

В.А. Резніченко

60 РОКІВ БАЗАМ ДАНИХ (частина друга)

Наводиться огляд досліджень і розробок баз даних з моменту їх виникнення в 60-х роках минулого століття і по сьогодні. 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, характеристичних властивостей, класифікації, принципів роботи, методів і технологій великих даних. Нарешті в останньому розділі дається короткий огляд досліджень і розробок із баз даних у 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, багатовимірна, машина баз даних, сховища даних, NoSQL, ключ-значення, стовпчикова, документно-орієнтована, графова, мультимодельна, хмарна, наукова, багатозначна, XML, NewSQL, онтологічна, великі дані.

Етап 4. Розширені реляційні бази даних (1980 – 2000 +)

Із моменту виникнення реляційна модель піддавалася критиці в зв'язку з простотою її структури даних. Відтак пропонувалися розвинутіші моделі, що дозволяли адекватніше представляти інформаційні моделі різних предметних областей. Однак їхньою особливістю було те, що всі вони створювалися на базі реляційної моделі й отримали назву розширених реляційних баз даних. Переважна більшість наведених у наступному абзаці БД або створювалися на основі реляційних БД, або мали варіанти такої реалізації.

У цей період активізувалися дослідження із взаємопроникнення технологій штучного інтелекту (ШІ) і БД. 1988 року відбулися два різних симпозіуми з інтеграції ШІ і БД. Відповідно до потреб у БД виникли два напрямки представлення в них правил (що породжують нові дані з існуючих). В результаті виникли БД двох типів: активні і дедуктивні. Крім того, потреба включення в БД часу і простору привела до появи темпоральних і просторових БД, а потреба застосування об'єктної техноло-

гії до БД привела до появи об'єктних БД. Прагнення суттєво підвищити продуктивність баз даних для роботи з великими обсягами даних викликало необхідність дослідження і розробки машин баз даних. А успіхи у створенні комп'ютерних мереж привели до появи розподільних і паралельних баз даних. Зрештою в цей же період було визнано, що БД мають використовуватись не лише для «рутинної» роботи зі збору, зберігання та пошуку ретельно відібраних і перевічених даних, а й для їх систематизації, узагальнення, статистичної та аналітичної обробки. Так з'явилися статистичні БД, БД для роботи з масивами, багатовимірні БД і, зрештою, сховища даних.

Темпоральні бази даних.

Темпоральні бази даних – це бази, що зберігають дані, дотичні до часу. Час як окремий тип даних присутній в усіх СУБД. Однак це не є підставою вважати їх темпоральними, адже інтерпретація часу і семантика взаємозв'язку між часом і даними залишається за розробником. У темпораль-

них базах даних мають існувати правила інтерпретації часу й можливості розкриття семантики взаємозв'язку даних із часом.

Дослідження з використання поняття часу в інформаційних системах були здійснені вже в 60-х роках минулого століття. Вважається [212], що вперше ідеї фіксації змінюваної в часі інформації в базах даних з'явилися 1976 року в праці Яніса А. Бубенка молодшого (Janis A. Bubenko, Jr) [213]. Згодом були проведені активні дослідження із розкриття семантики часу на концептуальному рівні [213 - 218], створення залежних від часу моделей даних для статистичних реляційних баз даних [219 - 222] і розробки темпоральних мов запитів [223 - 227].



Яніс Бубенко мол.

Основні поняття. Вже до середини 80-х років склалися основні положення темпоральних БД (ТБД). Їхня суть полягала в наступному. Темпоральний домен (temporal domain), загалом визначається як безліч темпоральних індивідів (temporal individuals), на яких задані темпоральні відношення (temporal relations). Темпоральний домен характеризується такими аспектами: структурним (лінійний час, розгалужений час), дискретним (безперервний час, дискретний час), граничним (обмежений час, безкінечний час).

Темпоральними індивідами можуть виступати моменти часу (часові точки) або часові інтервали. Для моментів часу задане відношення лінійного порядку (linear order), а для інтервалів – відношення Аллена [228].

Асоціація темпоральних індивідів із даними бази даних проводиться з допомогою часових позначок (timestamps).

Лінія часу – це часова вісь, задана на конкретному часовому домені й призначена для асоціації часових позначок із даними. Вирізняють дві лінії часу:

Лінія дійсного часу. Дійсний час (valid time) визначає період часу, протягом якого відбувається той чи інший факт моделювання реальності.

Лінія транзакційного часу. Транзакційний час – це період часу, протягом якого інформація про факт зберігається в базі даних.

Бази даних, які підтримують обидві лінії часу, називаються бітемпоральними.

Крім того, вірогідне існування користувального часу (user-defined time) – часу (інтервалу часу), що прив'язується в базі даних до факту самим користувачем.

На лінії часу присутній спеціальний момент часу, котрий називається ЗАРАЗ, і має специфічні особливості [229].

У темпоральній реляційній моделі час може прив'язуватись або до атрибутів, або до кортежів.

Темпоральні моделі даних. Темпоральна модель даних – це модель, в якій існує можливість прив'язувати дані до часу. Ці моделі відрізняються залежно від того, які лінії часу вони підтримують (модель дійсного часу, модель транзакційного часу, бітемпоральна модель), які темпоральні індивіди використовують (моменти часу – точкова модель або інтервали) і до яких даних прив'язується час (до значень атрибутів чи кортежів).

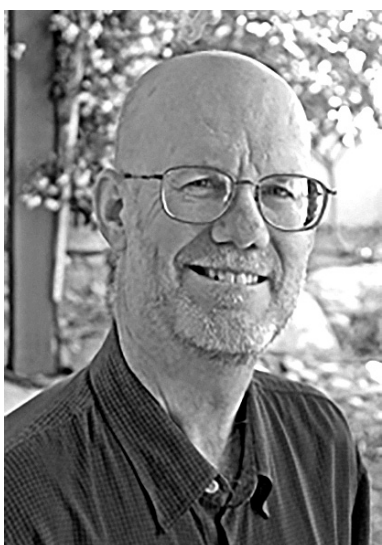
Пік досліджень із темпоральних моделей даних припадає на 80-і роки. Наведемо список праць, присвячених темпоральним моделям даних:

Яків бен-Зві (Jacob Ben-Zvi) [224, 225] – бітемпоральна інтервальна модель;

Джонс і Мейсон (Jones and Mason) [226] – інтервальна модель дійсного часу;

Річард Снодграсс (Richard Snodgrass) [227] – бітемпоральна точкова й інтервальна модель;

Лоренцос і Джонсон (Lorentzos and Johnson) [230, 231] – точкова й інтервальна модель дійсного часу.



Річард Снодграс

Всі ці моделі передбачали прив'язку часу до кортежів. Також пропонувалися темпоральні моделі з прив'язкою часу до атрибутів [232 - 237].

Із оглядом досліджень темпоральних моделей даних можна ознайомитись у статтях [238 - 240].

Темпоральні залежності.

Теорія проектування баз даних базується на понятті залежностей. Серед них фундаментальним поняттям є функціональна залежність.

Динамічний варіант функціональної залежності (DFD) вперше був висунутий Віану (Vianu) [241]. Передбачалося, що така FD має виконуватися в даному кортежі і в його оновленому варіанті. В цій же праці був досліджений взаємозв'язок між DFD і статичним FD. Ще один вид темпоральної FD був запропонований Війсеном (Wijsen) [242, 243]. В цьому випадку необхідно, аби FD виконувалась у поєднанні старого й нового відношення. Війсен визначив також тренд – залежність (trend dependency) [244], що є узагальненням визначеної ним же темпоральної FD.

Ще один різновид TFD був визначений у статті [245]. В ній пропонувалося розширення теорії нормалізації з урахуванням надлишковості, яка породжується TFD. У праці [246] було визначено залежність, викликану обмеженнями (constraint generating dependency - CGD). CGD означає, що кожен атрибут відношення приймає значення з

домену, визначеного обмеженням. Це характерно для атрибутів часових позначок темпоральних моделей даних. Нарешті, в статті [247] були досліджені темпоральні розширення відношень спеціалізації та узагальнення.

Мови. У проведених дослідженнях було запропоновано темпоральні реляційні алгебри [230 – 232, 236, 237, 248 - 250] і обчислення [233, 236] для різних темпоральних моделей даних. Була також визначена вкладена бітемпоральна модель даних та відповідна їй алгебра [234].

Тансел і Аркун розробили мову HQUEL [251], яка є розширенням мови QUEL «історичними» даними. Вони ж запропонували мову TBE (Time-By-Example) [252], в якій скористалися ідеєю графічної реляційної мови QBE. Снодграс також запропонував темпоральний варіант QUEL, який було названо TQUEL [227, 253].

Пропонувалися різні варіанти темпорального розширення SQL [235, 254 - 258].

TSQL2. Одним із ключових періодів у галузі досліджень темпоральних баз даних, часом їхнього «офіційного» представлення можна вважати 1992 – 1995 роки. Спершу Річард Снодграс (Richard T. Snodgrass) висунув ідею про можливе темпоральне розширення стандарту SQL-92, а згодом, 1993 року, був проведений семінар [259], який продемонстрував зацікавленість наукового співтовариства в розробці темпорального розширення стандарту SQL-92. В результаті було організовано комітет зі створення такої мови, названої Temporal Structured Query Language TSQL2. Провідну роль у роботі комітету відіграв Снодграс. Уже у вересні 1993 року було випущено перший черновий варіант мови, а в грудні – другий. В результаті плідної роботи в березні 1994 року з'явилася перша попередня версія специфікації мови [260], а у вересні навчальний посібник [261]. Нарешті 1995 року була опублікована остаточна специфікація мови запитів TSQL2 [262].

Подальша діяльність була пов'язана із включенням і розширенням основних ідей TSQL2 в SQL3. Ця мова була названа SQL/Temporal. Були опрацьовані питання

підтримки в SQL/Temporal дійсного і транзакційного часу [263, 264]. Остаточні пропозиції переходу від TSQL2 у SQL3 було сформульовано в [265].

Темпоральний SQL: 2011. 1995 року в ANSI/ISO було прийнято рішення про розгортання робіт зі створення нового стандарту SQL, який би включав темпоральні властивості. В зв'язку з цим США внесли пропозицію розширити відповідні можливості SQL, що базувалися на піонерських дослідженнях колективу під керівництвом Снодграса.

Ці пропозиції базувалися на детально на той час пропрацьованих групою Снодграса специфікаціях мови TSQL2, яка була темпоральним розширенням SQL-92, а також на пропозиціях переносу TSQL2 в SQL3. Однак деякі члени ISO піддали сумніву ці пропозиції США в зв'язку із наявними в них серйозними проблемами і протиріччями. Зі свого боку Великобританія внесла пропозицію, сформульовану на основі досліджень Нікоса Лоренцоса (Nikos Lorentzos) з університету Афін, Греція. Однак США не погодилися з позицією ISO щодо їхньої пропозиції і не підтримали Великобританію. Через це ANSI і ISO вирішили відкласти подальшу роботу по темпоральному SQL до офіційної публікації версії SQL-99.

Після публікації SQL-99 ані США, ані Великобританія не внесли жодних нових пропозицій, котрі б усунули виниклі раніше протиріччя. В зв'язку з цим 2001 року ANSI і ISO вирішили припинити діяльність зі створення стандарту темпорального SQL. Друга спроба додавання темпоральних властивостей в SQL була зроблена 2008 року. Вона почалася з обговорення, введення і прийняття пропозицій двох комітетів INCITS DM32.2 і ISO/IEC JTC1 SC32 WG3 із «системно – версійних таблиць» (systemversioned tables).

Ще одна темпоральна риса була додана в SQL 2010 року у вигляді «таблиць із прикладними періодами» (application-time period tables). Обидва ці поняття й розроблені для них відповідні мовні засоби були включені до стандарту SQL:2011. Із темпоральними особливостями SQL:2011 можна ознайомитися в [266, 267].

Просторові бази даних.

Просторова база даних (ПБД) – це база даних, призначена для зберігання, маніпулювання і виконання записів до даних про просторові об'єкти, представлені певними абстракціями. Тоді як традиційні БД призначені для зберігання й обробки числової і символічної інформації, ПБД дають можливість працювати з цілісними просторовими об'єктами, що об'єднують як традиційні види даних (описова частина або атрибутивна), так і геометричні (дані про розміри й розташування об'єктів у просторі).

Ще на початку 70-х років поняття обробки просторових даних використовували для позначення діяльності, пов'язаної з електронною обробкою даних з метою підвищення продуктивності під час складання і редагування карт, картографічних вимірів та аналізу просторових даних.

Моделі просторових даних.

Двома видами моделей просторових даних є: польова й об'єктна.

