

60 РОКІВ БАЗАМ ДАНИХ (частина третя)

Наводиться огляд досліджень і розробок баз даних з моменту їх виникнення в 60-х роках минулого століття і по теперішній час. Виділяються наступні етапи: виникнення і становлення, бурхливий розвиток, епоха реляційних баз даних, розширені реляційні бази даних, постреляційні бази даних і великі дані. На етапі становлення описуються системи IDS, IMS, Total і Adabas. На етапі бурхливого розвитку висвітлені питання архітектури баз даних ANSI/X3/SPARC, пропозицій КОДАСИЛ, концепції і мов концептуального моделювання. На етапі епохи реляційних баз даних розкриваються результати наукової діяльності Е. Кодда, теорія залежностей і нормальних форм, мови запитів, експериментальні дослідження і розробки, оптимізація та стандартизація, управління транзакціями. Етап розширених реляційних баз даних присвячений опису темпоральних, просторових, дедуктивних, активних, об'єктних, розподілених та статистичних баз даних, баз даних масивів, машин баз даних і сховищ даних. На наступному етапі розкрита проблематика постреляційних баз даних, а саме, NOSQL-, NewSQL- і онтологічних баз даних. Шостий етап присвячений розкриттю причин виникнення, характерних властивостей, класифікації, принципів роботи, методів і технологій великих даних. Нарешті, в останньому розділі дається короткий огляд досліджень і розробок по базах даних в Радянському Союзі.

Ключові слова. Типи баз даних: ієрархічна, мережева, реляційна, навігаційна, темпоральна, просторова, просторово-темпоральна, просторово-мережева, об'єктів, що переміщуються, дедуктивна, активна, об'єктно-орієнтована, об'єктно-реляційна, розподілена, паралельна, масивів, статистична, багатовимір-на, машина баз даних, сховища даних, NoSQL, ключ-значення, стовпчикова, документно-орієнтована, графова, мультимодельна, хмарна, наукова, багатозначна, XML, NewSQL, онтологічна, великі дані.

Розподілені бази даних.

Розподілена база даних (РобД) – це інтегрована сукупність баз даних, які фізично розподілені комп'ютерною мережею. А розподілена система управління базами даних (РосУБД) – це програмна система, яка управляє розподіленою базою даних таким чином, що аспекти розподілення стають прозорими (невидимими) для користувачів. РосУБД може мати спільний інтерфейс для доступу до розподілених даних [10].

Виникнення РобД обумовлене тим, що тут природно представляються організаційна структура даних підприємства, підвищується надійність, доступність і локальний контроль, зростає продуктивність, полегшується процедура розширення системи.

Розробка концепції і дослідження РаБД почалися в другій половині 70-х років. Серед численних дослідницьких систем найбільш відомими є три: система SDD-1 [439 - 442], створена в науково-дослідницькому відділенні корпорації Computer Corporation of America наприкінці 1970-х і початку 1980-х років, система

System R * [443 - 446], розподілена версія системи -протоколу System R, створена в дослідницькому відділенні компанії IBM на початку 1980-х років, і система Distributed Ingres [447 - 449], розподілена версія прототипу системи Ingres, створена також на початку 1980-х років у Каліфорнійському університеті в Берклі. Варто згадати також проект POLIPHEME у Франції [450]. У проєктах 70-х років було виявлено низку ключових проблем, пов'язаних із розробкою систем розподілених баз даних, подані підходи до їх вирішення. Той факт, що всього лише за кілька років у цій царині були отримані значні результати, підтверджується появою наприкінці 70-х оглядів на цю тему [451 - 453].

До кінця 80-х років були здійснені численні дослідження, експериментальні розробки, почали з'являтися перші промислові РобД. Було звернуто увагу на створення мультибаз даних і на надання більшої автономності індивідуальним системам [454 - 455].

У 1986 – 87 роках були представлені перші промислові РосУБД Ingres/STAR,

Oracle 7 s DB2. Тож постала необхідність формування основних принципів, вимог і функціональних можливостей РоБД. Відповіддю на ці вимоги в 90-му році стала стаття Дейта [456], де були сформульовані 12 правил РоБД, головне з них – прозорість для користувачів розподіленої структури баз даних. Ці правила були прийняті науковим товариством і ними досі користуються у розробці РоСУБД.

Типи РоБД. Існують два основні типи РоБД: однорідні (homogeneous) і неоднорідні (heterogeneous).

Однорідні РоБД. У них усі вузли перебувають під управлінням РоСУБД одного типу (і, можливо, під управлінням однієї операційної системи). Існують два типи однорідних РоБД: автономні і неавтономні. Автономні працюють незалежно, передаючи і приймаючи повідомлення одне одному для спільного оновлення даних. Неавтономні РоБД передбачають існування центральної (головної) РоСУБД, котра координує доступ до даних і їх оновлення в мережі. Звичайні розподілені (regular distributed) й паралельні бази даних належать до однорідних РоБД.

Неоднорідні РоБД. Вони працюють під управлінням різних операційних систем і типів РоСУБД. Існує чотири типи неоднорідних РоБД:

- федеративні (federated);
- із посередниками (mediators);
- мультибази даних (multidatabases);
- однорангові бази даних.

Федеративні РоБД. Являють собою об'єднання БД різних типів, якими володіють різні користувачі і які об'єднуються для спрощення спільного використання даних. Федеративна БД передбачає визначення глобальної інтеграційної схеми, що містить відображення в схеми баз даних учасників. Уперше федеративну БД визначили Маклеод і Хаймбігнер (McLeod, Heimbigner) 1985 року [457], вона досліджувалася у багатьох працях [458, 459]. У [460] наводиться огляд федеративних БД.

У разі суттєвого збільшення баз даних, що інтегруються, буває важко, а інколи і неможливо визначити глобальну

інтеграційну схему. Мультибази даних не передбачають існування глобальної схеми. Натомість мова запитів дає можливість специфікувати вирази, які дозволяють здійснювати пошук за об'єднувальними базами даних.

Посередники (mediators) [462, 463]. Вони розміщені між системами з однією глобальною схемою і взагалі без схем. Натомість користувачі визначають погляди – посередники, які об'єднують і узгоджують дані з різних джерел. Для таких поглядів необхідна мова запитів, яка здатна формулювати запити за багатьма базами даних, подібно до мови запитів мультибаз даних.

Однорангові БД (peer – to – peer databases – P2PDB) [464 - 466]. Вони являють собою сукупність автономних локальних репозиторіїв/баз даних, які взаємодіють між собою на рівноправній основі. Основне завдання P2PDB – розповсюджувати запити між гетерогенними вузлами у великій розподіленій мережі. Таке розповсюдження може зупинитися за кілька кроків. Це допустимо для деяких сучасних систем, які не потребують високої точності результатів. Як – от у пошукових машинах Інтернету.

Розподіл даних. Фрагментація. В РоБД існує задача розподілу логічно цілісної БД по вузлах розподіленої структури таким чином, щоб оптимізувати цільову функцію. Є два фундаментальні методи вирішення цієї задачі: фрагментація і реплікація.

Фрагментація (сегментація, декомпозиція) передбачає розподіл даних на сегменти (фрагменти) даних, що не перетинаються, для їх прив'язування до вузлів мережі. Реплікація передбачає запам'ятовування на різних вузлах ідентичних копій всієї або ж частини логічної бази даних. РоСУБД гарантує користувачам прозорість такого розподілу. Крім цього, існує задача розміщення фрагментованих/реплікованих даних у вузлах мережі.

Є два види фрагментації: горизонтальна і вертикальна. Під час горизонтальної фрагментації відношення розбивається на групи рядків, які розподіляються по вузлах. За вертикальної фрагментації відно-

шення розбиваються на групи стовпчиків. Допускається також гібридна фрагментація, яка передбачає одночасне використання двох попередніх фрагментацій.

Основні дослідження з фрагментації були здійснені на початку 80-х років [467, 469, 470, 10]. Одна з основних задач вертикальної фрагментації – визначення наборів атрибутів, що мають бути об'єднані в одну групу. В працях [471, 472] було запропоновано алгоритм енергетичного зв'язування (bond energy algorithm) для групування атрибутів. На його основі здійснюється вертикальна фрагментація. У праці [473] висунуто модифікований варіант цього алгоритму.

Що ж до задачі розміщення даних, то роботи в цьому напрямку почалися ще наприкінці 60-х років, коли досліджувалась проблема розміщення файлів [474]. У працях [475 - 477] була досліджена проблема складності задач розміщення. В [478, 479] досліджені динамічні алгоритми розміщення даних, які передбачають можливість зміни початкового розміщення для обліку змін у методах доступу й робочих навантаженнях. Пропонувались також методи інтеграції фрагментації та розміщення [467, 468].

Розподіл даних. Реплікація. Роботи із реплікації БД датуються початком 80-х років, коли були здійснені дослідження щодо доступності даних, а більшість пропонованих рішень забезпечувала узгодженість даних. Цікавим оглядом досліджень того часу є стаття [480].

Основна проблема реплікації даних полягає в тому, що оновлення будь – якого заданого логічного об'єкту має розповсюджуватися по всіх збережених копіях цього об'єкту. 1996 року Грей продовжив дослідження в цій царині [481] й запропонував варіанти негайного (eager) і відтермінованого (lazy) оновлень. Один із варіантів відтермінованого оновлення – використання первинної копії (master copy), коли основна копія оновлюється оперативно, а оновлення вторинних копій відкладається до зручного часу. Причому синхронна реплікація передбачає завершення розповсюдження змін до завершення транзакції, а синхронна реплікація допускає розповсю-

дження змін після завершення транзакції. Зрештою Грей запропонував дворівневе оновлення транзакцій.

Остання стаття активізувала подальші дослідження з реплікації. Серед напрямків досліджень – зменшення накладних витрат на комунікацію і координацію за рахунок затримки оновлень видалених копій. Однак у цьому випадку копії можуть містити застарілі або навіть неузгоджені дані. Через це з'явилися пропозиції щодо неузгодженості [482], встановлення обмежень на «застарілі» дані та виявлення й усунення неузгодженості [483].

Інший напрямок досліджень, які здійснювалися в контексті масштабованої кластерної реплікації, пов'язаний із розробкою методів забезпечення надійної узгодженості за прийнятних витрат [484, 485]. Із появою хмарних систем зберігання в дослідження були залучені внутрішньохмарні реплікації, концептуально схожі на кластерну реплікацію, а також міжхмарні й геореплікації [486, 487]. Що ж до розподілених систем, то спочатку дослідження концентрувалися на реплікаційних файлових системах [488, 489], згодом на веб-серверних реплікаціях [490] і файлових реплікаціях однорангових систем [491]. Також були отримані результати із відмовостійких реплікацій об'єктів [492, 493]. Як і в сфері баз даних, найостанніші результати щодо реплікації розподілених систем, належать до хмарної інфраструктури, а саме – реплікації в системах зберігання [494], таких, як HDFS і Cassandra, а також глобальна (wide – area) реплікація [484, 495].

Тупики в РoБД. У БД, що послуговуються протоколом блокувань для доступу до спільно використовуваних даних, можливі тупикові ситуації (deadlock), коли транзакція очікує події, яка може статися через наступні дії самої транзакції. Як-от, коли дві транзакції чекають одна на одну.

Існують такі категорії алгоритмів виявлення тупиків у РoБД [10]: централізовані, ієрархічні, розподілені. Централізовані алгоритми [496, 497] використовують центральний вузол для виявлення тупиків. Як зазначається в [439], цен-

тралізоване двофазне блокування (two – phase locking – 2PL) і виявлення тупиків стало позитивною природною комбінацією. Централізоване виявлення тупиків уперше було реалізовано в Distributed INGRES [498]. Ієрархічні алгоритми [497, 499] для виявлення тупиків покладаються на ієрархічну структуру вузлів РoБД. Розподілені алгоритми [499, 500] покладаються на кооперацію всіх вузлів РoБД для виявлення тупиків. Розподілене виявлення тупиків уперше було реалізовано в System R* [500].

У [10] представлено огляд методів управління розподіленими тупиками. В оглядах [501 - 504] обговорюються різні розподілені алгоритми виявлення тупиків. У праці [505] наводиться порівняльний аналіз додаткових алгоритмів виявлення тупиків : просування шляху (path – pushing) [499], зондовий (probe – based) [506], глобального стану (global state) [507, 508].

Розподілена обробка запитів. Це процедура виконання запиту в розподіленому середовищі, де дані розміщені в різних вузлах комп'ютерної мережі. Вона передбачає перетворення запиту, сформульованого високорівневою мовою (наприклад, SQL), у вираз процедурної мови низького рівня (приміром, реляційна алгебра), який названо «план виконання запиту». Далі цей план оптимізується з урахуванням розподіленості даних і, зрештою, відбувається послідовне виконання операторами отриманого оптимального плану.

Дослідження із розподіленої обробки запитів почалися наприкінці 70-х років. Тоді було розроблено три експериментальні системи, де були закладені фундаментальні методи розподіленої оптимізації і обробки запитів: SDD-1 [509] (1976), Distributed INGRES [510, 511] (1977) і System R* [512, 513] (1981). Вважається, що першим дистрибутивним алгоритмом оптимізації запитів є «скелелазіння» (hill climbing) Вонга [514], який згодом було поліпшено в SDD-1 через включення операції напівз'єднання. Оптимізаційний алгоритм SDD-1 є статичним, він спрямований на зменшення сумарних комунікаційних витрат і не підтримує фрагментацію і реплікацію. Дистрибутивний алгоритм

оптимізації запитів Distributed Ingres [510] у кожному наступному кроці детерміновано аналізує простір імовірних планів і робить висновок щодо локальної оптимізації. Він також підтримує горизонтальну фрагментацію. Цільова функція оптимізації є виваженою комбінацією вартості сумарного часу і часу реакції. Алгоритм є динамічним.

Дистрибутивний алгоритм оптимізації запитів System R* [513] всебічно аналізує межі пошуку всіх можливих планів виконання запитів. Алгоритм не підтримує фрагментацію і реплікацію. Цільова функція оптимізації враховує локальну обробку й комунікаційні витрати. Алгоритм є статичним.

Було здійснено дослідження з оптимізації виконання виражень реляційної алгебри із розподіленим середовищем включно. В статті [515] наведено огляд цих результатів. Запропоновано кілька підходів щодо динамічної оптимізації запитів для паралельних і розподілених баз даних [516]. Алгоритм у [517] передбачає зміну плану відпрацювання запиту в процесі його виконання, аби врахувати непередбачувані обставини. В системі Mariposa [518] вперше була висунута модель оптимізації розподіленого запиту. В монографіях [519, 10] детально висвітлюються результати в галузі технологій розподілених баз даних і оптимізації розподілених запитів, отримані у 80 – 90-х роках. Стаття [516] є найновішим оглядом у цій сфері.

Управління паралелізмом (Concurrency Control). Це процедура такого управління одночасною безконфліктною роботою багатьох транзакцій, за якої транзакції коректно виконують свою роботу без порушення обмежень цілісності БД (принципа ACID).

Дослідження з управління паралелізмом у розподілених системах зародилися на початку 80-х років. Вони спиралися на широко відому на той час статтю [520] із управління паралелізмом у централізованих БД. Грей розвинув ці ідеї для транзакцій [521], а Спектор і Шварц [522] дослідили транзакції у розподіленому середовищі.

Було запропоновано три механізми управління паралелізмом: блокування, оптимістичний протокол і впорядкування за часовими позначками (timestamp ordering).

Блокування. Це обмеження доступу до спільно використовуваних ресурсів (даних) із одночасним виконанням багатьох транзакцій.

Першим широко відомим механізмом блокувань було двофазне блокування (Two – Phase Locking – 2PL), яке вперше описане в [520]. Згодом була визначена велика кількість його різновидів: строга (strict), консервативна (conservative), первинної копії (primary copy), розподілена (distributed), точна (rigorous). У [523] описано варіант 2PL з урахуванням використання старих значень. Також були запропоновані гібридні блокування, що передбачають використання методів, відмінних від 2PL [524 - 526].

Наступний тип управління паралелізмом дістав назву «оптимістичний» у тому сенсі, що створюються локальні копії даних транзакції й оновлюються саме вони, а не власне дані. Вперше цей метод був висунутий у праці [527], і відтоді досліджено чимало його різновидів [528, 529].

