

Бази даних

ПРОФЕСОР КАФЕДРИ ПРОГРАМНИХ ЗАСОБІВ Г.В. ТАБУНЩИК
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»
ЗА МАТЕРІАЛАМИ ТРЕНІНГІВ LUXOFT

Класифікація по моделі даних

Ієрархічні

Мережеві

Реляційні

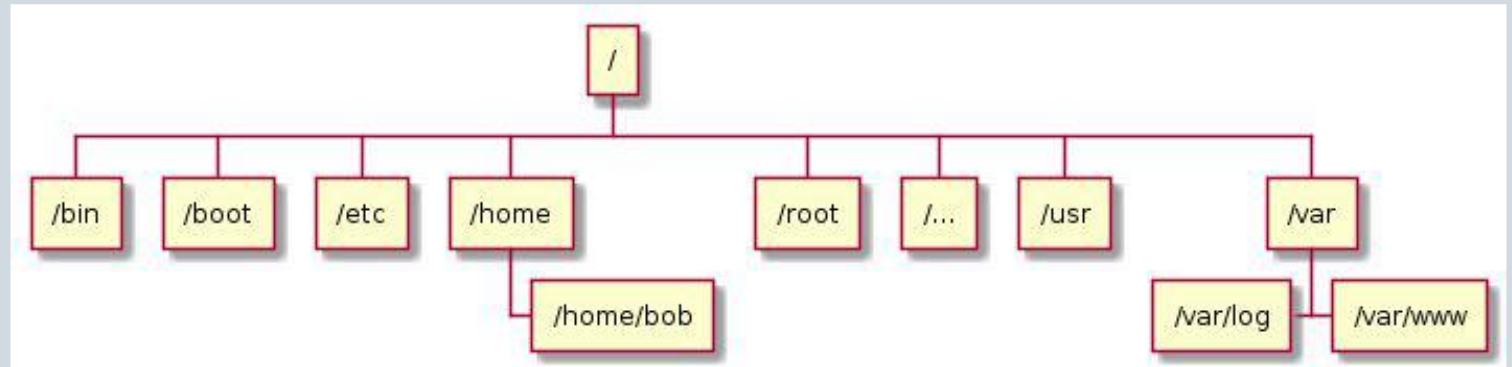
Об'єктно - реляційні

Об'єктно - орієнтовані

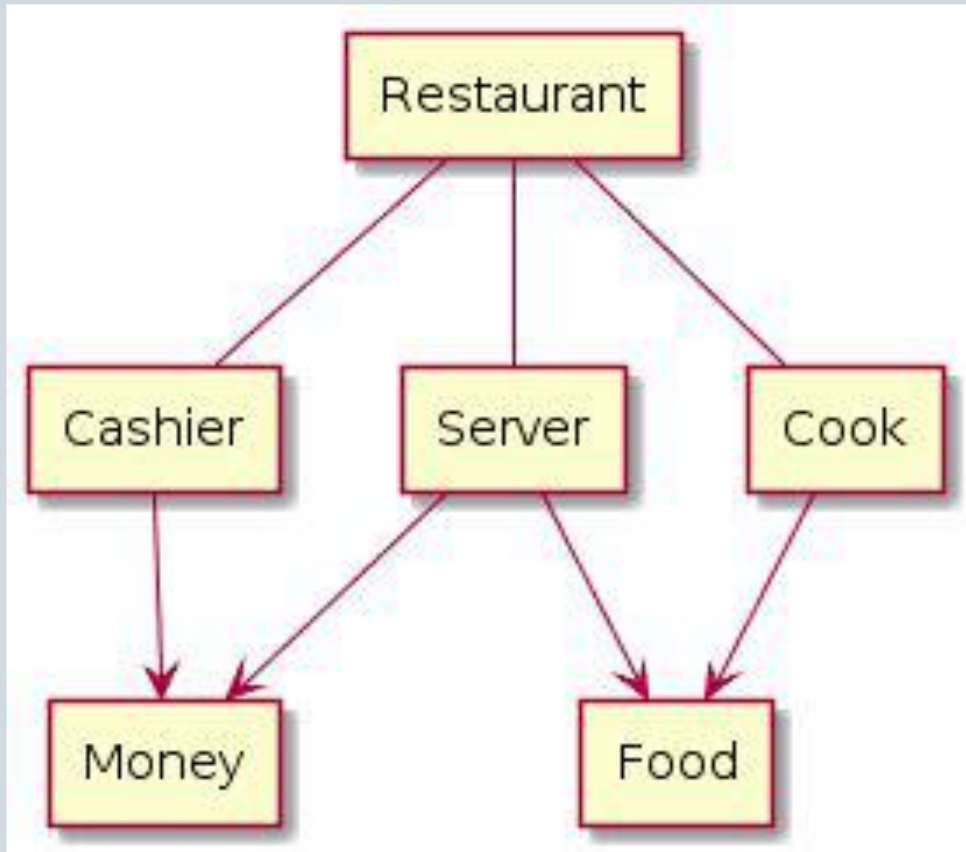
Ієрархічні моделі

Характеристики:

- модель у вигляді дерева залежностей
- кожен запис має тільки один батьківський вузол
- багато до багатьох - відсутнє, зв'язки побудовані
- у вигляді вказівників



Мережеві моделі

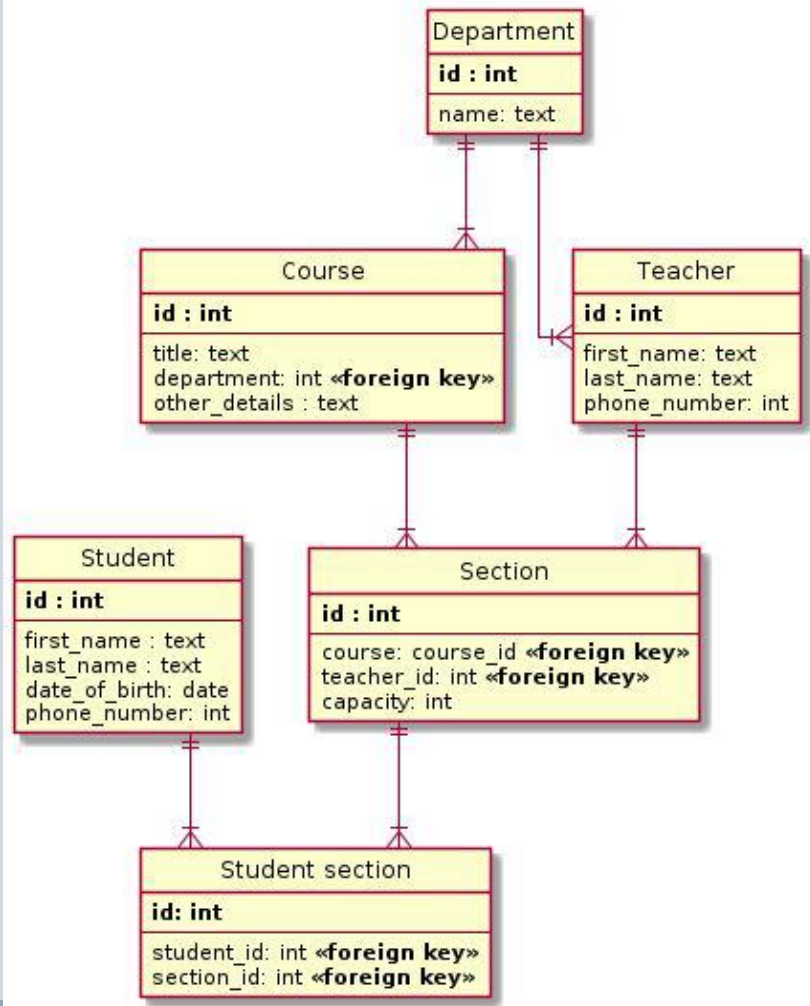


Характеристики:

Мають ті ж самі характеристики що і ієрархічні

представлені у вигляді графа

Реляційні моделі



Характеристики:

для доступу до даних використовується SQL

сутності представлені у вигляді таблиць

має високоорганізовану структуру і гнучкість

у використанні та побудові складних типів даних

наявність JOIN

Об'єктно - Реляційні моделі

Розрізняють, як зазначалося раніше, два різновиди ОРБД - гібридні і розширені

У **гібридних** ОРБД інтерфейс користувача і алгоритм додатки виконані з урахуванням об'єктно-орієнтованого підходу, тоді як власне БД є реляційної.