Польова модель. Використовується для представлення безперервних або аморфних явищ, зокрема, температури або хмарності. Ця модель підтримує функціональну точку зору, коли базисна система відліку (просторова система координат, як от, широта і довгота) функціонально відображаються в задану сферу знань. Наприклад, у градуси для температури. Комп'ютерною реалізацією польової моделі є растрова структура даних – рівномірна решітка, накладена на базисний простір. Іншими популярними структурами даних для представлення полів є триангульована нерегулярна мережа (triangulated irregular network TIN), лінійні контури, точкові решітки.

Операції польової моделі поділяються на три типи [268]:

локальні операції – значення функції в даній точці залежить лише від значення аргументу в цій точці (поточкова сума, різниця, максимум, середнє значення);

фокальні операції – значення функції в даній точці залежить від значень малого оточення цієї точки (нахил, середньозважене значення в довкіллі);

зональні операції – польова функція задається не на точках або їхньому малому оточенні, а на областях (багатокутниках) вцілому (сума, середнє, мінімальне польове значення кожної зони).

Об'єктна модель. Пояснює інформаційний простір як сукупність дискретних, ідентифікованих просторових сутностей. Вона представляється в комп'ютерах так званою векторною структурою даних. Основне питання об'єктної моделі – вибір базової множини типів просторових даних. Було проведено чимало досліджень, результатом яких став стандарт OGC [271], що визначив наступні типи (з певним спрощенням):

прості об'єкти – точки, криві, поверхні;
набори об'єктів – набір точок, набір кривих, набір поверхонь.

Операції. Здійснена велика кількість досліджень із визначення просторових операцій. Стаття [273] була однією з перших спроб загального опису операцій над картами (алгебра карт) з позицій растрового аналізу і стала базовою мовою для роботи з польовими моделями. З часом ця алгебра була уточнена в [274]. У [275] запропонована розширена мова запитів для географічних баз даних. У статті [276] вперше пропонувалося розширення алгебри реляційної моделі шляхом введення просторових об'єктів і операцій. У статті [277] пропонується мова SpatialSQL, де до мови SQL влючені просторові операції і відношення. Алгебра ROSE (**RO** bust **S**patial **E**xtension) [278] базується на реляційній моделі, використовує типи даних для представлення точок, ліній, областей і пропонує вичерпний набір операцій; семантика типів і операцій визначена формально. В праці [279] представлена просторова логіка, яка може використовуватись для міркувань щодо топологічних і просторових взаємозв'язків між об'єктами. Перевага даного підходу – строго визначена семантика й використання механізму логічного висновку.

Для об'єктної моделі були визначені наступні операції [269]:

операції на множинах (однаково, не однаково, дорівнює, не дорівнює, є членом, є пустим, об'єднання, різниця, кардинальність...);

топологічні операції (є границею, внутрішня частина, зовнішня частина, замикання, торкаються, пересікаються, знаходиться всередині, знаходиться зовні, охоплює...);

метричні операції (відстань, кут, довжина, площа периметр...);

операції напрямку (на північ, на схід, праворуч, зверху, спереду, між...);

мережеві операції (попередник, послідовник, з'єднані, шлях...);

динамічні операції (повернути, масштабувати, зсунути, розділити, злити...).

У термінах об'єктно – реляційних баз даних просторова модель даних реалізується визначенням просторових типів даних і операцій над об'єктами цих типів.

У зв'язку з цим чимало робіт із ПБД було спрямовано на розробку абстрактних типів даних (АТД) і їх впровадження в мови запитів. Завдяки створенню консорціуму Open Geospatial Consortium Inc. (OGC) вдалося серйозно просунути у створенні стандартів із геопросторових технологій [270, 272]. Зокрема, OGC представив специфікацію [272] вмонтування в SQL двомірних просторових АТД на основі об'єктної моделі, а також запропонував вичерпний список операцій. У монографії [269] подано глибокий аналіз проблематики ПБД.

Просторові типи даних. Просторові типи даних уможливають моделювання об'єктів у просторі, а також їхні взаємозв'язки, властивості й операції. Вони становлять особливий інтерес у ПБД [269, 280, 281]. Більшість найпопулярніших абстракцій просторових об'єктів належить до класу структурних просторових типів даних. Ці типи даних представляють простір у вигляді точок, ліній, областей, поверхонь, об'ємів, просторових розбиттів (spatial partitions), просторових мереж та інших подібних об'єктів. Тобто просторові об'єкти розглядаються з точки зору їхньої структурної форми і просторових розмірів. Просторові типи даних для точок, ліній і областей розглядаються в [276, 278, 282 - 286], для поверхонь і об'ємів у [287], для просторових розбиттів у [288] і для просторових

мереж у [289]. Оригінальний підхід для визначення просторових типів даних пропонується в [289], названий Realm. Realm – це кінцева множина точок і лінійних відрізків, що не пересікаються. Вони можуть знаходитися у вузлах рівномірно розділеної сітки. На основі цих примітивних понять визначаються складніші структури й операції над ними.

Відношення головного напрямку (Cardinal direction relationships). Поняття «напрямок» є однією з важливих характеристик просторових систем. На алгоритмах обчислення просторових напрямків базуються ПБД і ГІС.

Відношення головного напрямку (ВГН) – це просторове відношення, що вказує на знаходження одного об'єкту відносно іншого. Воно має як кількісне, так і якісне значення. Було запропоновано низку моделей ВГН. Спершу пропонувалися моделі, які просторові об'єкти представляли точками, а напрямок визначався відповідно до нанесеної сітки [292, 293]. В проєкційних моделях сітка наносилась паралельно всім координат, а в конусній моделі – під кутом. У наступних моделях об'єкти апроксимувалися так званими «представницькими» областями, серед яких найчастіше пропонувалися обмежувальні прямокутники [298, 299]. Однак цей метод давав невірний напрямок, коли об'єкти накладені один на один, переплетені або підковоподібні [293]. Згодом були запропоновані більш точні моделі ВГН, де висхідні об'єкти представлені своїми точними фігурами, а послідальні об'єкти апроксимуються обмежувальним прямокутником. Знову таки, залежно від нанесеної сітки розрізняють моделі та конусного відношення напрямку (cone-based directional relationships) [269]. Була також пропозиція моделювати ВГН тернарними відношеннями [291]. Щодо згаданих вище моделей ВГН досліджувалися наступні завдання:

ефективне визначення відношень, які є між множинами об'єктів [291, 299, 300, 303];




обчислення інверсних відношень [290, 291, 295, 299, 304];

обчислення композиції двох або більше відношень [295, 299, 301, 304];


перевірка узгодженості множин відношень [295, 296, 297, 299, 302].


Оригінальне вирішення з моделювання напрямку у вигляді просторового об'єкту подане в [305].


Концептуальне моделювання. Загальноновизнаним засобом концептуального моделювання є ER-мова. Пропонувалося чимало розширень цієї мови для представлення в ній просторових об'єктів із просторовими характеристиками [306]. Одним з таких розширень є ER-схема з піктограмами [307]. Піктограми в ній застосовуються для зазначення типів просторових сутностей і просторових зв'язків. Базовими типами геометричних фігур є: точка, (ламана) лінія, й багатокутник. Вони мають такі піктограми об'єктів:


-  точка
-  лінія
-  багатокутник


Мульти-фігури – це множини базових фігур. Надається

-  можливість вказати кількість елементів у мульти – множині за аналогією із зазначенням потужності закінчень зв'язків у звичайній ER-мові (M : N - не менше M і не більше N).

-  *Похідні фігури* – фігури об'єкту є похідною від фігур інших об'єктів. Як от, багатокутник країни є похідним від багатокутника областей.

-  *Альтернативні фігури* - об'єкт може бути представлений кількома фігурами. Наприклад, залежно від масштабу, місто може бути представлене точкою або багатокутником.

-  *Будь-яка можлива фігура* - об'єкт може бути представлений будь-якою допустимою фігурою.

-  *Фігура - об'єкт, що визначається користувачем*, являє собою не стандартну фігуру, а визначену користувачем.

Піктограми зв'язків:

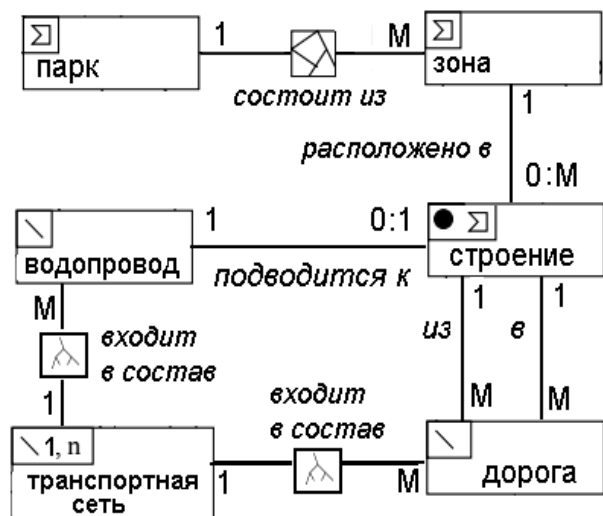


Є частиною в сенсі ієрархічної структури.



Є частиною в сенсі відношення розбиття (partition).

На рисунку нижче наведено приклад просторової ER-схеми.



Детальний опис цього розширення ER-мови наведено в працях [268, 308]. Були зроблені пропозиції щодо використання геопросторових онтологій для концептуального моделювання геоінформаційних систем [309], які б уможливили представлення у моделі ПО глибшу просторову семантику. Крім того, пропонується розширити геопросторові онтології темпоральними характеристиками з використанням OWL-Time [310].

Ці дослідження із просторових онтологій поділяються на дві категорії:

- онтології для інтеграції структур збереження просторових даних. В такому разі вирішення задачі інтеграції різних ПБД з метою вирішення задачі обміну даними між ними [311];

- онтології для більш точного представлення семантики даних. У цьому випадку просторова онтологія створюється або для відображення семантики конкретної предметної області з характерним саме для неї набором об'єктів і понять [312], або ж створюється певна універсальна онтологія, що охоплює максимально широке коло понять і характеристик геопросторових об'єктів [313 - 315].

Величезний обсяг геоінформаційних ресурсів у Інтернеті ініціював дослідження зі створення геопросторового семантичного веба [316, 317]. Так, приміром, була висунута пропозиція геопросторового розширення RDF (GeoRDF) [318].

Просторово – мережеві бази даних (ПМБД). Вони призначені для підтримки просторових мереж через надання необхідних моделі даних, мови запитів, структури зберігання і методів індексування. Модель просторової мережі може бути у вигляді графа, вершинами якого є точки у просторі. Характерними задачами, які вирішуються в ПМБД, є знаходження шляху між двома вершинами, який відповідає вказаним обмеженням.

Так само, як і в звичайних БД, в ПМБД виділяють три рівня моделювання: концептуальний, логічний та фізичний.

Завдання концептуальної моделі – адекватно представити численні об'єкти, їхні зв'язки, властивості й обмеження. Для цього пропонується використовувати піктографічну ER-модель [307] із завантаженими необхідною інформацією вершинами і ребрами. Досконалішою є транспортна модель даних UNETRANS [319].

Логічна модель даних передбачає використання моделі конкретної комерційної СУБД. Зазвичай застосовують об'єктно – реляційну модель. У праці [269] описано спеціальні операції на графах, які використовуються в ПСБД. Ще одна логічна модель системи GraphDB описана в роботі [320].

Фізична модель даних пов'язана із конкретною реалізацією ПМБД. На цьому рівні вирішуються задачі структур зберігання, методів індексування і доступу, управління пам'яттю тощо. Тут застосовують такі структури, як мережевий граф, матриця суміжності, список суміжності [269]. У праці [321] пропонується метод доступу ССАМ. У статтях [322, 323] обговорюються питання виконання таких стандартних запитів у ПМБД, як «найкоротший шлях», «найближчі пари», «найближчий сусід». Важливим транспортним обмеженням для просторової мережі є так звані «обмеження на виконання поворотів». Якщо їх не враховувати, то можлива

побудова нездійснених шляхів. Питанню побудови шляхів із урахуванням обмежень на повороти присвячені статті [324 - 327].

Просторово – часові мережеві БД.

Практично всі транспортні задачі на мережах залежать від часу доби, коли вони вирішуються. Іншими словами, просторово – мережеві моделі залежать від часу. Тому для вирішення транспортних задач необхідно застосовувати просторово – часові моделі мереж. У цьому напрямку також проводяться дослідження. Так, зокрема, для моделювання просторово – часових мереж у статті [328] пропонується робити копії всієї просторової мережі для кожного необхідного моменту часу, а в працях [329, 330] - зв'язувати з усіма вершинами і ребрами мінливі у часі атрибути.