Зрештою впорядкування за часовими позначками використовує Системний Час або ж певний логічний лічильник в ролі часових позначок для впорядкування виконання паралельних транзакцій. Транзакції присвоюється часова позначка здебільшого із урахуванням часу запуску транзакції. Старіша транзакція є пріоритетнішою. У разі конфлікту перевага надається пріоритетній транзакції. Цей протокол описано в [530 - 532]. У [532, 533] також описані багатоверсійні часові позначки. Цікавий огляд методів управління паралелізмом знаходимо в статті [534].

Машини баз даних.

Загалом машиною бази даних (МБД) прийнято називати апаратно-програмний мультимікропроцесорний комплекс, призначений для всіх або деяких функцій СУБД. Цей напрямок баз даних з'явився на початку 70-х років. На першому етапі протягом 10 – 12 років основна ідея досліджень і розробок МБД була спрямована на

створення спеціальних обчислювальних обладнань і розробку архітектур, де процес обчислення бази даних розміщувався ближче до дисків з метою значного збільшення продуктивності. На той час було реалізовано понад 50 проєктів. Основними критеріями оцінки того чи іншого проєкту були повнота виконуваних функцій СУБД і очікуване підвищення продуктивності. На основі експериментальних прототипів у багатьох країнах світу згодом сформувався виробництво різних зразків машин баз даних [535].

У цей період були запропоновані рішення, які дістали назву процесорів фільтрів. Їхньою задачею була перевірка передаваних даних із дисків на зовнішній сервер. У працях [536, 537] процесори фільтрів були розділені на такі групи:

процесор на доріжку (Processor-per-Track - PPT),

процесор на голівку (Processor-per-Head - PPH),

процесор на диск (Processor-per-Disk – PPD),

мультипроцесорний кеш (Multi-Processor Cashe – MPC),

процесор на комірку бульбашкової пам'яті (Processor-per-Bubble-cell – PPB).

Процесор на доріжку – PPT.

Згідно із [538] першим дослідником в галузі МБД був Даніель Слотник (Daniel L. Slotnick), який 1970 року опублікував статтю [539], де висунув пропозицію архітектури з процесором на кожному доріжку. В цій архітектурі запам'ятовувач складається з великої кількості комірок, кожна з яких має доріжку даних, зв'язану з процесором, котрий блискавично виконує функцію пошуку потрібних даних. Координацію роботи з комірками здійснює управляючий процесор. Основна ідея Слотника полягала в тому, щоби здійснювати пошук у базі даних безпосередньо на запам'ятовувачі. Тим самим обмежити обсяг даних, які передаються на основний процесор. У подальшому підхід Слотника розвинули Паркер (Parker) [540], Мінські (Minsky) [541] і Пархамі (Parhami) [542]. На основі цієї архітектури реалізовані МБД RAP [543], CASSM [544], RARES [545].



Даніель Слотник

Процесор на голівку – РРН. До цього класу належать МБД, в яких логіка обробки даних прив'язується до кожної голівки в диску з рухомими голівками. В РРН дані паралельно передаються від головок до багатьох процесорів. Кожен процесор застосовує функцію відбору до вихідного потоку даних і розміщує вибрані дані у вихідному буфері. За такої організації кожен циліндр диску з рухомою голівкою аналізується щокочен оберт. До цього класу належать МБД DBC [546], SURE [547].

Процесор на диск – РРД. На відміну від РРТ і РРН дана архітектура передбачає використання стандартних дисководів. Процесор (або багато процесорів) розміщується між диском і запам'ятовувачем, куди передаються відібрані дані. Цей процесор діє як фільтр, передаючи до основного процесора лише ті дані, які відповідають критерію відбору.

Мультипроцесорний кеш – МРС. До цього класу належать МБД, в яких спеціалізовані процесори відділяються від пристроїв зберігання великим дисковим кешем. Мета цього архітектурного вирішення – підтримувати паралелізм обробки під час використання традиційних пристроїв зберігання. Перед обробкою дані мають бути переміщені з диска в кеш, після чого вони стають доступними процесорам у паралельному режимі. Більше того, проміжні результати виконання запиту поміщаються процесорами в кеш і до них надається швидкий доступ для виконання наступних операцій запиту. Реалізовано багато МБД цього класу, як от RAP2 [548], DIRECT

[549], INFOPLEX [550], RDBM [551], DBMAC [552].

Процесор на комірку бульбашкової пам'яті – РРВ. З кожною коміркою зовнішньої пам'яті асоціюється процесор.

Варто зауважити, що в цей період більшість проектів розробки МБД концентрувалася навколо спеціалізованого апаратного забезпечення, яке перебувало ще в стадії розробки. Зокрема, такого, як CCD-пам'ять (charge-coupled device, пристрій із зарядовим зв'язком), бульбашкова пам'ять (bubble memory), диски з фіксованими голівками на кожному доріжку (hesd-per-track disks), оптичні диски (optical disks). Жодна з цих технологій уповні не виправдала себе. Тож після дванадцяти років активності в цьому напрямку майбутнє МБД виглядало непевно навіть для найбільших її прихильників. Так, наприклад, 1983 року стаття [538] передвікала стрімке зникнення МБД. Найвідомішими монографіями на тему машин баз даних першого етапу були праці Есена Озкарахан (1986р.), а також Калиниченка Л.А. і Ривкіна В.М. (1990р.) [9].



Есен Озкарахан

Попри песимістичні настрої, напрямком МБД вижив і успішно розвивається завдяки паралельним системам баз даних.

Як зазначається в [553], успіх паралельних баз даних пояснюється широким розповсюдженням реляційних баз даних. 1983 року вони лише почали з'являтися на ринку, сьогодні ж – домінують. Реляційні запити щонайкраще підходять для паралельного виконання; вони складаються

з однорідних операцій над однорідним потоком даних. Кожна операція утворює нове відношення, тож із операцій можуть складатися високо паралельні графи потоків даних. Дві операції можуть працювати послідовно, якщо направити вивід однієї операції на вхід іншої. Це так званий конвеєрний паралелізм (pipelined parallelism). Якщо розділяти введені дані між кількома процесорами і пам'яттю, часто-густо з'являється можливість розбити операцію на кілька незалежних операцій, кожна з яких працює з частиною даних. Такий розподіл даних і обробка мають назву «роздільний паралелізм» (partitioned parallelism).

Таким чином, історія демонструє, що вузькоспеціалізовані машини баз даних виявилися неспроможними, тоді як паралельні системи баз даних досягли величезних успіхів. Успішні паралельні системи баз даних базуються на звичайних процесорах, пам'яті і дисках. Саме в цих системах здебільшого відобразилися ідеї високо паралельних архітектур.

У 1980-х роках дослідження щодо машин баз даних були зосереджені на масивних паралельних обчисленнях (massive parallel computing). Процесори були з'єднані у вузли, і такі вузли потім об'єднувалися у високошвидкісні міжблокові зв'язки [553, 554]. Деякі з цих типів машин баз даних досягли значного успіху в промисловості.

У середині 80-х років Стоунбрейкер висунув наступну просту класифікацію паралельних мультипроцесорних систем [555].

Спільно використовувані пам'ять і диски (shared-everything – SE). Всі процесори мають прямий доступ до загальної глобальної пам'яті і до всіх дисків. Взаємодія між процесорами відбувається з використанням загальної пам'яті. Прикладами таких систем можуть бути XPRS [556], DBS3 [557]? Volcano [558].

Спільно використовувані диски (shared disks – SD). Кожен процесор має власну пам'ять і прямий доступ до всіх дисків. Усі процесори пов'язані один з одним через високошвидкісну мережу для передачі даних. Прикладами паралельних систем

баз даних SD-архітектури є IBM IMS [559], Oracle Parallel Server [560], nCUBE [561], VAXclusters [562], IBM Parallel Sysplex [563].

Відсутність спільного використання ресурсів (shared-nothing – SN). Кожна пам'ять і диск є у розпорядженні одного процесора, який працює як сервер збережених в них даних. Масовий запам'ятовуючий пристрій у таких архітектурах розподілений між процесорами через з'єднання одного чи більше дисків. Так само, як і в SD-архітектурі, всі процесори зв'язані один з одним через високошвидкісну мережу. Відсутність спільного використання ресурсів характерна для систем баз даних, які використані в проектах Teradata[564], Gamma [565], Tandem [566], Bubba [567], Arbre [568], і nCUBE [569]. Прикладами комерційних систем SD-архітектури є NonStop SQL [570], Informix PDQ [571], NCR/Teradata DBC [572], IBM DB2 PE [573].

Аналізу архітектур паралельних систем баз даних присвячена також стаття Соколинського Л.Б. [574].

Згодом з'явилися мультипроцесорні системи, які поєднували характеристики SE- і SN-архітектур, тому Коупленд і Келлер [575] запропонували розширити класифікацію Стоунбрейкера наступним чином:

- кластеризовано все (clustered everything – CE) – кластери з SE-архітектурою об'єднуються за принципом SN-архітектури;

- кластеризовані диски (clustered-disk – CD) – кластери з CD-архітектурою об'єднуються за принципом CN-архітектури.

Такі архітектури дістали назву ієрархічних [576]. Пропозиції Коупленда дозволяють створювати дворівневі ієрархії (ISE/SD-кластери першого рівня з'єднуються в SN-кластери другого рівня). Дворівнева архітектура Коупленда може бути легко розширена до архітектур із трьома чи більше ієрархічними рівнями. Дворівнева ієрархічна архітектура була досліджена в працях [575, 577 - 580].

Зазначимо, що в другій половині 90-х років з'явилися багатопроцесорні системи із компонентами складної конструкції. Вони увібрали в себе різні архітектурні

рішення, що не підпадають під класифікацію Стоунбрейкера і Коупленда. До них можна віднести мультипроцесорну систему серії MVS-100/1000 [581], мультипроцесорну систему SP-2 [582] компанії IBM, комп'ютери на основі технології Server Net компанії Tandem [583], гібридну архітектуру CDN [584].

За [585] першим кроком на шляху створення сучасних МБД була презентація 2000 року технології InfiniBand – високошвидкісної комутованої комп'ютерної мережі компанії Voltaire (партнер Oracle, починаючи з 2001 року), яка була використана в Oracle RAC (Real Application Cluster), починаючи з версії Oracle Database 9i. 2009 року серед Top 500 суперкомп'ютерів світу 29% використовували InfiniBand. Oracle Exadata VI була першою сучасною МБД, створеною Oracle HP (Hewlett-Packard) 2008 року. Тестування цієї МБД у CERN [586] показало високу ефективність за часом і пам'яттю в процесі розвантаження даних великого обсягу. Sun і Oracle створили МБД Exadata Database Mashine Version 2. Завдяки застосуванню сучасних технологій цих двох компаній МБД працює вдвічі ефективніше за Oracle Exadata VI.

Teradata Database – це система масової паралельної обробки (MPP) із колективною розподіленою архітектурою. Задача рівномірно розподіляється по всіх процесорах і паралельно обробляється. Підтримує архітектуру без спільного використання ресурсів. Має високу горизонтальну масштабованість, а також один із найрозвинутіших оптимізаторів на ринку. Автоматично рівномірно розподіляє дані по дисках. Підтримує стандарт SQL.

Бази даних, що підтримують роботу з масивами. БД масивів дає змогу представляти і маніпулювати багатомірними масивами однорідних даних. Вважається, що попередником БД масивів була створена 1982 року PICDMS [587] – СУБД для роботи з рисунками. Вона уможливило оперування двомірними масивами з допомогою процедурної мови.

1993 року Майєр і Венс [588] констатували, що технологія баз даних вкрай рідко використовується в наукових додат-

ках тому, що СУБД не підтримують структури з упорядкованими даними, зокрема, такими, як масиви. Ця заява співпала з початком активного розвитку досліджень і розробок із БД масивів.

Значний внесок у розвиток теорії і практики СУБД масивів зробив німецький учений Пітер Бауманн (Peter Baumann). Він перший 1994 року запропонував декларативну мову запитів для роботи з багатомірними масивами, які базуються на пропонованій ним же алгебрі багатомірних масивів [589, 590]. Розроблені алгебра і мова запитів стали основою створення 1996 року під його керівництвом першої СУБД масивів RasDaMan [591], яка підтримувала реляційну модель даних із додатковим типом даних «багатомірний масив» і спеціальною мовою запитів RASQL, що базується на SQL. Згідно з даними [592] обсяг даних на всіх пристроях RasDaMan наближається до петабайту.



Пітер Бауманн

Моделі і мови. Було запропоновано численні формальні моделі й мови баз даних масивів, аналіз яких можна знайти в [593, 594]. Наведемо деякі.

Алгебра карт (Map algebra) [595, 596] – алгебра, що базується на множинах, розроблена на початку 80-х років Даною Томлін (Dana Tomlin) для маніпулювання географічними даними. Представляє двомірні і тримірні растрові дані. В ній відбувається категоризація операцій над масивами залежно від кількості комірок вхідного масиву, які беруть участь у створенні комірки вихідного масиву.

AFATL Image Algebra [597] – це алгебра, розроблена для обробки зображень і отримання статистичної інформації.

AML (Array Manipulation Language) [598] – універсальна мова маніпулювання масивами, що базується на пропонованій авторами алгебрі багатовимірних масивів. Характерною рисою AML є поняття бітових шаблонів і шаблонно-орієнтованих функцій.

AQL (Array Query Language) [599, 600] – ця мова вмонтовує підтримку багатовимірних масивів у мову NCRA, яка є розширенням мови вкладеного реляційного обчислення NRC.

Array Algebra [589, 590] – пропонується алгебраїчна модель масиву, яка базується на трьох ортогональних примітивах, щодо яких надається набір допоміжних функцій. Цей набір обумовлюється використовуваною моделлю даних (об'єктивною або реляційною).

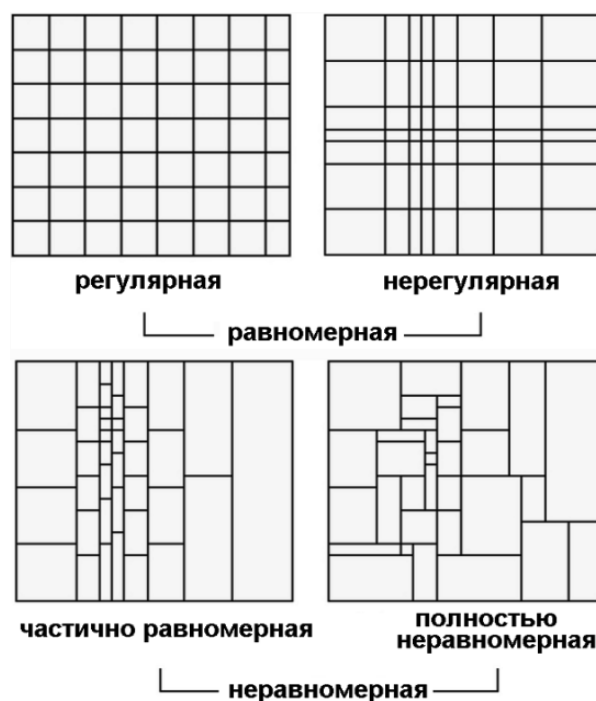
RAM [601, 602] – модель розроблена як розширення реляційної СУБД MonetDB [603].

Зберігання масивів. Зазвичай великі багатомірні масиви розбиваються на підмасиви, що утворюють одиниці доступу до них. Таке розбиття дістало назву мозаїки (tiling), а елементи мозаїки – плитки (tile) [618]. Мозаїка складається з плиток, що не перетинаються, кожна плитка – багатомірний підмасив висхідного масиву. Виділяють два основні види мозаїки – рівномірна (aligned) і нерівномірна (nonaligned). Рівномірна мозаїка для n -мірного масиву означає, що вона формується гіперплощинами, які є ортогональними всім n -мірному простору і розбивають весь масив на «плитки». Якщо всі площини знаходяться на одній відстані, то така мозаїка називається регулярною рівномірною, в іншому випадку – нерегулярною. В нерівномірній мозаїці (nonaligned tiling) деякі плитки мають сторони, які не є продовженням сторін сусідніх плиток. У частково рівномірній (partially aligned) мозаїці плитки вирівняні принаймні по одному з вимірів, а в повністю нерівномірній (totally nonaligned) таких вимірів немає. На рисунку, взятому з [618], наводиться приклад графічної інтерпретації

цих чотирьох категорій мозаїки для двомірного простору.

Архітектура реалізації. Виділяються наступні варіанти архітектури реалізації систем БД масивів:

- повнофункціональні системи БД масивів, реалізовані з нуля (RasDaMan [591], SciDB [604], MonetDB/SciQL [605]);
- реалізовані у вигляді додаткових рівнів у існуючих СУБД (EXTASICID [606, 607];
- реалізовані у вигляді об'єктно-реляційних розширень (PostGIS Raster [608], Teradata Arrays [609], Oracle GeoRaster [610]).



