність систем при обробленні запитів.

Сьогодні провідні постачальники SQL-орієнтованих СУБД (наприклад компанії “Oracle” і “Microsoft”) пропонують готові рішення для інтелектуального аналізу даних усередині бази даних, а також засоби розроблення додаткових серверних аналітичних механізмів. Нинішній попит на серверний інтелектуальний аналіз даних обумовлений потребою у перенесенні процедур з оброблення даних якомога ближче до місця їх зберігання.

Серверна аналітика дозволяє значно скоротити обсяг даних, що передаються від серверів баз даних на робочі станції, знизити рівень вимог до апаратного оснащення робочих станцій.

Але сучасні засоби та механізми інтелектуального аналізу даних дедалі більше орієнтуються на оброблення неструктурованих даних (текст, зображення, відео-, аудіодані тощо). Переваги СУБД переважно пов’язані з управлінням структурованими даними. Попит на засоби інтелектуального аналізу даних, збудовані в SQL-орієнтовані СУБД, сьогодні важко спрогнозувати. На жаль, часто спостерігається зворотна картина – у додатках для інтелектуального аналізу даних взагалі не використовуються СУБД.

А прогноз щодо розвитку мов запитів та їх оптимізаторів для підтримки інтелектуального аналізу даних загалом не виправдався.

(Продовження слідує)

Список використаних джерел

1. Bernstein P. A., Dayal U., DeWitt D. J., et al. Future Directions in DBMS Research // ACM SIGMOD Record, 1989. Vol. 18. No 1. P. 17–26.
2. Silberschatz A., Stonebraker M., Ullman J. D. Database Systems: Achievements and Opportunities // Communications of the ACM. 1991. Vol. 34. No 10. P. 110–120.
3. Silberschatz A., Stonebraker M., Ullman J. D. Database Research: Achievements and Opportunities into the 21st Century // SIGMOD Record. 1996. Vol. 25. No 1. P. 52–63.
4. Silberschatz A., Zdonik S., et al. Strategic Directions in Database Systems – Breaking Out of the Box // ACM Computing Surveys. 1996. Vol. 28. No 4. P. 764–778.
5. Bernstein P. A., Brodie M. L., Ceri S. et. al. The Asilomar Report on Database Research // SIGMOD Record. 1998. Vol. 27. No 4. P. 74–80.
6. Abiteboul S., Agrawal R., Bernstein P. A., et al. The Lowell Database Research Self-Assessment // Communications of the ACM. 2005. Vol. 48. No 5. P. 111–118.
7. Agrawal R., Ailamaki A., Bernstein P. A. et. al. The Claremont Report on Database Research // Communications of the ACM. 2009. Vol. 52. No 6. P. 56–65.
8. Abadi D., Agrawal R., Ailamaki A., et al. The Beckman Report on Database Research // ACM SIGMOD Record. 2014. Vol. 43. No 3. P. 61–70.
9. Keen P. G. W., Morton M. S. S. Decision support systems: an organizational perspective. Reading, Mass., Addison-Wesley Pub. Co., 1978.
10. Кузнецов С. Д. MapReduce: внутри, снаружи или сбоку от параллельных СУБД? // Труды ИСП РАН. 2010. Т. 19. С. 35–40.
11. Feigenbaum E. A., McCorduck P. The fifth generation: Japan’s computer challenge to the world // Creative Computing Magazine. 1984. Vol. 10. No 8. P. 103–111.
12. DeWitt D., Gray J. Parallel database systems: the future of high performance database systems // Communications of the ACM. 1992. Vol. 35. No 6. P. 85–98.
13. Winslett M. Jim Gray speaks out // ACM SIGMOD Record. 2003. Vol. 32. No 1. P. 53–61.
14. Gartner IT Glossary. Database Appliances. URL: <https://www.gartner.com/en/information-technology/glossary/database-appliances>
15. Exadata. URL: <https://ru.wikipedia.org/wiki/Exadata>
16. Stonebraker M. My Top 10 Assertions About Data Warehouses. BLOG@CACM, August 26, 2010.