Просторово – часові бази даних (ПЧБД). Вони представляють еволюцію в часі просторових об'єктів. Така еволюція може бути дискретною або безперервною в часі. У разі безперервного часу говорять про рухливі об'єкти, тож у зв'язку з цим вводять поняття рухливих точок, ліній, багатокутників.

Об'єкти, які переміщуються, мають свої операції, функції і предикати. Було висунуто декілька підходів для моделювання дискретних змін просторових об'єктів. Один з них – впровадження в темпоральні БД просторових типів даних [331]. Інший підхід [332] – залишити просторові об'єкти без змін, але доповнити кожну компоненту об'єкту (до прикладу, точку чи сегмент) темпоральною характеристикою. Просторово – часові типи даних дозволяють описувати динамічну поведінку просторових об'єктів у часі [333].

Було запропоновано просторово – часову мову запитів STQL [334] на базі SQL. У [277] також подається варіант просторового SQL. У [334 - 336] пропонуються просторово – часові предикати. У праці [337] подається огляд сучасного стану досліджень ПЧБД.

БД переміщуваних об'єктів (БДПО). Це просторово – часова база даних, призначена для фіксації та відстеження місцезнаходження об'єктів, що рухаються.

Дослідження БДПО були ініційовані в кінці 90-х років минулого стріччя [333, 338, 339, 340, 341]. Зазвичай БДПО послуговуються пласкою просторово – мережевою моделлю даних [342, 343].

Два дослідження практично дали життя цьому напрямку. По-перше, була запропонована модель MOST (Moving Objects Spatio-Temporal) [338, 339], яка дозволила відслідковувати в БД набір залежних від часу місцезнаходжень, наприклад, рух транспортного засобу. Було введено поняття динамічного атрибуту й визначена мова запитів FTL (Future Temporal Logic), яка дозволяла специфікувати змінні в часі взаємозв'язки між очікуваними місцезнаходженнями рухомих об'єктів. Нарешті були висунуті рішення щодо обліку невизначеності під час обробки запитів.

Другою важливою подією цього часу було відкриття 1996 року Європейського проекту Chorochronos [340], де була здійснена спроба інтегрувати концепції просторових і часових баз даних. У цьому проєкті для представлення переміщуваних об'єктів була запропонована так звана модель обмежень (constraint model) [344, 345] і розроблений прототип DEDALE [341].

Розрізняють два види БДПО: в першому випадку БДПО моделює, представляє і дає можливість формулювати запити до передісторії переміщення для проведення наступного просторово – часового аналізу [346, 347]. Другий різновид дозволяє моделювати, прогнозувати і робити запити на поточне і майбутнє переміщення [338, 348]. В цьому випадку доводиться обирати між неточністю прогнозних результатів і витратами на оновлення БД [339], що дає рішення задачі управління невизначеністю [349].

Поширеним підходом у дослідженнях із БДПО є створення спеціальних типів даних (точки, області й багатокутники, що переміщуються), а також спеціальних операцій і предикатів. До прикладу, в [333] пропонуються типи даних, які дозволяють задавати залежні від часу просторові об'єкти та операції над ними. Система типів даних для рухомих об'єктів була строго визначена в праці [350].

На завершення підкреслимо, що в монографіях [335, 351, 352] детально висвітлені практично всі питання, дотичні до рухомих об'єктів.

Просторові СУБД.

Чимало розповсюджених комерційних СУБД підтримують роботу з просторовими даними.

Серед реляційних СУБД до них належать: Oracle Database Spatial, MS SQL Server 2008, DB2 Spatial Extender, Informix Spatial Blade, MySQL Spatial, Spatial Query Server корпорації Boeing, розширення PostGIS СУБД PostgreSQL, розширення SpatiaLite для SQLite.

Серед NoSQL-систем підтримка просторових даних реалізована в MongoDB, RethinkDB, Cassandra.

Дедуктивні бази даних.

Із зростанням обсягів інформаційних ресурсів загострюється проблема їх розуміння й інтерпретації, особливо, якщо це стосується складних предметних областей. Для вирішення цієї проблеми необхідно мати механізми підтримки міркувань, аби робити складні висновки. Тож для цього почали залучати математичну логіку.

В кінці 70-х років почалося формування підходів до використання апарату логіки в базах даних [5, 6]. 1982 року Чандра і Харел [353] опублікували статтю, яка вважається першою працею в галузі теорії дедуктивних баз даних (ДБД). ДБД як один із напрямків теорії баз даних, почали стрімко розвиватися в середині 80-х років минулого століття. Висхідним пунктом появи і становлення ДБД була теорія логічного програмування, зокрема, Prolog.

ДБД – це результат об'єднання логічного програмування із реляційними базами даних. ДБД виразніші за реляційні бази даних, але менш виразні ніж системи логічного програмування. ДБД – це система баз даних, здатна робити висновки на основі правил і фактів, які зберігаються в базі даних. ДБД являє собою базу фактів і базу правил, де перша в теорії ДБД має назву екстенсійна база даних (ЕБД), а друга називається інтенсійною базою даних (ІБД). ЕБД подається у вигляді реляційних

відношень, а ІБД являє собою підмножину Prolog: без функціональних символів і спеціальних предикатів типу cut і var.

ІБД – це сукупність правил, котрі з логічної точки зору виглядають як хорновські диз'юнкти. Такі правила мають форму «якщо А, то В», де А називається «тілом», В – «головою». Тіло складається з кон'юнкції літералів (підцілей). Літераль – це атом або його заперечення. Атом – це предикат, який містить змінюваності або константи.

В багатьох статтях та енциклопедіях поняття ДБД і мова Datalog розглядаються як синоніми. Вважається, що найперша опублікована згадка терміну «Datalog» була у 1985 році в рукопису [354]. Пізніше термін «Datalog» згадувався у працях [355, 356]. Однак усе ж заведено вважати, що офіційно мова Datalog уперше була досліджена в книзі Майєра (David Maier) і Уоррена (David S. Warren) [7] як спрощений варіант Prolog без функціональних символів. Автори цю назву пояснювали тим, що предикати без функціональних символів нагадують відношення бази даних.



Давід Майєр

Зауважимо, що 1985 року термін Datalog також був вжитий як мова запитів до бази даних в системі, що підтримує природну мову [357]. Цей термін являв собою скорочення від «database dialogue» і ніякого відношення до Prolog не мав.

У будь – якому разі до середини 80-х років термін Datalog ствердився як мова дедуктивних даних, а не як мова логічного

програмування. 1989 року була опублікована чудова монографія Джеффри Ульмана (Jeffrey D. Ullman) [6], де окремий розділ був присвячений детальному викладу суті Datalog як логічної моделі даних. Багато чого з цього розділу використовується згодом у переліку принципів Datalog.



Джеффри Ульман

Безпечне правило. Щоб Datalog-правила інтерпретувалися операціями над кінцевими відношеннями, змінні правила мають бути обмежені, тобто мають знаходитися принаймні в одному атомі (літералі без заперечення). Правило безпечне за умови, якщо всі його змінні обмежені. На цей аспект було звернуто увагу в працях [358, 359].

Розрізняють три види правил:

прості – без рекурсії та заперечень;

рекурсивні – які містять рекурсивні визначення;

із запереченнями – які містять атомарні формули із запереченнями.

Прості правила. Для простих правил існує спосіб їх перетворення у вирази реляційної алгебри (див., наприклад, [6]). Отримані вирази визначають відношення для предикатів ІБД і являють собою єдину мінімальну модель.

Рекурсивні правила. Для рекурсивних Datalog-програм, що не містять заперечувальних під цілей, існує єдина мінімальна модель, яка містить задані ЕБД-відношення, і ця модель є єдиною мінімальною нерухомою точкою відносно ЕБД-відношень відповідних Datalog-правил. Відомо, що у разі, якщо функція монотонна, то її рекур-

сивне обчислення приводить до нерухомої точки. В реляційній алгебрі всі операції, за винятком різниці, є монотонними. Тому в реляційній алгебрі, розширеній циклами, можливо побудувати алгоритм, що обчислює найменшу нерухому точку рекурсивної Datalog-програми. В [6, 360] подані огляди з оптимізації рекурсивних запитів у ДБД.

Правила із запереченнями. Правила можуть мати в тілі заперчувальні підцілі. В такому разі виникають дві проблеми. По-перше, заперечення можуть викликати нескінченні інтерпретації. Тому було розширено вимогу існування безпекових правил із запереченнями – змінні, що зустрічаються в заперечуваних підцілях, обов'язково мають бути в під цілях без заперечень. Друга проблема пов'язана з тим, що за наявності заперечень Datalog-програма може мати безліч мінімальних моделей (мінімальних нерухомих точок). Тоді зміст програми із запереченнями задається вибором певної «переважаючої» моделі [361 - 368]. Заперечення викликають також проблеми в рекурсії. Через це було введено поняття стратифікованого заперечення [361, 362, 369, 370] і стратифікованих програм, що мають інтуїтивно зрозумілу семантику [371 - 374]. Стратифікація не гарантує існування найменшої нерухомої точки. Але це обмеження уможливорює вибір серед численних мінімальних нерухомих точок такої, яка буде інтерпретацією смислу Datalog-програми. Були також визначені більш специфічні класи стратифікованих програм, зокрема, локально стратифіковані програми [375], модульно стратифіковані програми [376].

Оптимізація. Однією з найскладніших проблем створення ДБД є оптимізація. Для нерекурсивних правил проблема оптимізації аналогічна традиційній реляційній оптимізації, а за наявності рекурсії і заперечення виникають додаткові проблеми і можливості. У цьому напрямку було здійснено численні дослідження. Серед них виділимо метод магічних множин (magic-sets) [6], а також низку методів, що на ньому базуються [377], алгоритм підрахунку (counting algorithm) [378], факторингову оптимізацію (factoring optimization) [379], метод видалення зайвих правил і літералів [380], метод оптимізації екзистенційних запитів [381],

метод «конвертів» (envelopes) [382] тощо. У праці [383] здійснено аналітичний огляд різних стратегій оптимізації і порівняльний аналіз їхньої продуктивності.

Дедуктивні системи баз даних.

Щодо створення дедуктивних систем баз даних, то тут можна виділити два напрямки. З одного боку проводилися дослідження й розроблялися проєкти зі створення самостійних систем. Із їх оглядом можна ознайомитися в [384, 385]. З іншого боку, 1999 року в чергову версію SQL (SQL99) було закладено можливість формулювати й виконувати рекурсивні запити. Як уже було вище згадано, прості правила й спеціальні правила із запереченнями повністю виражаються в реляційній алгебрі, а, отже, й в стандартному (без рекурсії) SQL. Введення рекурсії в SQL привело до того, що всі можливості дедуктивних Datalog-програм почали виявлятися в SQL99. Додатково виникали можливості вказувати напрямок пошуку (вшир, вглиб), фіксації та запобігання безкінечних циклів, створювати рекурсивні представлення, визначати пряму і взаємну рекурсію, лінійну і нелінійну рекурсію. З робочим варіантом стандарту рекурсивного SQL можна ознайомитись в [386], а з коротким списком практичного використання – в [387].

Слід зазначити, що до кінця 80-х років сформувався напрямок об'єктно-дедуктивних баз даних. Для них були розроблені спеціальні мови, як от O-Logic, F-Logic, ROL, IQL [388, 389]. У статті [390] подається аналітичний огляд об'єктно-дедуктивних баз даних.

Активні бази даних.

Традиційні бази даних пасивні. Дані розміщуються, оновлюються, переносяться і вибираються їхніми БД під впливом зовнішніх чинників (людина або програма). Бізнес-правила, що застосовуються до вмісту бази даних, також, керуються як правило, зовнішніми джерелами. Коротше кажучи, традиційні бази даних не є активними учасниками функціонування інформаційної системи і забезпечують лише функцію зберігання даних. Для подолання цього недоліку було введено концепцію активних баз даних. Активна база даних (АБД) – це база даних, стосовно якої

СУБД виконує не лише дії, що їх явно вказує користувач, а й додаткові дії, згідно із правилами, закладеними в саму БД.

Зародження ідей АБД пов'язують із появою концепції тригеру-механізму, який уперше був запропонований у дослідницькому проєкті System R компанії IBM. Підтримка концепції тригеру передбачалася в мові цієї системи SEQUEL. Однак, варто зазначити, що ідея тригеру раніше була втілена в мові визначення даних CODASYL [33, 34], (хоча сам термін «тригер» ще не вживався). В мові передбачалася підтримка концепції процедури бази даних, яка може асоціюватися з різними об'єктами бази даних у специфікації схеми. Процедура бази даних запускається автоматично в разі, якщо над об'єктом, з яким вона асоційована, виконується одна з даних у специфікації операцій. Водночас виконання процедури може передувати виконанню заданої операції, відбуватися після неї або мати місце у разі виникнення помилки.

