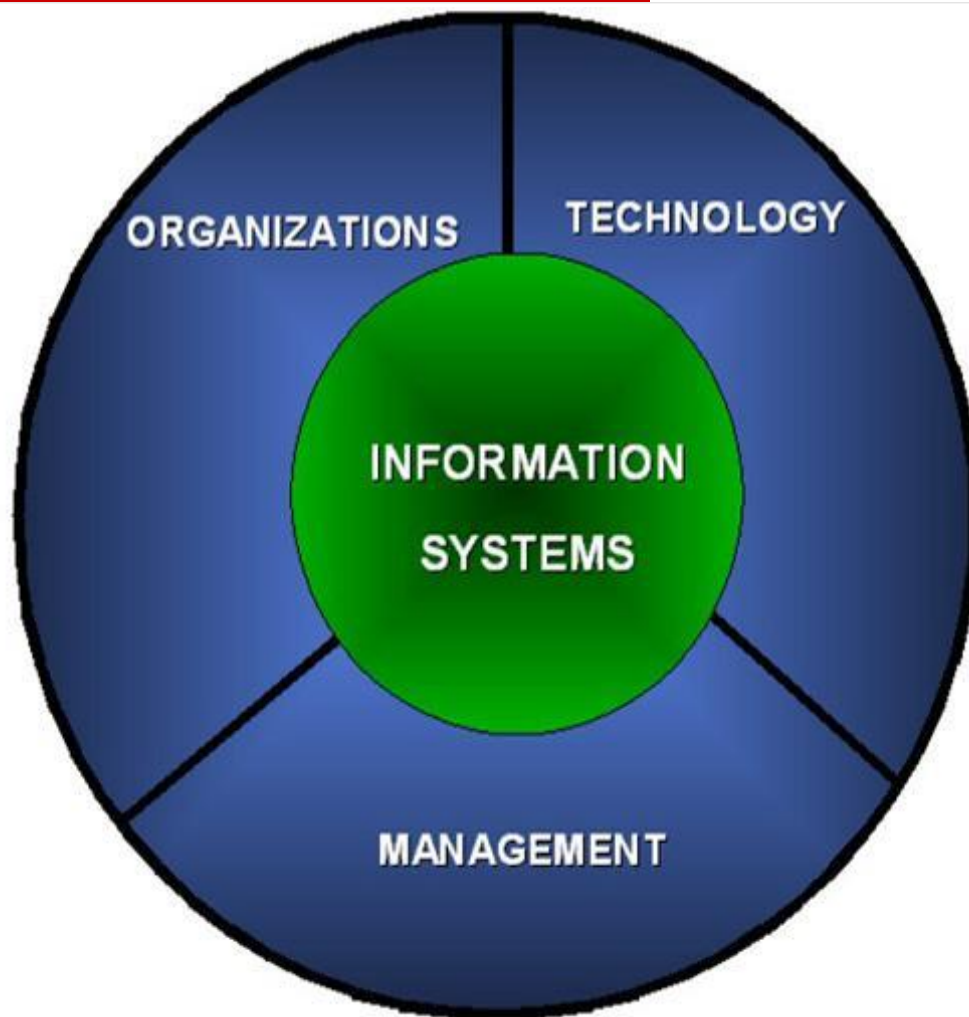


Тема лекції 1:

Системи баз даних. Основні поняття, архітектура, моделі даних.

- ❑ Основні визначення
 - ❑ Розподіл обов'язків в системах з базами даних
 - ❑ Функції СУБД
 - ❑ Трирівнева архітектура організації бази даних
 - ❑ Архітектура програмно-технічних засобів
 - ❑ Класифікації моделей даних
-

Інформаційні системи

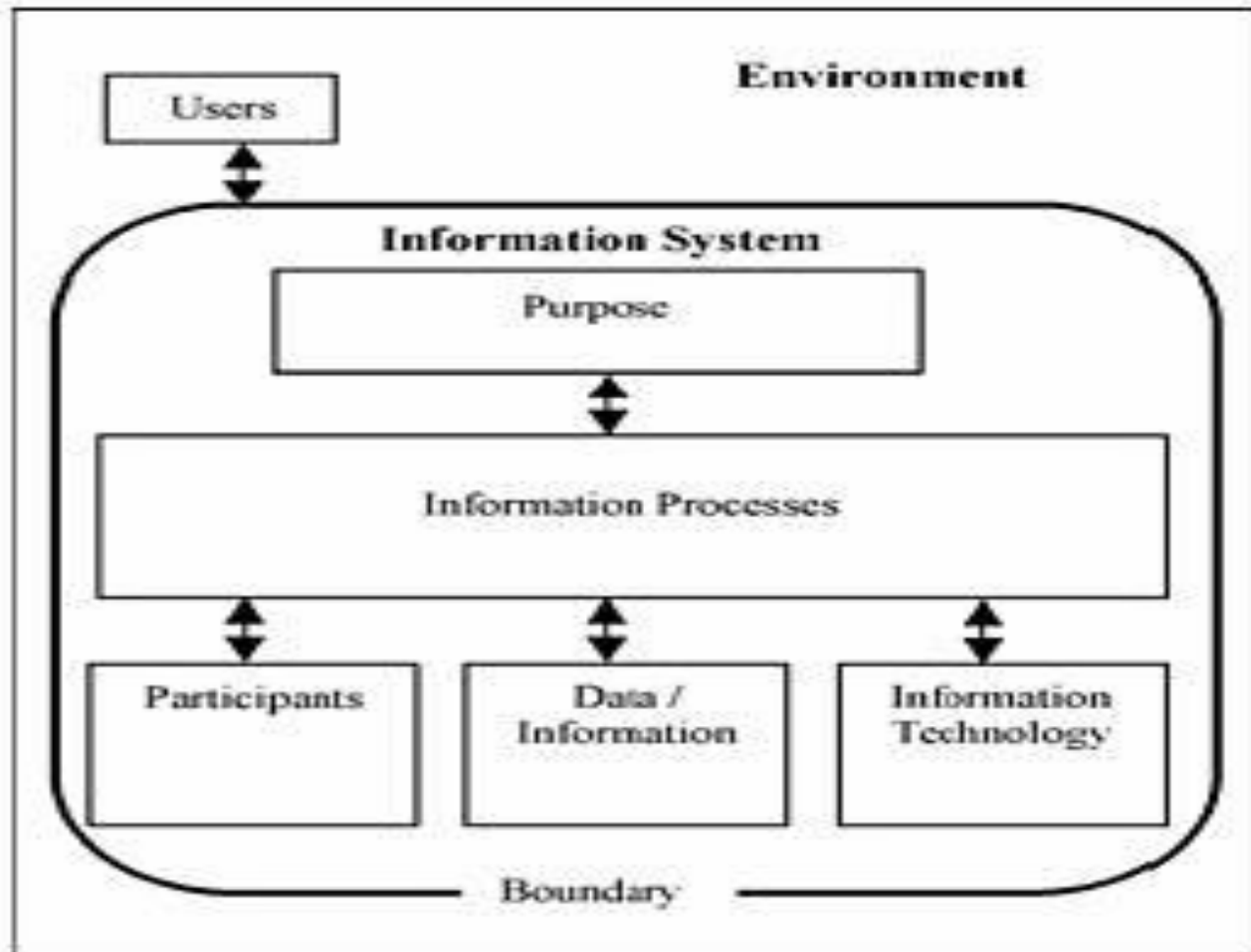


Інформаційна система – це програмний продукт

Функції ІС:

- ☐ підтримка надійного зберігання інформації;
 - ☐ виконання певних перетворень інформації або обчислень;
 - ☐ надання користувачам зручного інтерфейсу.
-

Інформаційна система

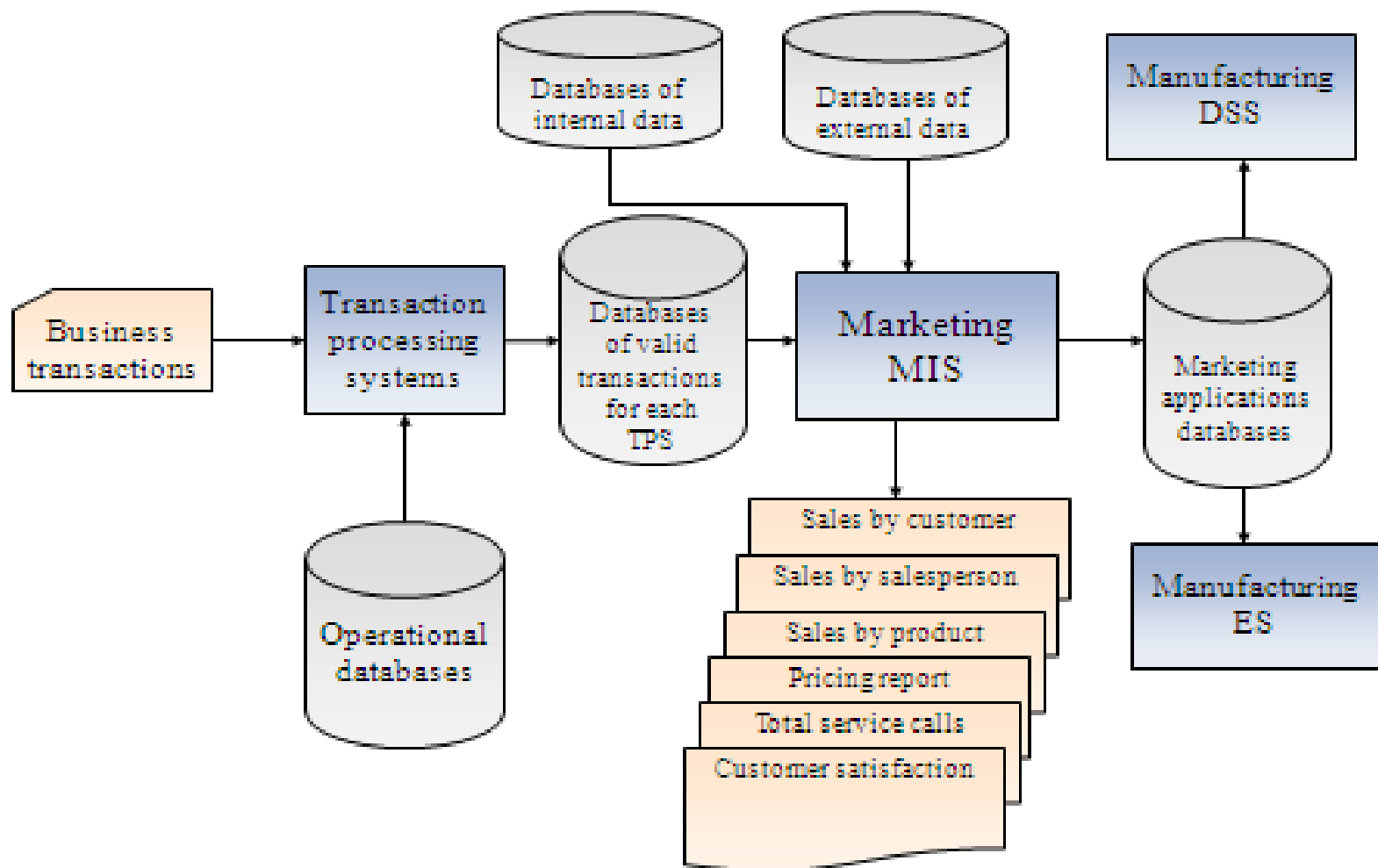


Визначення бази даних

База даних – це сукупність логічно зв'язаних даних (і опис цих даних), яка відображає стан об'єктів та їх зв'язків в певній предметній області, динамічно змінюється у процесі свого функціонування і є доступна багатьом користувачам [Дейт].

База даних – це єдине велике сховище даних, яке один раз визначається, а потім функціонує у багатокористувацькому режимі [Ульман].

Бази даних та інформаційні системи



Інші визначення

Схема (зміст) бази даних – це опис бази даних

Стан (деталізація) бази даних– це сукупність інформації, що зберігається в базі даних в будь-який певний момент часу

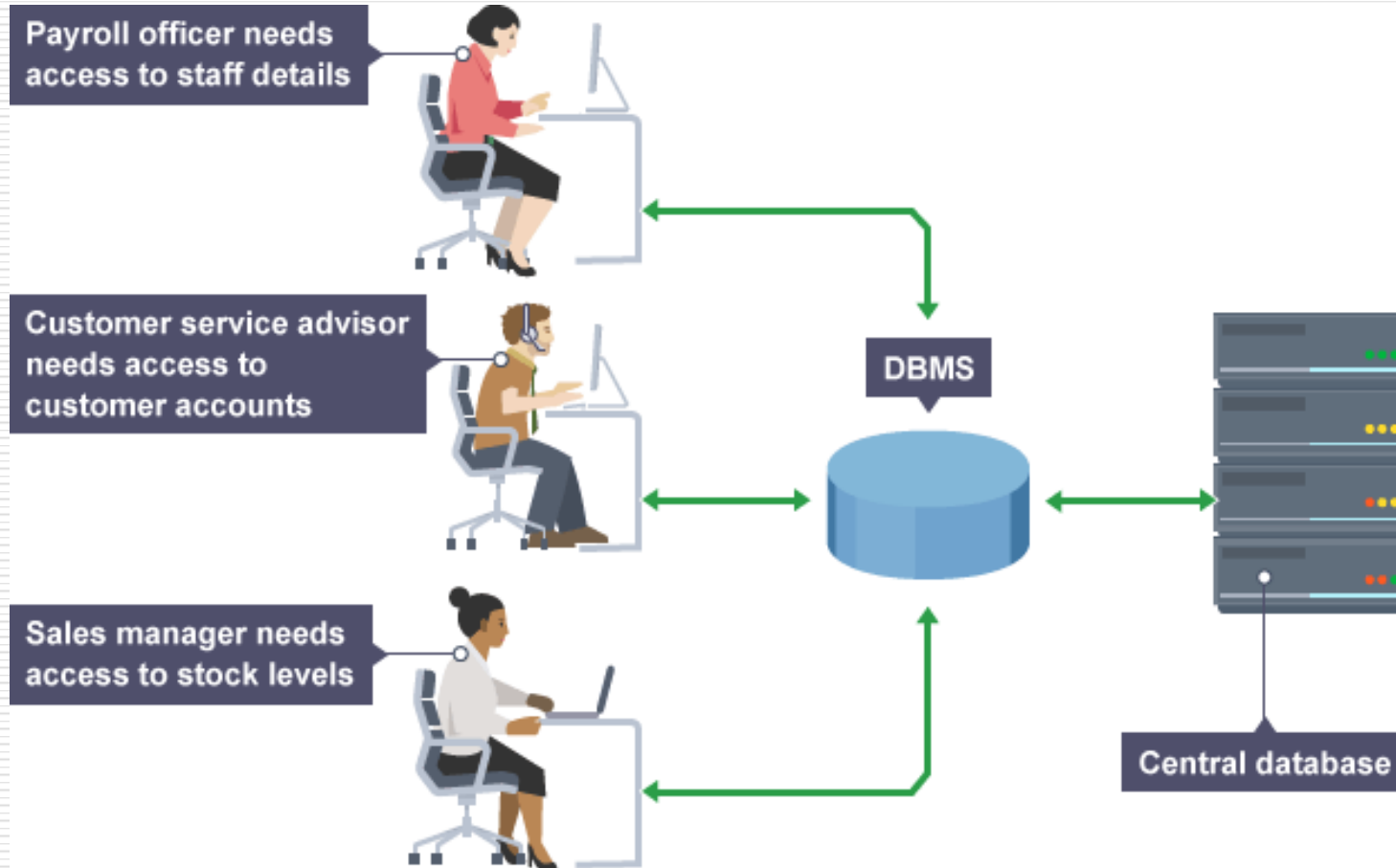
Прикладні програми, застосування, додатки баз даних

- можуть працювати паралельно і незалежно одна від одної
 - структура даних відділена від додатків
 - незалежність від даних
-

Система управління базами даних

СУБД – це програме забезпечення підтримки інтегрованої сукупності даних, призначене для створення, зберігання, ведення і використання бази даних багатьма користувачами (застосуваннями).