У **розширених (постреляційних)** ОРБД передбачається об'єктно-орієнтоване побудова власне бази даних шляхом використання відомих і введення нових типів даних, пов'язаних між собою. Цей зв'язок найчастіше здійснюється створенням методів за допомогою тригерів і збережених процедур

Характеристики:

складні дані

спадкування типу

поведінка об'єкта

Об'єкти, класи

Проміжний підхід між Реляційними та Об'єктно орієнтованими

Стандарт для моделі ОРБД був погоджений в 1999 році, після більш ніж шестирічної розробки (цей стандарт відомий під неофіційним іменем SQL3). Даний стандарт є результатом праці Американського національного інституту стандартів (American National Standards Institute - ANSI) і Міжнародної організації по стандартизації (International Organization for Standardization- ISO).

Об'єктно - орієнтовані моделі

В стандарті ODMG в якості базового засобу маніпулювання об'єктними базами даних пропонується мова OQL (Object Query Language). Це невелика, але досить складна мова запитів. Розробники в цілому характеризують її таким чином:

- OQL спирається на об'єктну модель ODMG (мається на увазі, що в ній підтримуються засоби доступу до всіх можливих структур даних, що допускаються в структурній частині моделі).
- OQL дуже близька до SQL/92. Розширення відносяться до об'єктно-орієнтованих понять, таких як складні об'єкти, об'єктні ідентифікатори, шляхові вирази, поліморфізм, виклик операцій і відкладене зв'язування.
- У OQL забезпечуються високорівневі примітиви для роботи з множинами об'єктів, але, крім того, є настільки ж ефективні примітиви для роботи зі структурами, списками і масивами.
- OQL є функціональною мовою, що допускає необмежену композицію операцій, якщо операнди не виходять за межі системи типів. Це є наслідком того факту, що результат будь-якого запиту володіє типом, що належить до моделі типів ODMG, і тому до результату запиту може бути застосований новий запит.
- OQL не є обчислювально повною мовою. Вона являє собою просту мову запитів.
- Оператори мови OQL можуть викликатися з будь-якої мови програмування, для якого в стандарті ODMG визначені правила зв'язування. І, навпаки, у запитах OQL можуть бути присутніми виклики операцій, запрограмованих на цих мовах.
- У OQL не визначаються явні операції оновлення, а використовуються виклики операцій, визначених в об'єктах для цілей оновлення.
- У OQL забезпечується декларативний доступ до об'єктів. З цієї причини OQL-запити можуть добре оптимізуватися.
- Можна легко визначити формальну семантику OQL

Характеристики:

високопродуктивна обробка даних

Об'єкт - колекція, структура

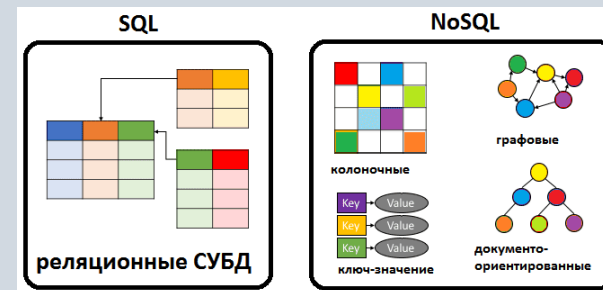
Підтримка інкапсуляції

Підтримка типів і класів і типів

Підтримка наслідування типів і класів від їх предків

Model	Data Element Organization	Relationship Organization	Identity	Access Language
Hierarchical	Files, Records	Logical Proximity in a Linearized tree	Record based	Procedural
Network	Files, Records	Intersecting Networks	Record based	Procedural
Relational	Tables	Identifiers of rows in one table are embedded as attribute values in another table	Value based	Non-procedural
Object-Oriented	Objects	Logical Containment - Related objects are found within a given object by recursively examining attributes of an object that are themselves objects	Record based	Procedural
Object-Relational	Object- infrastructure to the database system itself—user-defined data types, functions, and rules	Relational extenders that support specialized applications such as image retrieval, advanced text searching, and geographic applications.	Value based	Non-procedural
Deductive	Facts, Rules	Inference rules that permit related facts to be generated on demand.	Value based	Non-procedural

NoN - No - SQL



база даних "ключ/значення" є нічим більшим, ніж вище описана функція - двійковий об'єкт даних, асоційований з його ключем, зберігається у базі даних для подальшого його повернення.

документоорієнтована база даних трохи виходить за рамки попереднього принципу. Вона накладає певну структуру на двійковий об'єкт. Об'єктами повинні бути документи впізнаваного формату, XML чи PDF. Проте немає ніяких вимог щодо структури чи змісту документу. Кожен документ зберігається як значення (у структурі "ключ/значення") і може супроводжуватись метаданими, вбудованими в себе.