АБД має передбачати підтримку таких можливостей:

- містити логіку обробки даних (бізнес-правила) власне в базі даних, щоб вона управлялася через СУБД, а не прикладними програмами чи користувачами;
- забезпечити моніторинг подій і умов, які впливають на дані й можуть ініціювати обробку даних, що ними управляє СУБД;
- містити спосіб, із яким ці події й умови могли б запускати логіку обробки даних всередині бази даних.

ЕСА – правило. Для підтримки згаданих вище можливостей в активну базу даних було введено поняття ЕСА-правила – це конструкція із трьох складників: подія, умова й дія (Event-Condition-Action). Вперше воно було визначено в проєкті HiPAC (High Performance ACtive database system) [392]. Семантика правила проста: коли відбувається подія, перевіряється умова, і, в разі її істинності, відбувається дія.

Умовою ЕСА-правила можуть бути: запит до бази даних, логічне вираження, виклик підпрограми (процедури або функції), що повертає логічне значення).

Дія ЕСА-правила – довільний код, викликаний появою події та за істинності

умови. Дія виконується або у вигляді складової частини транзакції ЕСА-правила, або у вигляді самостійної транзакції залежно від режиму зв'язування. В межах одного ЕСА-правила може виконуватись кілька дій одночасно, тому варто відслідковувати конфліктні ситуації. Тіло дії може ініціювати подія, котрі викликають виконання іншого правила тощо. І, зрештою, ланцюжок послідовно ініційованих вкладених правил може бути рекурсивним.

Моделі ЕСА-правил. Для опису ЕСА-правил були висунуті дві моделі: модель знань (knowledge model) і модель виконання (execution model) [391, 392]. Модель знань описує, чим є активні правила, а модель виконання специфікує, яким чином інтерпретуються ЕСА-правила в процесі їх виконання.

Модель знань ЕСА-правил. У праці [393] було узагальнено і класифіковано характеристики моделі знань, що раніше були представлені в літературі [394 - 396]. Ці характеристики відносяться до всіх складових ЕСА-правил і наводяться далі.

Характеристики моделі знань події. До них належать наступні:

- *Джерела подій* (event sources): операції над структурними елементами бази даних (structure operations), поведінка зовнішнього середовища (behaviour invocation – дії користувачів або прикладних програм), команди транзакцій (begin, abort, rollback commit), часові характеристики;

- *Виявлення подій* – це процес аналізу потоку подій для знаходження тих, що відповідають заданому зразку. Зазвичай виявлення подій включає процедури фільтрації й агрегації. Основоположні дослідження із виявлення подій були здійснені під час виконання проєктів HiPAC [392, 397], Snoop [398, 399], ODE [400], SAMOS [401]. В [402] подано огляд досліджень із виявлення подій, а в статті [403] – специфікацій подій.

- *Гранулярність подій* (event granularity) – вказує, чи визначається подія для кожного об'єкту з множини, або з даної підмножини, або конкретних об'єктів множини.

- *Прості й складові події.* Подія, що об'єднує кілька подій, називається складовою. Піонерською працею в сфері складових подій вважається проєкт HiPAC [392].

Для опису складових подій у межах розроблених систем були визначені різні алгебри, як-от [395, 398, 399, 401, 404]. У праці [398] для складових подій введена характеристика «політика споживання» (consumption policy). Вона визначає ситуацію, де, як вважається, складова подія відбулася. В праці [405] подається загальний метод і мова EPL специфікації семантики складових подій.

Характеристики моделей знань умови. До них належать:

- *факультативність* – чи є умова обов'язковою в ЕСА-правилі, чи ні. Вважається, що в правилі має бути присутня або подія, або умова;

- *контекст* – загалом як контекст ЕСА-правила пропонуються такі варіанти: стан бази в період – запуску транзакції (DBT), -ініціювання події (DBE), - перевірки умови (DBC), - виконання дії (DBA), а також прив'язка умови до події (BindE), а також дії до умови (BindC). За контекст умови беруться DBT, DBE, DBC и BindE.

Характеристики моделей знань дії. До них належать:

- *види дій.* Пропонуються такі: робота зі структурою бази даних, ініціювання зовнішнього середовища, інформування, аварійне завершення, виконання іншої дії чимось, що ініціювало подію («do-instead» Стоунбрейкера [394]);

- *контекст* – аналогічно контексту умови специфікує, які саме дані доступні дії. Допустимими значеннями є DBT, DBE, DBC, DBA и BindC.

Модель виконання ЕСА-правил. Модель виконання (execution model) специфікує, яким чином трактуються ЕСА-правила в процесі їх виконання [391]. Вона стосується подій, умов і дій ЕСА-правил.

Модель виконання подій. Якщо активна база даних підтримує виявлення складових подій, то необхідні правила їх виявлення й відпрацювання. Для вирішення цієї ситуації були запропоновані так звані «режими споживання подій» (event consumption modes) [398, 404, 406]. Далі подаються характеристики моделі виконання умов і дій ЕСА-правил.

Правила перевірки умов і виконання дій. Режими зв'язування (Coupling modes). Вони визначають, як ініціюється

перевірка умови у відповідь на подію, що відбулася, і як планується, контролюється і виконується дія ЕСА-правила у разі позитивного результату перевірки умови. Вперше вони були досліджені у проєкті НіРАС [397]. Визначаються вони для пар подія-умова і умова-дія. Зв'язування подія-умова визначає, коли слід перевірити умову відносно події, а умова-дія – коли слід виконати дію щодо умови. Для обох видів зв'язування було запропоновано однакові варіанти зв'язування:

– *негайно* (immediately) – умова/дія перевіряється/виконується одразу ж після події/умови;

– *відстрочено* (deferred) – перевірка/виконання умови/дії відтермінується до завершення транзакції (до виконання Commit), де ініційований тригер. Були також запропоновані варіанти, коли відстрочка задавалася визначеним користувачем часом відстрочки [407] або ж виконанням спеціальних команд [396];

– *окремо* (detached) – перевірка/виконання умови/дії здійснюється в іншій транзакції, ніж подія/умова, причому виконання дії може залежати або бути незалежним від завершення транзакції, де відбулася подія чи було перевірено умову.

У джерелі [397] було встановлено, що в тригері не всі варіанти пар значень режимів зв'язування є допустимими. Зазначимо також, що в дослідницькому проєкті REACH (REal-time ACtive Heterogeneous System) [408] було висунуто ще два варіанти режиму зв'язування для підтримки побічних ефектів незворотніх дій ЕСА-правил.

Запуск подією кількох правил. Можлива ситуація, коли подія ініціює запуск кількох правил. У такому разі було запропоновано механізми планування порядку виконання правил [391, 394, 409].

Політика підсумкового ефекту (net effect policy). Для випадку, якщо в межах одного правила виконується кілька дій відносно одних і тих самих даних, пропонується політика підсумкового ефекту [391], коли виконується лише одна дія або навіть не виконується жодна.

Виклик правилом іншого правила. Правило може ініціювати виклик іншого правила тощо. При цьому виникають ситу-

ації, коли виконання внутрішнього правила суперечить виконанню зовнішнього правила. А також можливі цикли, коли правило ініціює виконання самого себе. Ці ситуації також розглядаються в літературі [391].

Системи активних баз даних. Розроблено системи активних реляційних баз даних (РБД) і об'єктно-орієнтованих баз даних (ООБД).

Активні РБД. Включення активних механізмів до РБД не є чимось новим. Переважна більшість комерційних систем підтримують механізми тригерів. Окрім того, досліджуються розробки розвинутіших засобів підтримки активних правил. Пропозиції із включення активної поведінки в реляційні системи, зазвичай, обмежуються можливостями традиційних пасивних реляційних систем. Приміром, подіями, які ініціюють правила, є лише операції над даними (вставка, видалення, заміна). Як правило, в реляційних системах не розглядаються складові події. Вони не мають розвинутих режимів зв'язування, і мова опису правил вбудовується в мову запитів. Серед «ранніх» реляційних систем, що мали механізми активації, можна виділити Starburst [396, 410], Postgress [411], Ariel [412]. Прикладами активного розширення реляційної моделі даних можуть бути праці [413 - 419]. Серед них особливий інтерес становлять ті, що досліджують взаємозв'язок активних і дедуктивних баз даних [415 - 419].

У стандарт SQL 1999 року були включені тригери [420]. Відтоді всі промислові реляційні СУБД як механізм активних правил включають щонайменше тригери SQL.

Активні ООБД. Щодо ООБД, то на відміну від реляційних, вони завжди підтримували тісний зв'язок між поведінкою користувачів і даними БД. Така поведінка представляється методами, приписуваними класам даних БД. Цей факт, а також інкапсуляція структури об'єкту, вказують, що деякі аспекти, котрі можуть представлятися в РБД за допомогою активної поведінки, в ООБД підтримуються за допомогою методів. Однак дослідження з активного розширення ООБД почалися практично одночасно з появою самих ООБД. А їхня суттєва відмінність полягає в тому, що в активних ООБД примітивні події часто асоціюють-

ся з викликами методів, а не з операціями над структурними елементами БД. Розроблена чимала кількість систем активних ООБД, серед яких можна відзначити HiPAC [392, 395, 421], EXACT [407], NAOS [422], Chimera [423], Ode [424], SAMOS [401], Sentinel [406], REACH [408]. Зі стислим описом цих систем можна ознайомитись у [391], там же є посилання на інші праці з активного розширення ООБД.

Система HiPAC. На завершення значимо, що HiPAC стала однією з передових систем активних баз даних свого часу і єдиною, орієнтованою на потреби додатків реального часу. Це привело до створення інноваційної моделі тригеру. Модуль ЕСА-правил, поданий в HiPAC, нині широко застосовується в активних обчислювальних системах, у системах обробки складних подій і в розподілених системах. Керівник проєкту HiPAC Умешвар Дайал (Umeshwar Dayal) 2010 року став лауреатом інноваційної премії SIGMOD імені Едгара Ф. Кодда за піонерські праці і суттєвий внесок у розподільчі гетерогенні бази даних, високопродуктивні активні бази даних, моделі довгочасних транзакцій та дослідження в галузі бізнес-процесів.



Умешвар Дайал

Об'єктні бази даних.

Поява напрямку об'єктних баз даних (ОБД) визначалася, передовсім, потребами практики: необхідністю розробки складних прикладних систем, для яких технологія попередніх систем баз даних не була цілком за-

довільною. Дослідження в галузі ОБД почалися в зв'язку з необхідністю розробки ефективного механізму, що дозволяв би об'єктно-орієнтованим додаткам зберігати об'єкти по закінченні своєї роботи і користуватися ними під час наступного запуску. Тобто необхідно було об'єктно-орієнтованому середовищу надати прозорий механізм збереження й вибірки об'єктних даних з баз даних.

ОБД виникли не на порожньому місці. Відповідний базис забезпечувався працями в галузі баз даних, напрямками мов програмування з абстрактними типами даних і об'єктно-орієнтованих мов програмування.

Перші об'єктні СУБД. На початку 80-х років чимало дослідницьких груп з університетів, наукових інститутів, провідних комп'ютерних компаній і невеликих фірм – початківців взяли за створення ООСУБД. Було випущено перші промислові ООСУБД G-Base (1986 г.), Gemstone (1987 г.), IRIS (1987), Stattice (1988 г.), Vbase (1988 г.), ObjectStore (1988 г.), Versant (1988 г.), O2 (1988 г.), ORION (1989 г.).

Два напрямки в ОБД. До кінця 80-х років визначилися два напрямки у створенні об'єктних баз даних: об'єктно-орієнтовані бази даних (ООБД) і об'єктно-реляційні бази даних (ОРБД). ООБД користуються об'єктно-орієнтованою мовою програмування як мовою бази даних і забезпечує цілісність об'єктів із наданням усіх функціональних можливостей, властивих традиційним базам даних. ООБД передбачають створення самостійних об'єктно-орієнтованих систем управління базами даних (ООСУБД). ООСУБД реалізує гнучку модель, базовану на тій же парадигмі, що й об'єктно-орієнтовані мови програмування. ООСУБД забезпечують глибшу інтеграцію з об'єктно-орієнтованими додатками, ніж реляційна база даних.

У свою чергу ОРБД розширює можливості реляційних баз даних засобами підтримки об'єктів.

Маніфест об'єктно-орієнтованих систем баз даних. 1989 року група провідних спеціалістів і дослідників баз даних написали «Маніфест об'єктно-орієнтованих систем баз даних» [425] (так званий Перший маніфест). Це був перший документ, де була зроблена спроба дати визначення системам об'єктно-орієнтованих баз даних.

Були описані основні властивості і характеристики, що їх мала мати технологія ООБД.