17. Приказчиков С. О., Костенецкий П. С. Применение графических ускорителей для обработки запросов над сжатыми данными в параллельных системах баз данных // Вестник Южно-Уральского государственного университета. Серия «Вычислительная математика и информатика». 2015. Т. 4. Вып. 1. С. 64–70.
18. Rozier M., Abrossimov V., Armand F., Boule I., Gien M., Guillemont M., Herrmann F., Kaiser C., Langlois S., Léonard P., Neuhaus W. CHORUS Distributed Operating Systems // Computing Systems. 1988. Vol. 1. No 4. P. 305–370.
19. Burdonov I., Ivannikov V., Kopytov G., Kosachev A., Kuznetsov S. The CLOS project: Towards an object-oriented environment for application development. Next Generation Information System Technology // Lecture Notes in Computer Science (LNCS). 1991. Vol. 504. P. 422–427. https://doi.org/10.1007/3-540-54141-1_23
20. Golub D. B., Julin D. P., Rashid R. F., Draves R. P., Dean R. W., Forin A., Barrera J., Tokuda H., Malan G., Bohman D. Microkernel operating system architecture and mach // Proceedings of the USENIX Workshop on Micro-Kernels and Other Kernel Architectures, 1992. P. 11–30.
21. QNX Neutrino Real-time Operating System (RTOS). Proven reliability, performance, security. URL: <https://blackberry.qnx.com/en/software-solutions/embedded-software/qnx-neutrino-rtos>
22. Tanenbaum A., Appuswamy R., Bos H., Cavallaro L., Giuffrida C., Hrubý T., Herder J., van der Kouwe E., van Moelenbroek D. MINIX 3: Status Report and Current Research // Login. 2010. Vol. 35. No 3. P. 7–13.
23. Кузнецов С. Д. Стандарты языка реляционных баз данных SQL: краткий обзор // СУБД. 1996. № 2. С. 6–36. URL: http://citforum.ru/database/articles/art_2.shtml
24. McJones P. R. (Ed.) The 1995 SQL Reunion: People, Projects, and Politics. August 20, 1997 (2nd edition). URL: https://www.mcjones.org/System_R/SQL_Reunion_95/SRC-1997-018.pdf
25. Frana P. L. Oral history interview with Donald D. Chamberlin. Charles Babbage Institute, 2001. URL: <https://conservancy.umn.edu/handle/11299/107215>
26. Atkinson M., Bancilhon F., DeWitt D., Dittrich K., Maier D., Zdonik S. The Object-Oriented Database System Manifesto // Proceedings of 1st International Conference on Deductive and Object-Oriented Databases, Kyoto, Japan. New York, N.Y.: Elsevier Science, 1989. P. 223–240.
27. Stonebraker M., Rowe L., Lindsay B., Gray J., Carey M., Brodie M., Bernstein Ph., Beech D. Third-Generation Data Base System Manifesto // ACM SIGMOD Record. 1990. Vol. 19. No 3. P. 31–44.
28. Stonebraker M. The Land Sharks Are on the Squawk Box // Communications of the ACM. 2016. Vol. 59. No 2. P. 74–83.
29. Кузнецов С. Д. Объектно-реляционные базы данных: прошедший этап или недооцененные возможности? // Труды ИСП РАН. 2007. Т. 13. Ч. 2. С. 115–140.
30. Grinev M. N., Kuznetsov S. D. UQL: A UML-based Query Language for Integrated Data // Programming and Computer Software. 2002. Vol. 28. No 4. P. 189–196. <https://doi.org/10.1023/A:1016366916304>
31. Stonebraker M., Abadi D. J., Batkin A., Chen X., Cherniack M., Ferreira M., Lau E., Lin A., Madden S. R., O’Neil E. J., O’Neil P. E., Rasin A., Tran N., Zdonik S. B. C-Store: A Column-Oriented DBMS // Proceedings of VLDB. 2005. P. 553–564.
32. Stonebraker M., Madden S., Abadi D. J., Harizopoulos S., Hachem N., Helland P. The End of an Architectural Era (It’s Time for a Complete Rewrite) // Proceedings of VLDB. 2007. P. 1150–1160.
33. Vertica Accelerator. The Fastest Analytics and Machine Learning – from Start to Finish. URL: <https://www.vertica.com/>
34. VoltDB. URL: <https://www.voltdb.com/>
35. Кузнецов С. Д., Посконин А. В. Системы управления данными категории NoSQL // Программирование. 2014. Т. 40. № 6. С. 34–47.
36. Newman D. Data as a Service: The Big Opportunity for Business. URL: <https://www.forbes.com/sites/danielnewman/2017/02/07/data-as-a-service-the-big-opportunity-for-business/?sh=7a98e3a724d9>
37. Кузнецов С. Д. Когда, как и зачем стоит применять теорему CAP? // Открытые системы. СУБД. 2012. № 4. С. 56–59.