Пропонувалося два способи «впровадження» масивів у реляційні БД:

- додавання масивів у вигляді нового типу стовпчика (Ras DaMan, Teradata, Oracle, PostGIS, Roster, ISOSQL);
- подання масивів у вигляді таблиці (SciQL і SciDB).

2007 року на симпозіумі з екстремально великих баз даних (XLDB) представники науки і промисловості дійшли висновку, що існуючі СУБД не спроможні маніпулювати обсягами даних, які з'являться в найближчому майбутньому. Була також підкреслена необхідність розробки СУБД нового покоління, що мають відповідати таким вимогам [611]:

- модель даних базується на багатовимірних масивах, а не на кортежах;
- модель зберігання базується на версійності, а не на оновленні значень;
- масштабованість до сотень петабайт і висока відмовостійкість;
- СУБД є вільно розповсюджуваним програмним забезпеченням.

Відповіддю на це звернення став запуск 2008 року міжнародного проєкту під керівництвом Майкла Стоунбрейкера із створення нової СУБД під назвою SciDB. 2010 року було випущено першу публічну версію SciDB [612]. Її архітектура заснована на модифікованому ядрі PostgreSQL. SciDB призначена для зберігання, обробки й аналізу надвеликих обсягів багатовимірних розподілених масивів наукових даних, масштабованих на тисячі серверів [613]. Зберігання даних організовано у вигляді багатовимірних вкладених масивів, для обробки яких розроблені мови AQL (Array Query Language) і AFL (Array Functional Language).

Інші системи БД масивів.

SciQL [605]. Мова запитів, що заснована на SQL і використовує масиви для наукових застосувань. Розширює колончаті СУБД MonetDB операторами над масивами [614, 615], тим самим дозволяючи MonetDB ефективно функціонувати як база даних масивів.

EXTASCID [606, 607]. Це повна і розширювана система для обробки наукових даних. Підтримує як масиви, так і реляційні дані. Створена на основі масивно-паралельної архітектури GLADE для агрегування даних.

PostGIS Raster [608] (раніше відома як WKT Raster) дає змогу підтримувати растрові дані в системі PostGIS. Це забезпечується визначенням нового типу даних RASTER і додаткового набору SQL-функцій, котрі працюють із векторними і растровими даними.

Oracle GeoRaster [610] – це вбудована в Oracle Spatial можливість збереження, індексування, аналізу й доставки растрових зображень (як-от супутникових знімків), даних типу grid-даних, а також зв'язаних з ними метаданих. Ці типи даних можна використовувати для зберігання багатовимірних grid-шарів і електронних зо-

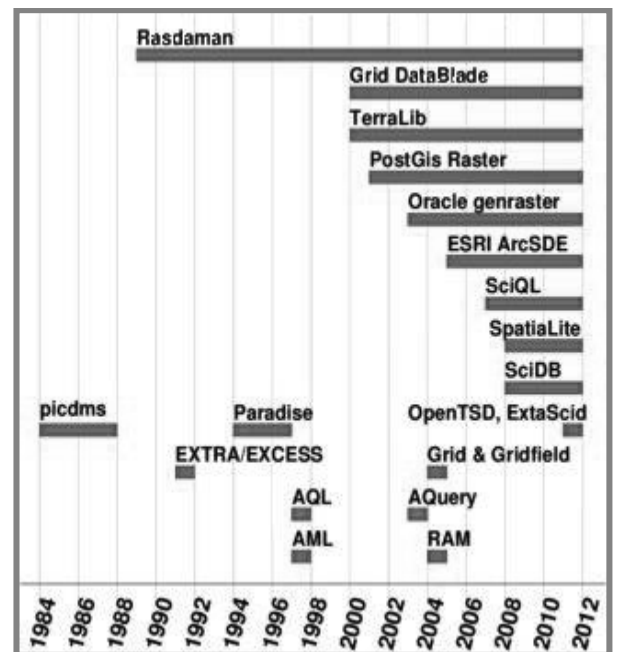
бражень, які можуть бути прив'язані для позиціонування на поверхні Землі або в локальній системі координат.

Teradata Arrays [609]. Нещодавно Teradata ввела в свою СУБД масиви у вигляді самостійного типу даних.

Для ілюстрації історії розвитку систем БД масивів наведемо рисунок із [616].

У 2018 році до ISO SQL було включено підтримку багатовимірних масивів даних [617] у вигляді спеціального типу даних.

На завершення зазначимо, що Альянс із Дослідницькими Даними (RDA – Research Data Alliance) надав 2021 року вичерпний огляд баз даних масивів і пов'язаних з ними Технологій [594].



Статистичні бази даних.

Під статистичними базами даних (СБД) маються на увазі такі БД, які дають можливість отримувати, зберігати і обробляти агреговані дані, тобто дані, отримані з допомогою різних способів узагальнення, групування, класифікації.

Дослідження статистичних баз даних почалися у 1970-х роках, а найбільшого розвитку набули в 1980-х роках ще до появи OLAP і їхній розвиток триває й дотепер.

Статистичні моделі даних.

Статистичні дані абстрактніші за звичайні, а операції мають іншу семанти-

ку. В СБД аналіз здійснюється з використанням агрегованих даних, отриманих із необроблених. Зведені дані мають бути різних форм, які не підтримуються традиційними СУБД. Більше того, реляційна модель у чистому вигляді теж не підходить для обробки таких даних. Основна причина – багатовимірність статистичних даних. Тож для СУБД потрібні або нові структури даних і операції над ними, або ж необхідно розширювати реляційну модель даних, аби мати можливість представляти відношення над множинами й нові оператори до них [630]. У СБД визначено три типи моделей даних: графічні, табличні й багатовимірні. Подамо короткий опис статистичних моделей даних, про які докладну інформацію знаходимо в працях [619, 620]:

SUBJECT [621] – подана графічна модель системи SUBJECT;

SAM (Semantic Association Model) [622] – модель було розроблено для моделювання як наукових статистичних даних, так і бізнесорієнтованих даних;

GRASS (Graphical Approach for Statistical Summaries) [623] – є розширенням SUBJECT. Для представлення моделі використовується орієнтований, направлений, ациклічний граф;

CSM (Conceptual Statistical Model) [624] – використовуються дві різні, одна доповнює одну одну моделі даних для опису елементарних і зведених даних, а саме ER-модель Чена і перевизначену модель GRASS;

STORM (Statistical Object Representation Model) [625] – графічна модель, в якій логічне представлення відокремлене від фізичної структури статистичних таблиць;

MEFISTO [626] – функціональна модель даних, що базується на структурі з назвою «статистична сутність», а також на численних операціях, які складають алгебру маніпулювання даними цієї структури;

Розширена реляційна модель даних із включенням у реляційну алгебру додаткових статистичних операторів [627];

Темпоральна статистична модель даних [628].

Статистичні оператори (алгебри). У статтях про СБД пропонується

чимало підходів щодо визначення операторів, які б відповідали обраній структурі моделі. В праці [629] вводяться статистичні оператори, аналогічні реляційній алгебрі, але із семантикою, характерною для багатовимірних об'єктів. У подальшому вводиться поняття повноти за аналогією з реляційною повнотою і показується повнота запропонованої алгебри. В праці [630] пропонується також розширення реляційної моделі даних шляхом введення відношень над множинами й оператори до них. Ще одним прикладом алгебри, залежної від обраної статистичної моделі, є праця [631], де задіяна двомірна модель презентації статистичних даних. 1997 року в [632] було висунуто варіант розширення SQL функціональними можливостями OLAP для отримання підсумкових значень у багатовимірному просторі. У працях [633, 634] пропонуються багатовимірні моделі даних і оператори. Статті [626, 635] також присвячені операторам у СБД. У [636] представлена алгебра статистичних даних.

Метадані. Вважається, що статистичні дані мають два типи атрибутів [637]: вільні атрибути, що являють собою результати застосування до висхідних даних агрегуючих функцій, і дескриптивні атрибути, які ці вільні дані описують, що також називаються метаданими. Правильно організовані, класифіковані й описані метадані дуже корисні для розуміння суті зведених даних. У зв'язку з цим ефективно використання метаданих є вкрай важливим у СБД. Додаткову інформацію про це можна отримати в працях [638 - 642].

Системи й мови запитів статистичних баз даних. В огляді [643] дано всебічний аналіз систем і мов запитів статистичних баз даних на основі таксономії, запропонованій в [644]. Перерахуємо їх, посилаючи зацікавлених читачів до вказаних статей для детального ознайомлення.

1) Статистичні системи управління базами даних (ССУБД), створені на основі традиційних СУБД. Більшість ССУБД даної категорії створені на основі реляційних СУБД. До них належать:

STRAND [645] базується на ER-моделі Чена, є похідним від CABLE [646]

і має за основу реляційну СУБД INGRES. Запити STRAND транслюються в мову QUEL і виконуються в INGRES.

HSDB [647] являє собою ССУБД, створену на основі реляційної системи Model 204 [648]. HSDB підтримує зведені таблиці й надає обмежений набір операцій над ними. Здатна виконувати процедури статистичного аналізу над реляційними і зведеними таблицями.

Розширена РМД [649]. Розширюється модель Кодда з метою представлення статистичних даних через запровадження «статистичної реляційної таблиці». Для неї розширюються реляційні операції і вводяться нові статистичні оператори. Пропонується мова запитів, що має подібні риси з QBE.

SYSTEM/K [650]. Об'єктно-орієнтована система управління базами знань створена на основі системи SQL/DS. Має широкі можливості з управління метаданими й обмежений перелік статистичних функцій.

GRAFSTAT [651]. Прикладна система призначена для аналізу даних із допомогою функцій прикладної статистики й графічного представлення результатів. Має інтерфейс із DB2 і SQL/DS через SQL.

SUBYL [652], PASTE [653], GPI [654], PEPIN-SICLA [655] є прикладами систем, що використовують традиційну СУБД, статистичний пакет і графічний пакет для створення системи управління статистичними даними.

2) Самостійно розроблені ССУБД.

Вони мають шість підкатегорій відповідно до використовуваної моделі й мови запитів:

- Системи на базі реляційної моделі й реляційних мов запитів. Вони пропонують власні методи фізичної організації даних, засоби концептуального моделювання, придатні для ССУБД, а також можливості використання агрегуючих функцій у мовах запитів. До них належать:

RAPID [656] і CAS SBD [657] – використовують реляційну алгебру;

ABE [658] – використовує реляційне обчислення;

SIR/SQL [659], GENISYS [660], CANTOR [661] - використовують SQL.

У [662] представлено мову запитів статистичної обробки неповної інформації.

У системі July [663] використовується універсальний реляційний інтерфейс для інтерпретації статистичних запитів.

У статті [664] описана статистична модель даних та її застосування в СБД.

- Системи на базі ієрархічної і мережевої моделей. Прикладами є: SIR/DBMS [665], TPL і TPLDCS [666], BROWSE [667].

- Формальні розширення реляційної моделі: ABE [658], SSDL [668], SSDB [669].

- ССУБД і мови з графічним зовнішнім інтерфейсом. Системи даної категорії мають графічні двомірні або табличні мови запитів. Прикладами є: SUBJECT [621], GRASS [623], ABE [658], GUIDE [670], STBE [671], ALDS [672], GRASP [673].

- Природномовний інтерфейс користувача: LIDS 86 [674].

- Мови запитів, що обчислюють агреговану інформацію з темпоральних даних. Приклади: TQUEL [675], HQUEL [676], TBE [677], TEER [678], розширена реляційна алгебра Тансела [679].

Підсумовуючи, відзначимо, що на підставі аналізу літератури можна стверджувати, що в розвиток дисципліни «статистичні бази даних» (принаймні на початковому етапі) значний внесок зробили турецько-американські вчені Зехра Мерал Озсойоглу (Zehra Meral Ozsoyoglu) і Гюльтекін Озсойоглу (Gultekin Ozsoyoglu).

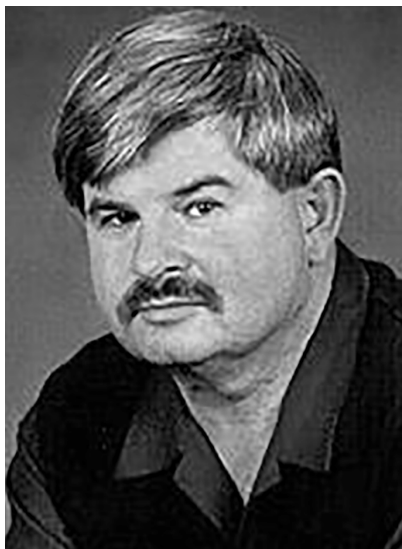


Мерал Озсойоглу

Мерал Озсойоглу спеціалізується на базах даних. 2011 року вона отримала звання «Дійсний член ACM» (ACM Fellow) за «великий внесок у системи управління базами даних». А 2018 року отримала премію ACM SIGMOD Contributions Award за «віддане служіння співтовариству баз даних». У нагороді згадується її діяльність як головного редактора ACM Transactions on Database Systems і Proceedings of the VLDB Endowment, а також як голови програмного комітету конференції VLDB та Симпозіуму з принципів систем баз даних.

Сховища даних. Завдання збору інформації з різних джерел не є новим. Наприкінці минулого сторіччя поширилася концепція створення сховищ даних (Data Warehouse).

Термін «сховище даних» (DWH – Data Warehouse) уперше з'явився на початку 1980-х років, коли дослідники IBM Пол Мерфі (Paul Murphy) й Баррі Девлін (Barry Devlin) розробили сховище бізнес-даних. Проте основоположником сховищ даних вважається Вільям Х. Інмон (William H. Inmon), який почав дослідження в цій царині 1983 року, а власне концепція сховищ даних була викладена ним 1990 року в монографії [682]. Ця монографія стала своєрідною біблією сховищ даних і започаткувала розвиток їх індустрії. Інмон організував першу конференцію, вперше створив тематичний розділ у журналі.



Вільям Інмон

Ще одним фундатором справедливо вважається Ральф Кімбелл (Ralph Kimball). Він є одним із перших архітекторів сховищ даних, і його методологія (відома також як просторове моделювання або методологія Кімбелла) стала фактичним стандартом у системі підтримки прийняття рішень. 1990 року компанія Red Brick System, заснована Кімбеллом, розробила Red Brick Warehouse – компактну стандартну реляційну базу даних на основі SQL для додатків DWH і бізнес-аналітики. Його монографія [683] є бестселером і нині.



Ральф Кімбелл

Попри те, що ці вчені часом притримувалися протилежних поглядів на DWH, вони заснували і суттєво збагатили науку DWH. Порівняння поглядів на DWH цих двох учених наведено в [684] й багатьох інших працях дослідників їхнього творчого доробку.

Дослідниками – новаторами у DWH, які опублікували свої монографії в середині 90-х років, є також Брекетт [685], Гілл і Рао [686], По [687].

За Інмоном сховище даних – це предметно орієнтована, інтегрована, хронологічно послідовна і незмінна (постійна) колекція даних, створена для підтримки процесу прийняття рішень керівництвом [682]. Також вважається, що в широкому сенсі сховище даних – це сукупність технологій, котрі дозволяють керівництву приймати рішення швидше і якісніше, а тому вони є складовими автоматизованих систем підтримки прийняття рішень.

DWH передбачають інтеграцію гетерогенних (неоднорідних) БД. Однак тут є відмінність від традиційного підходу. Останній передбачає створення оболонок та посередників, що приводять стандартні запити до вигляду, який сприймається кожною з інтегрованих БД. Натомість у DWH інформація з численних гетерогенних джерел попередньо перетворюється, інтегрується і зберігається в єдиному сховищі даних.

Основним завданням традиційних БД, що отримали назву операційних, є ефективне виконання транзакцій із урахуванням активного оновлення БД, аби підтримувати її цілісність. Ці БД належать до класу систем оперативної обробки транзакцій (online transaction processing OLTP). З іншого боку, системи DWH не передбачають динамічного оновлення. Для них немає проблеми підтримання цілісності й призначені вони для користувачів, які здійснюють аналіз даних для прийняття рішень. Системи такого класу отримали назву систем оперативної аналітичної обробки (online analytical processing – OLAP).

Термін OLAP було введено Едгаром Коддом у його публікації в журналі Computerworld 1993 року [688]. В цій статті він визначив OLAP як засіб динамічного аналізу, синтезу й консолідації великих обсягів багатовимірних даних, сформулював концептуальні положення OLAP, описав архітектуру, виділив фундаментальні компоненти й за аналогією до 12 правил для реляційних баз даних запропонував 12 принципів аналітичної обробки.