У ньому також зазначається, що поточний стан справ у проблематиці ООБД характеризується: відсутністю загальноприйнятої моделі даних, відсутністю формальної теорії і активної експериментальної діяльності.

Загальноприйнята об'єктно-орієнтована модель даних була відсутня не тому, що не було жодної розробленої повної моделі, а через відсутність спільного узгодження щодо прийняття будь-якої моделі. Що ж до формальної теорії, то для ООБД мало б бути щось подібне до того, що створив Ковальський для логічного програмування. Необхідність такої теорії очевидна: формальна семантика основних понять ООБД визначена недостатньо. Її відсутність практично унеможливлювала досягнення консенсусу відносно моделі даних.

Дослідження у сфері об'єктних баз даних особливо активно відбувалися у 80-і роки. Це привело в кінці 80-х до виникнення промислових компаній і ринку систем управління об'єктними базами даних (СУОБД). Водночас ринок об'єктних баз даних гостро потребував стандарту. Вирішальне слово в цій, як, власне, і в інших проблемах, пов'язаних із об'єктними базами даних, було сформульоване в стандарті ODMG.

Стандарт на зберігання об'єктів ODMG 3.0. Влітку 1991 року в США була створена Object Data Management Group (ODMG) – Група Управління Об'єктними даними – як консорціум виробників СУОБД. ODMG очолив Рік Кеттелл (Rick Cattell).



Рік Кеттелл

Завданням групи була розробка стандарту на збереження об'єктів у базах даних. У період із 1993 по 2001 рік ODMG опублікувала п'ять версій своєї специфікації. Останньою з них була версія 3.0 [434], після чого група завершила свою роботу.

Стандарт на зберігання об'єктів ODMG – 3.0 розроблено на основі трьох існуючих стандартів: управління базами даних (SQL), стандарти OMG - Object Management Group і стандарти на об'єктно-орієнтовані мови програмування (C++, Smalltalk, Java). ODMG сприяє можливості взаємодії з базами даних в об'єктно-орієнтовані мови програмування. Стандарт складається з таких частин:

- *Об'єктна модель* – уніфікована основа всього стандарту. Вона розширює об'єктну модель консорціуму OMG.

- *Мова визначення об'єктів* (ODL - Object Definition Language) – засіб визначення типів об'єктів, які відповідають об'єктній моделі даних ODMG. ODL використовується для підтримки перенесення об'єктних схем між відповідними системами управління об'єктними даними (СУОД).

- *Мова об'єктних запитів* (OQL - Object Query Language) - SQL –подібна декларативна мова, яка надає ефективні засоби для отримання об'єктів із бази даних.

- *Формат обміну об'єктами* (OIF - Object Interchange Format) – мова опису завантаження і розвантаження поточного стану СУОД в/з файлів. Використовується для обміну збереженими об'єктами між СУОД.

- *Зв'язування з ОО-мовами.* Стандарт зв'язування з C++, Smalltalk і Java визначає Object Manipulation Language (OML) – мова маніпулювання об'єктами, яка розширює базові ОО-мови шляхом маніпулювання і збереження об'єктів.

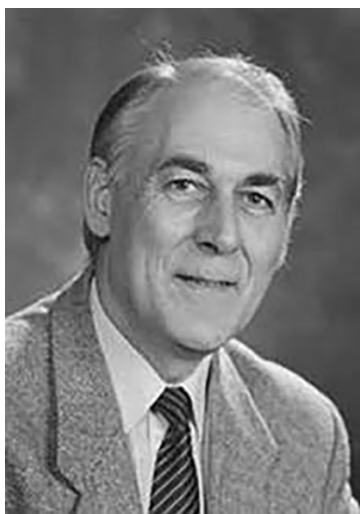
Другий маніфест. Товариство дослідників реляційних баз даних відповіло на активність у сфері ООБД своїм маніфестом на підтримку ORBD. 1990 року М. Стоунбрейкер та його колеги з комітету перспективних систем БД опублікували «Маніфест систем баз даних третього покоління» [435] (так званий Другий маніфест), де стверджувалось, що СУБД третього покоління, тобто ті, що прийдуть за реляційними, мають бути створені на основі реляційних техно-

логії. Прихильники цього напрямку дотримуються принципу еволюційного розвитку можливостей СУБД без докорінної відмови від попередніх підходів та збереження наступності з системами попереднього покоління. Цей принцип був підтриманий у процесі створення дедуктивних і темпоральних баз даних. Так само розвивалося створення об'єктно-реляційних баз даних.

На основі цієї ідеї під керівництвом М. Стоунбрейкера в університеті Берклі (Каліфорнія, США) було розроблено СУБД Postgres [436]. Це була перша практично реалізована об'єктно-реляційна система баз даних, де був продемонстрований підхід з інтеграції об'єктних і реляційних концепцій.

Варто також згадати Вона Кіма (Won Kim), який 1991 року випустив систему UniSQL [437], що також вважається однією з перших об'єктно-реляційних СУБД.

Третій маніфест. У березні 1995 року була опублікована стаття Хью Дарвена (Hugh Darwen) і Крістофера Дж. Дейта (Christopher J. Date), названа авторами «Третім маніфестом» [438].



Хью Дарвен

В ній висловлювалася думка щодо майбутніх систем управління базами даних і підхід з інтеграції реляційної та об'єктної технологій. У маніфесті піднята проблема щодо вирішення задачі невідповідності між об'єктно-орієнтованими мовами програмування і системами управління реляційними базами даних. Автори пропонують взяти за основу реляційну базу даних у вигляді визначених користувачем типів.



Крістофер Дж. Дейт

Схеми реалізації ОРБД. Пропонувалися різні схеми реалізації ОРБД. Як – от:

Об'єктно – реляційний шлюз (Object-Relational Gateway). Об'єктно – орієнтований додаток працює як звичайний користувач із об'єктною мовою, а шлюз виділяє і замінює всі об'єктно – орієнтовані елементи цієї мови на їх реляційні компоненти. Попри зниження продуктивності, такий варіант дозволяє програмістам цілком сконцентруватися на об'єктно – орієнтованій розробці.

Об'єктно – реляційний інтерфейс (Object-Relational Interface). Між ООБД і ОРБД знаходиться проміжний інтерфейс, що відображає об'єктні конструкції в реляційні і навпаки. Об'єктно – орієнтований додаток працює з ООСУБД, яка через інтерфейс взаємодіє з реляційною СУБД.

Уніфіковані СУБД (unified DBMS). Ще одним вирішенням є створення гібридних об'єктно – реляційних СУБД, здатних зберігати і традиційні табличні дані, й об'єкти.

Об'єктно – реляційні СУБД. У другій половині 90-х років провідні компанії почали випускати СУБД, які підтримували об'єктно – реляційну модель даних. Першою 1996 року з'явилася на ринку СУБД Informix, створена на основі системи Illustra Стоунбрейкера. 1997 року була випущена об'єктно – реляційна версія СУБД DB2 компанії IBM, в основі якої був дослідницький прототип Starburst IBM Almaden. Цього ж року компанія Oracle випустила продукт такого ж класу Oracle 8. Сьогодні практично всі сучасні реляційні СУБД є об'єктно – реляційними. Всі вони розширюють реля-

ційну базу даних засобами представлення об'єктів.

Позиції об'єктно – реляційного підходу зміцнилися завдяки прийнятій 1999 року версії стандарту SQL-3, де було введено підтримку об'єктно – орієнтованої концепції (структурні типи даних, типізовані таблиці, об'єкти, методи, які визначаються користувачем).

(Далі буде)

References

212. Snodgrass R.T., Ahn I. A taxonomy of time databases. ACM SIGMOD Record, 1985, Vol. 14, No 4, pp. 236-246
213. Bubenko J.A, Jr. The temporal dimension in information modeling. Technical Report RC 6187 #26479, IBM Thomal J. Watson Research Center, Nov. 1976
214. Bubenko J.A. Jr. The Temporal Dimension in Information Processing. In: Proceedings of IFIP WG 2.6 Working Conference on Architecture and Models in Data Base Management Systems, G M Nijssen, Ed, North Holland, 1977, pp. 93-118
215. Breutmann B., Falkenberg E., Mauer R. "CSL: a language for defining conceptual schemas". in Proceedings of the Database Architecture Conference, Venice, June 1979, pp. 237-256
216. Hammer M., McLeod D. Database Description with SDM A Semantic Database Model ACM Transactions on Database Systems, 6, No 3, Sep 1981, pp 351-386
217. Klopprogge M.R. TERM: An Approach to Include the Time Dimension in the Entity- Relationship Model. In: Proceedings of the Second International Conference on the Entity Relationship Approach, Washington, DC, pp. 477-512 (October 1981)
218. Anderson, T.L. Modeling Time at the Conceptual Level. In Improving Database Usability and Responsiveness, Ed. P. Scheuermann Jerusalem, Israel Academic Press, 1982, pp. 273-297
219. Codd, E.F. Extending the database relational model to capture more meaning. ACM Transactions on Database Systems, Vol. 4, No. 4, Dec 1979, pp 397-434
220. Sernadas A Temporal aspects of logical procedure definition. Information Systems, 1980, vol. 5, No 3, pp. 167-187
221. Clifford, J. and Warren D.S. Formal semantics for time in databases. ACM Transactions on Database Systems, vol. 8, No 2, June 1983, pp. 214-254
222. Ariav G. A temporally oriented data model. ACM Transactions on Database Systems (TODS), 1986 vol. 11, No 4, pp. 499-527
223. Ariav G., Morgan H.L., Zisman M.D. MDM: Embedding the time dimension in information systems, Technical Report 82-03-01 Department of Decision Sciences, Wharton School, University of Pennsylvania, 1982
224. Ben-Zvi J. "The Time Relational Model," PhD thesis, Computer Science Dept., UCLA, 1982
225. Gadia S. Ben-Zvi's Pioneering Work in Relational Temporal Databases. In: Tansel A. et al. Temporal Databases: Theory, Design, and Implementation (Redwood City, CA: The Benjamin/Cummings Publishing Company, 1993). pp. 202-207
226. Jones S., Mason P.J. Handling the Time Dimension in a Data Base. In Proceedings of the International Conference on Data Bases, Eds. S.M. Deen and P Hammersley British Computer Society University of Aberdeen, Heyden, July 1980 pp 65-83
227. Snodgrass R. The temporal query language TQuel. In PODS '84: Proceedings of the 3rd ACM SIGACT-SIGMOD symposium on Principles of database systems,
228. Allen J.F. "Maintaining knowledge about temporal intervals". Communications of the ACM, Nov. 1983, 26(11). pp.832-843
229. Clifford J., Dyreson C.E., Isakowitz T., Jensen C.S., Snodgrass R.T. On the semantics of "now.". ACM Trans Database Syst. 1997; 22(2), pp.171-214.
230. Lorentzos N.A., Johnson R.G. TRA a model for a temporal relational algebra. In: Rolland C, Bodart F, Leonard M, editors. Temporal aspects in information systems. North-Holland; 1988. p. 203-215.
231. Lorentzos N.A., Johnson R.G. Extending relational algebra to manipulate temporal data. Inf Syst. 1988;13(3):289-296.
232. Tansel A.U. Adding time dimension to relational model and extending relational algebra. Information Systems. 1986;11(4):343-355.
233. Tansel A.U. Temporal relational data model. IEEE Transactions on Knowledge and Data Engineering, 1997, 9(3): 464-479