38. Gray J., Chaudhuri S., Bosworth A., Layman A., Reichart D., Venkatrao M., Pellow F., Pirahesh H. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals // *Data Mining and Knowledge Discovery*. 1997. Vol. 1. No 1. P. 29–53.
39. Nolan C. Manipulate and Query OLAP Data Using ADOMD and Multidimensional Expressions // *Microsoft Systems Journal*. 1999. Vol. 14. No 8. P. 97–106.
40. Oracle Database, Data Warehousing Guide, 10g Release 2 (10.2), 2005.
41. Bernstein P. A., Umeshwar D. An Overview of Repository Technology // *VLDB '94, Proceedings of the 20th International Conference on Very Large Data Bases*, September 12–15, 1994. P. 705–713.
42. Gosling J., Joy B., Steele G. *The Java Language Specification*. Addison Wesley, 1996.
43. Sun Microsystems. *JavaBeans API specification, version 1.01*, August 1997.
44. Hamilton G., Cattell R. *JDBC: A Java SQL API, Version 1.20*. Sun Microsystems Inc., January 1997.
45. Chaudhuri S., Weikum G. Rethinking Database System Architecture: Towards a Self-Tuning RISC-Style Database System // *VLDB '00, Proceedings of the 26th International Conference on Very Large Data Bases*, September 10–14, 2000. P. 1–10.
46. Stonebraker M., Çetintemel U. “One Size Fits All”: An Idea Whose Time Has Come and Gone // *ICDE '05, Proceedings of the 21st International Conference on Data Engineering*, April 5–8, 2005. P. 2–11.
47. *Database SQL Tuning Guide*. Chapter 4, Query Optimizer Concepts.
48. Imielinski T., Mannila H. A database perspective on knowledge discovery // *Communications of the ACM*. 1996. Vol. 39. No 11. P. 58–64.

References

1. Bernstein P. A., Dayal U., DeWitt D. J. et al. (1989). Future Directions in DBMS Research. *ACM SIGMOD Record*, vol. 18, issue 1, 17–26.
2. Silberschatz A., Stonebraker M., & Ullman J. D. (1991). Database Systems: Achievements and Opportunities. *Communications of the ACM*, vol. 34, issue 10, 110–120.
3. Silberschatz A., Stonebraker M., & Ullman J. D. (1996). Database Research: Achievements and Opportunities into the 21st Century. *SIGMOD Record*, vol. 25, issue 1, 52–63.
4. Silberschatz A., Zdonik S. et al. (1996). Strategic Directions in Database Systems – Breaking Out of the Box. *ACM Computing Surveys*, vol. 28, issue 4, 764–778.
5. Bernstein P. A., Brodie M. L., Ceri S. et al. (1998). The Asilomar Report on Database Research. *SIGMOD Record*, vol. 27, issue 4, 74–80.
6. Abiteboul S., Agrawal R., Bernstein P. A., et al. (2005). The Lowell Database Research Self-Assessment. *Communications of the ACM*, vol. 48, issue 5, 111–118.
7. Agrawal R., Ailamaki A., Bernstein P. A. et al. (2009). The Claremont Report on Database Research. *Communications of the ACM*, vol. 52, issue 6, 56–65.
8. Abadi D., Agrawal R., Ailamaki A. et al. (2014). The Beckman Report on Database Research. *ACM SIGMOD Record*, vol. 43, issue 3, 61–70.
9. Keen P. G. W., & Morton M. S. S. (1978). *Decision support systems: an organizational perspective*. Reading, Mass., Addison-Wesley Pub. Co.
10. Kuznetsov S. D. (2010). MapReduce: vnutri, snaruzhi ili sboku ot paralelnyih SUBD? [MapReduce: within, outside, or on the side-by-side with parallel DBMSs?]. *Trudy ISP RAN – Proceedings of the Institute for System Programing of the RAS*, 19, 35–40 [in Russian].
11. Feigenbaum E. A., & McCorduck P. (1984). The fifth generation: Japan’s computer challenge to the world. *Creative Computing Magazine*, vol. 10, issue 8, 103–111.
12. DeWitt D., & Gray J. (1992). Parallel database systems: the future of high performance database systems. *Communications of the ACM*, vol. 35, issue 6, 85–98.
13. Winslett M. (2003). Jim Gray speaks out. *ACM SIGMOD Record*, vol. 32, issue 1, 53–61.
14. Gartner IT Glossary. Database Appliances. Retrieved from <https://www.gartner.com/en/information-technology/glossary/database-appliances>
15. Exadata. Retrieved from <https://ru.wikipedia.org/wiki/Exadata>
16. Stonebraker M. (2010). My Top 10 Assertions About Data Warehouses. *BLOG@CACM*, August 26.