колоноподібна база даних є гібридом NoSQL та реляційних баз даних. Вона передбачає певну структуру рядок-колонка, але не має жорстких правил реляційної структури.

****база даних на основі графів **** зберігає дані у кортежах з багатьма атрибутами, які відображають відношення в інший спосіб. Наприклад, "дружні" зв'язки у соціальних мережах можуть зберігатись у вигляді запису, який включає всього лиш двоє користувачів, які є друзями.

No Sql рішення



Переваги та недоліки

лінійна масштабованість - додавання нових вузлів в кластер збільшує загальну продуктивність системи;

гнучкість, що дозволяє оперувати полуструктурізовані дані, реалізуючи, в. т.ч. повнотекстовий пошук по базі;

можливість працювати з різними уявленнями інформації, в т.ч. без завдання схеми даних;

висока доступність за рахунок реплікації даних і інших механізмів відмовостійкості, зокрема, шаринга - автоматичного розподілу даних за різними вузлів мережі, коли кожен сервер кластера відповідає тільки за певний набір інформації, обробляючи запити на його читання і запис. Це збільшує швидкість обробки даних і пропускну здатність додатки.

продуктивність за рахунок оптимізації для конкретних видів моделей даних (документної, графової, колонкової або «ключ-значення») і шаблонів доступу;

широкі функціональні можливості - власні SQL-подібні мови запитів, RESTful-інтерфейси, API і складні типи даних, наприклад, map, list і struct, що дозволяють обробляти відразу безліч значень [2].

обмежена ємність вбудованого мови запитів [5].

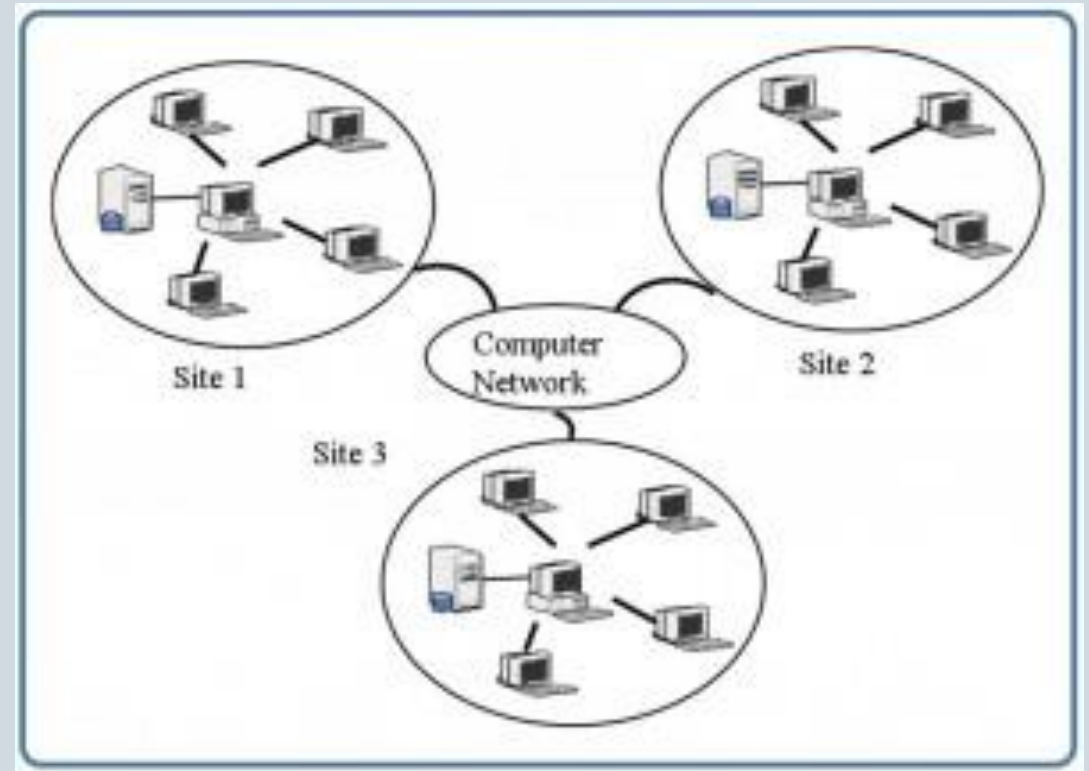
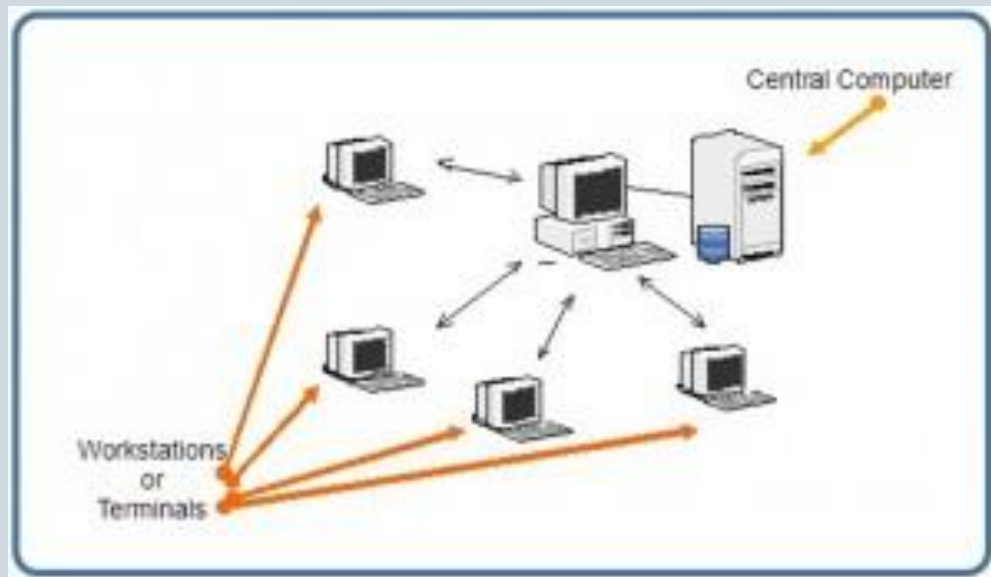
складності в підтримці всіх ACID-вимог до транзакцій (атомарність, консистентність, ізоляція, довговічність) через те, що NoSQL-СУБД замість CAP-моделі (узгодженість, доступність, стійкість до поділу) скоріше відповідають моделі BASE (базова доступність, гнучке стан і підсумкова узгодженість) [1]. Втім, деякі нереляційні СУБД намагаються обійти це обмеження за допомогою налаштованих рівнів узгодженості, про що ми розповідали на прикладі Cassandra. Аналогічним чином Riak дозволяє налаштовувати необхідні характеристики доступності-узгодженості навіть для окремих запитів за рахунок завдання кількості вузлів, необхідних для підтвердження успішного завершення транзакції [1].