На початку 1995 року Найджел Пендс (Nigel Pendse), не згідний із критеріями Кодда, запропонував альтернативні 5 правил приналежності систем до категорії OLAP [689]. Вони дістали назву «тести FASMI» - аббревіатура від перших букв слів фрази “Fast Analysis of Shared Multidimensional Information” (швидкий аналіз спільно використовуваної багатовимірної інформації). Це визначення є також доволі популярним серед спеціалістів OLAP.



Найджел Пендс

Зрештою, рада OLAP (OLAP Council), створена 1995 року, дала таке розгорнуте визначення OLAP:

«Оперативна аналітична обробка (OLAP) – це категорія програмних технологій, які дозволяють аналітикам, менеджерам і керівникам мати уявлення про дані за рахунок швидкого, погодженого, інтерактивного доступу до представленої в різному вигляді інформації, перетвореної з вихідних даних, щоб вони таким чином усвідомлювали реальний стан речей на підприємстві» [690].

Архітектура DWH. Пропонувалося чимало різних архітектурних рішень DWH [691-702], кожне з яких має свої специфічні особливості. В праці [703] здійснено аналіз 73 архітектур DWH. На базі пропозицій, висловлених у [691, 692] подано узагальнену архітектуру DWH. DWH має трирівневу архітектуру.

Нижчий рівень являє собою БД DWH. Вона підтримує обрану модель даних DWH і надає засоби ведення цієї БД.

Середній рівень виконує функції OLAP. Зазвичай він представлений наступними чотирма типами [693, 694]:

Реляційний OLAP (ROLAP) – розширена реляційна СУБД, яка відображає операції багатовимірної моделі даних у стандартні операції реляційної алгебри.

Багатовимірний OLAP (MOLAP) – СУБД, яка безпосередньо підтримує багатовимірну модель даних та її операції.

Гібридний OLAP (HOLAP) поєднує в собі якості попередніх двох видів.

Спеціалізований SQL – сервер має розвинуті можливості мови запитів SQL

для роботи з DWH-схемами (зірка, сніжинка, сузір'я фактів) у режимі лише читання.

Зовнішній рівень містить інструментальні засоби підтримки прикладних завдань DWH, включно з:

- бізнес аналітикою (business intelligence),
- оперативною аналітичною обробкою (OLAP),
- інтелектуальним аналізом даних (data mining),
- системами підтримки прийняття рішень (decision support systems),
- мовами запитів і створення звітів.

Окрім цих трьох рівнів архітектури DWH включає:

- Репозиторій метаданих, який містить інформацію про дані DWH;
- Вітрини даних (data marts), що містять підмножину корпоративних даних, цікавих для певної групи користувачів.
- Засоби управління і контролю.
- Інструментальні засоби завантаження даних із зовнішніх джерел (бази даних, файли, електронні таблиці тощо) у БД DWH.

Ця компонента дістала назву ETL (Extract, Transform, Load). Вона виконує функції одержання даних із джерел, їх перевірки і очистки, перетворення до належного вигляду, інтеграції й завантаження або оновлення БД DWH [704]. Концепція ETL виникла в 1970-х роках у зв'язку з використанням централізованих репозиторіїв даних. Але лише в кінці 1980-х і початку 1990-х років вона набула великої популярності в зв'язку з появою DWH.

Моделі DWH. Із архітектурної точки зору виділяють наступні три типи моделей DWH [705]:

- корпоративне сховище із консолідованими даними, взятими з кількох операційних джерел – це DWH усієї корпорації [706];

- вітрина даних – містить підмножини корпоративних даних;

- віртуальне сховище – це множинність поглядів (views) операційних БД [707, 708].

Також існує точка зору [709], що архітектура DWH включає: архітектуру моделі даних, процесну архітектуру, ін-

формаційну архітектуру, технологічну та ресурсну архітектуру.

Вітрина даних (data mart). Концепція вітрин даних була запропонована Forrester Research ще 1991 року. Це предметно спрямована база даних, яка зазвичай містить дані одного з напрямків діяльності компанії. Вона орієнтована на користувачів однієї робочої групи або департаменту. У вітрині інформація зберігається оптимізовано з точки зору вирішення конкретних задач.

Існує три типи вітрин даних, які відрізняються залежно від їх відношення до сховища даних.

Залежні вітрини даних – це сегменти в корпоративному сховищі даних. Цей низхідний підхід починається зі збереження всіх бізнес даних в одному центральному місці. Новостворені вітрини даних отримують певну підмножину первинних даних щоразу за потреби аналізу.

Незалежні вітрини даних діють як автономна система, яка не покладається на сховище даних. Аналітики можуть діставати дані щодо конкретного предмету або бізнес-процесу із внутрішніх або зовнішніх джерел даних, обробляти їх, а відтак зберігати в репозиторії вітрини даних доти, доки вони знадобляться групі.

Гібридні вітрини даних об'єднують дані з існуючих сховищ та з інших операційних джерел. Цей уніфікований підхід використовує швидкість і зручний інтерфейс низхідного підходу, а також пропонує інтеграцію незалежного методу на рівні підприємства.

Ідея об'єднати дві концепції – сховищ даних і вітрин даних, очевидно, належить Марку Демаресту (Marc Demarest) [710], який 1994 року запропонував об'єднати дві концепції та використати сховище даних як єдине інтегроване джерело даних для вітрин даних.

Для взаємодії між собою вітрини даних можуть об'єднуватися в мережу, тим самим створюючи віртуальне сховище даних.

Багатовимірна модель даних DWH. Куб даних. Було запропоновано безліч багатовимірних моделей даних. Їх класифікація, аналіз і порівняння наведені в праці [711]. Коротко опишемо одну з них, яка використовується найбільше, а саме, куб даних [712].

Куб даних передбачає моделювання і подання даних, використовуючи поняття багатовимірного простору. Куб даних визначається через поняття «факт» і «вимір».

Згідно [713] терміни «факт» і «вимір» виникли в кінці 1960-х років у результаті виконання спільного дослідницького проєкту корпорації General Mills і Дартмутського університету. В 1970-х роках маркетингові компанії AC Nielsen і IRI постійно вживали ці терміни для опису своїх агрегованих даних і прагнули використати просторові моделі для презентації аналітичної інформації.

Вимір (dimension) – це характеристика, відносно якої представлено дані, що агрегуються. Використовуючи n вимірів, отримуємо n – мірний куб. Вимір – це вісь куба.

Вимір може ділитися на підвиміри. Приміром, вимір «країна» - на під виміри «області», а області – на «міста» тощо, таким чином утворюючи ієрархічну структуру виміру.

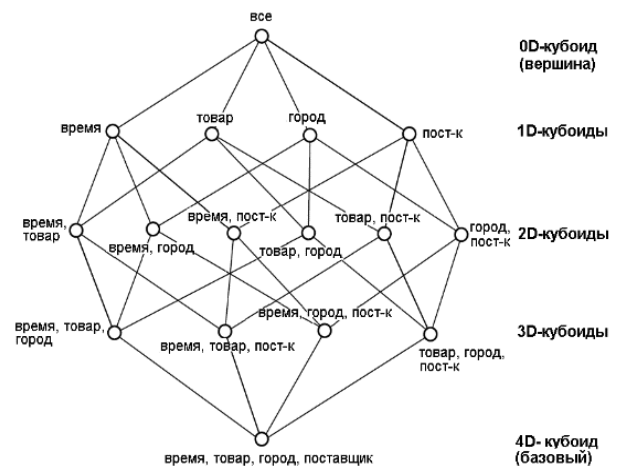
Факт – це характеристика, відносно якої представлені агреговані дані. Факт може мати властивості (атрибути).

Міра (measures) – це власне агреговані значення. Міри знаходяться в комірках кубу.

Багатовимірна модель має графічне зображення, що отримало назву кубоїда (cuboid) [714]. Кубоїд із найнижчим рівнем агрегованих даних (по всіх вимірах), називається базовим кубоїдом. На рисунку нижче подано трирівневий базовий кубоїд.

| | | | | | | | | |
|-----------------|----------|---------|------------|----|-----|-----|------|-----|
| Город | Чикаго | 854 | 882 | 89 | 623 | | | |
| | Нью-Йорк | 1087 | 968 | 38 | 872 | | | |
| | Торонто | 818 | 746 | 43 | 591 | | | |
| | Ванкувер | | | | | | | |
| Время (квартал) | Q1 | 605 | 825 | 14 | 400 | 682 | 925 | 698 |
| | Q2 | 680 | 952 | 31 | 512 | 728 | 1002 | 789 |
| | Q3 | 812 | 1023 | 30 | 501 | 784 | 984 | 870 |
| | Q4 | 927 | 1038 | 38 | 580 | | | |
| | | Монитор | Клавиатура | | | | | |
| | | Память | Процессор | | | | | |
| | | Товар | | | | | | |

Із цього трирівневого кубоїда можна отримати три двомірні кубоїди шляхом агрегування даних по кожному з трьох вимірів. Із них – три одномірних і, зрештою, один нуль мірний кубоїд (кубоїд – вершина). Тобто одне агреговане значення всіх мір вихідного кубоїда. Така структура є решіткою кубоїдів (cuboids lattice) [715] і називається кубом. На рисунку нижче наводиться куб (решітка кубоїдів) для чотиривимірного базового кубоїда з вимірами: товар, час, місто, постачальник. Іноді кубоїди називають також кубами чи підкубами.



Операції над OLAP – кубами. Існує п'ять основних операцій над кубами OLAP:

Згортання (roll-up), що також називається узагальненням (drill-up). Приводить до агрегування куба даних чи-то переміщенням вгору по ієрархічній структурі виміру (перехід від окремого поняття до загальнішого), чи-то видаленням виміру шляхом агрегування всіх мір цього виміру.

Розгортання (roll-down), що також має назву деталізація (drill-down). Операція протилежна згортанню/узагальненню – перехід від узагальнених даних до детальніших через переміщення вниз по ієрархічній структурі виміру чи введення нових вимірів.

Зріз (slice) – добування з куба підмножини комірок, пов'язаних із якимось певним значенням одного з його вимірів. Тобто отримуємо куб, де один з вимірів містить одне значення. Жодна агрегація мір не відбувається.

Фрагментація (dice) – є узагальненням зрізу. Із куба добувається підкуб, що має лише ті значення кожного з вимірів, які вказані в операції. Жодна агрегація мір не відбувається.

Обертання (pivot) – дозволяє змінювати просторову орієнтацію осей вимірів куба, обираючи найзручніше для аналітика представлення. В OLAP-технологіях куб – це передусім засіб візуалізації багатовимірних даних. Тому, використовуючи його, необхідно вирішувати задачу відображення інформації у зручному й інтерпретованому для людини вигляді.

Крім того, були запропоновані наступні додаткові операції.

Об'єднання (drill across) – дозволяє об'єднувати багато кубів, які мають один або більше спільних вимірів.

Проникнення (drill through) – дозволяє переходити від даних на нижньому рівні куба (базовий куб) до вихідних даних, звідки куб було добуто. Операція зазвичай використовується для визначення причини «викидів» у кубі даних.

Агрегуючі функції. Невід'ємна частина OLAP-моделі – завдання функцій агрегування. Оскільки мета OLAP – створення багаторівневої моделі аналізу, дані на всіх рівнях включно з базовим мають бути відповідно агреговані. По кожному виміру можливо задавати власну (і не одну) функцію агрегації.

Такі функції включають: функції агрегування, статистичні функції, функції ранжування Top N, Bottom N тощо. В [716] наведено класифікацію агрегуючих функцій з точки зору складності розпаралелювання.

У зв'язку із складністю структури куба опубліковано багато статей щодо його ефективної реалізації. Огляд досліджень у цій царині наведено в працях [715, 717]. Крім того, вичерпний огляд із реалізації ROLAP-кубів наведено в [718].

На закінчення відзначимо, що куб даних використовується не лише для представлення багатовимірних даних, а й інших складних типів даних. Серед них просторові, темпоральні, текстові, мультимедійні, мережеві і графічні [719, 720].

Багатовимірні бази даних (ББД) – це різновид БД, що створюється для схо-

вищ даних і оперативної аналітичної обробки даних (OLAP). OLAP, що працюють з ББД, називаються багатомірними OLAP (MOLAP). Як правило, ББД використовують модель багатовимірних кубів для представлення висхідних даних. У [721] стверджується, що математичний апарат багатовимірних БД було розроблено видатним американським математиком Доном Нельсоном (Don Nelson) у 60-х роках на замовлення міністерства оборони США.

Концептуальні схеми DWH. За аналогією із ER-схемою концептуальної моделі ПЗ, прийнятою в традиційній технології проектування реляційних OLTP-баз даних, у технології проектування DWH було запропоновано наступні OLAP-схеми: зірка, сніжинка й сузір'я фактів [714].

Схема зірки. Найпопулярніша схема, яка містить:

- одну велику центральну таблицю фактів із даними по всіх мірах;

- безліч невеликих за розміром таблиць вимірів, по одній на кожен вимір. Ця таблиця містить відомості (атрибути) виміру. Графічне зображення цієї схеми нагадує зірку, де таблиці вимірів знаходяться радіально навколо таблиці фактів.

Схема сніжинки. Є узагальненням схеми зірки. В даному випадку, якщо таблиця вимірів містить багато «різнопланових» атрибутів (приміром, вона містить атрибути не лише країни, а й міст), така таблиця нормалізується, тобто розбивається на кілька «додаткових» таблиць (таблиць підвимірів). Граф результуючої схеми нагадує сніжинку.

Сузір'я фактів. Припускає існування багатьох таблиць фактів, що мають спільні таблиці вимірів. Графічно ця схема представлена безліччю зв'язаних схем зірок.

Враховуючи складність процесу концептуального моделювання DWH, було здійснено чимало досліджень із питання оцінки якості цього процесу, огляд яких подається в [42].

Методології проектування. Пропонуються такі три методології проектування.

Проектування знизу – вгору. Ця методологія висунута Кімбеллом і передбачає попереднє проектування вітрин даних із конкретним тематичним напрямком. Остан-

ні представлені самостійними продуктами із наступним їх обслуговуванням в DWH.

Проектування зверху – вниз. Запропонована Інмоном методологія передбачає спершу створення централізованого репозиторію DWH із використанням «нормалізованої» моделі даних ПЗ. Далі на основі DWH створюються вітрини даних для конкретних додатків або підрозділів підприємства.

Гібридне проектування. Передбачає поєднання попередніх підходів і забезпечує всебічне й надійне проектування.

Інструментальні засоби. Було зроблено багато інструментальних засобів DWH. За адресою <https://www.guru99.com/top-20-etl-database-warehousing-tools.html> наводиться стислий опис 26 найпопулярніших інструментальних систем класу DWH.

Активні DWH. На початку цього століття була висунута концепція активних DWH [723, 724] для того, щоб DWH підтримували автоматичне прийняття рішень. В активних DWH розширюється технологія, яка є основою активних БД. А саме, вводяться «правила аналізу», що імітують роботу аналітика під час прийняття рішення. Водночас з'явилися перші комерційні продукти DWH із обмеженими можливостями активних правил [725, 726].

DWH реального часу [727]. Ця концепція розрахована на те, що вихідні дані надходять до DWH одразу, як тільки вони були породжені їхнім джерелом і стають доступними для аналізу. Про такі системи говорять, що вони є DWH «з нульовою затримкою». Популярність даної концепції сприяла тому, що багато виробників, включно з IBM [728] та Oracle [729], почали виробництво DWH цього класу. Стислий аналіз досліджень із цього напрямку наведено в [730].

Еволюція DWH. DWH уможливають збереження й аналіз даних за значний проміжок часу. Через те, що реальний світ, відображений у DWH, змінюється, те саме має відбуватися в DWH. Кімбалл, ймовірно, був першим, хто звернув на це увагу 1996 року й запропонував низку рішень [731]. З легкої руки Кімбалла ця проблема дістала назви «виміри, що повільно змінюються» (Slowly Changing Dimensions –

SCD). Відтоді в цьому напрямку було здійснено чимало досліджень, стислий огляд деяких з них наведено в [732].