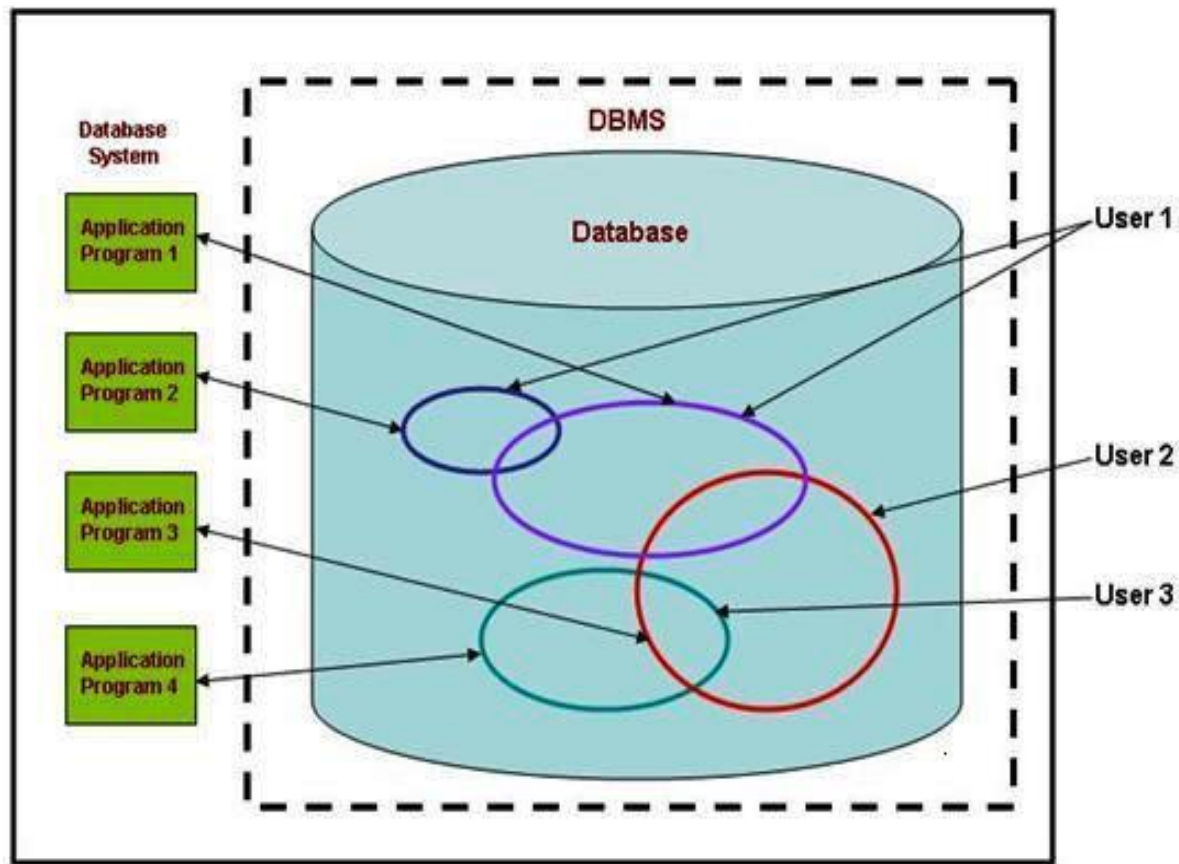
Система управління базами даних



Система баз даних

- це певна прикладна система, яка використовує базу даних і СУБД, що підтримує цю БД, призначена для вирішення конкретних завдань зберігання та опрацювання даних
-

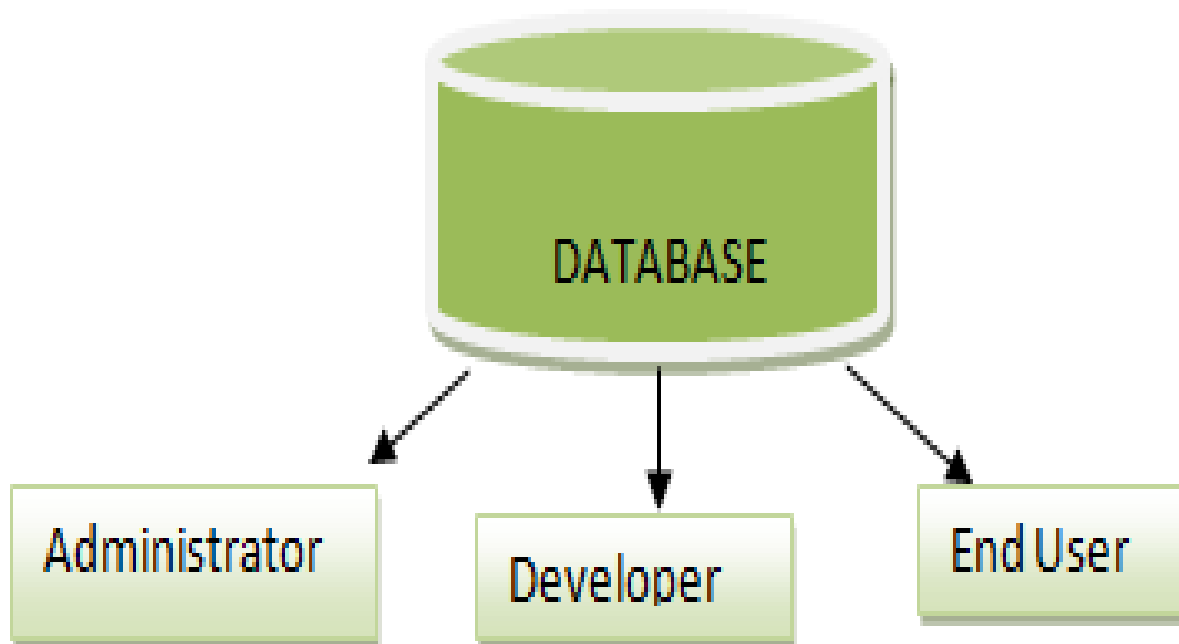
Система баз даних



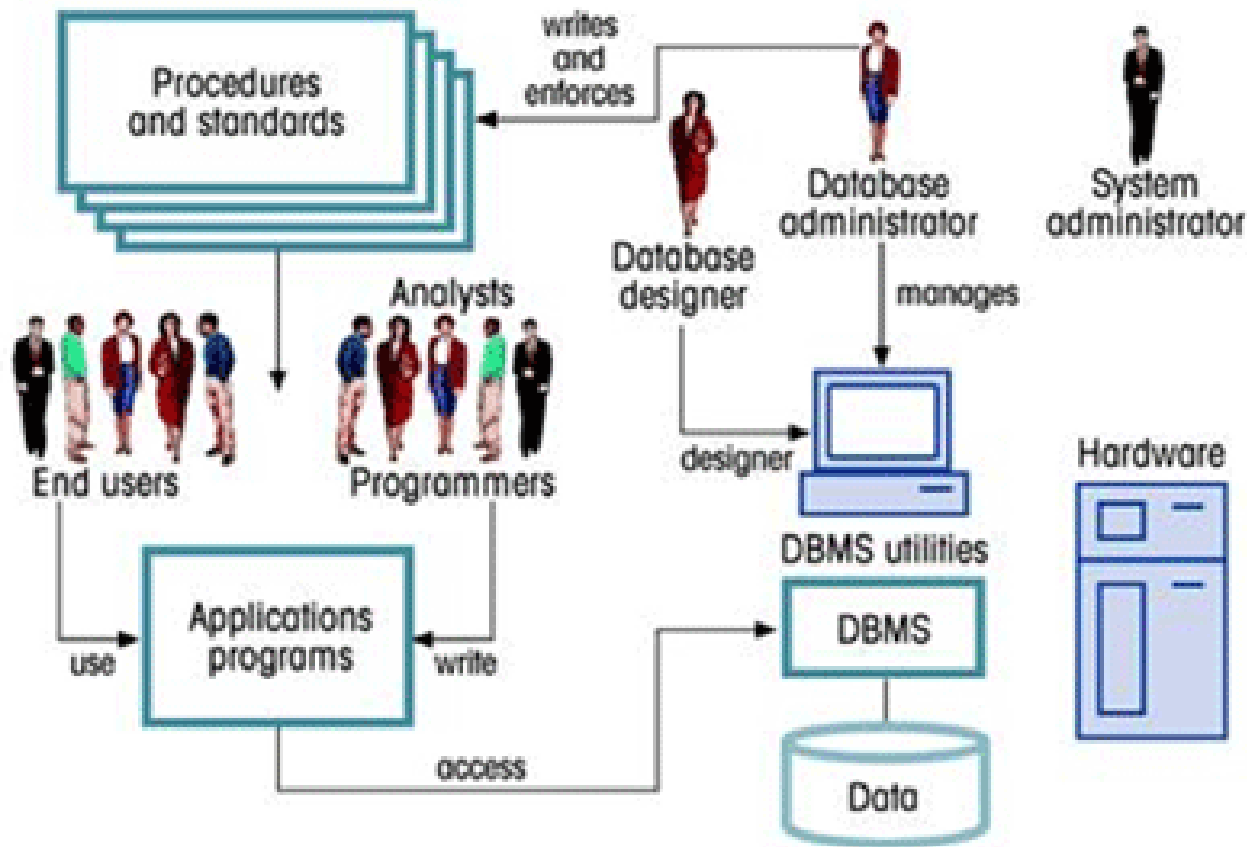
Розподіл обов'язків в системах з базами даних

1. Адміністратори даних та адміністратори баз даних
 2. Розробники баз даних
 3. Розробники додатків
 4. Користувачі-клієнти
 - Пересічні користувачі
 - Досвідчені користувачі
-

Розподіл обов'язків в системах з базами даних



Розподіл обов'язків в системах з базами даних



THE DATABASE SYSTEM ENVIRONMENT

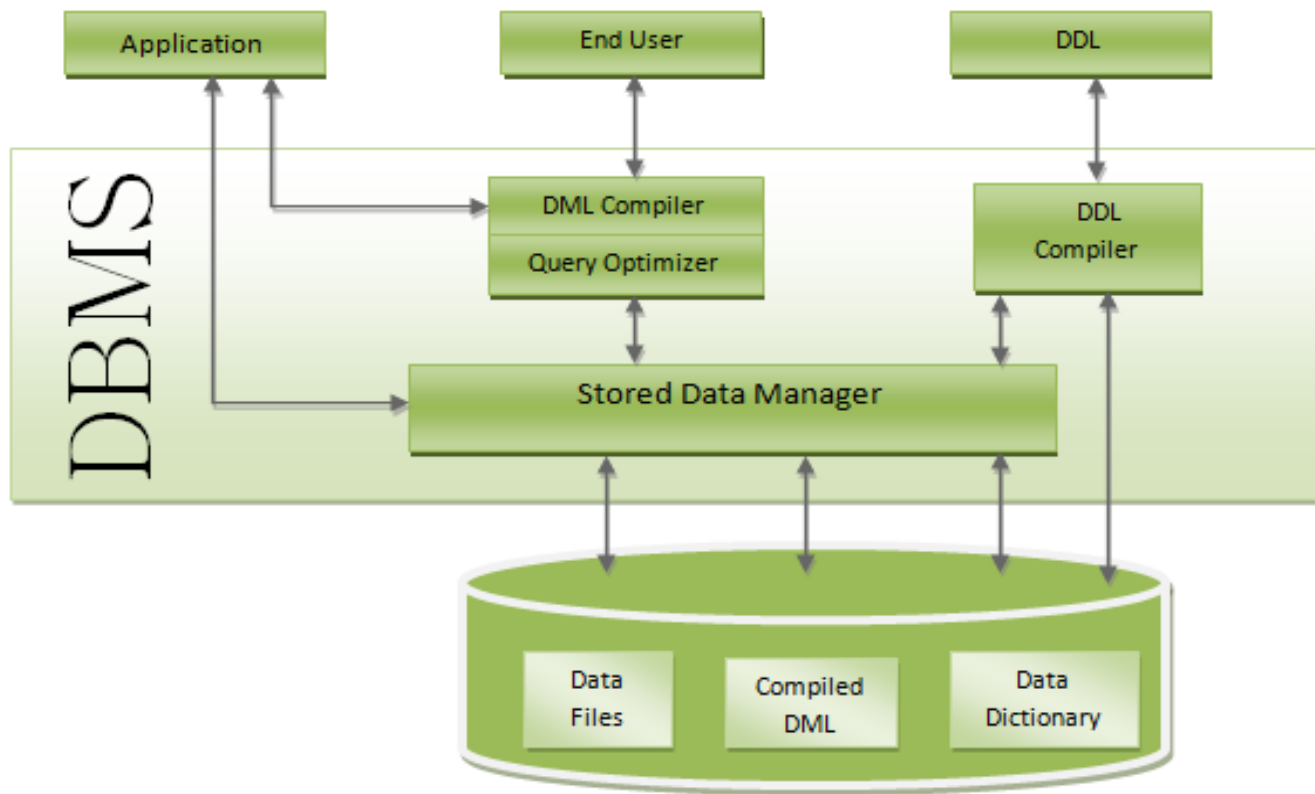
Функції СУБД

- ❑ Зберігання, вибір та оновлення даних
 - ❑ Визначення структури бази даних, її ініціалізація і проведення початкового завантаження даних
 - ❑ Наявність каталогу (бази метаданих)
 - ❑ Служба підтримки цілісності даних
 - ❑ Підтримка транзакцій
-

Функції СУБД (продовж.)

- ☐ Сервіс контролю за доступом до даних
 - ☐ Служба підтримки незалежності програм від даних
 - ☐ Підтримка обміну даними
 - ☐ Сервіс відновлення бази даних
 - ☐ Сервіс керування паралельністю
-

Функції СУБД



Функції СУБД.

Додаткові служби, утиліти

- ❑ утиліти імпортування та утиліти експортування бази даних;
 - ❑ засоби моніторингу, які слідкують за функціонуванням та використанням бази даних;
 - ❑ програми статистичного аналізу, які дозволяють оцінити продуктивність системи;
 - ❑ інструменти збору сміття та перерозподілу пам'яті.
-

Популярні реляційні СУБД



ORACLE®

PostgreSQL



SYBASE®

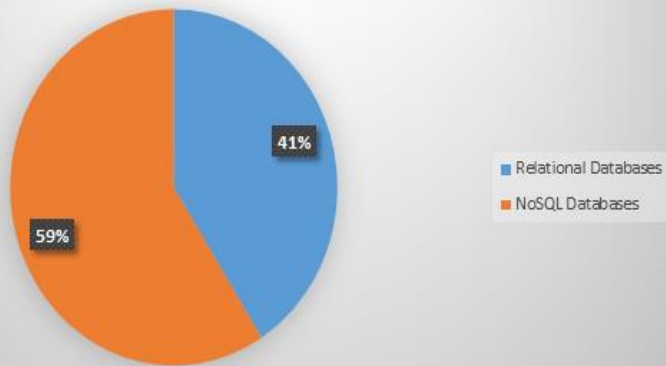


Популярні NoSQL СУБД



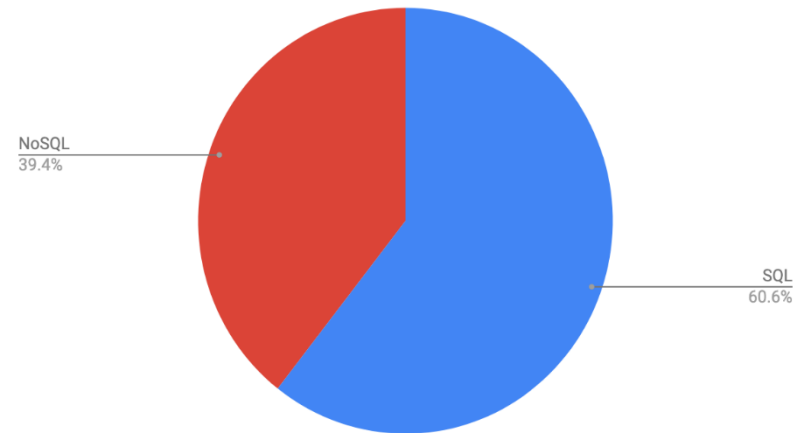
SQL vs. NoSQL популярність

Database Popularity (March 2014-2015)