234. Tansel A.U., Atay C.E. Nested bitemporal relational algebra. Conference: Computer and Information Sciences - ISCIS 2006, 21th International Symposium, Istanbul, Turkey, November 1-3, 2006, Proceedings, pp. 622-633
235. Navathe S.B., Ahmed R. A temporal relational model and a query language. *Information Sciences: an International Journal*. 1989;49(1-3):147-175.
236. Gadia S.K. A homogeneous relational model and query languages for temporal databases. *ACM Trans Database Syst*. 1988;13(4):418-448.
237. Clifford J, Croker A. The historical relational data model (HRDM) and algebra based on lifespans. In: *Proceedings of the 3rd International Conference on Data Engineering*; 1987. p. 528-537.
238. Gregersen H., Jense C.S. Temporal Entity-Relationship Models—a Survey. *IEEE Transactions on Knowledge and Data Engineering*, 1999, Vol. 11, No. 3, pp. 464 - 497
239. Arora S. A comparative study on temporal database models: A survey, 2015 International Symposium on Advanced Computing and Communication (ISACC), 2015, pp. 161-167,
240. Gandhi L. Literature survey of temporal data models. *International Journal of Latest Trends in Engineering and Technology*. 2017, Vol. 8 No. 4-1, pp.294-300
241. Vianu V. Dynamic functional dependencies and database aging. *J ACM*. 1987;34(1):28-59.
242. Wijzen J. Design of temporal relational databases based on dynamic and temporal functional dependencies. In: Clifford J, Tuzhilin A, editors. *Temporal databases. Workshops in computing*. Berlin/Heidelberg/New York: Springer; 1995. p. 61-76.
243. Wijzen J. Temporal FDs on complex objects. *ACM Trans Database Syst*. 1999;24(1):127-176.
244. Wijzen J. Reasoning about qualitative trends in databases. *Information Systems*. 1998;23(7):463-487.
245. Wang X.S., Bettini C., Brodsky A., Jajodia S. Logical design for temporal databases with multiple granularities. *ACM Trans Database Syst*. 1997;22(2):115-170.
246. Baudinet M., Chomicki J., Wolper P. Constraint generating dependencies. *J Comput Syst Sci*. 1999;59(1):94-115.
247. Jensen C.S., Snodgrass R.T. Temporal specialization and generalization. *IEEE Trans Knowl Data Eng*. 1994;6(6):954-974
248. Sarda N.L. Algebra and query language for a historical data model. *The Computer Journal*. 1990;33(1):11-18
249. Lorentzos N.A., Johnson R.G. Extending relational algebra to manipulate temporal data. *Information Systems*. 1988;13(3):289-296.
250. Tuzhilin A, Clifford J. A temporal relational algebra as basis for temporal relational completeness. In: *Proceedings of the 16th International Conference on Very Large Data Bases*; 1990. p. 13-23.
251. Tansel A.U., Arkun M.E. HQUEL, a Query Language for Historical Relational Databases. *SSDBM'86: Proceedings of the 3rd international workshop on Statistical and scientific database management*, 1986 pp. 135-142
252. Tansel A.U., Arkun M.E, Ozsoyoglu G. Time-by-example query language for historical databases. *IEEE Trans Softw Eng*. 1989;15(4):464-478.
253. Snodgrass S. The temporal query language TQUEL. *ACM Trans Database Syst*. 1987;12(2): 247-298.
254. Lorentzos N.A., Mitsopoulos Y.G. SQL extension for interval data. *IEEE Trans Knowl Data Eng*. 1997;9(3):480-99.
255. Sarda N.L. Extensions to SQL for historical databases. *IEEE Trans Knowl Data Eng*. 1990;2(2):220-230.
256. Navathe S.B., Ahmed R. TSQL: a language interface for history databases. In: Rolland C, Bodart F, Leonard M, editors. *Temporal aspects in information systems*. North-Holland; 1988. p. 109-122.
257. Toman D. Point-based temporal extensions of SQL and their efficient implementation. In: Etzion O, Jajodia S, Sripada S, editors. *Temporal databases: research and practice*. Springer; 1997, p. 211-237.
258. Böhlen M.H., Jensen C.S., Snodgrass R.T. Temporal statement modifiers. *ACM Trans Database Syst*. 2000;25(4):407-456.
259. Snodgrass R.T. editor. In: *Proceedings of the ARPA/NSF International Workshop on an Infrastructure for Temporal Databases*, 1993.
260. Snodgrass R.T., Ahn I., Ariav G., Batory D.S., Clifford J., Dyreson C.E., Elmasri R., Grandi F., Jensen C.S., Käfer W., Kline N., Kulkarni K., Leung T.Y.C., Lorentzos N., Roddick J.F.,

- Segev A., Soo M.D., Sripada S.M. TSQL2 language specification. *ACM SIGMOD Rec.* 1994;23(1):65–86.
261. Snodgrass R.T., Ahn I., Ariav G., Batory D.S., Clifford J., Dyreson C.E., Elmasri R., Grandi F., Jensen C.S., Käfer W., Kline N., Kulkarni K., Leung T.Y.C., Lorentzos N., Roddick J.F., Segev A., Soo M.D., Sripada S.M. A TSQL2 tutorial. *ACM SIGMOD Rec.* 1994;23(3):27–33
262. Snodgrass R.T. Editor. *The TSQL2 temporal query language.* Kluwer Academic; 1995.
263. Snodgrass R.T., Böhlen M.H., Jensen C.S., Steiner A. Adding valid time to SQL/temporal. Change proposal, ANSI X3H2-96-501r2, ISO/IEC JTC1/SC21/WG3 DBL MAD-146r2, Nov 1996.
264. Snodgrass R.T., Böhlen M.H., Jensen C.S., Steiner A. Adding transaction time to SQL/temporal. Change proposal, ANSI X3H2-96-502r2, ISO/IEC JTC1/SC21/ WG3 DBL MAD-147r2, Nov 1996.
265. Snodgrass R.T., Böhlen M.H., Jensen C.S., Steiner A. Transitioning temporal support in TSQL2 to SQL3. In: Ezion O, Jajodia S, Sripada SM, editors. *Temporal databases: research and practice.* Berlin: Springer; 1998. p. 150–194.
266. Kulkarni K, Michels J-E. Temporal features in SQL:2011. *ACM SIGMOD Rec.* 2012;41(3):34–43.
267. Reznichenko V.A. *Temporal SQL:2011 (Rus).* Software Engineering, 2013, vol. 15, No 3-4, pp. 48-65
268. Worboys MF, Duckham M. *GIS: a computing perspective.* Boca Raton: CRC press; 2004.
269. Shekar S, Chawla S. *Spatial databases: a tour.* Englewood Cliffs: Prentice-Hall; 2003
270. The Open Geospatial Consortium Date: 2011-12-19 OGC Reference Model. 44 p.
271. Open Geospatial Consortium Inc. Date: 2011-05-28 Editor: John R. Herring *OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture.* 93 p.
272. Open Geospatial Consortium Inc. Date: 2010-08-04 Editor: John R. Herring *OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option.* 111 p.
273. Tomlin C.D. A map algebra. In: *Proceedings of the Harvard Computer Graphic Conference;* 1983
274. Chan K.K.L., Tomlin C.D. Map Algebra as a Spatial Language. In D. M. Mark and A. U. Frank, editors, *Cognitive and Linguistic Aspects of Geographic Space,* pp. 351–360. Kluwer Academic Publishers, Dordrecht, 1991
275. Scholl M., Voisard A. Thematic map modeling. In: *Proceedings of the 1st International Symposium on Advances in Spatial Databases;* 1989. p. 167–190.
276. Güting R.H. Geo-relational algebra: a model and query language for geometric database systems. In: *Advances in Database Technology, Proceedings of the 1st International Conference on Extending Database Technology;* 1988. p. 506–527.
277. Egenhofer M.J. Spatial SQL: a query and presentation language. *IEEE Trans Knowl Data Eng.* 1994;6(1): 86–95.
278. Güting R.H, Schneider M. Realm-based spatial data types: the ROSE algebra. *VLDB J.* 1995;4(2):243–286.
279. Cui Z., A.G. Cohn & D.A. Randell, *Qualitative and Topological Relationships in Spatial Databases.* 3rd Int. Symp. on Advances in Spatial Databases (SSD'93), LNCS 692, 296-315, 1993.
280. Güting RH. An introduction to spatial database systems. *VLDB J.* 1994;3(4):357–99.
281. Rigaux P, Scholl M, Voisard A. *Spatial databases - with applications to GIS.* San Francisco: Morgan Kaufmann Publishers; 2002.
282. Clementini E., Di Felice P. A model for representing topological relationships between complex geometric features in spatial databases. *Inf Sci.* 1996; 90(1–4):121–136.
283. Schneider M. Spatial data types for database systems - finite resolution geometry for geographic information systems, vol. LNCS 1288. Berlin/New York: Springer; 1997.
284. Schneider M, Behr T. Topological relationships between complex spatial objects. *ACM Trans Database Syst.* 2006;31(1):39–81
285. Worboys M.F, Bofakos P. A canonical model for a class of areal spatial objects. In: *Proceedings of the 3rd International Symposium on Advances in Spatial Databases;* 1993. p. 36–52.
286. Egenhofer M.J. & R.D. Franzosa, *Point-Set Topological Spatial Relations.* *Int. Journal of Geographical Information Systems,* 5(2), 161-174, 1991.
287. Schneider M., Weinrich B. An abstract model of three dimensional spatial data types. In:

- Proceedings of the 12th ACM International Symposium on Geographic Information Systems; 2004. p. 67–72.
288. Erwig M., Schneider M. Partition and conquer. In: Proceedings of the third international conference on spatial information theory; 1997. p. 389–408.
 289. Güting R.H., Schneider M. Realms: A Foundation for Spatial Data Types in Database Systems. 3rd Int. Symp. on *Advances in Spatial Databases*, LNCS 692, 14-35, 1993.
 290. Cicerone S., Di Felice P. Cardinal directions between spatial objects: the pairwise-consistency problem. *Inf Sci.* 2004;164(1-4):165–88.
 291. Clementini E., Billen R. Modeling and computing ternary projective relations between regions. *IEEE Trans Knowl Data Eng.* 2006;18(6):799–814.
 292. Freksa C. Using orientation information for qualitative spatial reasoning. In: Proceedings of the International Conference on Spatial Information Theory; 1992. p. 162–78.
 293. Goyal R. Similarity assessment for cardinal directions between extended spatial objects. PhD Thesis, Department of Spatial Information Science and Engineering, University of Maine; 2000.
 294. Hernández D. Qualitative representation of spatial knowledge. LNCS, vol. 804. Berlin: Springer; 1994.
 295. Ligozat G. Reasoning about cardinal directions. *J Visual Lang Comput.* 1998;9(1):23–44.
 296. Liu W., Li S. Reasoning about cardinal directions between extended objects: the NP-hardness result. *Artif Intell.* 2011;175(18): 2155–2169.
 297. Liu W., Zhang X, Li S., Ying M. Reasoning about cardinal directions between extended objects. *Artif Intell.* 2010;174(12–13):951–983
 298. Mukerjee A, Joe G. A qualitative model for space. In: Proceedings of 7th National Conference on AI; 1990. p. 721–727.
 299. Papadias D. Relation-based representation of spatial knowledge. PhD Thesis, Department of Electrical and Computer Engineering, National Technical University of Athens; 1994.
 300. Peuquet D.J., Ci-Xiang Z. An algorithm to determine the directional relationship between arbitrarilyshaped polygons in the plane. *Pattern Recognit.* 1987;20(1):65–74.
 301. Skiadopoulos S, Koubarakis M. Composing cardinal direction relations. *Artif Intell.* 2004;152(2):143–71
 302. Skiadopoulos S, Koubarakis M. On the consistency of cardinal directions constraints. *Artif Intell.* 2005;163(1):91–135.
 303. Skiadopoulos S, Giannoukos C, Sarkas N, Vassiliadis P, Sellis T, Koubarakis M. Computing and managing cardinal direction relations. *IEEE Trans Knowl Data Eng.* 2005;17(12):1610–23.
 304. Skiadopoulos S, Sarkas N, Sellis T, Koubarakis M. A family of directional relation models for extended objects. *IEEE Transactions on Knowledge and Data Engineering*, vol.17, No. 12, 2005, pp 1610–1623
 305. Shekhar S., Liu X. Direction as a Spatial Object: A Summary of Results. In R. Laurini, K. Makki, and N. Pissinou, editors, *ACM-GIS '98*, Proceedings of the 6th international symposium on Advances in Geographic Information Systems, November 6-7, 1998, Washington, DC, USA, pp. 69–75. ACM, 1998.
 306. Thanasis Hadzilacos, Nectaria Tryfona. An Extended Entity-Relationship Model for Geographic Applications. *ACM SIGMOD Record*, Vol. 26, No. 3, 1997, pp. 24–29
 307. Shekhar S., Vatsavai R.R., Chawla S., Burke T.E. Spatial pictogram enhanced conceptual data models and their translation to logical data models. In: *ISD '99: Selected Papers from the International Workshop on Integrated Spatial Databases, Digital Images and GIS*, 1999 pp. 77–104
 308. Gandhi V., Kang J., Shekhar S. *Spatial Databases*. Technical Report; 07-020, 2007. Retrieved from the University of Minnesota Digital Conservancy, <https://hdl.handle.net/11299/215734>
 309. Frederico T. Fonseca, Max J. Egenhofer. Ontology-driven geographic information systems. In Claudia Bauzer Medeiros, editor, *ACM-GIS '99*, Proceedings of the 7th International Symposium on Advances in Geographic Information Systems, November 2-6, 1999, Kansas City, USA, pages 14–19. ACM, 1999.
 310. Simon Jonathan David Cox, Chris Little. *Time Ontology in OWL*. Technical Report • July 2016. - https://www.researchgate.net/publication/305810003_Time_Ontology_in_Owl
 311. Bennacer N., Aufaure M.A., Cullot N., Sotnykova A., Vangenot C. (2004). Representing