Темпоральні DWH. Темпоральні DWH містять такі ж структурні компоненти, що й традиційні DWH. А саме, виміри, ієрархії змін, факти й міри. Основною ж відмінністю є те, що в нетемпоральних DWH час може асоціюватися лише з фактами, які зазвичай представляють теперішній час (в термінах темпоральних БД). А в темпоральних DWH є можливість відслідковувати еволюцію вимірів, фактів і мір. Окрім того, темпоральні DWH, як і темпоральні БД, можуть бути бітемпоральними. Дослідження з темпоральних DWH охоплюють різні аспекти. Зокрема, темпоральні типи [733], концептуальне моделювання й проектування [734], логічне моделювання й запити [735, 736], затримка в отриманні вимірів [737], багатовимірна агрегація [738], коректна агрегація за наявності змін у даних і структурі [739], еволюція багатовимірних схем [740]. У праці [741] подається огляд темпоральних DWH.

Просторові DWH. Просторові DWH (Spatial DWH – SDWH) виникли в зв'язку з бурхливим розвитком додатків, що мають відношення до оперування просторовими даними і передовсім географічних інформаційних систем (geographic information system – GIS). SDWH – це такі DWH, які уможливають оперування просторовими об'єктами задля підтримки просторово-орієнтованої ділової активності й прийняття рішень.

У праці [742] вперше було введено поняття просторового OLAP (SOLAP), яке відображає застосування методів інтелектуального аналізу щодо обробки просторових даних. У працях [743, 744] було введено поняття просторових вимірів і запропоновано їх класифікацію. В статті [745] пропонується розширення концептуальної багатовимірної моделі просторовими вимірами, ієрархіями і мірами, а також введенням у модель топологічних зв'язків і операторів. Були досліджені способи представлення просторових мір для геометричних об'єктів із використанням системи координат [742, 743, 745, 746] і сукупності точок [744].

SOLAP застосовують переважно до дискретних просторових даних, однак чимало складних задач GID-аналізу передбачають використання безперервних просторових даних, що зазвичай називаються просторовими полями. Просторові поля, або просто поля, описують фізичні явища, які безперервно змінюються в просторі або в часі. Як-от температура і тиск повітря, узвишся землі, поширення буревію. Поля зазвичай представляються у вигляді функцій, які надають певні значення кожній точці простору. В зв'язку з цим здійснюються дослідження і розробки із створення польових DWH. Серед перших у цій царині праць була стаття [747], де пропонується куб даних із безперервними вимірами. В статті [748] також пропонується багатовимірною модель даних із безперервними вимірами і з набором операцій, котра може застосовуватися для OLAP-аналізу польових даних. У працях [749-751] подана модель і алгебра для роботи з просторово-часовими безперервними полями та їх використання для OLAP-аналізу просторових даних.

Було здійснено чимало інших досліджень із SDWH. Добрим вступом до просторових DWH є стаття [752]. У статті [753] подано аналітичний огляд фундаментальних методів і концепцій, що становлять основу просторових DWH.

SQL і OLAP. 1995 року група дослідників на чолі з Джеймсом Греєм запропонувала розширення мови SQL – фразу CUBE BY, завдання якої – створення OLAP-кубів [754]. CUBE BY створює групування за всіма можливими комбінаціями вказаних у ньому вимірів, із різними рівнями агрегації даних. Ця ідея була прийнята в SQL:1999.

В SQL:1999 з'явилися можливості роботи з OLAP-кубами. Для цього фраза GROUP BY була розширена фразами ROLLUP, CUBE і GROUPING SETS, а також додана функція GROUPING.

Фраза ROLLUP створює умови для багаторівневого ієрархічного групування за вказаними в ній стовпчиками й створює проміжні суми (subtotals) у відповідності зі збільшуваним рівнем агрегації. Від найдеталізованіших рівнів представлення даних до більш узагальнених сум.

Фраза CUBE дозволяє в одній команді вирахувати всі можливі комбінації проміжних сум. Висловлюючись термінами решітки кубів, вказані в цій фразі стовпчики формують базову таблицю, й для неї створюється решітка. Фраза CUBE здатна генерувати інформацію, необхідну для перехресних звітів (cross – tabulation reports), в одному запиті.

Фраза GROUPING SETS формує результати угруповань за вказаними в ній стовпчиками й об'єднує їх в одну таблицю. Інакше кажучи, вона еквівалентна конструкції UNION ALL до вказаних груп.

Функція GROUPING повертає істину в разі, якщо вказаний вислів є статистичним (тобто має підсумкове значення), і – неправду, якщо вислів нестатистичний.

References

439. Rothnie J.B. Jr, Bernstein P.A., Fox S., Goodman N., Hammer M., Landers T.A., Reeve C.L., Shipman D.W., Wong E. Introduction to a system for distributed databases (SDD-1). ACM Trans. on Database Syst. 1980;5(1):1–17.
440. Bernstein P.A., Shipman D.W., Rothnie J.B. Concurrency Control in a System for Distributed Databases (SDD-1) ACM Transactions on Database Systems, Vol. 5, No. 1, March 1980, Pages 19-51.
441. Hammar M., Shipman D. Reliability mechanism for SDD-1: a system for distributed database. ACM Trans. Database Syst., 5 (4) (Dec. 1980), pp. 431-466
442. Bernstein P.A., Goodman N., Wong E., Reeve C.L., Rothnie J.B. Query Processing in a System for Distributed Databases (SDD-1). ACM Transactions on Database Systems, Vol. 6, No. 4, December 1961, Pages 602-625
443. Selinger P.G. An architectural overview of R*: a distributed database management system. In: Proceedings of the 5th Berkeley Workshop on Distributed Data Management and Computer Networks; 1981, p. 187.
444. Williams R., Daniels D., Haas L., Lapis G., Lindsay B., Ng P. Obermarck R., Selinger P., Walker A., Wilms P., Yost R. R*: An Overview of the Architecture. IBM Research Report RJ3325, IBM Research Laboratory, San Jose, CA, Dec. 1981.

445. Lohman G.M., Mohan C., Haas L.M., Daniels D., Lindsay B.G., Selinger P.G., Wilms P.F. Query Processing in R*. Lohman G.M. et al. (1985) Query Processing in R*. In: Kim W., Reiner D.S., Batory D.S. (eds) Query Processing in Database Systems. Topics in Information Systems. Springer, Berlin, Heidelberg, pp. 31-47
446. Daniels D. et al. An Introduction to Distributed Query Compilation in R*. Distributed Data Bases (ed. H.-J. Schneider): Proc. 2nd Int. Symposium on Distributed Data Bases. - New York, N.Y.: North-Holland, 1982.
447. Stonebraker M.R., Neuhold E.J. A Distributed Data Base Version of INGRES // Proc. 2nd Berkley Conf. On Distributed Data Management and Computer Networks. — Lawrence Berkley Laboratory, May 1977.
448. Epstein R., Stonebraker M., Wong E. Distributed Query Processing in a Relational Database System // Proc. 1978 ACM SIGMOD Int. Conf. on Management of Data. — Austin, Tex. — May-June 1978.
449. Stonebraker M. The design and implementation of distributed INGRES. The INGRES Papers, Reading; 1986, p. 187–196.
450. Adiba M.E., Andrade J.M., Fernandez F., Gia Toan Nguyen. POLIPHEME: An experience in distributed database system design and implementation. Proc. of Int. Symposium on Distributed Data Bases. Paris, France, 1980, pp. 475-479
451. Epstein R., Stonebraker M., Wong E. Distributed Query Processing in a Relation Data Base System. SIGMOD '78: Proceedings of the 1978 ACM SIGMOD international conference on management of data May 1978 Pages 169–180.
452. Davenport R.A. Distributed database technology — a survey. Computer Networks (1976), Volume 2, Issue 3, 1978, Pages 155-167,
453. Rothnie J.B., Goodman N. A Survey of Research and Development in Distributed Database Management. VLDB '77: Proceedings of the third international conference on Very large data bases - Volume 3 October 1977 Pages 48–62
454. Sheth A.P., Larson J.A. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Comput Surv. 1990;22(3):183–236.
455. Stonebraker M., Aoki P.M., Pfeffer A, Sah A, Sidell J, Staelin C, Yu A. Mariposa: a wide-area distributed database system. VLDB J. 1996;5(1):48–63.
456. Date C.J. What is a Distributed Database System? In: Date C. J. Relational Database Writings 1985-1989. — Reading, Mass.: Addison-Wesley, 1990.
457. Heimbigner D., McLeod D. “A Federated Architecture for information management”. ACM Transactions on Information Systems, 1985,.Volume 3, Issue 3. pp. 253–278.
458. Sheth A.P., Larson J.A. “Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases”. ACM Computing Surveys, 1990, Vol. 22, No.3. pp. 183–236.
459. Masood N., Eaglestone B. “Component and Federation Concept Models in a Federated Database System”. Malaysian Journal of Computer Science, 2003, 16 (2): 47–57.
460. Sheth A., Larson J.A. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. ACM Computing Surveys 1990, 22(3):183--236
461. Litwin W., Mark L., Roussopoulos N. Interoperability of multiple autonomous databases. ACM Comput. Surv. 1990; 22(3):267–293.
462. Wiederhold G. Mediators in the architecture of future information systems. IEEE Comput. 1992;25(3): 38–49.
463. Risch T., Josifovski V., Katchaounov T. (2004) Functional Data Integration in a Distributed Mediator System. In: Gray P.M.D., Kerschberg L., King P.J.H., Poulouvasilis A. (eds) The Functional Approach to Data Management. Springer, Berlin, Heidelberg. pp. 211-238
464. Gribble S.D., Halevy A.Y., Ives Z.G., Rodrigo M., Suci D. What Can Database Do for Peer-to-Peer? In Processing of Int'l Workshop on the WEB and Databases (WebDB), 2001, pp. 31-36
465. Bonifati A., Chrysanthis P.K., Ouksel A.M., Sattler K.-U. Distributed databases and peer-to-peer databases: Past and present, ACM SIGMOD Record, 2008, 37(1): 5-11
466. Beng Chin Ooi, Kian-Lee Tan, guest editors. Introduction: special section on peer-to-peer-based data management. IEEE Trans Knowl Data Eng. 2004;16(7):785–786.

467. Sacca D., Wiederhold G. Database partitioning in a cluster of processors. *ACM Trans Database Syst.* 1985;10(1):29–56.
468. Yoshida M., Mizumachi K., Wakino A., Oyake I., Matsushita Y. Time and cost evaluation schemes of multiple copies of data in distributed database systems. *IEEE Trans. Softw. Eng.* 1985; 11(9) :954–958.
469. Ceri S., Negri M., Pelagatti G. Horizontal data partitioning in database design. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*; 1982. p. 128–136.
470. Ceri S., Pelagatti G. *Distributed databases: principles and systems*. New York: McGraw-Hill; 1984.
471. Navathe S., Ceri S., Wiederhold G., Dou J. Vertical partitioning of algorithms for database design. *ACM Trans Database Syst.* 1984;9(4):680–710.
472. McCormick W.T., Schweitzer P.J., White T.W. Problem decomposition and data reorganization by a clustering technique. *Oper Res.* 1972;20(5): 993–1009
473. Shikha Mehta, Parul Agarwal, Praxhar Shrivastava, Jharna Barlawala, Differential bond energy algorithm for optimal vertical fragmentation of distributed databases, *Journal of King Saud University - Computer and Information Sciences*, 2018,
474. Chu W.W. Optimal file allocation in a multiple computer network. *IEEE Trans Comput.* 1969;- 18(10):885–889.
475. Apers P.M. Data allocation in distributed database systems. *ACM Trans Database Syst.* 1988;13(2): 263–304.
476. Bell D.A. Difficult data placement problems. *Comput J.* 1984;27(4):315–320.
477. Chang C.C, Shieh J.C. On the complexity of file allocation.problem. In: *Proceedings of the International.Conference on the Foundations of Data Organization*; 1985. p. 177–181.
478. Brunstrom A., Leutenegger S.T, Simha R. Experimental evaluation of dynamic data allocation strategies in a distributed database with changing workloads. In: *Proceedings of the 40) 4th International Conference on Information and Knowledge Management*; 1995. p.395–402.
479. Karlapalem K., Ng M.P. Query-driven data allocation algorithms for distributed database systems. In: *Proceedings of the 8th International Conference Database and Expert Systems Applications*; 1997. p. 347–356.
480. Bernstein P.A., Hadzilacos V., Goodman N. *Concurrency control and recovery in database systems*. Reading: Addison Wesley; 1987.
481. Gray J., Helland P., O’Neil P., Shasha D. The dangers of replication and a solution. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*; 1996. p. 173–182.
482. Breitbart Y., Komondoor R., Rastogi R., Seshadri S., Silberschatz A. Update propagation protocols for replicated databases. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*; 1999. p. 97–108.
483. Saito Y., Shapiro M. Optimistic replication. *ACM Comput Surv.* 2005;37(1):42–81.
484. Lin Y., Kemme B., Patiño-Martínez M., Jiménez-Peris R. Middleware based data replication providing snapshot isolation. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*; 2005. p. 419–430.
485. Wiesmann M., Schiper A. Comparison of database replication techniques based on total order broadcast. *IEEE Trans Knowl Data Eng.* 2005;17(4):551–566
486. Corbett J.C., Dean J., Epstein M., Fikes A., Frost C., Furman J.J., Ghemawat S., Gubarev A., Heiser C., Hochschild P., Hsieh W.C., Kanthak S., Kogan E., Li H., Lloyd A., Melnik S., Mwaura D., Nagle D., Quinlan S., Rao R., Rolig L., Saito Y., Szymaniak M., Taylor C., Wang R., Woodford D. Spanner: Google’s globally distributed database. *ACM Trans Comput Syst.* 2013;31(3):8:1-8:22.
487. Mahmoud H.A., Nawab F., Pucher A., Agrawal D., El Abbadi A. Low-latency multi-datacenter databases using replicated commit. *Proc VLDB Endow.* 2013;6(9):661–672.
488. Satyanarayanan M., Kistler J.J., Kumar P., Okasaki M.E., Siegel E.H., Steere D.C. Coda: a highly available file system for a distributed workstation environment. *IEEE Trans Comput.* 1990;39(4):447–459.
489. Terry D.B., Theimer M., Petersen K., Demers A.J., Spreitzer M., Hauser C. Managing update conflicts in Bayou, a weakly connect-

- ed replicated storage system. In: Proceedings of the 15th ACM Symposium on Operating System Principles; 1995. p. 172–183.
490. Sivasubramanian S., Szymaniak M., Pierre G., van Steen M. Replication for web hosting systems. *ACM Comput Surv.* 2004;36(3):291–334.
491. Lv Q., Cao P., Cohen E., Li K., Shenker S. Search and replication in unstructured peer-to-peer networks. In: Proceedings of the 16th Annual International Conference on Supercomputing; 2002. p. 84–95.
492. Budhiraja N, Marzullo K, Schneider FB, Toueg S. The primary-backup approach. In: Mullender S, editor. *Distributed systems*. 2nd ed. Reading: Addison Wesley; 1993. p. 199–216.
493. Schneider F.B. Replication management using the state-machine approach. In: Mullender S, editor. *Distributed systems*. 2nd ed. Reading: Addison Wesley; 1993. p. 169–198.
494. Almeida S., Leitão J., Rodrigues L.E.T. Chainreaction: a causal+ consistent datastore based on chain replication. In: Proceedings of the 8th ACM SIGOPS/EuroSys European Conference on Computer Systems; 2013. p. 85–98.
495. Sovran Y., Power R., Aguilera M.K., Li J. Transactional storage for geo-replicated systems. In: Proceedings of the 23rd ACM Symposium on Operating System Principles; 2011. p. 385–400.
496. Gray J. Notes on data base operating systems. In: *Advanced Course: Operating Systems*; 1978. p. 393–481.
497. Ho G.S, Ramamoorthy C.V. Protocols for deadlock detection in distributed database systems. *IEEE Trans Softw Eng.* 1982;8(6):554–557.
498. Stonebraker M. The design and implementation of distributed ingres. In: *The INGRES papers: anatomy of a relational database system*; 1986. p. 187–96.
499. Menascé D.A., Muntz R. Locking and deadlock detection in distributed data bases. *IEEE Trans Softw Eng.* 1997;5(3):195–202.
500. Mohan C., Lindsay., Bruce G., Obermarck R. Transaction management in the R* distributed database management system. *ACM Trans Database Syst.* 1986;11(4):378–396.
501. Abonamah A.A., Elmagarmid A. A survey of deadlock detection algorithms in distributed database systems. In: *Advances in distributed and parallel processing. System paradigms and methods*, vol. 1; 1994. p. 310–341.
502. Elmagarmid A.K. A survey of distributed deadlock algorithms. *ACM SIGMOD Rec.* 1986;15(3):37–45.
503. Knapp E. Deadlock detection in distributed databases. *ACM Comput Surv.* 1987;19(4): 303–328.
504. Singhal M. Deadlock detection in distributed systems. *Computer.* 1989;22(11):37–48.
505. Krivokapic N, Kemper A, Gudes E. Deadlock detection in distributed database systems: a new algorithm and a comparative performance analysis. *VLDB J.* 1999;8(2):79–100.
506. Roesler M., Burkhard W.A., Cooper K.B. Efficient deadlock resolution for lock-based concurrency control schemes. In: Proceedings of the 18th International Conference on Distributed Computing Systems; 1998. p. 224–233.
507. Bracha G., Sam T. Distributed deadlock detection. *Distrib Comput.* 1985;2(3):127–138.
508. Chandy K.M., Lamport L. Distributed snapshots: determining global states of distributed systems. *ACM Trans Comput Syst.* 1986;3(1):63–75.
509. Bernstein P.A., Goodman N., Wong E., Reeve C.L., Rothnie Jr.J.B. Query processing in a system for distributed databases (SDD-1). *ACM Trans Database Syst.* 1981;6(4):602–625.
510. Epstein RS, Stonebraker M, Wong E. Distributed query processing in a relational database system. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 1978. p. 169–180.
511. Stonebraker M. The design and implementation of distributed INGRES. In: Stonebraker M, editor. *The INGRES papers*. Reading: Addison-Wesley; 1986. 4) Williams R., Daniels D., Hass L., Lapis G., Lindsay B., Ng P., Obermarck R., Selinger P., Walker A., Wilms P., Yost R. R*: an overview of the architecture. IBM Research Lab, San Jose, Technical Report RJ3325; 1981.
512. Williams R., Daniels D., Hass L., Lapis G., Lindsay B., Ng P., Obermarck R., Selinger P., Walker A., Wilms P., Yost R. R*: an overview of the architecture. IBM Research Lab, San Jose, Technical Report RJ3325; 1981.
513. Haas L.M., Selinger P.G., Bertino E., Daniels D., Lindsay B.G., Lohman G.M., Masu-