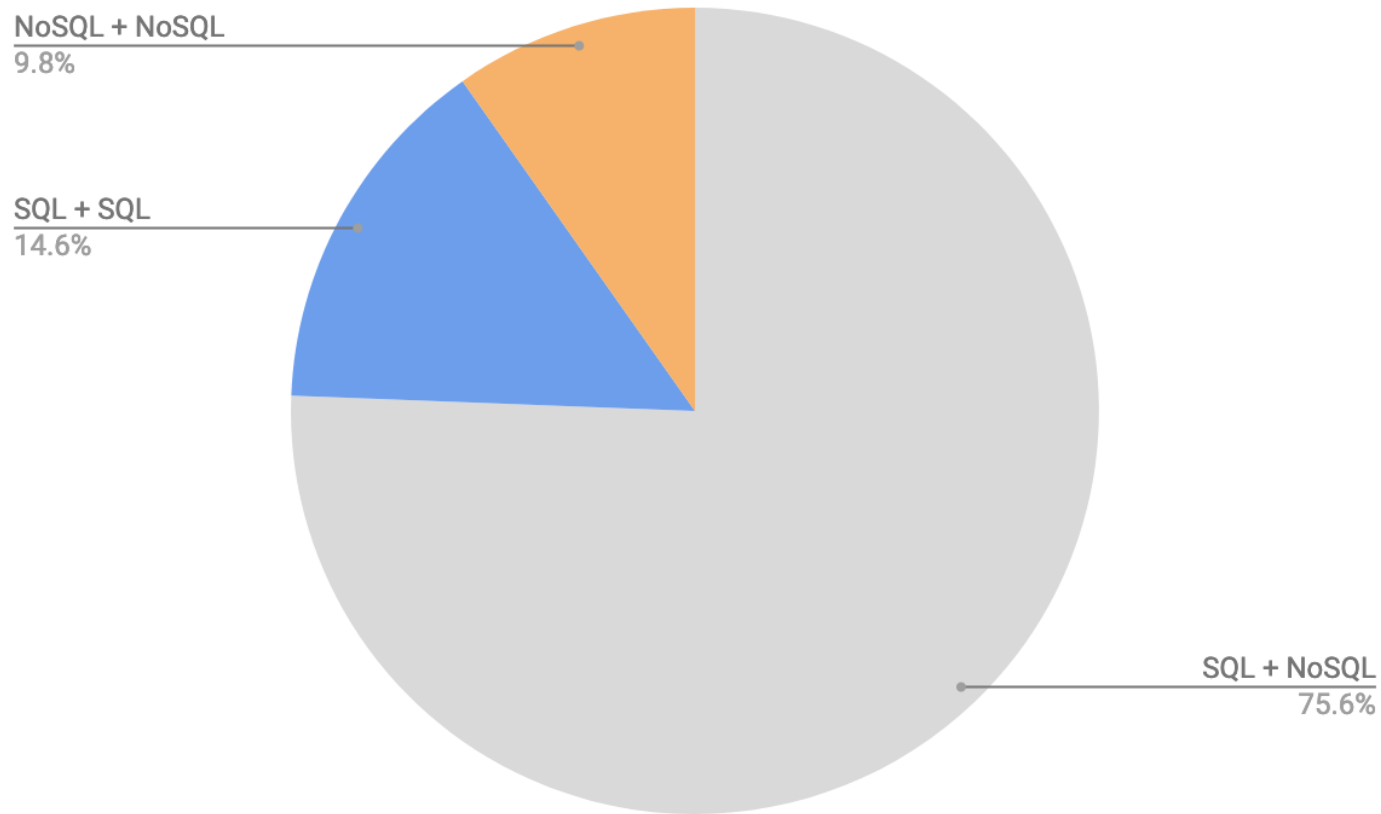


Source: <http://www.techrepublic.com>

SQL vs. NoSQL Open Source Database Popularity



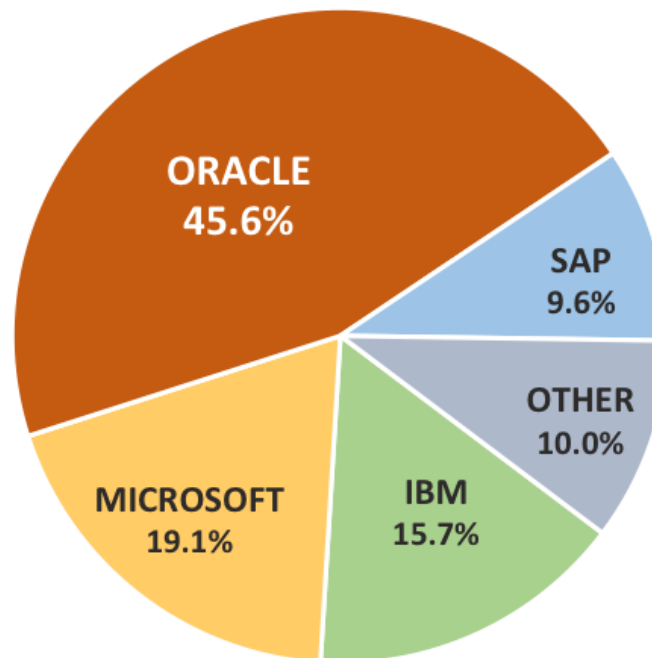
SQL & NoSQL Multiple Database Combinations Trends



Популярні комерційні РСУБД

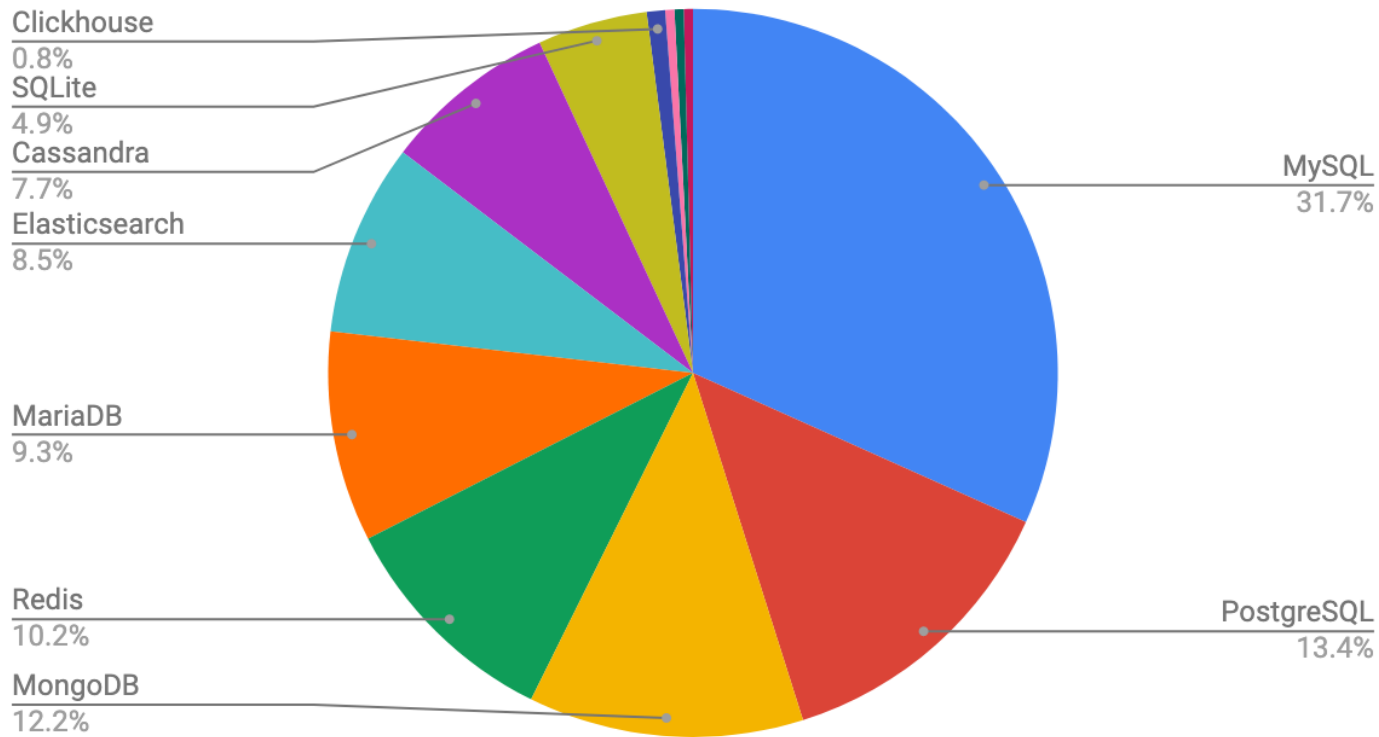
2016 COMMERCIAL DATABASE MARKET SHARE

(Source: Gartner, Inc. 2016)

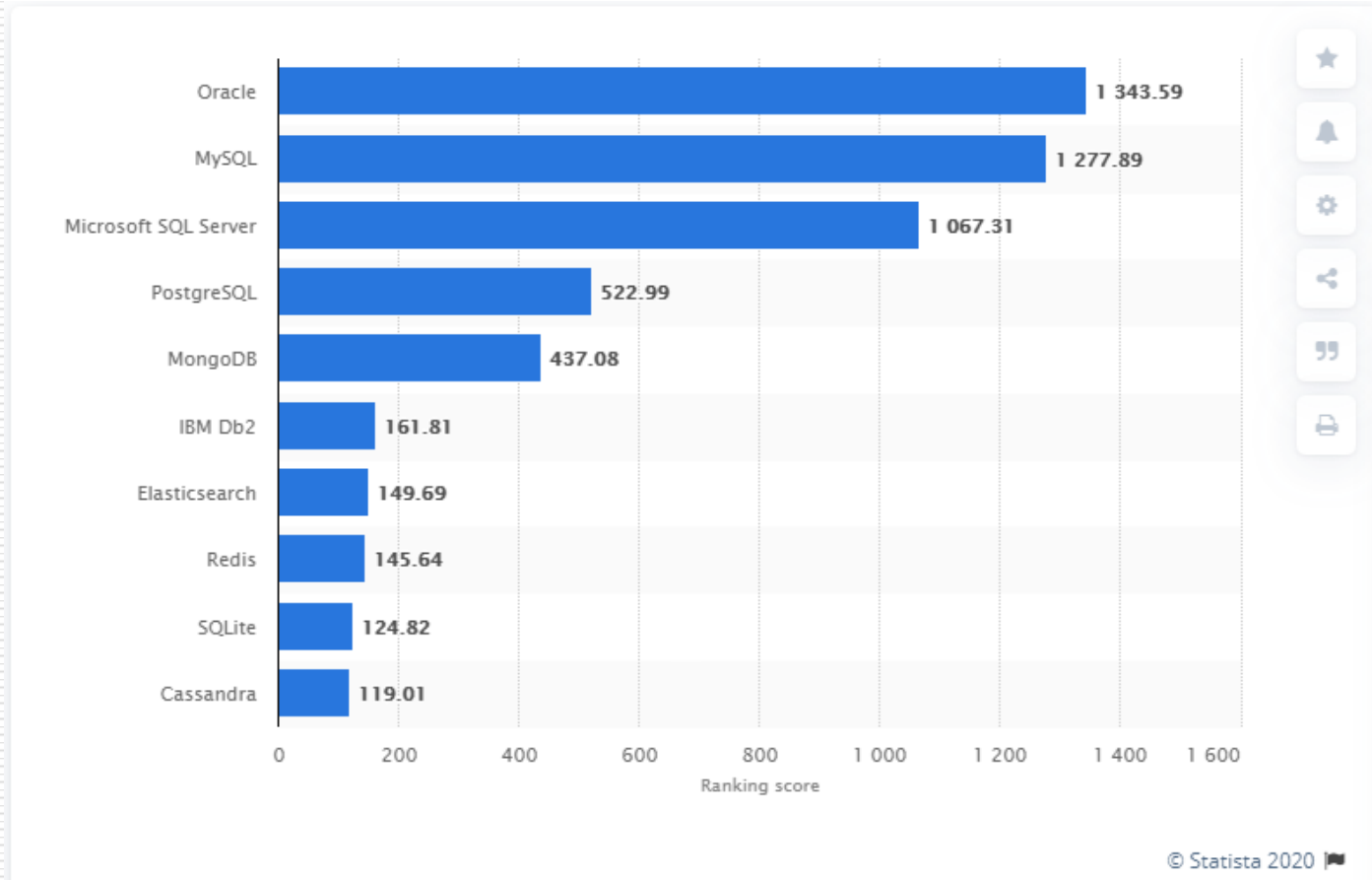


Популярні Open Source СУБД




2019 Most Popular Open Source Databases



Популярні СУБД



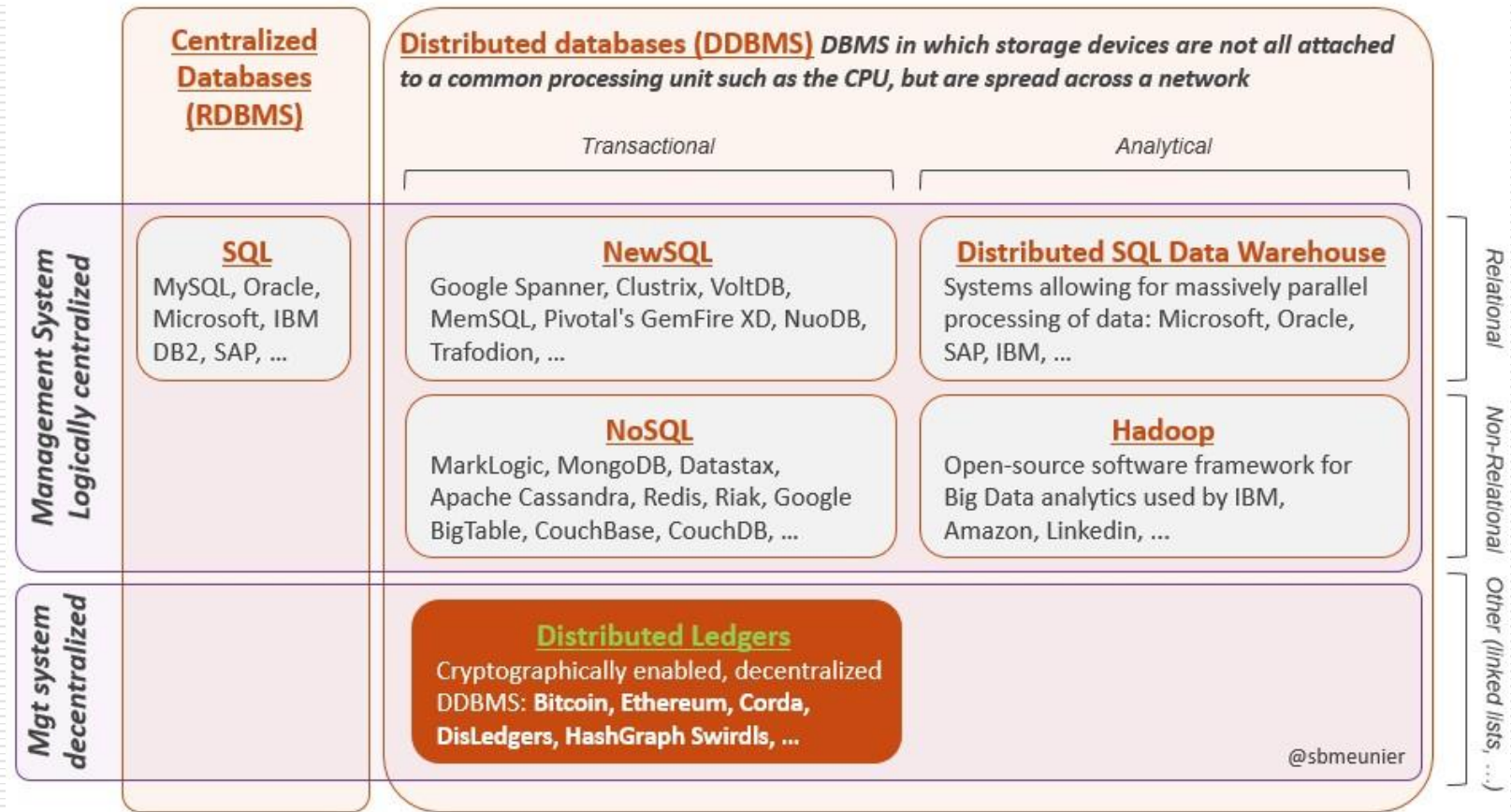
Популярні СУБД в розробці ІС

| Most Popular Database Platforms  Love  Dread  Want | | | | | | | | | |
|---|------|------|---------|------|------|------|------|------|------|
| | 2019 | 2018 | %Change | 2019 | 2018 | 2019 | 2018 | 2019 | 2018 |
| MySQL | 52% | 59% | -7% | 54% | 49% | 46% | 51% | 8% | 8% |
| PostgreSQL | 36% | 33% | 3% | 70% | 62% | 30% | 38% | 14% | 11% |
| MS SQL Server | 34% | 42% | -8% | 58% | 52% | 43% | 48% | 3% | 4% |
| SQLite | 30% | 20% | 10% | 56% | 48% | 45% | 52% | 7% | 53% |
| MongoDB | 26% | 26% | 0% | 60% | 55% | 41% | 45% | 18% | 45% |
| Redis | 20% | 19% | 1% | 71% | 65% | 29% | 36% | 11% | 35% |
| MariaDB | 17% | 14% | 3% | 59% | 53% | 41% | 47% | 4% | 47% |
| Oracle | 16% | 11% | 5% | 38% | 37% | 62% | 63% | 3% | 63% |
| Elasticsearch | 16% | 14% | 2% | 63% | 60% | 67% | 40% | 11% | 40% |
| Firebase | 12% | | | 61% | | 39% | | 8% | |
| DynamoDB | 7% | | | 55% | | 45% | | 4% | |
| Cassandra | 4% | | | 47% | | 53% | | 6% | |
| Couchbase | 2% | | | 37% | | 63% | | 2% | |