сильна прив'язка застосування до конкретної СУБД через специфіку внутрішнього мови запитів і гнучкою моделі даних, орієнтованої на конкретний випадок [5];

недолік фахівців з NoSQL-баз в порівнянні з реляційними аналогами

Класифікація по архітектурі

1. Локальні
2. Розподілені



12 властивостей розподілених баз даних

Локальна автономія — управління даними на кожному з вузлів розподіленої системи виконується локально.

Незалежність вузлів — всі вузли рівноправні і незалежні, а розташовані на них БД є рівноправними постачальниками даних в загальний простір даних.

Безперервні операції — можливість безперервного доступу до даних в рамках розподіленої БД незалежно від їх розташування і незалежно від операцій, що виконуються на локальних вузлах.

Прозорість розташування — користувач, що звертається до БД, нічого не повинен знати про реальне, фізичне розміщення даних у вузлах інформаційної системи.

Прозора фрагментація — можливість розподіленого (тобто на різних вузлах) розміщення даних, логічно поєднаних в єдине ціле. Існує фрагментація двох типів: горизонтальна і вертикальна.

Прозоре тиражування — тиражування даних — це асинхронний процес перенесення змін об'єктів вихідної бази даних в бази, розташовані на інших вузлах розподіленої системи

Обробка розподілених запитів — можливість виконання операцій вибірки даних з розподіленої БД, за допомогою запитів, сформульованих на мові SQL

Обробка розподілених транзакцій — можливість виконання операцій оновлення розподіленої бази даних, які не порушують цілісність і узгодженість даних.

Незалежність від устаткування — як вузли розподіленої системи можуть виступати ПК будь-яких моделей і виробників

Незалежність від операційних систем — різноманіття операційних систем, керуючих вузлами розподіленої системи

Прозорість мережі — спектр підтримуваних конкретно СУБД мережевих протоколів не має бути обмеженням системи, заснованої на розподіленій БД

Незалежність від баз даних — в розподіленій системі можуть працювати СУБД різних виробників, і можливі операції пошуку і оновлення в базах даних різних моделей і форматів.