- naga Y., Mohan C., Ng P., Wilms P.F., Yost R.A. R*: a research project on distributed relational DBMS. *IEEE Database Eng Bull.* 1982;5(4):28–32.
514. Wong E. Retrieving dispersed data from SDD-1: a system for distributed databases. In: *Proceedings of the 2nd Berkeley Workshop on Distributed Data Management and Computer Networks*; 1977. p. 217–235.
515. Yu C.T. and Chang C.C. Distributed query processing. *ACM Comput. Surv.*, 16(4):399–433, 1984.
516. Kossmann D. The state of the art in distributed query processing. *ACM Comput Surv.* 2000;32(4):422–469.
517. Urhan T., Franklin M.J., Amsaleg L. Cost based query scrambling for initial delays. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*; 1998. p. 130–141.
518. Stonebraker M., Devine R., Kornacker M., Litwin W., Pfeffer A., Sah A., Staelin C. An economic paradigm for query processing and data migration in Mariposa. In: *Proceedings of the 3rd International Conference Parallel and Distributed Information Systems*; 1994. p. 58–67.
519. Ceri S., Pelagatti G. *Distributed databases principles and systems*. New York: McGraw-Hill; 1984.
520. Eswaran K.P., Gray J.N., Lorie R.A., Traiger I.L. The notion of consistency and predicate locks in a database system. *Commun ACM.* 1976;19(11):624–633.
521. Gray J.N. The transaction concept: virtues and limitations. In: *Proceedings of the 7th International Conference on Very Data Bases*; 1981. p 144–154.
522. Spector A.Z., Schwarz P.M. Transactions: a construct for reliable distributed computing. *ACM Operat Syst Rev.* 1983;17(2):18–35
523. Stearns R.E., Rosenkrantz D.J. Distributed database concurrency controls using before-values. *SIGMOD '81: Proceedings of the 1981 ACM SIGMOD international conference on Management of data*, 1981, pp 74–83
524. Boral H., Gold I. Towards a self-adapting centralized concurrency control algorithm. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*; 1984. p. 18–32.
525. Lausen G. Concurrency control in database systems: a step towards the integration of optimistic methods and locking. In: *Proceedings of the ACM Annual Conference*; 1982. p. 64–68.
526. Salem K., Garcia-Molina H., Shands J. Altruistic locking. *ACM Trans Database Syst.* 1994;19(1):17–165.
527. Kung H.T. Robinson J.T. “On Optimistic Methods for Concurrency Control”. *ACM Transactions on Database Systems*, Vol. 6, No. 2, 1981, pp. 213–226.
528. Rahm E. Concepts for Optimistic Concurrency Control in Centralized and Distributed Database Systems. *IT Informationstechnik*, (in German), 1988, vol. 30, no. 1, pp. 28–47.
529. Thomasian A. Distributed optimistic concurrency control methods for high-performance transaction processing *IEEE Transactions on Knowledge and Data Engineering*, 1998, 10(1):173 - 189
530. Bernstein P., Goodman N. [1980] “Timestamp-Based Algorithms for Concurrency Control in Distributed Database Systems,” in *VLDB '80: Proceedings of the sixth international conference on Very Large Data Bases - Volume 6* October, 1980, pp. 285–300
531. Bernstein PA., Goodman N., Rothnie J.B. Jr., Papadimitriou C.H. Analysis of serializability of SDD1: a system of distributed databases (the fully redundant case). *IEEE Trans. On Software Engineering*, SE-4: 3 (1978), pp. 154–168.
532. Reed D.P. “Implementing Atomic Actions on Decentralized Data,” *TOCS*, 1:1, February 1983, pp. 3–23.
533. Reed D.P. Naming and synchronization in a decentralized computer system. Ph. D. Thesis, MIT, Cambridge, Mass., 1977
534. Thomasian A. Concurrency Control: Methods, Performance, and Analysis, *ACM Computing Surveys*, 1998, 30(1):70–119
535. Ozkarahan E. *Database Machines and Database Management*. Englewood Cliffs, N.J.; Prentice-Hall, 1986 - 636 p.
536. DeWitt D.J., Hawthorn P.B. A performance evaluation of data base machine architectures. In: *Proceedings of the 7th International Conference on Very Data Bases*; 1981. p. 199–214.
537. Boral H., DeWitt D.J., Wilkinson W.K. Performance evaluation of four associative disk

- designs Information Systems, Volume 7, Issue 1, 1982, Pages 53-64
538. Boral H., DeWitt D. Database machines: An idea whose time has passed? A critique of the future of database machines. In Proceedings of the 1983 Workshop on Database Machines. H.-O. Leilich and M. Missikoff, Eds., Springer-Verlag, 1983, pp. 16--187
 539. Slotnik D.L. "Logic per Track Devices" in Advances in Computers, Vol. 10., Frantz Alt, Ed., Academic Press, New York, 1970, pp. 291 - 296. TODS, Vol 1, No. 3. September 1976.
 540. Parker J.L. "A Logic per Track Retrieval System," IFIP Congress, 1971. J.L. Parker, "A Logic per Track Retrieval System", Proc. IFIP Congress 1971, pp. TA-4-146 to TA-4-150
 541. Minsky N., "Rotating Storage Devices as Partially Associative Memories" Proc. 1972 FJCC. N. Minsky: Rotating Storage Devices as Partially Associative Memories, FJCC 1972, AFIPS Conf. Proc., pp. 587-595
 542. Parhami B. "A Highly Parallel Computing System for Information Retrieval" Proceedings of the Fall Joint Computer Conference, 1972. pp. 681-690
 543. Ozkarahan E.A., Schuster S.A., Smith K.S. RAP: An Associative Processor for Data Base Management. Proc. AFIPS 44, NCC, 1975, pp. 379-387.
 544. Su S.Y.W., Lipovski G.J."CASSM: A Cellular System for Very Large Data Bases", VLDB '75: Proceedings of the 1st International Conference on Very Large Data Bases September 1975 Pages 456-472
 545. Lin S.C., Smith D.C.P., Smith J.M. "The Design of a Rotating Associative Memory for Relational Database Applications," TODS Vol. 1, No. 1, pages 53 - 75, Mar. 1976.
 546. Kannan K. "The Design of a Mass Memory for a Database Computer," Proc. Fifth Annual Symposium on Computer Architecture. Palo Alto, CA. April 1978, pp. 44-51
 547. Leilich H.-O., Stiege G., Zeidler H.Ch. "A Search Processor for Data Base Management Systems" VLDB '78: Proceedings of the fourth international conference on Very Large Data Bases - Volume 4, September 1978, Pages 280-287
 548. Schuster S.A., Nguyen, H.B., Ozkarahan, E.A. K.C. Smith, "RAP.2 - An Associative Processor for Databases and its Applications," IEEE Transactions on Computers, C-28, No. 6, June 1979. pp. 446-458
 549. DeWitt D.J., "DIRECT - A Multiprocessor Organization for Supporting Relational Database Management Systems," IEEE Transactions on Computers. June 1979, pp. 395-406.
 550. Madnick S.E. "The Infoplex Database Computer: Concepts and Directions," Proceedings of the IEEE Computer Conference, Feb. 1979, pp. 168-176
 551. Hell W. "RDBM - A Relational Database Machine: Architecture and Hardware Design," Proceedings of the 6th Workshop on Computer Architecture for Non-Numeric Processing, June 1981,
 552. Missikoff M. "An Overview of the project DBMAC for a relational machine," Proceedings of the 6th Workshop on Computer Architecture for Non-Numeric Processing, Hyeres, France, June 1981.
 553. DeWitt D.J., Gray J. Parallel database systems: the future of high performance database systems. Commun ACM. 1992;36(6):85-98.
 554. Hurson A.R., Miller L.L., Pakzad S.H., Eich M.H., Shirazi B. Parallel architectures for database systems. Advances in Computers, Vol. 28, 1989, pp. 107-151.
 555. Stonebraker M. "The Case for Shared Nothing," Database Engineering, Vol. 9, No. 1, 1986. pp. 4-9
 556. Stonebraker M., Katz, R.H., Patterson, D.A., Ousterhout, J.K., The Design of XPRS, Fourteenth Int. Conf. on Very Large Data Bases, (Los Angeles, 1988), Morgan Kaufmann, 1988, pp. 318-330.
 557. Bergsten B., Couprie, M., Lopez, M., DBS3: A Parallel Data Base System for Shared Store (Synopsis), in Issues, Architectures, and Algorithms (Proc. of the 2nd Int. Conf. on Parallel and Distributed Information Systems (PDIS 1993), San Diego, 1993), IEEE Comput. Soc., 1993, pp. 260-262.
 558. Graefe G., Volcano—An Extensible and Parallel Query Evaluation System, IEEE Trans. Knowledge Data Engineering, 1994, vol. 6, no. 1, pp. 120-135.
 559. Strickland J.P., Uhrowczik, P.P., Watts, V.L., IMS/VS: An Evolving System, IBM Systems J., 1982, vol. 21, no. 3, pp. 490-510.
 560. Linder B., Oracle Parallel RDBMS on Massively Parallel Systems, in Issues, Architec-

- tures, and Algorithms (Proc. of the 2nd Int. Conf. on Parallel and Distributed Information Systems (PDIS 1993), San Diego, 1993), IEEE Comput. Soc., 1993, pp. 67–68.
561. Dubova N., Supercomputers nCube, Otkrytye sistemy, 1995, no. 2, pp. 42–47.
562. Kronenberg N.P., Levy, H.M., Strecker, W.D., VAXclusters: A Closely-Coupled Distributed System, ACM Trans. Comput. Systems, 1986, vol. 4, no. 2, pp. 130–146.
563. Nick J.M., Moore B.B., Chung J.-Y., Bowen N.S., S/390 Cluster Technology: Parallel Sysplex, IBM Systems J., 1997, vol. 36, no. 2, pp. 172–201.
564. Teradata: DBC/1012 Data Base Computer Concepts & Facilities, Teradata Corp. Document No. C02-0001-00, 1983.
565. Dewitt D.J., Ghandeharizadeh S., Schneider D.A., Bricker A. Hsiao H.-I., Rasmussen R. “The Gamma Database Machine Project,” IEEE Knowledge and Data Engineering, Vol. 2, No. 1, March, 1990, pp. 44-62
566. Tandem Performance Group, “A Benchmark of Non-Stop SQL on the Debit Credit Transaction,” Proceedings of the 1988 SIGMOD Conference, Chicago, IL, June 1988.
567. Alexander W., Copeland G.P. Process And Dataflow Control In Distributed Data-Intensive Systems. Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, June 1-3, 1988. ACM Press, 1988. P. 90-98
568. Lorie R., Daudenarde J., Hallmark G., Stamos J., Young H., “Adding Intra-Transaction Parallelism to an Existing DBMS: Early Experience”, IEEE Data Engineering Newsletter, Vol. 12, No. 1, March 1989., pp. 2–8.
569. Gibbs J, “ Massively Parallel Systems, Rethinking Computing for Business and Science,” Oracle, 1991, Vol. 6, No.1
570. Engler, S., Glasstone R., Hasan W., Parallelism and Its Price: A Case Study of NonStop SQL/MP, ACM SIGMOD Record, 1995, vol. 24, no. 4, pp. 61–71.
571. Clay D. Informix Parallel Data Query (PDQ), in Issues, Architectures, and Algorithms (Proc. of the 2nd Int. Conf. on Parallel and Distributed Information Systems (PDIS 1993), San Diego, 1993), IEEE Comput. Soc., 1993, pp. 71–72.
572. Page J., A Study of a Parallel Database Machine and Its Performance: The NCR/Teradata DBC/1012. Advanced Database Systems, Lecture Notes in Computer Science (Proc. of the 10th British Natl. Conf. on Databases. BNCOD 10, Aberdeen, 1992), Springer, 1992, vol. 618, pp. 115–137.
573. Baru C.K. et al. DB2 Parallel Edition, IBM System J., 1995, vol. 34, no. 2, pp. 292–322.
574. Sokolinsky L.B. Survey of Architectures of Parallel Database Systems (Rus). Programming and Computer Software volume 30, No 6, pages 337–346 (2004)
575. Copeland G.P., Keller T., A Comparison of High-Availability Media Recovery Techniques, Proc. of the 1989 ACM SIGMOD Int. Conf. on Management of Data (Portland, 1989), ACM, 1989, pp. 98–109.
576. Graefe G., Query Evaluation Techniques for Large Databases, ACM Computing Surv., 1993, vol. 25, no. 2, pp. 73–169.
577. Hua K.A., Lee C., Peir J.-K., Interconnecting Shared-Everything Systems for Efficient Parallel Query Processing, Proc. First Int. Conf. on Parallel and Distributed Information Systems (PDIS 1991) (Miami Beach, 1991), IEEE-CS, 1991, pp. 262–270.
578. Pramanik S., Tout W.R. The NUMA with Clusters of Processors for Parallel Join, IEEE Trans. Knowledge Data Eng., 1997, vol. 9, no. 4, pp. 653–666.
579. Bouganim L., Florescu D., Valduriez P. Dynamic Load Balancing in Hierarchical Parallel Database Systems, Proc. 22th Int. Conf. on Very Large Data Bases (VLDB’96) (Mumbai, India, 1996), Morgan Kaufmann, 1996, pp. 436–447.
580. Xu Y., Dandamudi S.P. Performance Evaluation of a Two-Level Hierarchical Parallel Database System, Proc. Int. Conf. Computers and Their Applications, Tempe, Arizona, 1997, pp. 242–247.
581. Korneev V.V. Parallel’nye vychislitel’nye sistemy (Parallel Computing Systems), Moscow: Nolidzh, 1999.
582. Shmidt V. IBM SP2 Systems (Rus), Otkrytye Sistemy, 1995, no. 6, pp. 53–60.
583. Shnitman V. Fault-Tolerant Servers ServerNet (Rus), Otkrytye Sistemy, 1996, no. 3, pp. 5–11.
584. Sokolinsky L.B. Organization of Parallel Query Processing in Multiprocessor Database Machines with Hierarchical Architecture, Programmirovanie, 2001, no. 6, pp. 13–29.