Gartner's Magic Quadrant for Operational Database Management Systems



СУБД за типами зберігання та опрацювання даних

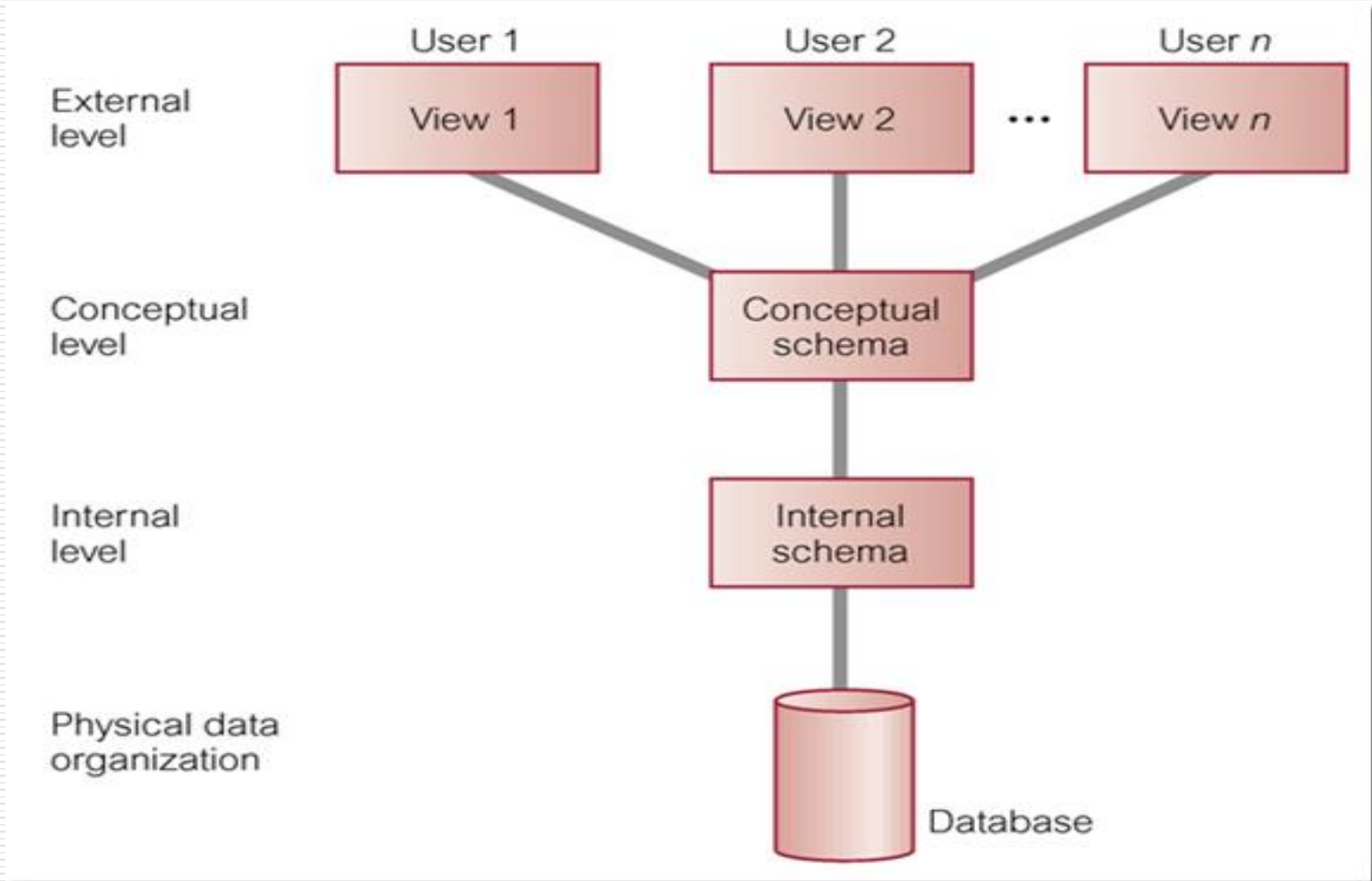


Трирівнева архітектура організації бази даних

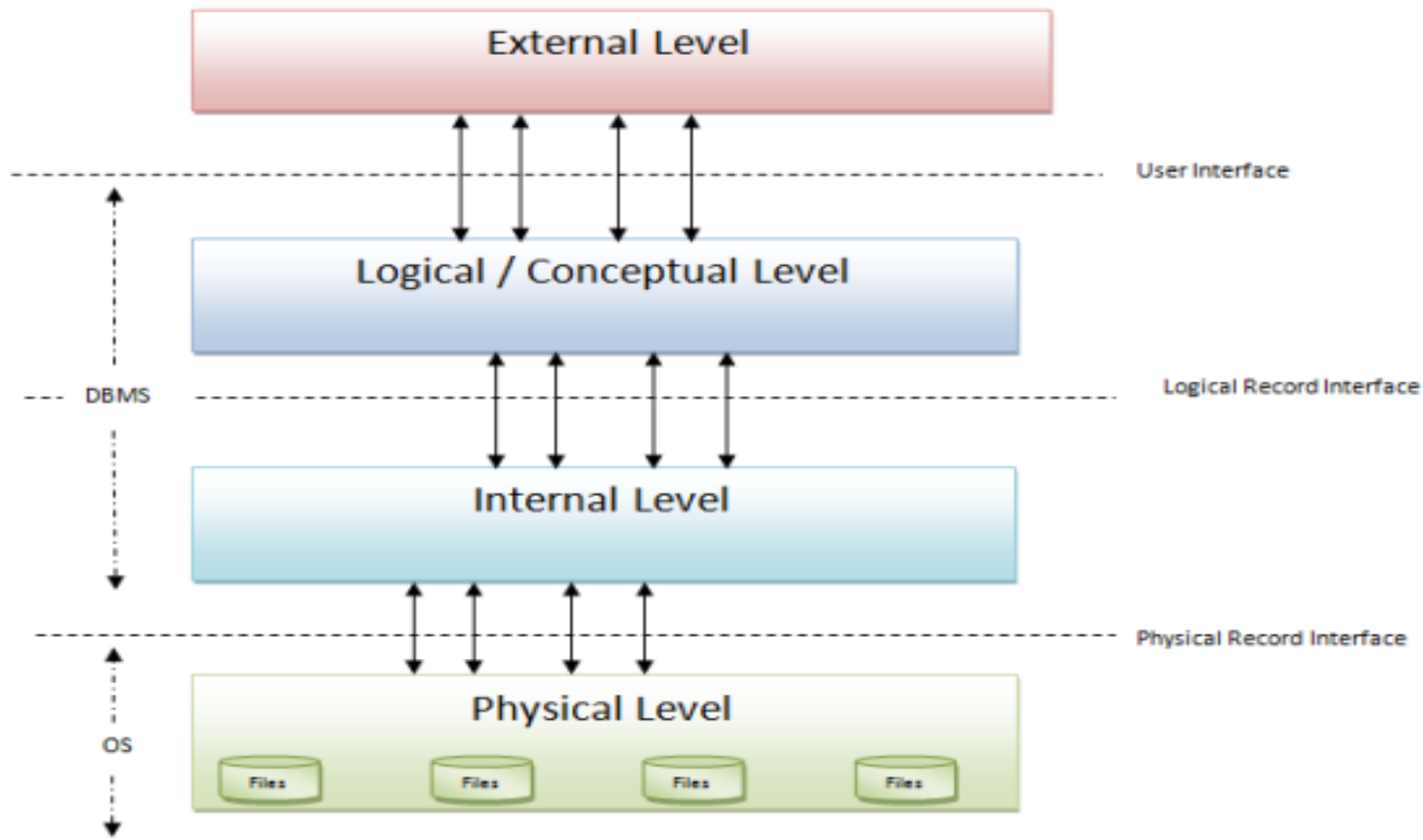
Вперше специфікована в 1975 році дослідницькою групою ANSI/X3/SPARC

- ANSI (American National Standard Institute) – Національний Інститут Стандартизації США,
 - X3 – його комітет обчислювальної техніки та обробки інформації,
 - SPARC (Standards Planning and Requirements Committee) – підкомітет ANSI/X3 з планування стандартів
-

Трирівнева архітектура ANSI-SPARC



Трирівнева архітектура ANSI-SPARC



Зовнішній рівень (external level)

- описує ту частину бази даних, яка відноситься до кожного користувача
 - складається з декількох різних зовнішніх схем (представлень) бази даних
 - зовнішні схеми можуть по-різному відображати одні і ті ж дані
-

Концептуальний рівень (conceptual level)

- ❑ здійснюється інтегрований опис предметної області
 - ❑ містить логічну структуру всієї бази даних
 - опис усіх сутностей, їх атрибутів і зв'язків;
 - підтримку цілісності даних;
 - обмеження, які накладаються на дані;
 - семантичну інформацію про дані;
 - ❑ підтримує кожну зовнішню схему
 - ❑ об'єднує дані, які використовуються усіма додатками, що працюють з базою даних
 - ❑ не містить ніяких відомостей про методи зберігання даних
-

Внутрішній рівень (internal level)

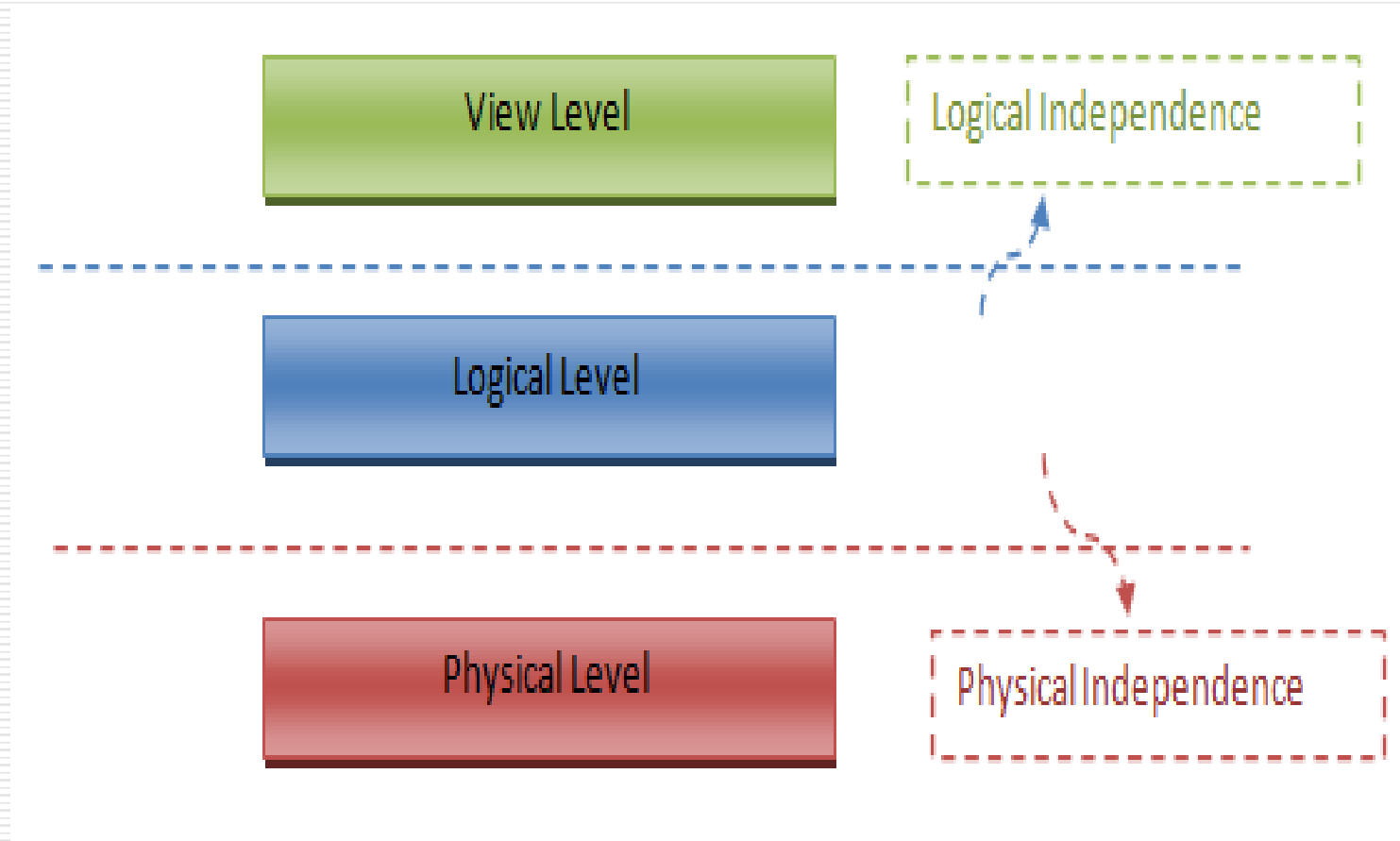
- описує фізичну реалізацію бази даних з урахуванням способів зберігання даних і методів доступу до них
 - містить наступну інформацію:
 - розподіл дискового простору для зберігання даних та індексів;
 - відомості про розміщення записів;
 - інформація про міри безпеки даних;
 - відомості про стиснення даних та методи їх шифрування.
-

Відображення

«концептуальний – зовнішній» і
«концептуальний – внутрішній»
дають змогу вирішувати проблему
логічної та фізичної незалежності
даних:

*будь-які зміни на одному з рівнів не
повинні спричиняти зміни на інших
рівнях, а лише має змінитися
відповідне відображення*

Незалежність даних



Логічна незалежність даних

- повна захищеність зовнішніх схем від змін, що вносяться в концептуальну схему
 - можливість змінювати логічне представлення бази даних без необхідності змінювати фізичні структури зберігання даних
-

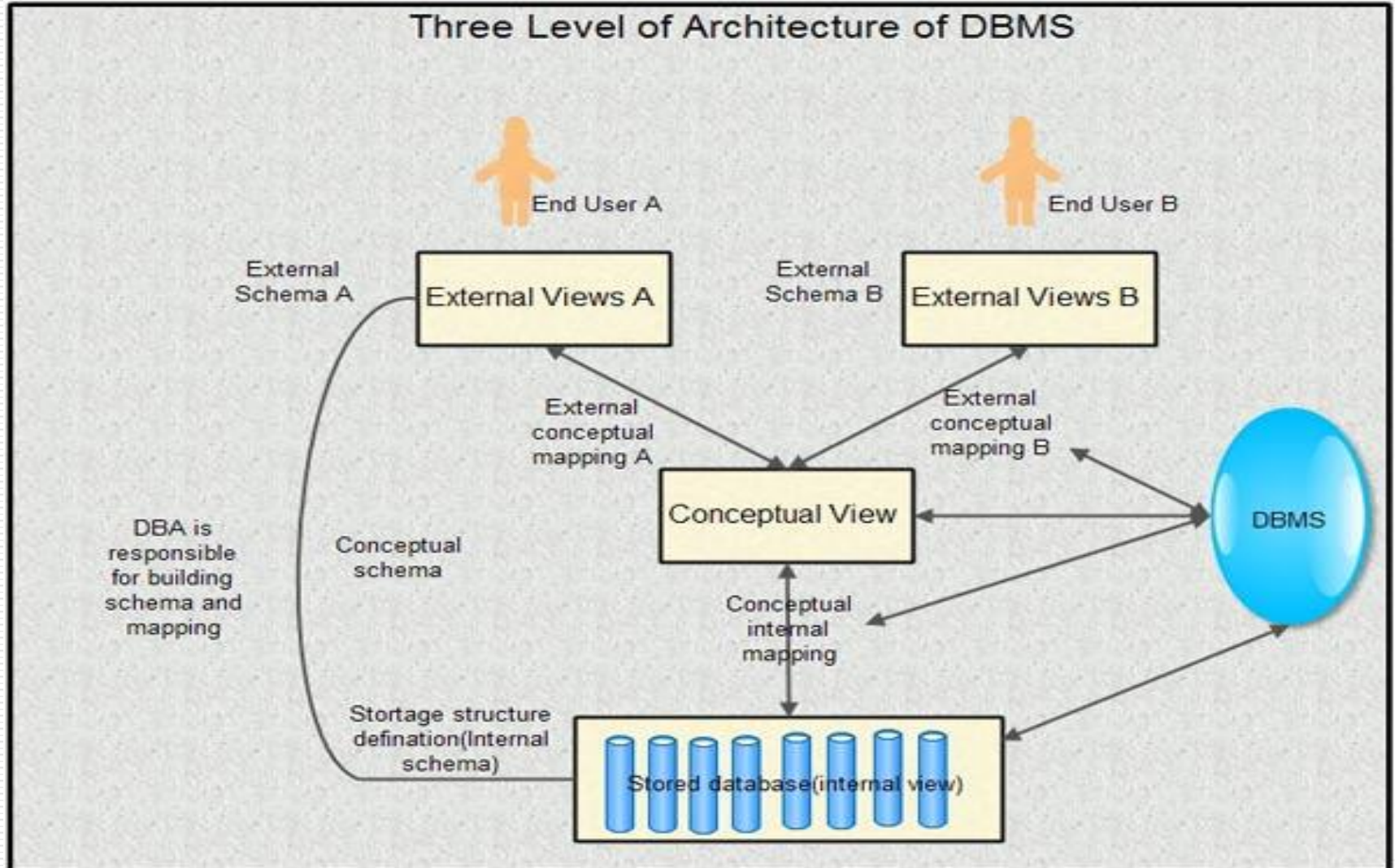
Фізична незалежність

- захищеність концептуальної схеми від змін, що вносяться у внутрішню схему
 - зміни помітні лише в загальній продуктивності системи
-