Крістофер Дейт

Класифікація за можливістю доступу

1. Мейнфреймові
2. Клієнт - серверні
3. Файл - серверні
4. Вбудовувані

Класифікація за швидкістю обробки даних

1. Операційні
2. Сховища даних

Індекси

1. Структура індекса
2. Кластерні індекси
3. Некластерні індекси

ACID

Atomicity

Consistency

Isolation

Durability

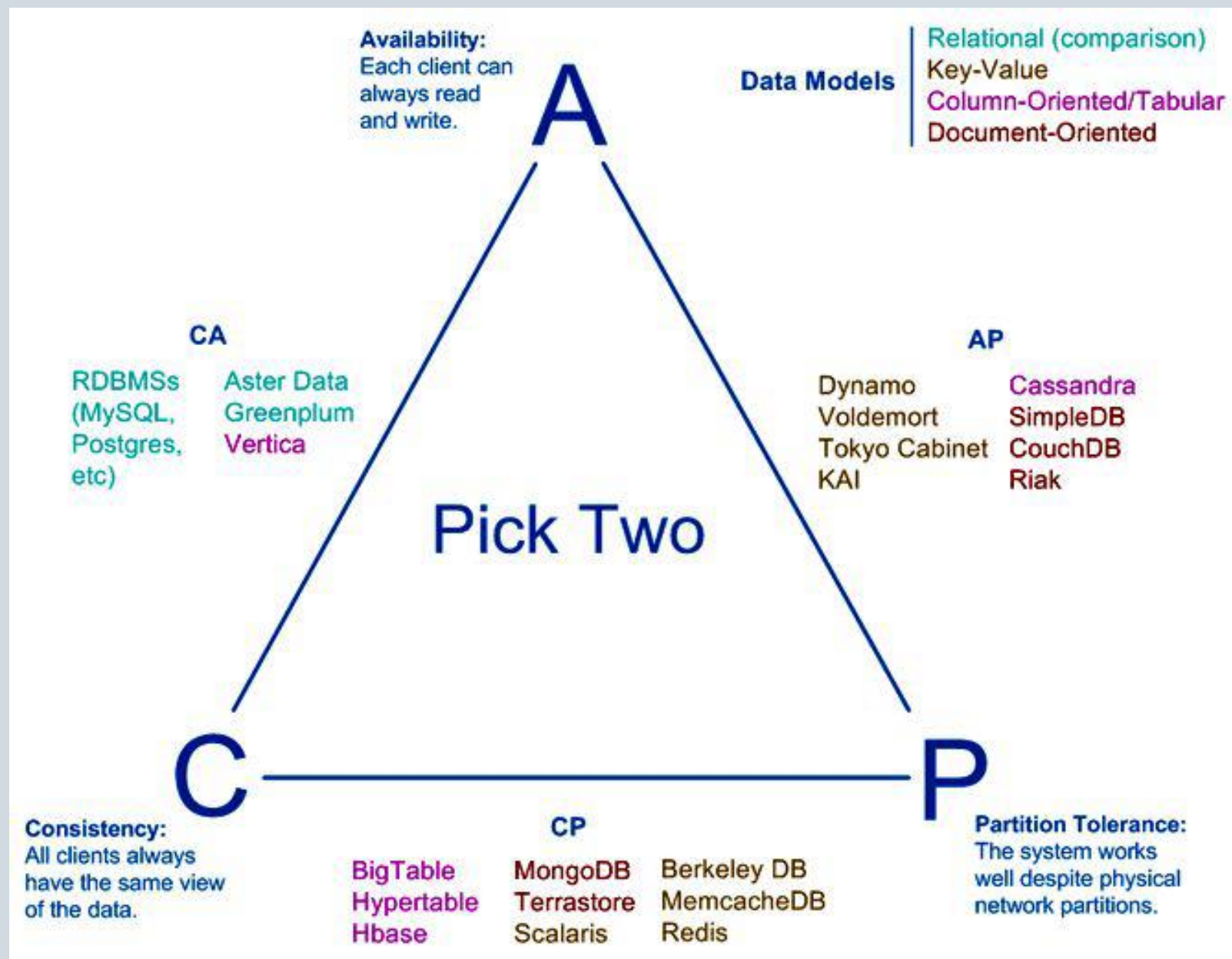
BASE

Basically available

Soft - state

Eventually consistent

Теорема CAP



Теорема PACELC