17. Prikazchikov S. O., & Kostenetskiy P. S. (2015). Primenenie graficheskikh uskoriteley dlya obrabotki zaprosov nad szhatyimi dannymi v parallelnykh sistemah baz dannykh [Applications of graphic accelerators for processing of queries over compressed data in parallel database systems] *Vestnik Yuzhno-Uralskogo gosudarstvennogo universiteta. Seriya: Vyichislitel'naya matematika i informatika – Bulletin of the South Ural State University. Series “Computational Mathematics and Software Engineering”*, vol. 4, issue 1, 64–70 [in Russian].
18. Rozier M., Abrossimov V., Armand F., Boule I., Gien M., Guillemont M., et al. (1988). CHORUS Distributed Operating Systems. *Computing Systems*, vol. 1, issue 4, 305–370.
19. Burdonov I., Ivannikov V., Kopytov G., Kosachev A., & Kuznetsov S. (1991). The CLOS project: Towards an object-oriented environment for application development. *Next Generation Information System Technology. Lecture Notes in Computer Science (LNCS)*, 504, 422–427. https://doi.org/10.1007/3-540-54141-1_23
20. Golub D. B., Julin D. P., Rashid R. F., Draves R. P., Dean R. W., Forin A., et al. (1992). Microkernel operating system architecture and mach. *Proceedings of the USENIX Workshop on Micro-Kernels and Other Kernel Architectures*, 11–30.
21. QNX Neutrino Real-time Operating System (RTOS). Proven reliability, performance, security. Retrieved from <https://blackberry.qnx.com/en/software-solutions/embedded-software/qnx-neutrino-rtos>
22. Tanenbaum A., Appuswamy R., Bos H., Cavallaro L., Giuffrida C., Hrubý T., et al. (2010). MINIX 3: Status Report and Current Research. *Login*, vol. 35, issue 3, 7–13.
23. Kuznetsov S. D. (1996). Standartyi yazyika relyatsionnykh baz dannykh SQL: kratkiy obzor [Language standards for SQL relational databases: a short review]. *Otkryitiye sistemyi. SUBD – Open Systems. Database Management Systems*, 2, 6–36 [in Russian]. Retrieved from http://citforum.ru/database/articles/art_2.shtml
24. McJones P. R. (Ed.) (1997). *The 1995 SQL Reunion: People, Projects, and Politics. August 20 (2nd edition)*. Retrieved from https://www.mcjones.org/System_R/SQL_Reunion_95/SRC-1997-018.pdf
25. Frana P. L. Oral history interview with Donald D. Chamberlin. Charles Babbage Institute, 2001. Retrieved from <https://conservancy.umn.edu/handle/11299/107215>
26. Atkinson M., Bancillon F., DeWitt D., Dittrich K., Maier D., & Zdonik S. (1989). *The Object-Oriented Database System Manifesto. Proceedings of 1st International Conference on Deductive and Object-Oriented Databases*. New York, N.Y.: Elsevier Science, 223–240.
27. Stonebraker M., Rowe L., Lindsay B., Gray J., Carey M., Brodie M., et al. (1990). *Third-Generation Data Base System Manifesto. ACM SIGMOD Record*, vol. 19, issue 3, 31–44.
28. Stonebraker M. (2016). *The Land Sharks Are on the Squawk Box. Communications of the ACM*, vol. 59, issue 2, 74–83.
29. Kuznetsov S. D. (2007). Obiektno-relyatsionnyie bazyi dannyih: proshedshiy etap ili nedootsenennyye vozmozhnosti [Object-relational databases: the past stage or unappreciated opportunities?]. *Trudy ISP RAN – Proceedings of the Institute for System Programming of the RAS*, vol. 13, issue 2, 115–140 [in Russian].
30. Grinev M. N., & Kuznetsov S. D. (2002). UQL: A UML-based Query Language for Integrated Data. *Programming and Computer Software*, vol. 28, issue 4, 189–196. <https://doi.org/10.1023/A:1016366916304>
31. Stonebraker M., Abadi D. J., Batkin A., Chen X., Cherniack M., Ferreira M., et al. (2005). C-Store: A Column-Oriented DBMS. *Proceedings of VLDB*, 553–564.
32. Stonebraker M., Madden S., Abadi D. J., Harizopoulos S., Hachem N., & Helland P. (2007). *The End of an Architectural Era (It's Time for a Complete Rewrite)*. *Proceedings of VLDB*, 1150–1160.
33. Vertica Accelerator. *The Fastest Analytics and Machine Learning – from Start to Finish*. Retrieved from <https://www.vertica.com/>
34. VoltDB. Retrieved from <https://www.voltdb.com/>
35. Kuznetsov S. D., & Poskonin A. V. (2014). Sistemyi upravleniya dannymi kategorii NoSQL [NoSQL Data Management Systems] // *Programmirovanie – Programming and Computer Software*, vol. 40, issue 6, 34–47 [in Russian].