Сучасна трирівнева архітектура бази даних

1. Рівень **представлення інформації** забезпечує інтерфейс з користувачем (людиною або програмою)
 2. Рівень обробки даних (рівень **бізнес-логіки**) визначає функціональність і працездатність системи в цілому
 3. Рівень **зберігання даних** забезпечує фізичне зберігання, додавання, модифікацію і вибірку даних; перевіряється цілісність і узгодженість даних; реалізація транзакцій
-

Сучасна трирівнева архітектура бази даних

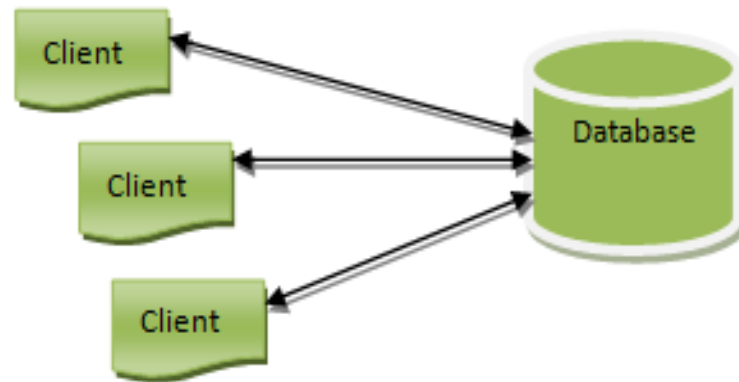
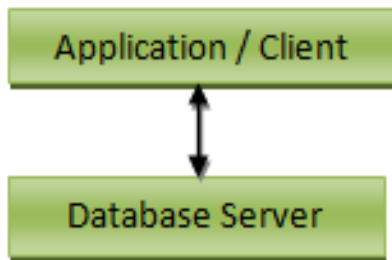


Архітектури програмно-технічних засобів

- ☐ файл-сервер
 - ☐ клієнт-сервер з бізнес-логікою на клієнті
 - ☐ клієнт-сервер з бізнес-логікою на сервері
 - ☐ N-рівнева архітектура
-

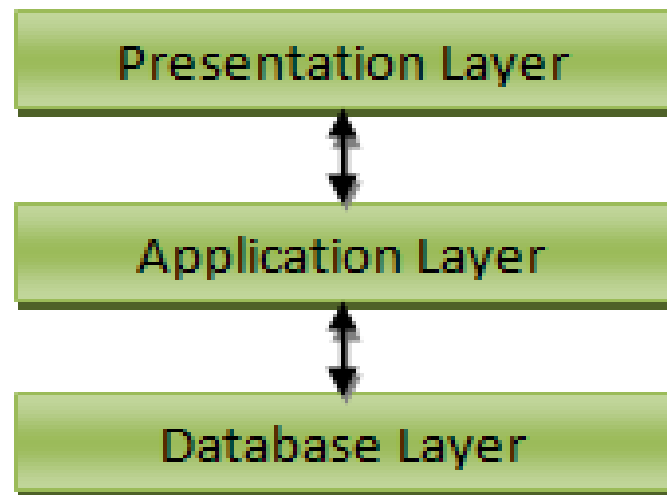
Архітектури програмно-технічних засобів

- ❑ клієнт-сервер з бізнес-логікою на клієнті
- ❑ клієнт-сервер з бізнес-логікою на сервері



Архітектури програмно-технічних засобів

□ N-рівнева архітектура



Відповідність логічної архітектури БД і програмно-технічної архітектури

| Тип архітектури | <i>Файл-сервер</i> | <i>Клієнт-сервер (бізнес-логіка на клієнті)</i> | <i>Клієнт-сервер (бізнес-логіка на сервері)</i> | <i>N- рівнева архітектура</i> |
|---------------------------------|--|---|--|---|
| | | | | |
| <i>Представлення інформації</i> | Клієнт | Клієнт | Клієнт | Клієнт |
| <i>Бізнес-логіки</i> | Клієнт | Клієнт | Сервер БД | Сервер прикладних програм (комп. кластер) |
| <i>Зберігання даних</i> | Файл-сервер (або клієнт) | Сервер БД | Сервер БД | Сервер БД |
| Реалізація | Усі три рівні утворюють єдиний програмний модуль | Інтерфейс користувача і бізнес логіка утворюють єдиний модуль. Дані зберігаються на сервері | Усі бізнес-логіка реалізована у вигляді збережених процедур, які виконуються на сервері БД | Усі рівні реалізовані на різних машинах |

Архітектура файл-сервер

- ❑ усі рівні системи представляють єдине і неподільне ціле
 - ❑ БД зберігається у вигляді файлу або набору файлів на файл-сервері
 - ❑ уся логіка вибірки, зберігання і забезпечення узгодженості даних покладається на клієнтську частину
 - ❑ обробка інформації ведеться на рівні окремих кортежів (записів)
-

Архітектура файл-сервер

□ **Переваги:**

- простота логіки;
- низькі вимоги до апаратного забезпечення і малий об'єм необхідної пам'яті;
- не вимагає надійних багатозадачних і багатокористувацьких ОС;
- невисока ціна СУБД.

□ **Недоліки**

- обмеженість мови і негнучкість середовища розробки додатків;
 - слабка масштабованість;
 - не забезпечує багатокористувацький режим роботи;
 - важко підтримувати цілісність і узгодженість даних;
 - необхідно вручну блокувати записи або таблиці;
 - низький рівень захищеності як зовнішньої (від зламу), так і внутрішньої (від програмних помилок), наприклад, індекси окремо від таблиць;
 - не має засобів шифрації мережевого трафіку;
 - створює велике навантаження на мережу.
-

Архітектура клієнт-сервер з бізнес-логікою на клієнті

- ❑ Зберігання, вибірка та підтримка узгодженості даних покладається на сервер БД
 - ❑ уся бізнес-логіка і логіка представлення інформації виконуються на клієнтських машинах
 - ❑ продуктивність і збереженість даних залежить від сервера БД
 - ❑ клієнтська частина обмінюється даними з сервером за допомогою запитів SQL
 - ❑ опрацювання інформації ведеться на рівні множини кортежів (записів)
 - ❑ процес розробки розділяється на створення БД і написання клієнтської частини з бізнес-логікою
-

Архітектура клієнт-сервер з бізнес-логікою на клієнті

□ **Переваги:**

- висока продуктивність, стабільність і надійність при багатокористувацькому режимі роботи;
- легко організовується захист даних (шифрування мережевого трафіку SSH, SSL);
- універсальність мови визначення та маніпулювання даними.

□ **Недоліки**

- більша ціна СУБД (сервер БД продається окремо);
 - достатньо високі вимоги до кваліфікації розробників;
 - необхідні навички адміністрування сервера БД;
 - підвищені вимоги до пропускної здатності мережі;
 - підвищені вимоги до клієнтських місць (на них виконується рівень бізнес-логіки).
-

Архітектура клієнт-сервер з бізнес-логікою на сервері

- на сервер переноситься максимально можлива частина бізнес-логіки
 - можливість сучасних серверів БД виконувати збережені процедури на сервері
-

Архітектура клієнт-сервер з бізнес-логікою на сервері

□ **Переваги:**

- знижені, порівняно з попереднім класом систем, вимоги до пропускної здатності мережі та до клієнтських машин;
- простіший процес створення бізнес-логіки.

□ **Недоліки**

- підвищені вимоги до сервера БД, оскільки кожний сеанс «з'їдає» пам'ять з розрахунком граничного навантаження;
 - невисока мобільність системи на інші сервери БД.
-

N- рівнева архітектура

- ❑ основними елементами є сервери БД, сервер (кластер) прикладних програм і клієнтська частина
 - ❑ максимальне спрощення клієнта і сервера БД
 - ❑ тонкий клієнт являє собою деякий термінал типу браузера
 - ❑ уся бізнес-логіка оформляється у вигляді набору прикладних програм, які запускаються на сервері-кластері
 - ❑ сервери БД займаються лише задачами зберігання, додавання, модифікації та підтримки узгодженості даних
 - ❑ сервер програм з'єднаний з сервером БД окремим високошвидкісним сегментом мережі
-

N- рівнева архітектура

□ **Переваги:**

- підвищена захищеність;
- висока продуктивність;
- гнучкість розвитку та модифікації;
- простота адміністрування;
- можливість створення системи з масовим паралелізмом, оскільки серверів БД може бути декілька, а сервером програм можуть служити декілька з'єднаних в кластер комп'ютерів.

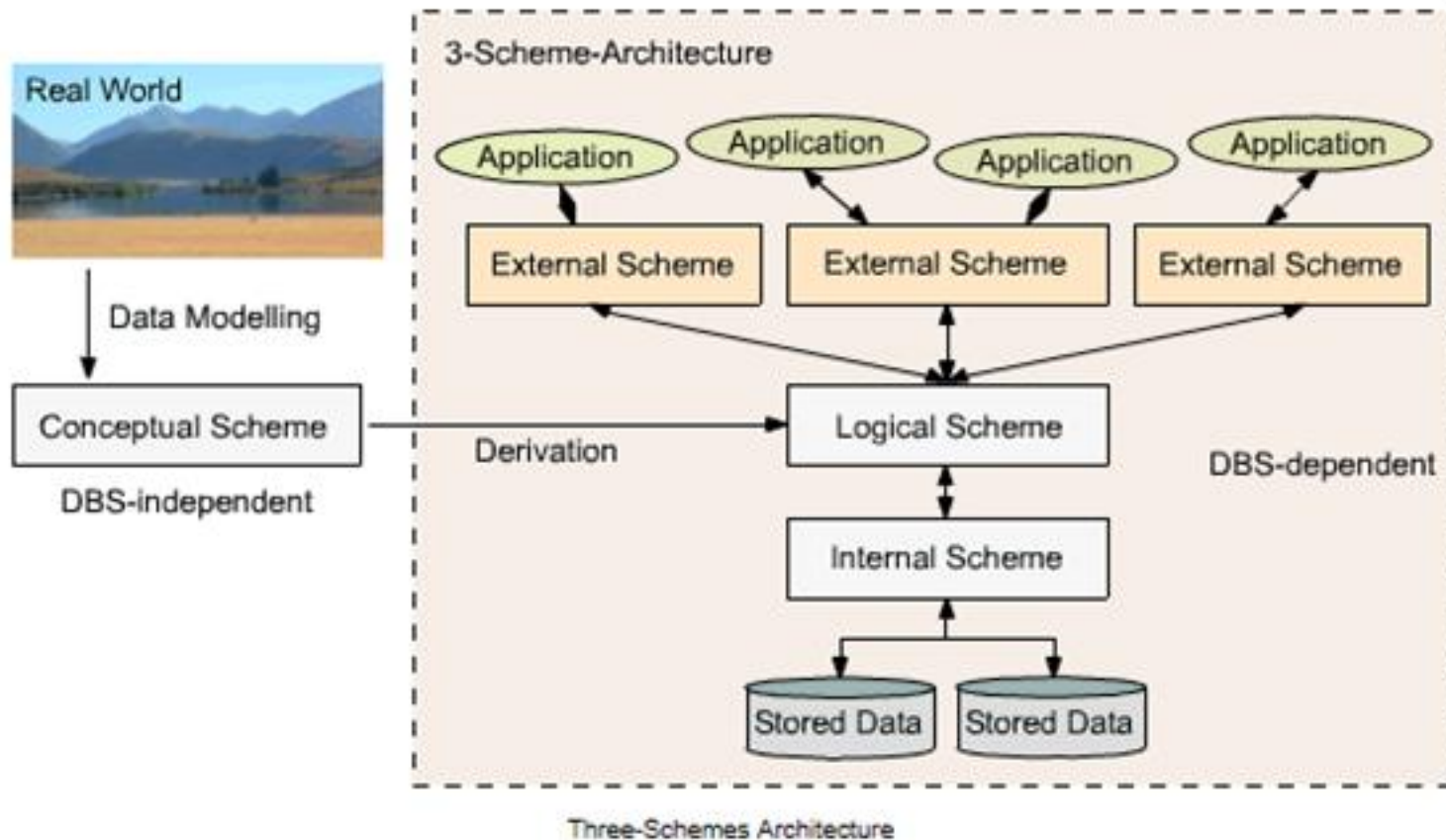
□ **Недоліки**

- велика складність архітектури;
 - висока ціна рішення;
 - у деяких випадках поступається в продуктивності клієнт-серверним системам з бізнес-логікою на сервері.
-

Визначення даних і моделей даних

- ❑ «**Дані**» в концепції баз даних – це набір конкретних значень, параметрів, які характеризують об'єкт, умову, ситуацію або інші фактори.
 - ❑ Модель даних – це представлення «реальних» об'єктів, подій та існуючих між ними зв'язків. Це деяка абстракція, яка застосовується до певних даних, і в якій акцент робиться на найважливіших аспектах, а всі другорядні властивості ігноруються.
 - ❑ «**Модель даних**» в концепції баз даних – це інтегрований набір понять для опису даних, зв'язків між ними та обмежень, які накладаються на дані, в деякій інформаційній системі.
-

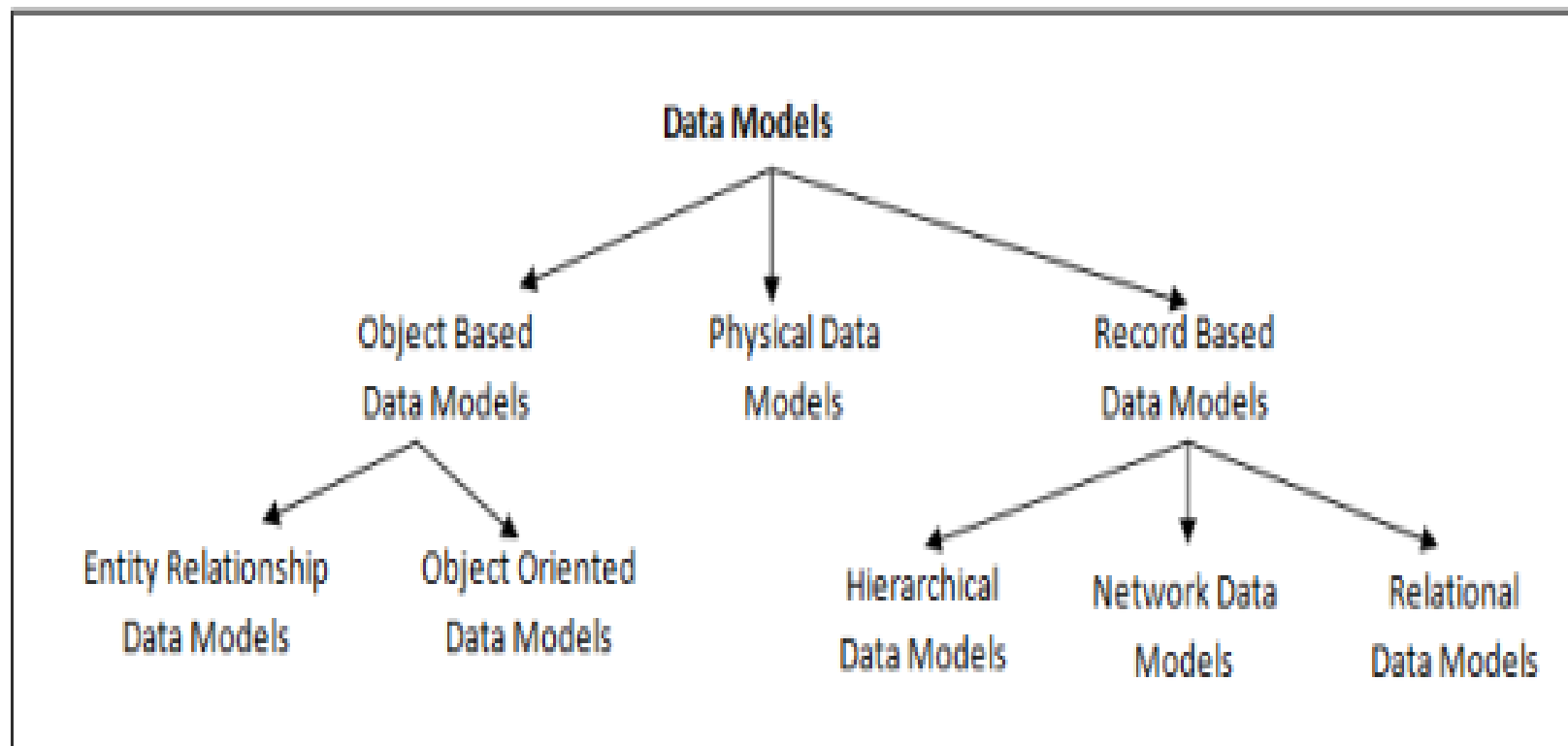
Модель даних в архітектурі БД



Класифікація 1 моделей даних

- ❑ об'єктні (object-based) моделі даних - опис даних на концептуальному та зовнішньому рівнях
 - ❑ моделі даних на основі записів (record-based) - опис даних на концептуальному та зовнішньому рівнях
 - ❑ фізичні моделі даних - опис даних на внутрішньому рівні
-

Класифікація 1 моделей даних



Класифікація 1 моделей даних.