585. Velicanu M., Litan D., Mocanu (Virgolici) A.-M., 2010. "Some Considerations about Modern Database Machines," *Informatica Economica, Academy of Economic Studies - Bucharest, Romania*, vol. 14(2), pages 37-44.
586. Eric G. Oracle and storage IOs, explanations and experience at CERN," 17th International Conference on Computing in High Energy and Nuclear Physics, Prague, Czech Republic, March 2009, pp. 21 – 27.
587. Chock M., Cardenas A., Klinger A. Database structure and manipulation capabilities of a picture database management system (PICDMS). *IEEE ToPAMI*, 6(4):484–492, 1984
588. Maier D., Vance B. A call to order. In *PODS '93: Proceedings of the twelfth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, 1993, pp. 1–16
589. Baumann P. Management of Multidimensional Discrete Data. *VLDB Journal, Special Issue on Spatial Database Systems*, 1994, Vol 4, No. 3, pp. 401–444
590. Baumann P. A Database Array Algebra for Spatio-Temporal Data and Beyond, 4th International Workshop on Next Generation Information Technologies and Systems (NGITS '99), July 5–7, 1999, Zikhron Yaakov, Israel, *Lecture Notes on Computer Science 1649*, Springer Verlag, pp. 76 – 93.
591. Baumann P., Dehme A., Furtado P., Ritsch R., Widmann N. The Multidimensional Database System RasDaMan. Conference: SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA. *ACM SIGMOD Record*, 1998, Vol. 27, No. 2, pp 575–577
592. EarthServer: The EarthServer Initiative. www.earthserver.eu
593. Baumann P., Holsten S. (2011) A Comparative Analysis of Array Models for Databases. In: Kim T. et al. (eds) *Database Theory and Application, Bio-Science and Bio-Technology. BSBT 2011, DTA 2011. Communications in Computer and Information Science*, vol 258. pp 80-89 Springer, Berlin, Heidelberg.
594. Baumann, P., Misev, D., Merticariu, V., Bang Pham Huu. Array databases: concepts, standards, implementations. *Journal of Big Data*, vol 8, No. 1 (2021).
595. Tomlin D. A Map Algebra. Harvard Graduate School of Design, 1990.
596. Mennis, J., Viger, R., Tomlin, C.D.: Cubic Map Algebra Functions for Spatio-Temporal Analysis. *Cartography and Geographic Information Science*, Vol. 32, No. 1, 2005, pp. 17-32.
597. Ritter G, Wilson J, Davidson J. Image Algebra: An Overview. *Computer Vision, Graphics, and Image Processing*. Vol. 49, No. 3, 1990, pp. 297-331.
598. Marathe A, Salem K. A language for manipulating arrays. In: *Proceedings of the 23th International Conference on Very Large Data Bases*; 1997. p. 46–55.
599. Libkin, L., Machlin, R., Wong, L.: A query language for multidimensional arrays: design, implementation and optimization techniques. *Proc. ACM SIGMOD'96, Montreal, Canada/ ACM SIGMOD Record*, 1996, vol. 25, No. 2, pp. 228–239
600. Machlin R. Index-based multidimensional array queries: safety and equivalence. In L. Libkin, editor, *PODS*, pp. 175–184. ACM, 2007.
601. van Ballegooij A.R., de Vries A.P., Kersten M. RAM: Array processing over a relational DBMS. Technical Report INS-R0301, CWI (March 2003)
602. van Ballegooij A.R. RAM: A multidimensional array DBMS. In W. Lindner, M. Mesiti, C. Turker, Y. Tzitzikas, and A. Vakali, editors, *EDBT Workshops*, volume 3268 of *Lecture Notes in Computer Science*, pp. 154–165. Springer, 2004.
603. Cornacchia R., Heman S., Zukowski M., de Vries A., Boncz P. Flexible and efficient IR using array databases, *VLDB Journal*. 7(1): 151–168.
604. Stonebraker M., Brown P., Poliakov A., Raman S. (2011) The Architecture of SciDB. In: Bayard Cushing J., French J., Bowers S. (eds). *Proceedings of the 23rd International Conference on Scientific and Statistical Database Management*; 2011 pp 1-16
605. Kersten M.L, Zhang Y., Ivanova M., Nes N. SciQL, a query language for science applications. *Proceedings, EDBT/ICDT 2011 Workshop on Array Databases: Uppsala, Sweden, March 25, 2011*, pp
606. Cheng, Y., Rusu, F. Formal representation of the SS-DB benchmark and experimental evaluation in EXTASCID. *Distrib Parallel Databases*, 2015, vol. 33, No. 3, pp. 277–317

607. Cheng Y., Rusu, F. Astronomical data processing in EXTASCID. SSDBM: Proceedings of the 25th International Conference on Scientific and Statistical Database Management, 2013 Article No.: 47, pp. 1–4
608. Tollefsen, Andreas Forø (2013) PostGIS 2.0 og Raster, Kart og Plan 73(3), pp. 159–164.
609. Teradata. Multidimensional array options. - <https://docs.teradata.com/r/VrFCOAaniAlfr-JsA51oQJA/ZMY8sE8cSytuSPtp8QnuFA>
610. Oracle: GeoRaster. - http://docs.oracle.com/cd/B19306_01/appdev.102/b14254/geor_intro.htm.
611. Becla J., Lim K.T. Report from the first Workshop on Extremely Large Databases, Data Science Journal, 2008, Vol. 7, pp. 1-13
612. Stonebraker M., Brown P., Poliakov A., Raman S. (2011) The Architecture of SciDB. In: Bayard Cushing J., French J., Bowers S. (eds). Proceedings of the 23rd International Conference on Scientific and Statistical Database Management; 2011 pp. 1-16
613. Bauman National Library. SciDB. - <https://ru.bmstu.wiki/SciDB>
614. Ivanova M, Kersten M.L, Manegold S. Data vaults: a symbiosis between Database technology and scientific file repositories. Proc. Intl. Conference on Scientific and Statistical Database Management (SSDBM). Athens. 2012, pp. 485-:494.
615. Zhang Y, Kersten M.L, Ivanova M, Nes N. SciQL, bridging the gap between science and relational DBMS. In: Desai B.C, Cruz I.F, Bernardino J, editors. Proceedings of the 15th Symposium on International Database Engineering and Applications; 2011. pp. 124–133.
616. Baumann P., Stamerjohanns H. (2014) Towards a Systematic Benchmark for Array Database Systems. In: Rabl T., Poess M., Baru C., Jacobsen H.A. (eds) Specifying Big Data Benchmarks. pp 94-102.
617. “ISO/IEC DIS 9075-15 Information technology -- Database languages -- SQL -- Part 15: Multi-dimensional arrays (SQL/MDA)”
618. Furtado P., Baumann P. Storage of multidimensional arrays based on arbitrary tiling. In Proceedings of the 15th International Conference on Data Engineering, pp. 328–336. IEEE Computer Society, 23-26 March 1999
619. Srivastava J., Ngo H.Q. Statistical Databases. Technical Report TR 99-009, 1999, Department of Computer Science and Engineering, University of Minnesota. - <https://conservancy.umn.edu/bitstream/handle/11299/215365/99-009.pdf?sequence=1&isAllowed=y>
620. Michalewicz Z. (ed.) Statistical and Scientific Databases. Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, 11991, 544 p.
621. Chan P., Shoshani A. SUBJECT: A directory driven system for large statistical databases. In VLDB '81: Proceedings of the seventh international conference on Very Large Data Bases - Volume 7, 1981, pp. 553–563
622. Su S. SAM: A semantic association model for corporate and scientific-statistical databases. Journal of Information Science, pp. 151–199, 1983.
623. Rafanelli M., Ricci F.L. A visual interface for browsing and manipulating statistical entities. In Proceedings of the Fifth International Conference on Scientific and Statistical Database Management, pp. 1990, 163–182,
624. Battista G.D., Batini C. Design of statistical databases: a methodology for the conceptual step. The Journal of Information Systems, vol. 13, no. 4, pp. 407–422, 1988
625. Rafanelli M., Shoshani A. STORM: A statistical object representation model. In Proceedings of the Fifth International Conference on Scientific and Statistical Database Management, pp. 14–29, 1990.
626. Rafanelli M., F.L. Ricci, “Mefisto: A functional model for statistical entities,” IEEE Transactions on Knowledge and Data Engineering, vol. 5, No. 4, pp. 670–681, Aug. 1993.
627. Ghosh S.P. (1989) Statistical relational model. In: Rafanelli M., Klensin J.C., Svensson P. (eds) Statistical and Scientific Database Management. SSDBM 1988. Lecture Notes in Computer Science, vol 339. Springer, Berlin, Heidelberg. pp. 338-355
628. Shoshani A., Kawagoe K. Temporal data management. In Proceedings of the Twenty-Second International Conference on Very Large Data Bases (VLDB), 1986 pp. 79–90.
629. Meo-Evoli L., Ricci F.L., Shoshani A., On the Semantic Completeness of Macro-Data Operators for Statistical Aggregation, SSDBM 1992, pp. 239-258.
630. Ozsoyoglu G., Ozsoyoglu Z.M., Matos V., Extending relational algebra and relational

- calculus with set-valued attributes and aggregate functions,” *ACM Transactions on Database Systems*, 1987, vol. 12, pp. 566–592.
631. Ozsoyoglu G., Ozsoyoglu Z.M., Malta F. A Language and a Physical Organization Technique for Summary Tables. *SIGMOD*, 1985: pp. 3-16.
632. Gray J., Bosworth A., Layman A., Pirahesh H. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. *Data Mining and Knowledge Discovery*, 1997, Vol. 1, No. 1, pp. 29–53
633. Agrawal R., Gupta A., Sarawagi S. Modeling Multidimensional Databases. *ICDE '97: Proceedings of the Thirteenth International Conference on Data Engineering* April, 1997, pp. 232–243.
634. Vassiliadis P. “Modeling multidimensional databases, cubes and cube operations,” *Proceedings. Tenth International Conference on Scientific and Statistical Database Management (Cat. No.98TB100243)*, 1998, pp. 53-62,
635. Gentle J.E., Bell J. Special Data Types and Operators for Statistical Data. *IEEE Database Eng. Bull.*, 1984, Vol. 7, No. 1, pp. 34-37
636. Fortunato E., Rafanelli M., Ricci F., Sebastio A., An algebra for statistical data. In *SSDBM'86: Proceedings of the 3rd international workshop on Statistical and scientific database management*, 1986, pp. 122–134
637. Bezenchek A., Rafanelli M., Tininini L. A data structure for representing aggregate data. In *Proceedings: Eighth International Conference on Scientific and Statistical Database Systems*, Stockholm, Sweden (P. P. Svensson and J. C. J. C. French, eds.), IEEE Computer Society Press, 1996, pp. 22–31
638. van den Berg G.M., E. de Feber. Definition and use of meta-data in statistical data processing. In *Proceedings of the 6th International Conference on Statistical and Scientific Management*, (Ascona, Switzerland), 1992, pp. 290–306
639. Kent J.P., Schuerhoff M. Some thoughts about a metadata management system, in *Proceedings of the 9th International Conference on scientific and Statistical Databases*, (Olympia, WA), pp. 155–164, IEEE Press, Aug. 1997.
640. Westlake A. “A simple structure for statistical meta-data,” in *Proceedings of the 9th International Conference on scientific and Statistical Databases*, (Olympia, WA), pp. 186–195, IEEE Press, Aug. 1997.
641. Ghosh S. P. Statistical Metadata. In *Kotz-Johnson Encyclopedia of Statistical Science*, Vol.8, John Wiley & Sons Inc. Publ., 1988
642. Signore M., Scanu M., Brancato G. Statistical metadata: a unified approach to management and dissemination. *Journal of Official Statistics*, 2015, Vol. 31, No 2, pp. 325-347
643. Tansel A. Query languages for statistical databases. *Statistics and Computing*. 1995. Vol. 5, No. 1, pp. 59-72
644. Ozsoyoglu G., Ozsoyoglu, Z. M. Statistical database query languages. *IEEE Transactions on Software Engineering*. 1985, vol 11, No. 10, pp. 1071-1080.
645. Johnson R. Modeling summary data. In *Proceedings of the ACM SIGMOD Conference*, (Ann Arbor, Michigan), pp. 93–97, 1981.
646. Shoshani A. CABLE: A Chain-Based Language for the Entity-Relationship Model. *Proceedings of the 1st International Conference on the Entity-Relationship Approach to Systems Analysis and Design*, 1980, pp. 465–466
647. Ikeda H., Kobayashi Y. Additional facilities of a conventional DBMS to support interactive statistical analysis. In *Proceedings of the 1st LBL Workshop on Statistical Database Management*, Lawrence Berkeley Lab, Berkeley, CA, Dec. 1981, pp. 25–36
648. Computer Corporation of America. File Manager's Technical Reference Manual, Model 204 Database Management System. Computer Corporation of America, Cambridge, MA, 1979
649. Ghosh S.P. Statistical relational tables for statistical database management, *IEEE Transactions of Software Engineering*, 1986, vol. SE-12, No. 12, pp. 1106–1116.
650. Maier D., Cirilli C. SYSTEM/K: A knowledge based management system. In *Proceedings of the Second Int. Workshop on Statistical Database Management*, Los Altos, CA, Sept. 1983, pp. 287–294
651. Stein D.M. A database interface to an integrated dataanalysis and plotting tool. In *Proceedings of the 3rd International Workshop on Statistical and Scientific Database Management*, Luxemburg, 1986, pp. 98–106
652. Heiler S., Bergman R.F. SIBYL: An economist'sworkbench. In *SSDBM'83: Pro-*

- ceedings of the 2nd International Workshop on Statistical Database Management, Los Altos, CA., 1983, pp. 73–79
653. Weiss S.E., Weeks P.L. PASTE—a tool to put application systems together easily. In SSD-BM'83: Proceedings of the 2nd International Workshop on Statistical Database Management LosAltos CA, 1983, pp. 119–123
654. Hollabaugh L.A., Reinwald L.T. GPI: a statistical package/database interface. SS-DBM'81: In Proceedings of the 1st International Workshop on Statistical Database Management MenloPark CA, 1981, pp. 78–87
655. Boufares P., Elkabbaj Y., Joiner G., Ounally H. Laversion SM90 du SGBD relationnel PEPIN. Journes SM90, Versailles, France, 1985
656. Turner M. T., Hammond R. Cotton P. A DBMS for large statistical databases. In VLDB '79: Proceedings of the fifth international conference on Very Large Data Bases - Volume 5, 1979, pp. 319–327
657. Johji S., Sato H. Statistical database research project in Japan and the CAS SDB project. In SSDBM'83: Proceedings of the 2nd international workshop on Statistical Database Management, 1983 pp. 325–330.
658. Klug A. ABE – a query language for constructing aggregates-by-example. In SSD-BM'81: Proceedings of the 1st LBL Workshop on Statistical Database Management, 1981, pp. 190–205
659. Anderson G., Snider T., Robinson B., Toporek J. An integrated research support system for inter-package communication and handling large volume output from statistical database analysis operation. In SSDBM'83: Proceedings of the 2nd international workshop on Statistical Database Management, 1983, pp. 104–110
660. Dintelman S.M., Maness A.T. An implementation of a query language supporting path expressions. In Proceedings of the 1982 ACM SIGMOD International Conference on Management of Data, (Orlando, Florida), 1982., pp. 87–93
661. Karasolo I., Severson P. An overview of CANTOR – a new system for data analysis. In SSDBM'83: Proceedings of the 2nd international workshop on Proceedings of the Second International Workshop on Statistical Database Management September 1983 pp. 315–324
662. Chan C., Michalewicz Z. A query language capable of handling incomplete information and statistics. In SSDBM'86: Proceedings of the 3rd international workshop on Statistical and scientific database management, 1986, pp. 107–115
663. D'attri A., Ricci F.L. Interpretation of statistical queries to relational databases. In SS-DBM'1988: Proceedings of the 4th international conference on Statistical and Scientific Database Management, 1988, pp. 246–258
664. Chen M., McNamee L., Melkanoff M. A model of summary data and its applications in statistical databases. In SSDBM'1988: Proceedings of the 4th international conference on Statistical and Scientific Database Management, 1988, pp 356–387
665. Anderson G., Snider T., Robinson B., Toporek J. An integrated research support system for inter-package communication and handling large volume output from statistical database analysis operation. In SSDBM'83: Proceedings of the Second International Workshop on Statistical Database Management, 1983, pp. 104–110.
666. Weiss W., Weeks P., Byrd P. Must we navigate through databases. In SSDBM'81: Proceedings of the 1st LBL Workshop on Statistical Database Management, Lawrence Berkeley Lab, Berkeley, CA, Dec. 1981, pp. 111–122
667. Hendrix G.G., Sacerdoti E.D., Sagalowicz D., Slocum J. Developing a natural language interface to a complex system. ACM Transactions on Database Systems, 1978, Vol. 3, No. 2., pp. 105–147
668. Brown W., Navathe S., Su S. Complex data types and a data manipulation language for scientific and statistical databases. In SSD-BM'83: Proceedings of the 2nd international workshop on Statistical Database Management, 1983, pp.188–195
669. Ozsoyoglu G., Ozsoyoglu Z.M. Features of a system for statistical databases. In SSD-BM'83: Proceedings of the 2nd international workshop on Statistical Database Management, 1983, pp. 9–18.
670. Wong H.K.T., Kuo I. GUIDE: Graphical user interface for database exploration. In Proceedings of the 8th Conference on Very