- and reasoning for spatiotemporal ontology integration. In R. Meersman, Z. Tari, & A. Corsaro (Eds.), OTM int. conf. on the move to meaningful internet systems (pp. 30–31). Springer.
312. Baglioni M., Masserotti M.V., Renso C., Spinsanti L. (2007). Building geospatial ontologies from geographical databases. In F. Fonseca, M. A. Rodríguez, & S. Levashkin (Eds.), International conference on geospatial semantics (pp. 195–209). Springer.
313. Hogenboom F., Borgman B., Frasinca, F. & Kaymak U. (2010). Spatial knowledge representation on the semantic web. Proceedings of the IEEE 4th International Conference on Semantic Computing (ICSC 2010), pp. 252-259, September 2010.
314. Parent C., Spaccapietra S., Zimányi E. (2006). Conceptual modeling for traditional and spatio-temporal applications: The MADS approach. Springer.
315. Spaccapietra S., Cullot N., Parent C., Vangenot, C (2004). On spatial ontologies. Database Laboratory, Swiss Federal Institute of Technology. 9 p.
316. Egenhofer M.J. Toward the semantic geospatial web. Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems, 2002, pp. 1–4.
317. Fonseca F., Rodriguez M.A. From geo-pragmatics to derivation ontologies: New directions for the geospatial semantic web. Transactions in GIS, 2007, vol. 11, No. 3, pp. 313-316.
318. Subbiah G., Alam A., Khan L. Thuraisingham B. An integrated platform for secure geospatial information exchange through the semantic web. Proceedings of ACM Workshop on Secure Web Services (SWS), 20066 George Mason University, Fairfax, VA, USA
319. Curtin K., Noronha V., Goodchild M., Grise S. ARCGIS transportation model (UNETRANS), UNETRANS data model reference, 2003.
320. Gutting R.H. GraphDB: modeling and querying graphs in databases. In: VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases, 1994, pp.297–308
321. Shekhar S., Liu D.R. CCAM: a connectivity-clustered access method for networks and network computations. IEEE Trans Knowl Data Eng. 1997;9(1): 102–119.
322. Jensen C.S., Kolar J., Pederson T.B., Timko I. Nearest neighbor queries in road networks. In: GIS '03: Proceedings of the 11th ACM International Symposium on Advances in Geographic Information System; 2003. pp. 1–8
323. Papadias D, Zhang J, Mamoulis N, Tao Y. Query processing in spatial network databases. In: VLDB '03: Proceedings of the 29th international conference on Very large data bases - Vol. 29, 2003, pp. 802–813
324. Miller H.J., Shaw S.L. GIS-T data models, geographic information systems for transportation: principles and applications. Oxford: Oxford University Press; 2001.
325. Anez J., de la Barra T., Perez B. Dual graph representation of transport networks. Transp Res. 1996;30(3):209–216.
326. Winter S. Modeling costs of turns in route planning. GeoInformatica. 2002; 6(4):345–361.
327. Hoel E.G., Heng W.L., Honeycutt D. High performance multimodal networks. In: Proceedings of the 9th International Symposium on Advances in Spatial and Temporal Databases; 2005, pp. 308-327.
328. Kohler E., Langtau K., Skutella M. Time-expanded graphs for flow-dependent transit times. In: Proceedings of the 10th Annual European Symposium on Algorithms; 2002, pp. 599–611
329. George B., Shekhar S. Spatio-temporal network databases and routing algorithms: a summary of results. In: Proceedings of the 10th International Symposium on Advances in Spatial and Temporal Databases; 2007. p. 460–477.
330. George B., Shekhar S. Time-aggregated graphs for modeling spatio-temporal networks - an extended abstract. In: Proceedings of the 25th International Conference on Conceptual Modeling; 2006 p. 85–99.
331. Tansel A.U, Clifford J., Gadia S., Jajodia S., Segev A., Snodgrass R.T, editors. Temporal databases: theory, design, and implementation. Benjamin-Cummings Publishing Co., 1993, 633 p.
332. Worboys M.F. A unified model for spatial and temporal information. Comput J. 1994;37(1): 25–34.
333. Erwig M., Güting R.H., Schneider M., Vazirgiannis M. Spatio-temporal data types: an approach to modeling and querying moving objects in databases. Geoinformatica. 1999;3(3):265–291.

334. Erwig M., Schneider M. Developments in spatiotemporal query languages. In: Proceedings of the IEEE International Workshop on Spatio-Temporal Data Models and Languages; 1999. p. 441–449.
335. Güting R.H., Schneider M. Moving objects databases. San Francisco: Morgan Kaufmann; 2005
336. Erwig M., Schneider M. Spatio-temporal predicates. *IEEE Trans Knowl Data Eng.* 2002; 14(4):1–42.
337. Jitkajornwanich K., Pant N., Fouladgar M., Elmasri R. A survey on spatial, temporal, and spatio-temporal database research and an original example of relevant applications using SQL ecosystem and deep learning, *Journal of Information and Telecommunication*, 2020, 4:4, 524-559,
338. Sistla A. P., Wolfson O., Chamberlain S., Dao S. Modeling and Querying Moving Objects. *ICDE '97: Proceedings of the Thirteenth International Conference on Data Engineering*, 1997, pp. 422–432
339. Wolfson O., Chamberlain S., Dao S., Jiang L., Mendez G. Cost and imprecision in modeling the position of moving objects. In: Proceedings of the 14th International Conference on Data Engineering; 1998. p. 588–596.
340. Frank A., Grumbach S., Güting R.H., Jensen C.S., Koubarakis M., Lorentzos N., Manolopoulos Y. Chorochronos: a research network for spatiotemporal database systems. *ACM SIGMOD Record*, Vol. 28I, No. 3., 1999, pp 12–21
341. Grumbach S., Rigaux Ph., Segoufin L. The DEDALE system for complex spatial queries *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, 1998 pp. 213–224
342. Vazirgiannis M., Wolfson O. A Spatiotemporal Model and Language for Moving Objects on Road Networks. *SSTD '01: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*, 2001, pp. 20–35
343. Güting R.H., Victor Teixeira de Almeida, Zhiming Ding. Modeling and querying moving objects in networks *The International Journal on Very Large Data Bases*, 2006, vol. 15, No. 2, pp 165–190
344. Belussi A., E. Bertino & B. Catania, Manipulating Spatial Data in Constraint Databases. 5th Int. Symp. on Advances in Spatial Databases (SSD'97), LNCS 1262, 115-141, 1997.
345. Rigaux P., Scholl M., Segoufin L., Grumbach S. Building a constraint-based spatial database system: model, languages, and implementation. *Inf Syst.* 2003;28(6):563–595.
346. Erwig M., Güting R.H., Schneider M., Vazirgiannis M. Spatio-temporal data types: an approach to modeling and querying moving objects in databases. *Geoinformatica.* 1999;3(3):265–291.
347. Güting R.H., Böhlen M.H., Erwig M., Jensen C.S., Lorentzos N.A., Schneider M., Vazirgiannis M. A foundation for representing and querying moving objects. *ACM Trans Database Syst.* 2000;25(1): pp. 1–42.
348. Sistla A.P., Wolfson O., Chamberlain S., Dao S. Querying the uncertain position of moving objects. In: Etzion O, Jajodia S, Sripada S, editors. *Temporal databases: research and practice*, LNCS, vol. 1399. Berlin: Springer; 1998. p. 310–37.
349. Trajcevski G., Wolfson O., Hinrichs K., Chamberlain S. Managing uncertainty in moving objects databases. *ACM Trans Database Syst.* 2004;29(3): 463–507.
350. Güting R.H., Böhlen M.H., Erwig M., Jensen C.S., Lorentzos N.A., Schneider M., Vazirgiannis M. A foundation for representing and querying moving objects in databases. *ACM Trans Database Syst.* 2000;25(1):1–42.
351. Pelekis N, Theodoridis Y. *Mobility data management and exploration*. New York: Springer; 2014.
352. Renso C, Spaccapietra S, Zimányi E. *Mobility data: modeling, management, and understanding*. Cambridge, UK: Cambridge University Press; 2013.
353. Chandra A.K., Harel D. Horn clauses and the fixpoint hierarchy *Proc. ACM Symp. on the Principles of Database Systems (PODS)* (1982), pp. 158-163
354. Porter H.H., Oct. 1985. Optimizations to Earley deduction for DATALOG programs. Available at: <http://www.cs.pdx.edu/~harry/earley/datalog.pdf>
355. Afrati C.H. Papadimitriou Ch. Papageorgiou G. Roussou A. Sagiv Y, Ullman J.D. 1986. Convergence of sideways query evaluation. In *ACM Symposium on Principles of Database Systems*, pp. 24–30
356. Bancilhon, R. Ramakrishnan. 1986. An amateur's introduction to recursive query process-

- ing strategies. In Proc. of the 1986 ACM SIGMOD International Conference on Management of Data, SIGMOD '86, pp. 16–52.
357. Hafner C.D., Godden K. Portability of syntax and semantics in DATALOG. *ACM Trans. on Information Systems*, 1985, 3(2):141–164.
358. Zaniolo, C. [1986]. “Safety and compilation of nonrecursive Horn clauses,” Proc. First Intl. Conf. on Expert Database Systems, pp. 167-178, Benjamin-Cummings, Menlo Park, CA.
359. Ramakrishnan R., Bancilhon F., Silberschatz A. [1987]. “Safety of recursive Horn clauses with infinite relations,” Proc. Sixth ACM Symp. on Principles of Database Systems, pp. 328-339.
360. Bancilhon F., Ramakrishnan R. An amateur’s introduction to recursive query processing strategies. *SIGMOD Record*, v. 15, no.2, 1986, pp. 16-52
361. Apt K.R., Blair H., Walker A., Towards a Theory of Declarative Knowledge, in: J. Minker (ed.), *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann, San Mateo, CA, 1988, pp. 89-148.
362. Chandra A.K., Harel D. Horn Clause Queries and Generalizations, *J. Logic Programming* 2(1):1-15 (Apr. 1985).
363. Gelfond M., Lifschitz V. The Stable Model Semantics for Logic Programming, in: *Proceedings of the Fifth International Conference and Symposium on Logic Programming*, 1988.
364. Przymusinska H., Przymusinski T.C. Weakly Perfect Model Semantics for Logic Programs, in: *Proceedings of the Fifth International Conference/Symposium on Logic Programming*, 1988.
365. Przymusinski, T.C. On the Declarative Semantics of Stratified Deductive Databases in: J. Minker (ed.), *Foundations of Deductive Databases and Logic Programming*, 1988, pp. 193-216.
366. Przymusinski T.C. Extended Stable-Semantics for Normal and Disjunctive Programs, in: *Seventh International Conference on Logic Programming*, 1990, pp. 459-477.
367. Ross K. Modular Stratification and Magic Sets for DATALOG Programs with Negation, in: *Proceedings of the ACM Symposium on Principles of Database Systems*, 1990, pp. 161-171.
368. Van Gelder A., Ross K., Schlipf J.S. The Well-Founded Semantics for General Logic Programs, *Journal of the ACM* 38(3):620- 650 (1991)
369. Naqvi S. A Logic for Negation in Database Systems, in: J. Minker (ed.), *Proceedings of the Workshop on Foundations of Deductive Databases and Logic Programming*, 1986, pp. 378-387.
370. Van Gelder A. Negation as Failure Using Tight Derivations for General Logic Programs, *Journal of Logic Programming* 6(1):109-133 (1989).
371. Balbin I., Port G.S., Ramamohanarao K., Meenakshi K. Efficient Bottom-Up Computation of Queries of Stratified Databases, *Journal of Logic Programming* 11:295-345 (1991).
372. Bayer R. Query Evaluation and Recursion in Deductive Database Systems, Technical Report 18503, Technische Universitaet Muenchen, Feb. 1985.
373. Beerl C., Naqvi S., Ramakrishnan R., Shmueli O., Tsur S. Sets and Negation in a Logic Database Language, in: *Proceedings of the ACM Symposium on Principles*
374. Kerisit J.M., Pugin J.M. Efficient Query Answering on Stratified Databases, in: *Proceedings of the International Conference on Fifth Generation Computer Systems*, Tokyo, Japan, Nov. 1988, pp. 719-725.
375. Przymusinski T. On the Declarative Semantics of Stratified Deductive Databases, in J. Minker (ed.), *Foundations of Deductive Databases and Logic Programming*, 193-216, Morgan-Kaufmann, Los Altos, 1988.
376. Ross K.A. Modular Stratification and Magic Sets for Datalog Programs with Negation. *Proceedings of the ACM Symposium on Principles of Database Systems*, 161-171, Nashville, 1990.
377. Warren D.S. Memoing for Logic Programs, *Communications of the ACM* 35 (3): 93-111 (Mar. 1992)
378. Sacca D., Zaniolo C. The Generalized Counting Methods for Recursive Logic Queries, in: *Proceedings of the First International Conference on Database Theory*, 1986.
379. Naughton J.F., Ramakrishnan R., Sagiv Y., Ullman J.D. Argument Reduction Through Factoring, in: *Proceedings of the Fifteenth International Conference on Very Large Databases*, Amsterdam, The Netherlands, Aug. 1989, pp. 173-182.