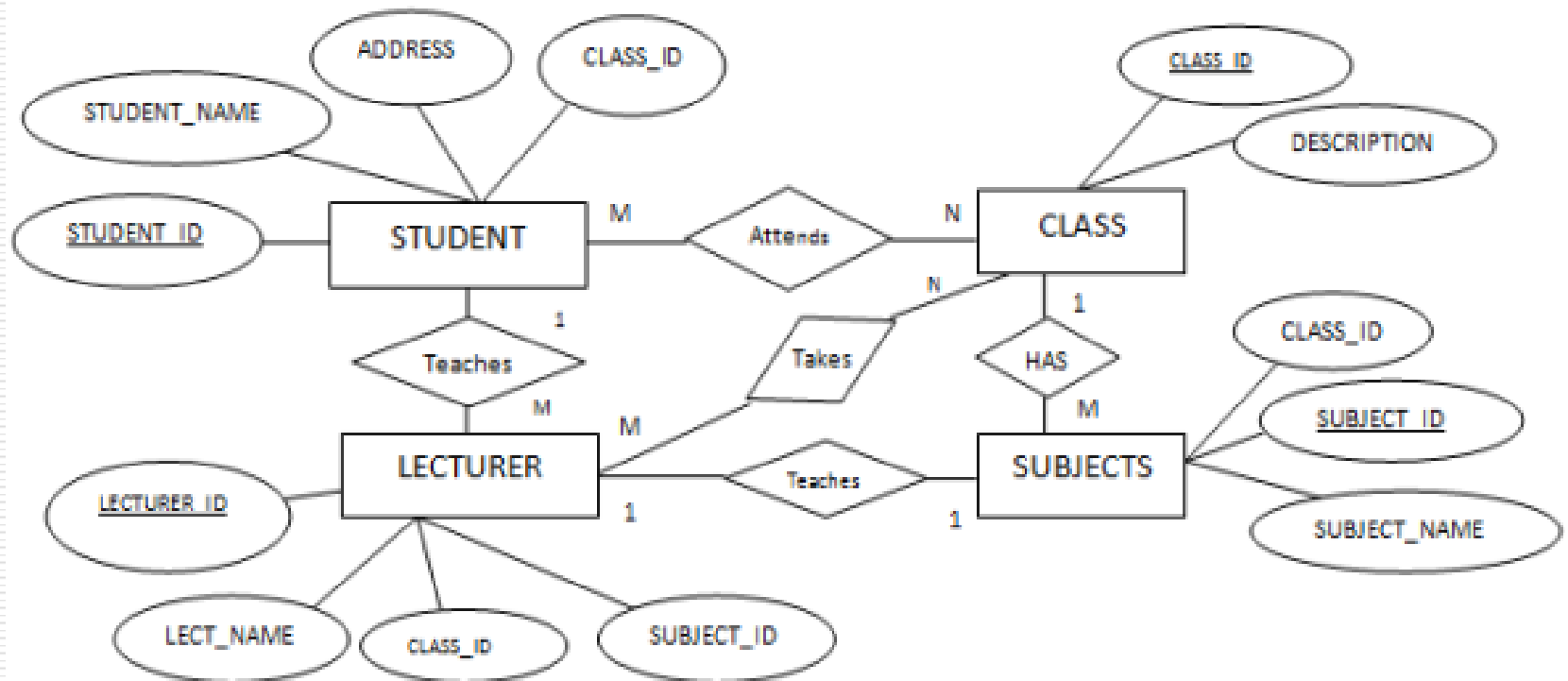
Об'єктні (object-based) моделі даних

- використовуються такі поняття як сутності, атрибути і зв'язки.

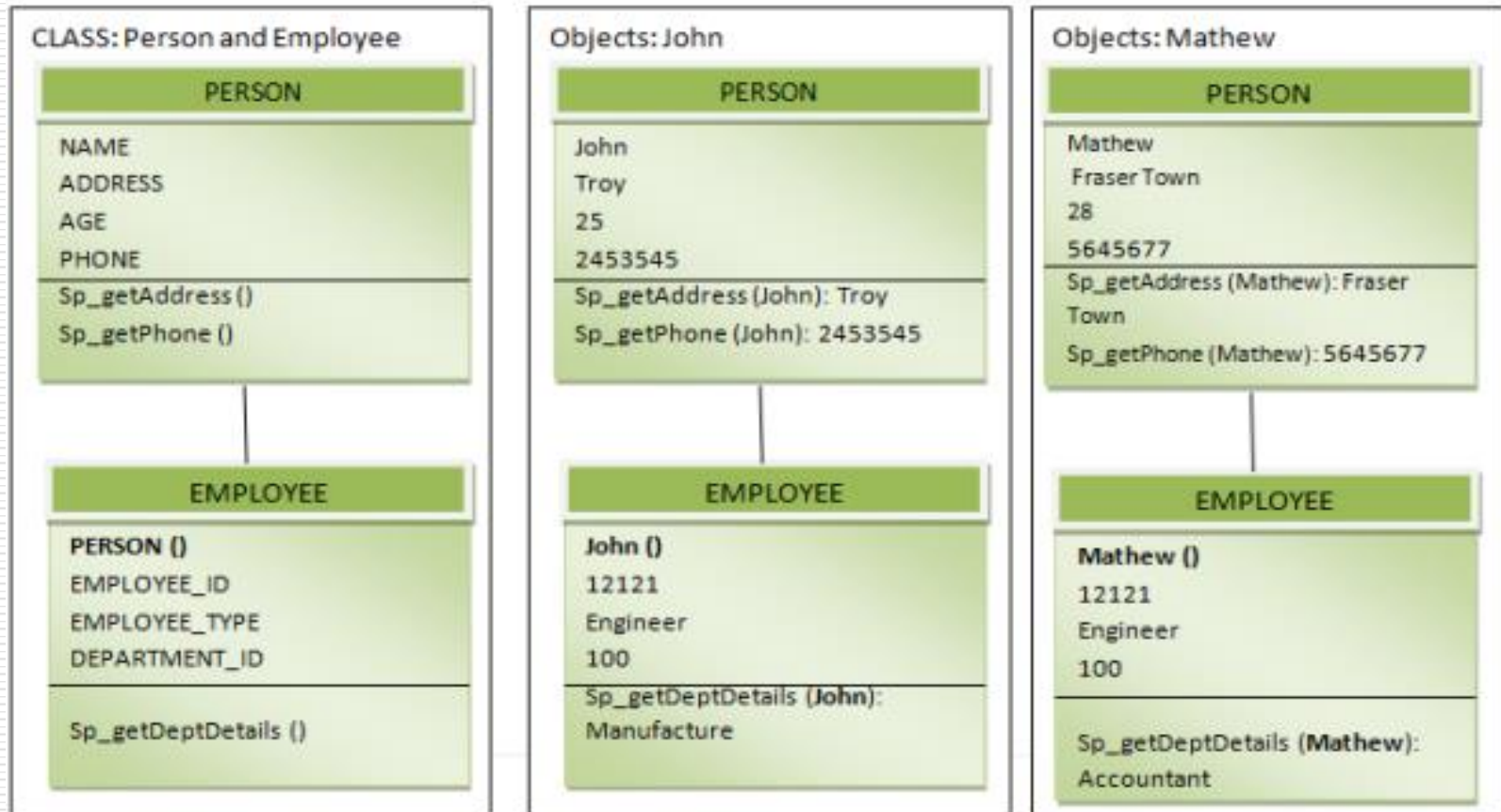
Загальні типи об'єктних моделей даних:

- ☐ модель типу „сутність-зв'язок“, або ER-модель (Entity-Relationship model);
 - ☐ семантична модель;
 - ☐ функціональна модель;
 - ☐ об'єктно-орієнтована модель.
-

Об'єктні (object-based) моделі даних. Entity-Relationship model



Об'єктні (object-based) моделі даних. Об'єктно-орієнтована модель



Класифікація 1 моделей даних.

Моделі даних на основі записів (record-based)

- База даних складається з декількох записів фіксованого формату, які можуть мати різні типи. Кожен тип запису визначає фіксовану кількість полів, кожне з яких має фіксовану довжину.

Загальні типи логічних моделей даних на основі записів:

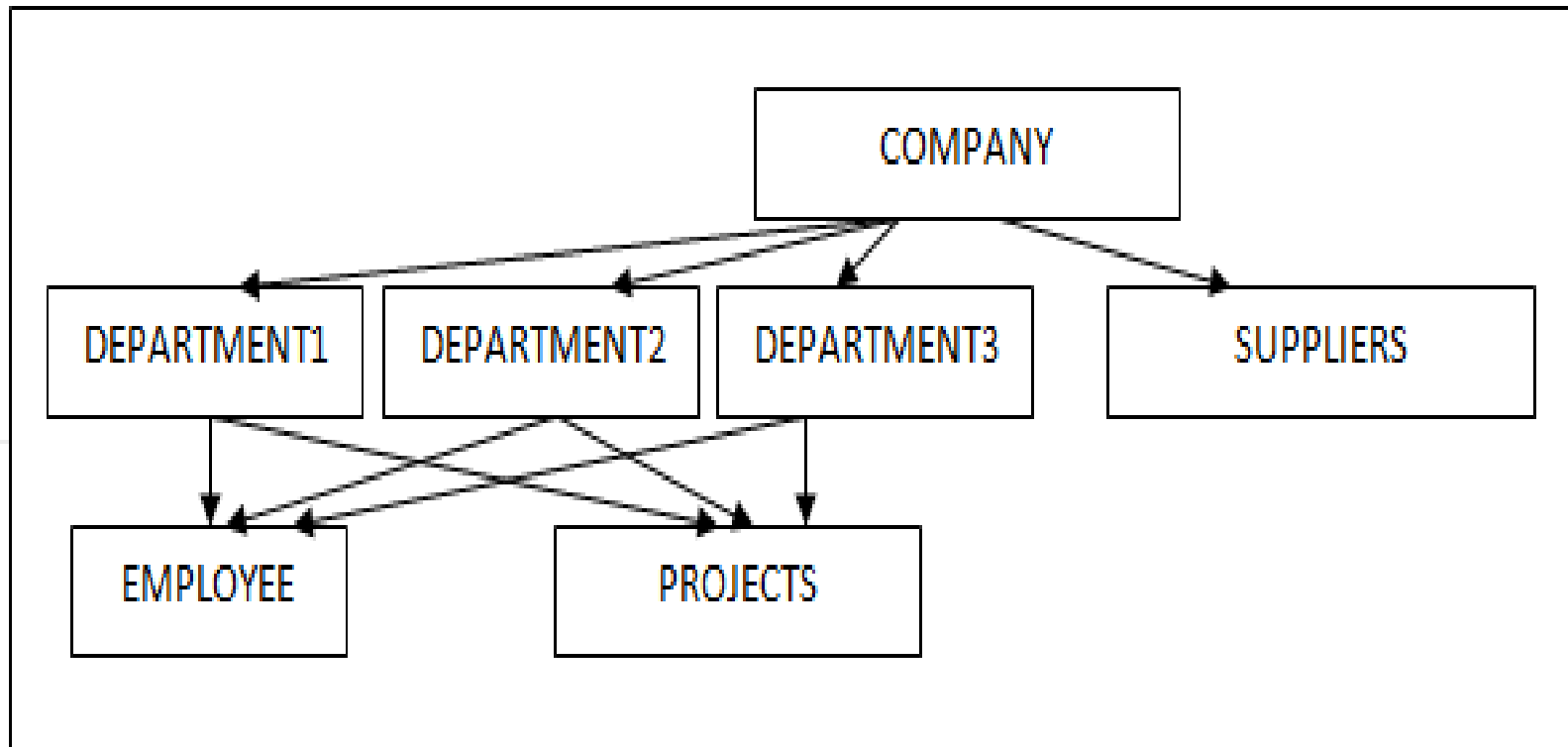
- ☐ реляційна модель даних (relational data model);
 - ☐ мережева модель даних (network data model);
 - ☐ ієрархічна модель даних (hierarchical data model).
-

Класифікація 1 моделей даних.

Моделі даних на основі записів (record-based)

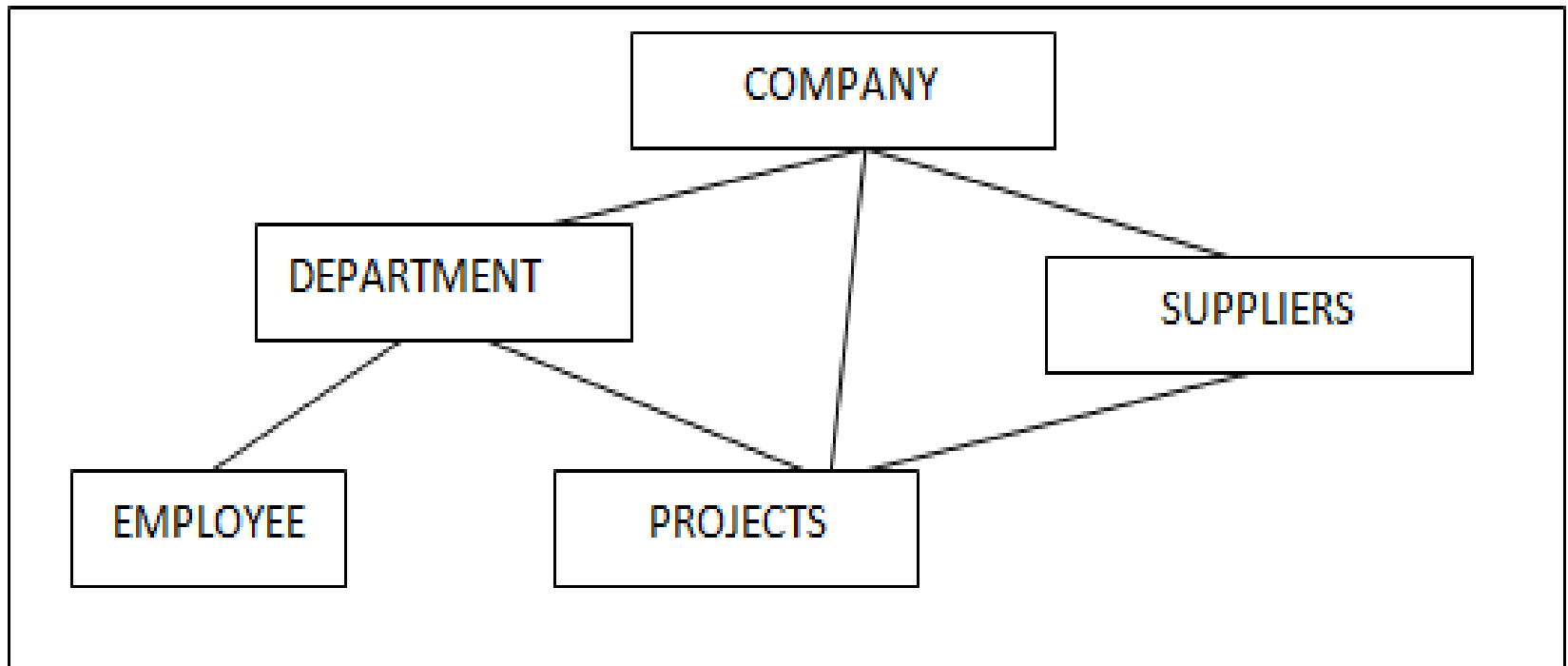
- **Реляційна модель даних** базується на понятті математичних відношень. Єдина вимога в реляційній моделі даних – це щоб база даних з точки зору користувача виглядала як набір таблиць, зв'язаних відношеннями. Однак це відноситься тільки до логічної структури бази даних, тобто до зовнішнього та концептуального рівня архітектури ANSI-SPARC. Дана вимога не відноситься до фізичної структури бази даних, яка може бути реалізована за допомогою різних структур зберігання.
 - В **ієрархічній моделі** дані представляються у вигляді деревовидної (ієрархічної) структури. Подібна організація даних є зручною для роботи з ієрархічно впорядкованою інформацією. Однак, при оперуванні складними логічними зв'язками ієрархічна модель стає дуже громіздкою.
 - В **мережевій моделі** дані організовуються у вигляді довільного графа. На відміну від реляційної моделі, зв'язки тут моделюються наборами, які реалізуються за допомогою вказівників. Недоліком мережевої моделі є жорсткість структури і складність її організації
-

Класифікація 1 моделей даних. Моделі даних на основі записів (record- based). Ієрархічна модель



Класифікація 1 моделей даних.