- Large Databases, Morgan Kaufman pubs. (Los Altos CA), McLeod and Villasenor, Mexico City, 1982, pp. 22-32
671. Ozsoyoglu Z. M., Ozsoyoglu G. Summary-table-by-example: A database query language for manipulating summary data. In Proceedings of the International Conference on Data Engineering, (Los Angeles, CA), 1984, pp. 193–202.
672. Thomas J., Hall D. ALDS project: Motivation, statistical database management issues, perspectives, and directions, In SSDBM'83: Proceedings of the 2nd international workshop on Statistical Database Management-September, 1983, pp.82–88
673. Catarci T., Santucci G. GRASP: A graphical system for statistical databases. In Proceedings of the Fifth International Conference on Scientific and Statistical Database Management, (Charlotte, NC), 1990, pp. 148–162
674. Sato H. A data model, knowledge base and natural language processing for sharing a large statistical database. In Proceedings of the 4th International Working Conference SSDBM on Statistical and Scientific Database Management, 1988, pp. 207–225
675. Snodgrass R. T., The temporal query language TQuel. In Symposium on Principles of Database Systems, 1984, pp. 204–213.
676. Tansel A.U., Arkun M.E. HQUEL, A query language for historical relational databases. In SSDBM'86: Proceedings of the 3rd international workshop on Statistical and scientific database management, 1986, pp. 135–142
677. Tansel A., Arkun M.E., Ozsoyoglu G. Time-by-example query language for historical databases. IEEE Transactions on Software Engineering (SE), 1989, vol. 15, No. 4, pp.464-478
678. Elmasri R., Kouramajian V. A temporal query language based on conceptual entities and roles. ER '92: Proceedings of the 11th International Conference on the Entity-Relationship Approach: Entity-Relationship Approach, 1992, pp. 375–388
679. Tansel A.U. A statistical interface for historical relational databases. In Proceedings of the Third International Conference on Data Engineering February, 1987, pp 538–546
680. Reznichenko V.A. Workig with windows in SQL (Rus). Software Engineering, 2011, vol. 7, No 3, pp. 35-48
681. Chandra P., Gupta M.K. Comprehensive survey on data warehousing research. International Journal of Information Technology. 10, pp. 217–224 (2018). <https://doi.org/10.1007/s41870-017-0067-y>
682. Inmon, W.H. 'Building the data warehouse', 5th Edition, John Wiley & Son. 2005
683. Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition. John Wiley & Sons, Inc. 2013. 600 p.
684. Breslin M. Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models. In Business Intelligence Journal. 2004. pp. 6-20
685. Brackett M.H. The Data Warehouse Challenge: Taming Data Chaos. John Wiley & Sons, 1996, 579 pages.
686. Gill S.H., Rao P.C. The Official Client/Server Computing Guide to Data Warehouse. QUE Corporation, 1996, 382 pages.
687. Poe V. Building a Data Warehouse for Decision Support. Prentice Hall. 1995
688. Codd E.F. Providing OLAP to User-Analysts: An IT Mandate // Computerworld. — T. 27, № 30
689. Pendse N. What is OLAP? - <http://dssresources.com/papers/features/pendse04072002.htm>
690. Ponniah P. 'Data warehousing fundamentals', John Wiley & Sons, 2001, 516 p.
691. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques, 3rd ed. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 2011, 703 p.
692. Chaudhuri S., Dayal U. An Overview of Data Warehousing and OLAP Technology. ACM SIGMOD Record, Volume 26, Issue 1, March 1997 pp 65–74. - <https://doi.org/10.1145/248603.248616>
693. Jensen C.S, Pedersen T.B, Thomsen C. Multidimensional databases and data warehousing. Synthesis lectures on data management. San Rafael: Morgan Claypool; 2010. 111 p.
694. Vaisman A, Zimányi E. Data Warehouse Systems: Design and Implementation (Data-Centric Systems and Applications) 2014th Edition. Springer; 2014.
695. Muhammad Arif, Ghulam Mujtaba. A Survey: Data Warehouse Architecture. International Journal of Hybrid Information Technology Vol.8, No. 5 (2015), pp. 349-356.

696. Astriani W., Trisminingsih R. Extraction, Transformation, and Loading (ETL) module for hotspot spatial data warehouse using Geokettle. *Procedia, Environmental Science*, Elsevier, The 2nd International Symposium on LAPAN-IPB Satellite for Food Security and Environmental Monitoring 2015, pp 626-634
697. Chaudhary S., Murala D.P., Srivastav V.K. (2011) 'A critical review of data warehouse', *Global Journal of Business Management and Information Technology*, Volume(1):No.(2), pp. 95-103.
698. Oliveira B., Belo O. (2015) A Domain-Specific Language for ETL Patterns Specification in Data Warehousing Systems. In: Pereira F., Machado P., Costa E., Cardoso A. (eds) *Progress in Artificial Intelligence. EPIA 2015. Lecture Notes in Computer Science*, vol 9273. Springer, Cham. pp 597-602 https://doi.org/10.1007/978-3-319-23485-4_60
699. Data Warehouse Architecture, Concepts and Components. - <https://www.guru99.com/data-warehouse-architecture.html>
700. Data Warehouse Architecture: Types, Components, & Concepts. - <https://www.astera.com/type/blog/data-warehouse-architecture/>
701. Enterprise Data Warehouse: Concepts and Architecture. - <https://www.altexsoft.com/blog/enterprise-data-warehouse-concepts/>
702. Bhadresh Pandya, Dr. Sanjay Shah. Proposed Local Data Mart Approach for Data Warehouse Architecture. *International Journal of Emerging Technology and Advanced Engineering*. 2014. Vol. 4, No. 2. pp. 101-104
703. Yang Q., Ge M. Helfert M. Analysis of Data Warehouse Architectures: Modeling and Classification. In *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019)*, pages 604-611.
704. Kimball R., Caserta J. *The Data Warehouse ETL Toolkit*. Wiley Publ., 2004, 526 p.
705. Chandra P., Gupta M.K. Comprehensive survey on data warehousing research. *International Journal of Information Technology*.
706. Scabora L.C., Brito J.J., Ciferri R.R., Ciferri C.D.D.A. Physical data warehouse design on NoSQL databases – OLAP Query Processing over HBase. *Proc. 18th Intern. Conf. SCITEPRESS*. 2016, pp. 111–118. DOI: 10.5220/0005815901110118.10, pp. 217–224
707. Khan F.A., Ahmad A., Imran M., Alharbi M., Jan B. Efficient data access and performance improvement model for virtual data warehouse. *Sustainable cities and society*. 2017, vol. 35, pp. 232–240. DOI: 10.1016/j.scs.2017.08.003.
708. Gupta A., Mumick I.S. 'Maintenance of materialized views: problems, techniques, and applications', *IEEE Data Engineering Bulletin, Special Issue on Materialized Views and Data Warehousing*, 1995, Vol.18, No. 2, pp. 3-18
709. Sachin Chaudhary, Devendra Prasad Murala, V.K. Srivastav. A Critical Review of Data Warehouse. *Global Journal of Business Management and Information Technology*. 2011, Vol. 1, No.2, pp. 95-103
710. Demarest, "Building The Data Mart", *DBMS Magazine*, 1994. — Vol 7, No.8. — p. 44—50.
711. Pedersen T.B, Jensen C.S. Multidimensional data modeling for complex data. In: *Proceedings of the 15th International Conference on Data Engineering*; 1999. p. 336–345.
712. Vassiliadis P. Modeling multidimensional databases, cubes and cube operations. In: *Proceedings of the 10th International Conference on Scientific and Statistical Database Management*; 1998. p. 53–62.
713. Kimball R., Ross M. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, Third Edition. John Wiley & Sons, Inc. 2013, 600 p.
714. Han J., Kamber M., Pei J. *Data Mining: Concepts and Techniques*, 3rd ed. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 2011, 703 p. - УЖЕ ЕСТЬ ВЫИШЕ
715. Harinarayan V, Rajaraman A, Ullman J.D. Implementing data cubes efficiently. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*; 1996. p. 205–216.
716. Sanjay Goil and Alok Choudhary. High performance OLAP and data mining on parallel computers. Center of Parallel and Distributed Computing Technical Report TR9705, 1997
717. Morfonios K., Ioannidis Y. Cube Implementations. In *Encyclopedia of Database Systems*, Ling Liu, M. Tamer Özsu Editors. pp. 710-716.

718. Morfonios K, Konakas S, Ioannidis Y, Kotsis N. ROLAP implementations of the data cube. *ACM Computing Surveys*, 2007. vol. 39, No. 4, Article 12, 53 pages/
719. Pedersen TB. Managing complex multidimensional data. In: Aufaure M-A, Zimányi E, editors. *Business intelligence – second European summer school, eBISS 2012*. Brussels: Springer LNBI; 2013, 15–21 July 2012, Tutorial Lectures.
720. Vaisman A, Zimányi E. *Data warehouse systems – design and implementation*. Springer; 2014.
721. Multidimensional DBMS. - https://tadviser.com/index.php/Article:Multidimensional_DBMS
722. Gosain A., Heena. Literature Review of Data model Quality metrics of Data Warehouse. *International Conference on Intelligent Computing, Communication & Convergence (ICCC-2015)*. *Procedia Computer Science* 48 (2015) 236 – 243
723. Schrefl M, Thalhammer T. On Making Data Warehouses Active. In M. Mohania and A. Min Tjoa, editors, *DaWaK 2000: Proceedings of the Second International Conference on Data Warehousing and Knowledge Discover*, Greenwich, London (UK), September 4-6, 2000. Springer LNCS, pp. 34–46
724. Thalhammer T, Schrefl M, Mohania M. Active data warehouses: complementing OLAP with analysis rules. *Data & Knowledge Engineering*, 2001, Vol. 39, No. 3, pp. 241–269.
725. Brobst S. Active data warehousing: a new breed of decision support. In: *Proceedings of the 13th International Workshop on Data and Expert System Applications*; 2002. p. 769–772.
726. Borbst S, Rarey J. The five stages of an active data warehouse evolution. *Teradata Mag.* 2001;3(1):38–44.
727. Syed Ijaz Ahmad Bukhari: Real Time Data Warehouse. *CoRR abs/1310.5254* (2013)
728. IBM Data Warehousing. -<https://www.ibm.com/analytics/us/en/data-management/data-warehouse>.
729. Best practices for Real-time Data Warehousing. An oracle white paper. 2014. <http://www.oracle.com/us/products/middleware/data-integration/realtime-datawarehousing-bp-2167237.pdf>
730. Mohania M., Nambiar U., Tam H., Schrefl M., Vincent M. Active, Real-Time, and Intellective Data Warehousing. In *Encyclopedia of Database Systems*, Ling Liu, M. Tamer Özsu Editors. pp 41-49
731. Kimball R. Slowly changing dimensions. *DBMS Mag.* 1996;9(4):14.
732. Eder J., Koncilia C., Wiggisser K. Data Warehouse Maintenance, Evolution, and Versioning. In *Encyclopedia of Database Systems*, Ling Liu, M. Tamer Özsu Editors. pp. 884-890
733. Bruckner R, Min Tjoa A. Capturing delays and valid times in data warehouses: towards timely consistent analyses. *J. Intell. Inf. Syst.* 2002;19(2):169–190.
734. Malinowski E, Zimányi E. Advanced data warehouse design: from conventional to spatial and temporal applications. Berlin/Heidelberg: Springer; 2008.
735. Ahmed W, Zimányi E, Wrembel R. Temporal data warehouses: logical models and querying. In: *Proceedings of the Journées francophones sur les Entrepôts de Données et l'Analyse en ligne, EDA*. Editions Hermann; 2015. p. 33–48.
736. Mendelzon A, Vaisman A. Time in multidimensional databases. In: Rafanelli M, editor. *Multidimensional databases: problems and solutions*. Hershey: Idea Group; 2003. p. 166–199.
737. Golfarelli M, Rizzi S. Managing late measurements in data warehouses. *Int J Data Wareh Min.* 2007;3(4):51–67.
738. Böhlen M, Gamper J, Jensen C. Towards general temporal aggregation. In: *Proceedings of the 25th British National Conference on Databases*; 2008. p. 257–169.
739. Golfarelli M, Lechtenböcker J, Rizzi S, Vossen G. Schema versioning in data warehouses: enabling cross-version querying via schema augmentation. *Data Knowl Eng.* 2006;59(2):435–459.
740. Ahmed W, Zimányi E, Wrembel R. A logical model for multiversion data warehouses. In: *Proceedings of the 16th International Conference on Data Warehousing and Knowledge Discovery*; 2014. p. 23–34.
741. Golfarelli M, Rizzi S. A survey on temporal data warehousing. *Int J Data Wareh Min.* 2009;5(1):1–17.
742. Rivest S, Bédard Y, Marchand P. Toward better support for spatial decision making: defining the characteristics of spatial on-line

- analytical processing (SOLAP). *Geomatica* 2001;55(4):539–555.
743. Bédard Y, Merrett T, Han J. Fundamentals of spatial data warehousing for geographic knowledge discovery. In: Miller H, Han J, editors, *Geographic data mining and knowledge discovery*. London: Taylor & Francis; 2001. p. 53–73.
744. Stefanovic N, Han J, Koperski K. Object-based selective materialization for efficient implementation of spatial data cubes. *IEEE Trans Knowl Data Eng.* 2000;12(6):938–958.
745. Malinowski E, Zimányi E. Representing spatiality in a conceptual multidimensional model. In: *Proceedings of the 12th ACM Symposium on Advances in Geographic Information Systems*; 2004. p. 12–22.
746. Bimonte S, Tchounikine A, Miquel M. Towards a spatial multidimensional model. In: *Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP*; 2005. p. 39–46.
747. Shanmugasundaram J, Fayyad U, Bradley P. Compressed data cubes for OLAP aggregate query approximation on continuous dimensions. In: *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 1999. p. 223–232.
748. Ahmed TO, Miquel M. Multidimensional structures dedicated to continuous spatio-temporal phenomena. In: *Proceedings of the 22nd British National Conference on Databases*; 2005. p. 29–40.
749. Gómez L, Gómez S, Vaisman A. Analyzing continuous fields with OLAP cubes. In: *Proceedings of the 14th ACM International Workshop on Data Warehousing and OLAP*; 2011. p. 89–94.
750. Gómez L, Gómez S, Vaisman A. A generic data model and query language for spatio-temporal OLAP cube analysis. In: *Proceedings of the 15th International Conference on Extending Database Technology*; 2012. p. 300–311.
751. Gómez L, Gómez S, Vaisman A. Modeling and querying continuous fields with OLAP cubes. *Int J Data Wareh Min.* 2013;9(3):22–45.
752. A.A. Vaisman, Zimányi E. *Spatial Data-warehousing. Encyclopedia of Database Systems*, Ling Liu, M. Tamer Özsu Editors, Second Edition, 2018, pp. 3587-3592
753. Bédard, Y., T. Merrett & J. Han, 2001, *Fundamentals of Spatial Data Warehousing for Geographic Knowledge Discovery, Geographic Data Mining and Knowledge Discovery*, Taylor & Francis, Vol. Research Monographs in GIS, No. Chap. 3, p. 53-73
754. Gray J, Chaudhuri S, Bosworth A, Layman A, Venkatrao, M, Reichart D, Pellow F, Pirahesh H. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab and Sub-Totals. *Data Mining and Knowledge Discovery*, 1997, 1(1):29–54.

Отримано: 27.07.2021

Про автора:

Резніченко Валерій Анатолієвич,
кандидат фізико-математичних наук,
заступник завідувача відділом.
Кількість публікацій
в українських виданнях – 61.
Кількість зарубіжних публікацій – 4.
Індекс Хірша – 12.
<http://orcid.org/0000-0002-4451-8931>.

Місце роботи автора:

Інститут програмних систем
НАН України, 03187, м. Київ-187,
проспект Академіка Глушкова, 40.
Тел.: (044) 526 3559.
E-mail: reznich@isofts.kiev.ua