380. Sagiv Y., Optimizing Datalog Programs, in: J. Minker (ed.), *Foundations of Deductive Databases and Logic Programming*, Los Altos, CA, Morgan Kaufmann, 1988, pp. 659-698.
381. Ramakrishnan R., Beeri C., Krishnamurthy R. Optimizing Existential Datalog Queries, in: *Proceedings of the ACM Symposium on Principles of Database Systems*, Austin~ TX, Mar. 1988, pp. 89-102.
382. Sippu S., Soisalon-Soinen E. An Optimization Strategy for Recursive Queries in Logic Databases, in: *Proceedings of the Fourth International Conference on Data Engineering*, Los Angeles, CA, 1988.
383. Bancilhon F., Ramakrishnan R. An amateur's introduction to recursive query processing strategies. *SIGMOD Record*, Vol. 15, No.2, 1986, pp. 16-52
384. Gallaire H., Minker J. and Nikolas J.M. Logic and databases: a deductive approach. *Computing Surveys*, 16:1, 1984, pp. 154-185
385. Ramakrishnan R., Ullman J.D. A survey of deductive database systems. *The Journal of Logic Programming*, 1995, Vol. 23, No 2, pp. 125-149
386. Finkelstein S.J., Mattos N., Mumick I., Pirahesh H. Expressing Recursive Queries in SQL SO/IEC JTC1/SC21 WG3 DBL MCI-X3H2-96-075 Tech. Rep., March, 1996
387. Reznichenko V.A. Recursive SQL (Rus). *Software Engineering*, 2010, vol. 4, No 4, pp. 48-65.
388. Kifer M., Lausen G. F-Logic: A Higher Order Language for Reasoning about Objects, Inheritance, and Schema. *SIGMOD Record*, v. 18, no.2, 1989, pp. 139- 146.
389. Liu M. *Deductive Database Languages: Problems and Solutions*. ACM Computing Surveys, v. 31, no. 1, 1999. pp. 27-62
390. Falcone Sampaio P.R., Paton N.W. (1997) Deductive object-oriented database systems: A survey. In: Geppert A., Berndtsson M. (eds) *Rules in Database Systems*. RIDS 1997. Lecture Notes in Computer Science, vol 1312. Springer, Berlin, Heidelberg. pp 1-19
391. Paton N.W., Díaz O. Active database systems. *ACM Computing Surveys*. 1999, vol. 31, No 1, pp. 63–103
392. Chakravarthy S, Blaustein B, Buchmann A.P, Carey M, Dayal U, Goldhirsch D, Hsu M, Jauhari R, Ladin R, Livny M, McCarthy D, McKee R, Rosenthal A. HiPAC: a research project in active, time-constrained database management. Technical report. CCA-88-02. Cambridge, MA: Xerox Advanced Information Technology; 1988
393. Paton N., Diaz O., Williams M., Campin J., Dinn A., Jaime A. Dimensions of active behaviour. In N. Paton and M. Williams Eds., *Proc. 1st Int. Workshop on Rules In Database Systems*, Springer- Verlag., 1994, pp. 40-57.
394. Stonebraker M., Jhingran A., Goh J., Potamianos S. On rules, procedures, caching and views in database systems. In *Proc. ACM SIGMOD 1990*, pp. 281-290
395. Dayal U., Buchmann A., McCarthy D. Rules are objects too: A knowledge model for an active object oriented database system. In K. Dittrich Ed., *Proc. 2nd Inti Workshop on OODBS*, Volume 334, 1988, pp. 129-143. Springer-Verlag. Lecture Notes in Computer Science
396. Widom J., Finkelstein S. Set-Oriented Production Rules in Relational Database Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1990, pp. 259-270.
397. Dayal U, Blaustein B, Buchmann A, Chakravarthy S, Hsu M, Ladin R, McCarty D, Rosenthal A, Sarin S, Carey M.J, Livny M, Jauhari R. The HiPAC project: combining active databases and timing constraints. *ACM SIGMOD Rec.* 1988;17(1):51–70.
398. Chakravarthy S, Krishnaprasad V, Anwar E, Kim S.K Composite events for active database: semantics, contexts, and detection. In: *Proceedings of the 20th International Conference on Very Large Data Bases*; 1994. p. 606–617.
399. Chakravarthy S, Mishra D. Snoop: an expressive event specification language for active databases. *Data Knowl Eng.* 1994;14(1):1–26.
400. Gehani N.H., Jagadish H.V., Schmueli O. Gehani N., Jagadish H.V., Shmueli O. COMPOSE: A system for composite specification and detection. In: Adam N.R., Bhargava B.K. (eds) *Advanced Database Systems*. 1993, pp. 3-15. Lecture Notes in Computer Science, vol 759. Springer, Berlin.
401. Gatzui S., Dittrich K. Events in an active object-oriented database. In N. Paton and M. Williams Eds., *Proc. 1st Int. Workshop on Rules in Database Systems*, 1994, pp. 23-39. Springer-Verlag

402. Mellin J., Berndtsson M. Event Detection. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 1361-1366
403. Mellin J., Berndtsson M. Event Specification. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 1389-1393
404. Gehani N., Jagadish H.V., Smueli O. Event specification in an active object-oriented database. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 1992. p. 81-90
405. Motakis I., Zaniolo C. Composite Temporal Events in Active Databases: A Formal Semantics. In: Clifford J., Tuzhilin A. (eds) Recent Advances in Temporal Databases. Workshops in Computing. Springer, London, 1995), 332-352
406. Chakravarthy S., Anwar E., Maugis L., Mishra D. Design of Sentinel: an object-oriented DBMS with event-based rules. Information and Software Technology, 1994, 36, 9, 555-568.
407. Diaz O., Jaime, A. EXACT: an EXtensible approach to ACTIVE object-oriented databases. VLDB Journal 1997, 6, 4, 282-295
408. Branding H, Buchmann A, Kudrass T, Zimmermann J. Rules in an open system: the REACH rule system. In: Proceedings of the 1st International Workshop on Rules in Database Systems, Workshops in Computing; 1994. p. 111-126
409. Agrawal R., Cochrane R., Lindsay B. On maintaining priorities in a production rule language. In G. Lohman, A. Sernadas, and R. Camps Eds., Proc. 17th VLDB, 1991, pp. 479-487. Morgan-Kaufmann.
410. Widom J. The Starburst Rule System: Language Design, Implementation, and Applications. In: IEEE Data Engineering Bulletin, Special Issue on Active Databases, 1992, 15(4): 15-18
411. Stonebraker M., Kemnitz, G. The POSTGRES Next-generation Database Management System. Communications of the ACM 1991, Vol. 34, No.10, pp. 78-92
412. Hanson E.N. The Design and Implementation of the Ariel Active Database Rule System. IEEE Trans. Knowl. Data Eng. 1996, 8(1): 157-172
413. Kotz A., Dittrich K., Mülle J. Supporting semantic rules by a generalized event/trigger mechanism. In Advance in Database Technology, EDDT, Venice 1988, pp. 76-91.
414. Reddi S., Pouloussis A., Small C. Extending a Functional DBPL With ECA-Rules. In T. Sellis Ed., Proc. 2nd Int. Wshp. on Rules in Database Systems 1995, pp. 101-115. Springer-Verlag.
415. Kiernan G., de Maindreville C., Simon E. Making Deductive Databases a Practical Technology: a step forward. In II. Garcia-Molina and II. Jagadish Eds., Proc. ACM SIGMOD Conf. 1990., pp. 237-246
416. Zaniolo C. A Unified Semantics for Active and Deductive Databases. In: Paton N.W., Williams M.H. (eds) Rules in Database Systems. Workshops in Computing. Springer, London. 1994, pp 271-287
417. Harrison J., Dietrich. S. Integrating active and deductive rules. In N. Paton and M. Williams Eds., Proc. 1st Int. Workshop on Rules In Database Systems, 1994, pp. 288-305. Springer-Verlag.
418. Widom J. Deductive and Active Databases: Two Paradigms or Ends of a Spectrum? In N. Paton and M. Williams Eds., Proc. 1st Int. Workshop on Rules In Database Systems 1994, pp. 306-315. Springer-Verlag.
419. Bayer P., Jonker W. A framework for supporting triggers in deductive databases. In N. Paton and M. Williams Eds., Proc. 1st Int. Workshop on Rules In Database Systems 1994, pp. 316-330. Springer-Verlag.
420. Kulkarni K., Mattos N., Cochrane R. Active Database Features in SQL3. In: Paton N.W. (eds) Active Rules in Database Systems. 1999, pp. 197-219 Monographs in Computer Science. Springer, New York, NY.
421. Chakravarthy S. Rule management and evaluation: an active DBMS perspective. SIGMOD RECORD 1989, 18, 3, 20-28.
422. Collet C., Coupaye T. and Svensen T. NAOS: Efficient and modular reactive capabilities in an object-oriented database system. In J. Bocca, M. Jarke, and C. Zaniolo Eds., Proc. 20th VLDD Conf, 1994, pp. 132-143. Morgan-Kaufmann.
423. Ceri S., Fraternali P., Paraboscii S., Tanca, L. Active Rule Management in Chimera. In J. Widom and S. Ceri Eds., Active Database Systems: Triggers and Rules for Active Database Processing, 1996, pp. 151-175. Morgan Kaufmann.

424. Gehani N. and Jagadish H. ODE as an Active Database: Constraints and Triggers. In R. C. G.M. Loiiiman. A. Sernadas Ed., 17th Intl. Conf. on Very Large Data Bases, Barcelona, 1991, pp. 327-336. Morgan Kaufmann
425. Atkinson M., Bancilhon F., DeWitt D., Dittrich K., Maier D., Zdonik S. The object-oriented database system manifesto. In: Proceedings of the 1st International Conference on Deductive and Object-Oriented Databases; 1989. p. 223–240.
426. Hull R., Tanaka K., Yoshikawa M. Behavior Analysis of Object-Oriented Databases: Method Structure, Execution Trees, and Reachability // Lect. Notes Comput. Sci.- 367.- 1989.- 372-388
427. Mozaffari M., Tanaka Y. ODM: An Object-Oriented Data Model // New. Generat. Comp.- 7, N 1.- 1989.- 4-35
428. Beerl C. A Formal Approach to Object-Oriented Databases // Data and Knowledge Eng.- 5.- 1990.- 353-382
429. Zicari R. Incomplete Information in Object-Oriented Databases // ACM SIGMOD Record.- 19, N 3.- 1990.- 5-16
430. Hong Sh., Maryanski F. Using a Meta Model to Represent Object-Oriented Data Models // 6th Int. Conf. Data Eng., Los Angeles, Calif., USA, Febr. 5-9, 1990.- 11-19
431. Cornelio, Shamkant B. Navathe, Keith L. Doty. Extending Object-Oriented Concepts to Support Engineering Applications // 6th Int. Conf. Data Eng., Los Angeles, Calif., USA, Febr. 5-9, 1990.- 220-227
432. Gunter Saake. Descriptive Specification of Database Object Behaviour // Data and Knowledge Eng.- 6, N 1. 1991.- 47-73
433. Lellani S.K., Spiratos N. Towards a Categorical Data Model Supporting Structured Objects and Inheritance // Proc. 1st Int. East/West Database Workshop, Kiev, Oct. 1990, Lect. Notes Comput. Sci.- 540.- 1991
434. Cattell R.G.G., Barry D.K.(eds.). The Object Data Standard: ODMG 3.0. — San Francisco, Calif.: Morgan Kaufmann, 2000.
435. Stonebraker M., Rowe L.A., Lindsay B., Gray, Carey M., Brodie M., Bernstein Ph., Beech D. Third-Generation Database System Manifesto. SIGMOD Record 19(3), September, 1990. pp 31-43
436. Rowe L, Stonebraker M. The Postgres data model. In: Proceedings of the 13th International Conference on Very Large Data Bases; 1987. p. 83–96.
437. Won Kim. UniSQL/X unified relational and object-oriented database system. SIGMOD '94: Proceedings of the 1994 ACM SIGMOD international conference on Management of data May 1994, p. 481
438. Darwen H., Date C.J. (March 1995). “The third manifesto”. ACM SIGMOD Record. New York, NY, USA: ACM Press. 24 (1): 39–49.

Одержано 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