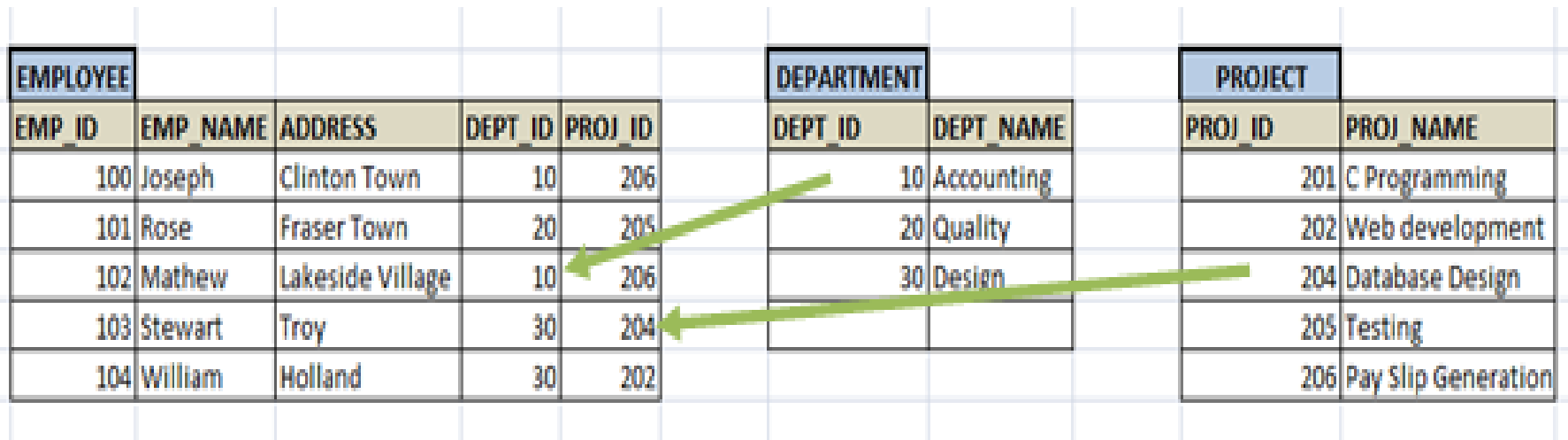
Моделі даних на основі записів (record-based). Мережева модель



Класифікація 1 моделей даних.

Моделі даних на основі записів (record-based). Реляційна модель

| EMPLOYEE | | | | | DEPARTMENT | | PROJECT | |
|----------|----------|------------------|---------|---------|------------|------------|---------|---------------------|
| EMP_ID | EMP_NAME | ADDRESS | DEPT_ID | PROJ_ID | DEPT_ID | DEPT_NAME | PROJ_ID | PROJ_NAME |
| 100 | Joseph | Clinton Town | 10 | 206 | 10 | Accounting | 201 | C Programming |
| 101 | Rose | Fraser Town | 20 | 205 | 20 | Quality | 202 | Web development |
| 102 | Mathew | Lakeside Village | 10 | 206 | 30 | Design | 204 | Database Design |
| 103 | Stewart | Troy | 30 | 204 | | | 205 | Testing |
| 104 | William | Holland | 30 | 202 | | | 206 | Pay Slip Generation |



Класифікація 1 моделей даних.

Фізична модель даних

- оперує категоріями, які відносяться до організації зовнішньої пам'яті та структур зберігання даних
 - використовуються різні методи розміщення даних, що базуються на файлових структурах:
 - організація файлів прямого та послідовного доступу;
 - індексні файли;
 - інвертовані файли;
 - файли, які використовують різні методи хешування;
 - взаємозв'язані файли
 - сторінкова організація даних (сучасні СУБД). Фізичні моделі, що базуються **на сторінковій організації** є найбільш перспективними.
-

Класифікація 1 моделей даних.

Фізична модель даних

- ☐ Sequential File Organization
 - ☐ Heap File Organization
 - ☐ Hash/Direct File Organization
 - ☐ Indexed Sequential Access Method
 - ☐ B+ Tree File Organization
 - ☐ Cluster File Organization
-

Класифікація 2 моделей даних

- інфологічні (семантичні) моделі
 - даталогічні моделі
 - фізичні моделі
-

Класифікація 2 моделей даних.

Інфологічні (семантичні) моделі

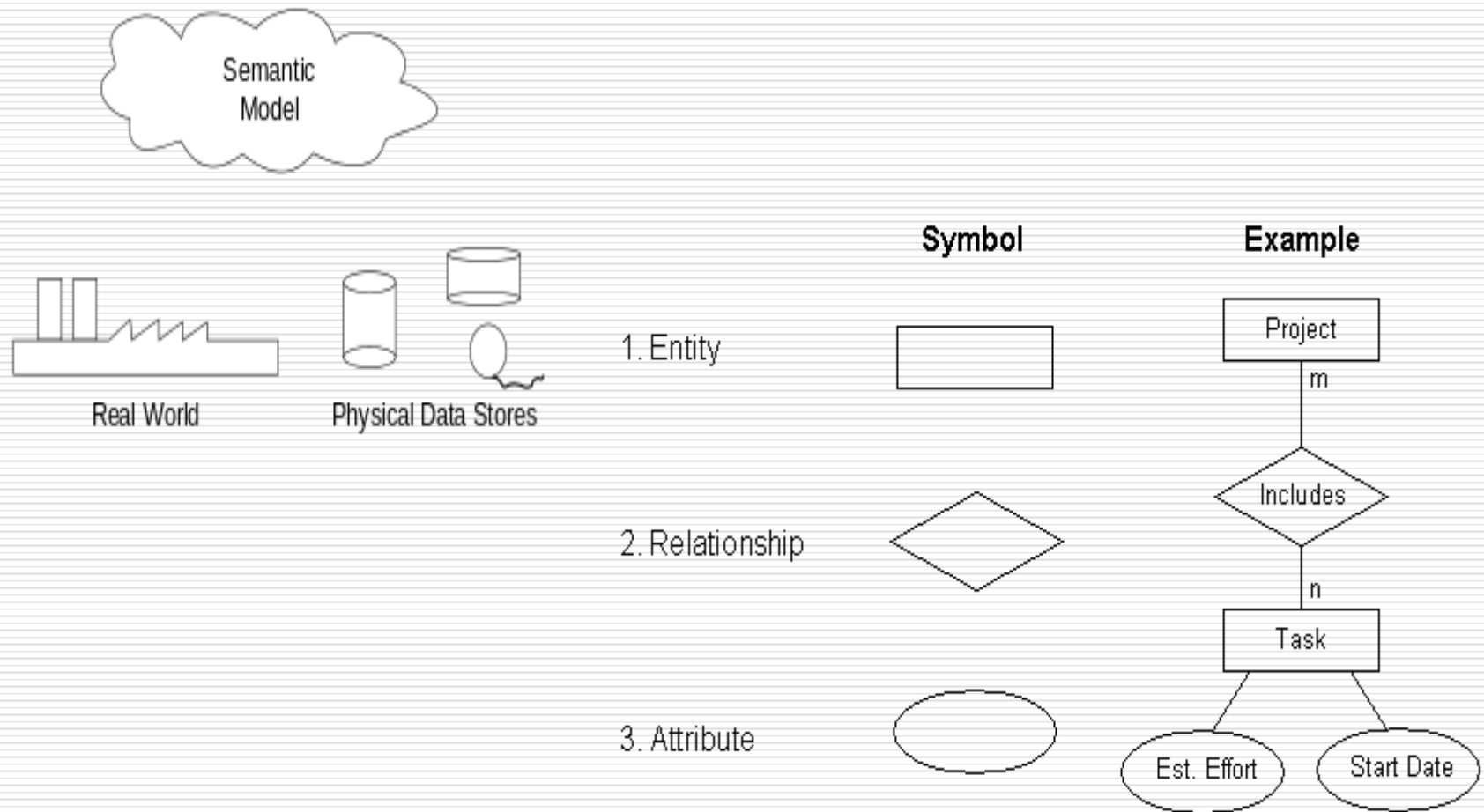
- виражають інформацію про предметну область у вигляді, який не залежить від вибраної СУБД
- відображають інформаційно-логічний рівень абстрагування, який пов'язаний з фіксацією та описом об'єктів ПО, їх властивостей і зв'язків у зручній для розробників та інших користувачів формі
- використовуються на ранніх стадіях проектування БД

Найпоширеніші типи інфологічної моделі:

- діаграми Бахмана
 - модель „сутність-зв'язок” (*ER-модель*)
-

Класифікація 2 моделей даних.

Інфологічні (семантичні) моделі



Класифікація 2 моделей даних. Даталогічні моделі

- підтримуються конкретною СУБД

Типи даталогічних моделей:

- ☐ документальні моделі
 - ☐ фактографічні моделі
-

Класифікація 2 моделей даних. Даталогічні → Документальні моделі

відповідають відображенню
слабоструктурованої інформації,
орієнтованої на вільні формати
документів та тексти на первинній мові

Типи документальних моделей:

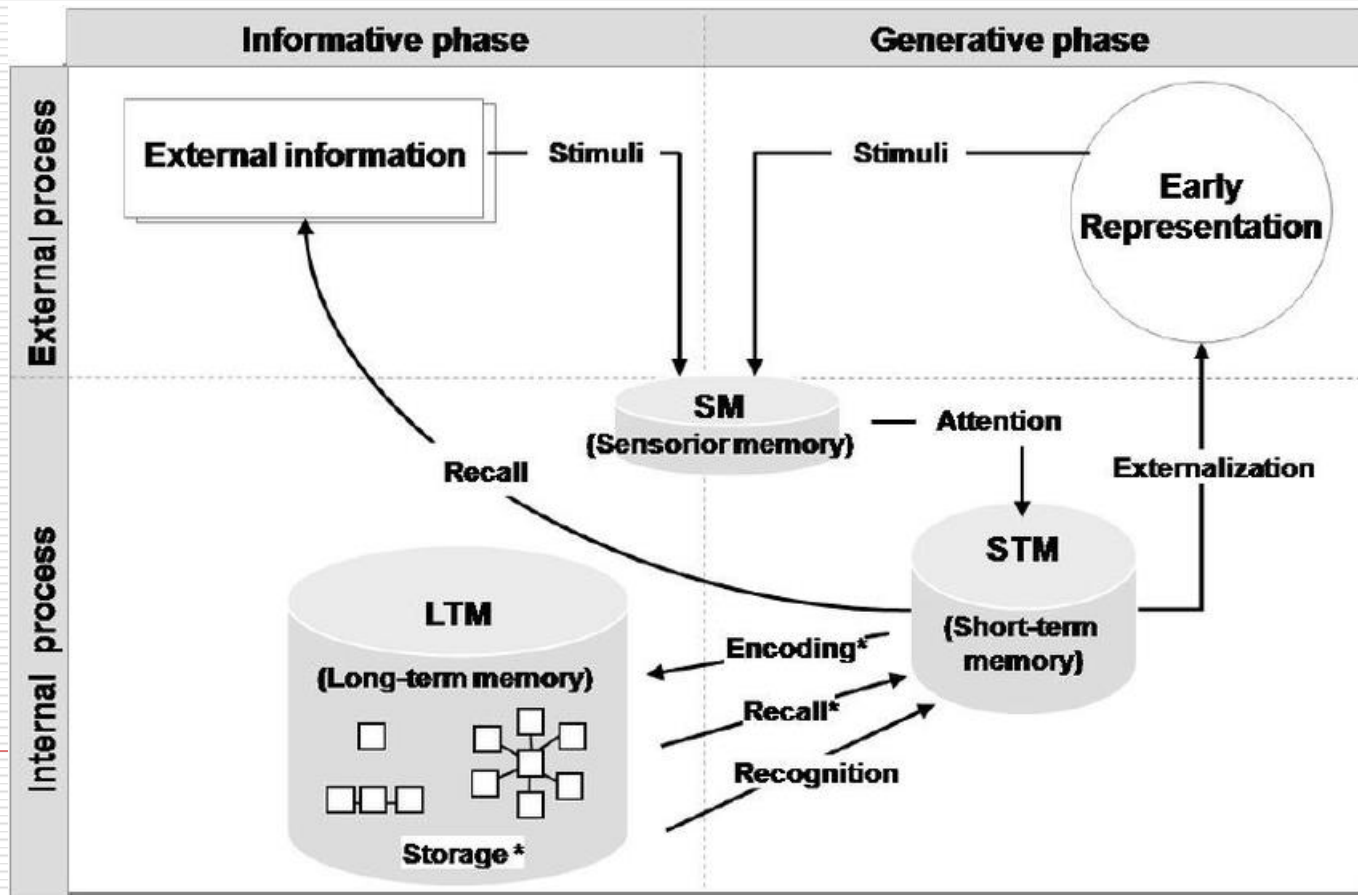
- ☐ *дескрипторні моделі*
 - ☐ *моделі, орієнтовані на формат документу*
 - ☐ *тезаурусні моделі*
-

Класифікація 2 моделей даних. Даталогічні → Документальні → ***Дескрипторні*** моделі

- ❑ Найпростіші з документальних моделей, використовувались на ранніх стадіях використання документальних баз даних.
 - ❑ Кожному документу відповідає дескриптор. Він має жорстку структуру і описує документ у відповідності з тими характеристиками, які потрібні для роботи з документальною БД.
-

Класифікація 2 моделей даних. Даталогічні → Документальні → ***Дескрипторні*** моделі

- ❑ Descriptive model of information processing to generate ideas



Класифікація 2 моделей даних.

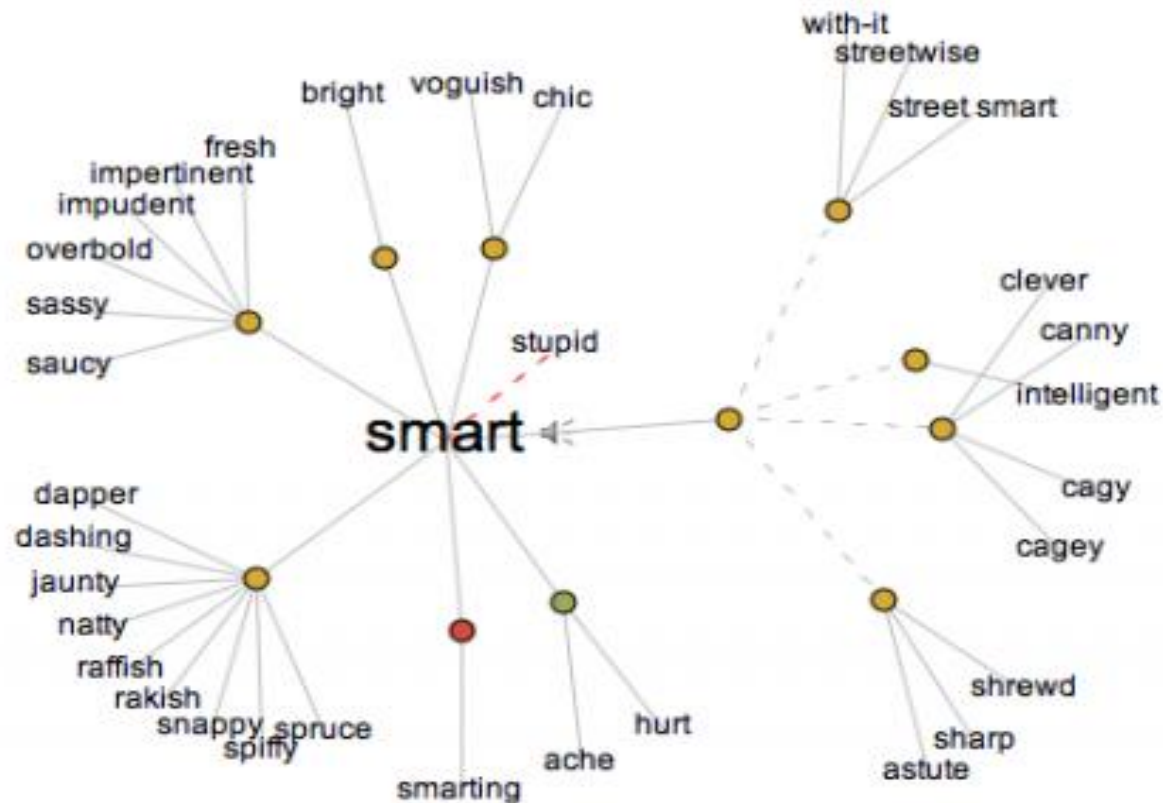
Даталогічні → Документальні → Моделі,
орієнтовані на формат документу

- пов'язані зі стандартними мовами розмітки:
 - HTML: у якості елементу гіпертекстової бази даних, яка описується HTML, використовується текстовий файл, який може легко передаватись по мережі з використанням протоколу HTTP.
 - XML описує цілий клас об'єктів даних, що називаються XML-документами. Вони використовуються у якості засобу для контролю за правильністю складання документів.
 - реалізовані за допомогою підходу NoSQL:
 - спеціально призначені для зберігання ієрархічних структур даних (документів), в основі яких лежать документні сховища (document store), що мають структуру дерева.
 - використовуються у системах керування вмістом, видавничій справі, документальному пошуку
-

Класифікація 2 моделей даних.
Даталогічні → Документальні →
Тезаурусні моделі

- ❑ основані на принципі організації словників
 - ❑ містять певні мовні конструкції і принципи їх взаємодії у заданій граматиці
 - ❑ ефективно використовуються в системах-перекладачах, в яких за тезаурусними моделями реалізовано принцип зберігання інформації
-

Класифікація 2 моделей даних. Даталогічні → Документальні → ***Тезаурусні*** моделі



Класифікація 2 моделей даних. Даталогічні → Фактографічні моделі

- графові моделі
 - теоретико-множинні моделі
 - об'єктно-орієнтована модель
-

Класифікація 2 моделей даних.