36. Newman D. Data as a Service: The Big Opportunity for Business. Retrieved from <https://www.forbes.com/sites/danielnewman/2017/02/07/data-as-a-service-the-big-opportunity-for-business/?sh=7a98e3a724d9>
37. Kuznetsov S. D. (2012). Kogda, kak i zachem stoit primenyat teoremu CAP? [When, how and why CAP theorem should be applied?] Otkryitiye sistemyi. SUBD – Open Systems. Database Management Systems, 4, 56–59 [in Russian].
38. Gray J., Chaudhuri S., Bosworth A., Layman A., Reichart D., Venkatrao M., et al. (1997). Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. Data Mining and Knowledge Discovery, vol. 1, issue 1, 29–53.
39. Nolan C. (1999). Manipulate and Query OLAP Data Using ADOMD and Multidimensional Expressions. Microsoft Systems Journal, vol. 14, issue 8, 97–106.
40. Oracle Database, Data Warehousing Guide, 10g Release 2 (10.2), 2005.
41. Bernstein P. A. & Umeshwar D. (1994). An Overview of Repository Technology. VLDB '94, Proceedings of the 20th International Conference on Very Large Data Bases, 705–713.
42. Gosling J., Joy B. & Steele G. (1996). The Java Language Specification. Addison Wesley.
43. Sun Microsystems. JavaBeans API specification, version 1.01, August 1997.
44. Hamilton G. & Cattell R. (1997). JDBC: A Java SQL API, Version 1.20. Sun Microsystems Inc.
45. Chaudhuri S. & Weikum G. (2000). Rethinking Database System Architecture: Towards a Self-Tuning RISC-Style Database System. VLDB '00, Proceedings of the 26th International Conference on Very Large Data Bases, 1–10.
46. Stonebraker M. & Çetintemel U. (2005). “One Size Fits All”: An Idea Whose Time Has Come and Gone. ICDE '05, Proceedings of the 21st International Conference on Data Engineering, 2–11.
47. Database SQL Tuning Guide. Chapter 4, Query Optimizer Concepts.
48. Imielinski T. & Mannila H. (1996). A database perspective on knowledge discovery. Communications of the ACM, vol. 39, issue 11, 58–64.

Посилання на статтю:

Єршова О. Л., Ставицький О. В. Дослідження сучасних тенденцій у технологіях баз і сховищ даних як технологічної та архітектурної основи для створення програмних і інтелектуальних систем засобами сучасних мов програмування. Частина 1. *Науковий вісник Національної академії статистики, обліку та аудиту: зб. наук. пр.* 2021. №3-4. С. 94-108. doi: 10.31767/nasoa.3-4-2021.11.