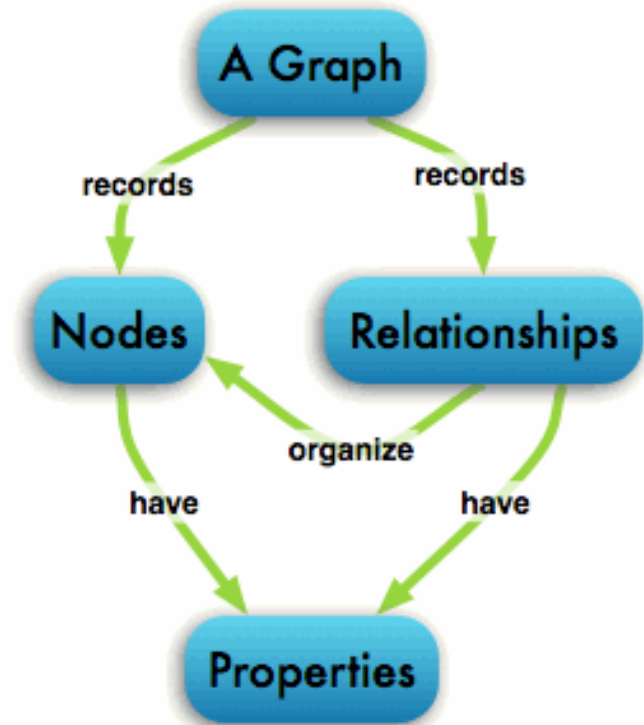
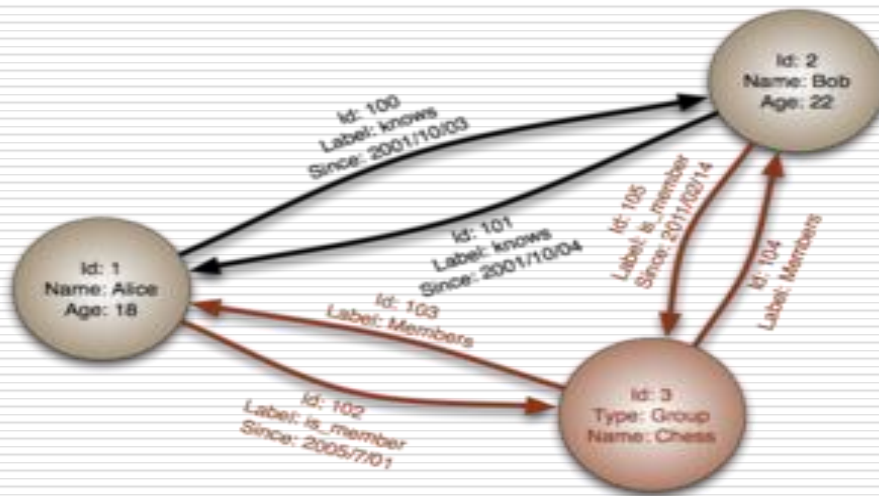
Даталогічні → Фактографічні → **Графові**
моделі

- відображають сукупність об'єктів реального світу у вигляді графа взаємозв'язаних інформаційних об'єктів.

В залежності від типу графа виділяють:

- ☐ ієрархічну модель
 - ☐ мережеву модель
-

Класифікація 2 моделей даних. Даталогічні → Фактографічні → **Графові** моделі



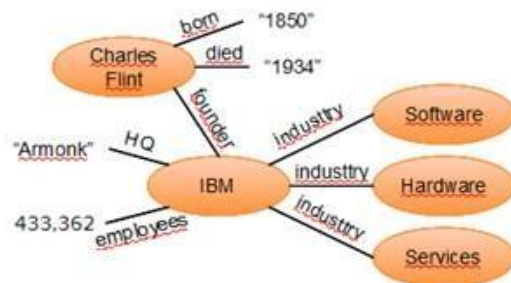
Класифікація 2 моделей даних. Даталогічні → Фактографічні → **Графові** моделі

Types of Graph

Graph Database

RDF / Property Graph

Attributes



| subject | predicate | object |
|---------------|-----------|----------|
| Charles Flint | born | "1850" |
| Charles Flint | died | "1934" |
| Charles Flint | founder | IBM |
| IBM | HQ | "Armonk" |
| IBM | employees | 433,362 |
| IBM | industry | Software |
| IBM | industry | Hardware |
| IBM | industry | Services |

Contextual Analysis

Topological Analytics

Collective Graph

Macro

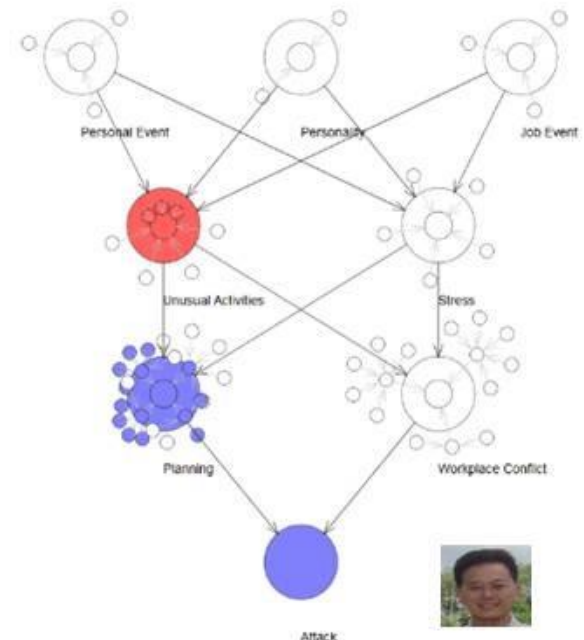


Collective Analysis

Graphical Models

Activity Graph

Micro & Reasoning



Cognitive Understanding

Класифікація 2 моделей даних. Даталогічні → Фактографічні → ***Теоретико-множинні*** моделі

- перехід від роботи з елементами даних (графові моделі) до роботи з макрооб'єктами

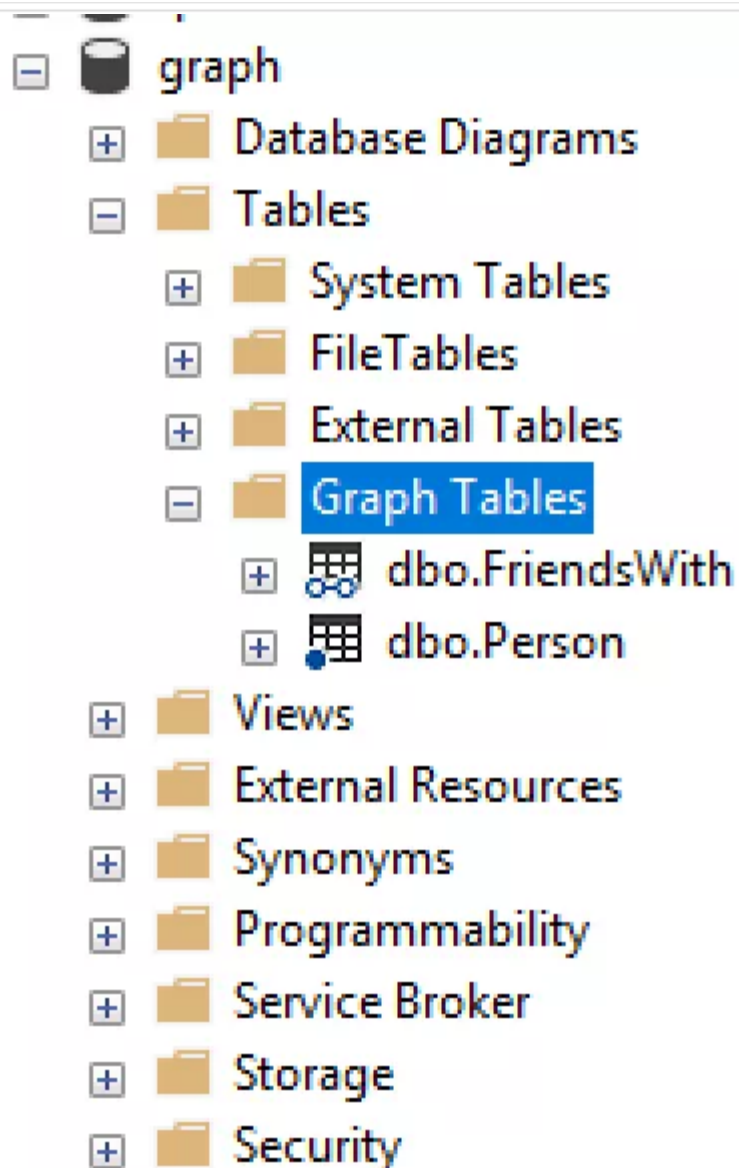
Основні моделі:

- ☐ реляційна модель
 - ☐ модель бінарних асоціацій
-

Класифікація 2 моделей даних.
Даталогічні → Фактографічні →
Об'єктно-орієнтована модель

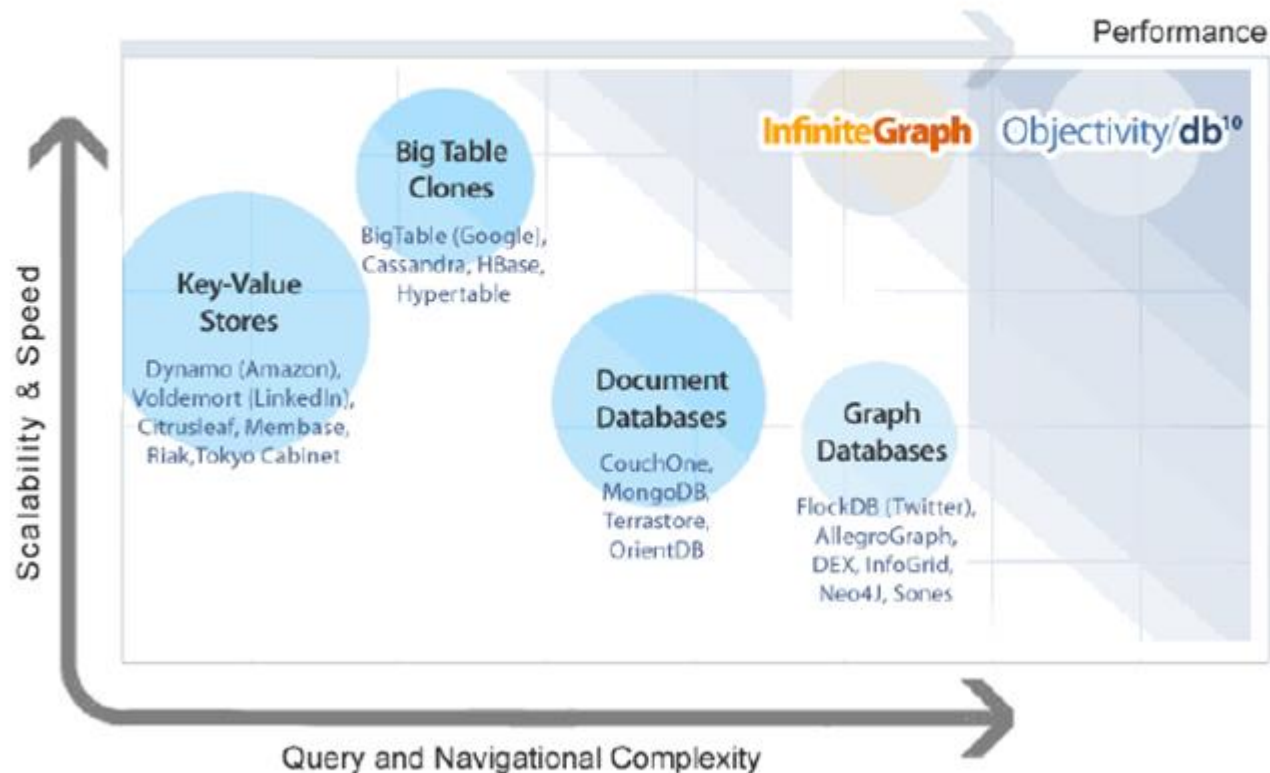
Розширяє визначення сутності з метою включення в нього не тільки атрибутів, які описують стан об'єкта, а й дій, які з ним пов'язані, тобто його поведінку. В такому випадку говорять, що об'єкт інкапсулює стан та поведінку .

Поеднання моделей даних.



Поеднания моделей даних.

The "NOSQL" Technology Landscape



Поеднання моделей даних.



Path exists in social network

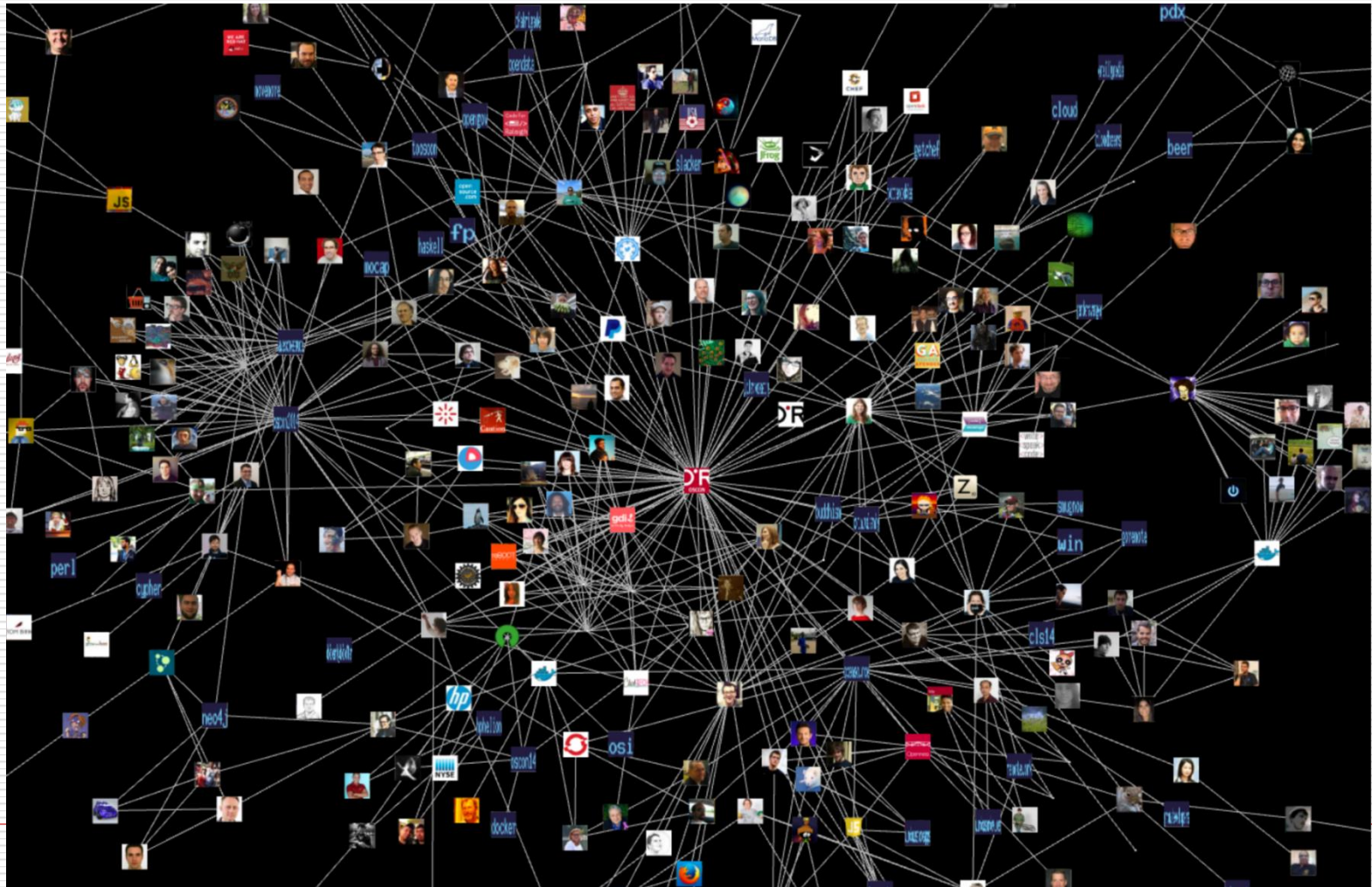
- Each person has on average 50 friends



The performance impact in Neo4j depends only on the degree of each node. In an RDBMS it depends on the number of entries in the tables involved in the join(s).

| Database | # persons | query time |
|----------------------|-----------|-----------------|
| Relational database | 1 000 | 2 000 ms |
| Neo4j Graph Database | 1 000 | 2 ms |
| Neo4j Graph Database | 1 000 000 | 2 ms |
| Relational database | 1 000 000 | way too long... |

Так виглядає модель Twitter



Дякую за увагу
