

# Вступ до організації баз даних

# Introduction to databases

# Введение в организацию баз данных

Тема 1

Торіс 1

## 1.1 Бази даних та СУБД

## 1.1 Databases and DBMS

## 1.1 Базы данных и СУБД

- Поява у 1960-х запам'ятовуючих пристроїв відносно великої ємності відкрило широкі можливості для створення складних структур довгострокового зберігання даних.
- Appearance in 1960-s memory devices of relatively big capacity allowed wide opportunities for development of complex structures of long-term data storage.
- Появление в 1960-х запоминающих устройств относительно высокой емкости открыло широкие возможности для создания сложных структур долговременного хранения данных.

## 1.1 Бази даних та СУБД

## 1.1 Databases and DBMS

## 1.1 Базы данных и СУБД

- Ці можливості призвели до значного ускладнення коду програм, подорожчання їх розробки та зниженню надійності. З'явилась ідея централізації функцій управління даними та створення систем, що надають програмам послуги з обробки даних – систем управління базами даних (СУБД).
- These possibilities led to increase in complexity of application code, development cost, and decrease of reliability. Appeared an idea of centralization of data management functions and creation of systems that provide data processing services to applications – database management systems (DBMS).
- Эти возможности привели к значительному усложнению кода программ, подорожанию их разработки и снижению надежности. Появилась идея централизации функций управления данными и создания систем, которые предоставляют приложениям услуги обработки данных – систем управления базами данных (СУБД).

## 1.1 Бази даних та СУБД

## 1.1 Databases and DBMS

## 1.1 Базы данных и СУБД

- СУБД забезпечують високу надійність зберігання та ефективність обробки даних, недосяжні при розробці індивідуальних засобів управління даними для кожної програми.
- DBMS provide high reliability of data storage and high efficiency of data processing, that are not possible with development of individual data management tools for each application.
- СУБД обеспечивают высокую надежность хранения и эффективность обработки данных, недостижимые при разработке индивидуальных средств управления данными для каждого приложения.

## 1.1 Бази даних та СУБД

## 1.1 Databases and DBMS

## 1.1 Базы данных и СУБД

- Особливості ранніх СУБД були обумовлені вимогами програм оперативної обробки даних (online transaction processing, OLTP) у банківський та фінансовій сферах.
- Features of early DBMS were defined by requirements of online transaction processing (OLTP) applications in banking and financial industries.
- Особенности ранних СУБД были обусловлены требованиями приложений оперативной обработки данных (online transaction processing, OLTP) в банковской и финансовой сферах.

## 1.1 Бази даних та СУБД

## 1.1 Databases and DBMS

## 1.1 Базы данных и СУБД

- СУБД – програмний комплекс, що забезпечує централізоване зберігання даних та надає програмам послуги з обробки даних.
- DBMS – software system that ensures centralized data storage and provides data processing services to applications.
- СУБД – программный комплекс, обеспечивающий централизованное хранение данных и предоставляющий приложениям услуги по обработке данных.

## 1.1 Бази даних та СУБД

## 1.1 Databases and DBMS

## 1.1 Базы данных и СУБД

- База даних (БД) – сукупність даних, що зберігаються під управлінням СУБД. БД – фундамент, на якому будуються майже будь-які програми. СУБД, пов'язана з деякою конкретною БД та готова виконувати запити на обробку цієї БД, називається екземпляром (instance) або сервером БД.
- Database (DB) – dataset stored under DBMS management. DB – baseline of almost any applications. DBMS, which is related to some specific DB and ready to run DB processing queries, is called instance or DB server.
- База данных (БД) – совокупность данных, которые хранятся под управлением СУБД. БД – фундамент, на котором строятся практически любые приложения. СУБД, связанная с некоторой конкретной БД и готовая выполнять запросы на обработку этой БД, называется экземпляром (instance) или сервером БД.

## 1.2 Вимоги до СУБД

## 1.2 Requirements to DBMS

## 1.2 Требования к СУБД

- Ранні СУБД дуже сильно відрізнялися своєю внутрішньою організацією та можливостями. Знадобилось декілька років, для того щоб визначити основні функції СУБД та вимоги, які слід ставити до таких систем.
- Early DBMS were very different by their internal organization and features. It took several years in order to define main DBMS functions and requirement for such systems.
- Ранние СУБД очень сильно отличались своей внутренней организацией и возможностями. Понадобилось несколько лет, чтобы определить основные функции СУБД и требования, которые следует предъявлять к таким системам.



## 1.2 Вимоги до СУБД

## 1.2 Requirements to DBMS

## 1.2 Требования к СУБД

- Системи OLTP характеризуються тим, що:
  - Кожна операція займає дуже мало часу (мілісекунди)
  - Дані спільно використовуються багатьма програмами
  - Операції використовують незначну долю загального обсягу даних
- OLTP systems are characterized by the following features:
  - Each operation takes very small (milliseconds)
  - Data used together by multiple applications
  - Operations are using insignificant piece of total amount of data
- Системы OLTP характеризуются тем, что:
  - Каждая операция занимает очень мало времени (миллисекунды)
  - Данные совместно используются несколькими приложениями
  - Операции используют незначительную долю общего объема данных

## 1.2 Вимоги до СУБД

## 1.2 Requirements to DBMS

## 1.2 Требования к СУБД

- Процеси обробки та структури даних в областях, де використовувались ранні СУБД, були формалізовані задовго до появи обчислювальних систем (наприклад, бухгалтерський облік). Це привело до того, що СУБД, як правило, орієнтовані на обробку структурованих даних.
- Data processing workflows and data structures, where early DBMS were used, were formalized long before occurrence of computing systems (e.g., accounting). It led to the fact that DBMS are mostly oriented on structured data processing.
- Процессы обработки и структуры данных в областях, где использовались ранние СУБД, были формализованы задолго до появления вычислительных систем (например, бухгалтерский учет). Это привело к тому, что СУБД, как правило, ориентированы на обработку структурированных данных.

## 1.2 Вимоги до СУБД

## 1.2 Requirements to DBMS

## 1.2 Требования к СУБД

Основні вимоги до СУБД:

- Розділення програм і даних. Опис структури даних відділений від коду програм. Система повинна допускати незалежну зміну структури даних та коду програми.
- Високорівнева мова запитів. Система повинна надавати засоби обробки даних, незалежні від якоїсь програми.
- Цілісність. Система повинна запобігати запису даних, що порушують заздалегідь визначені обмеження.
- Узгодженість. Система повинна запобігати пошкодженню даних внаслідок паралельної роботи декількох програм.
- Відмовостійкість. СУБД не повинна допускати втрати даних.
- Захист та розмежування доступу. Система повинна запобігати несанкціонованому доступу до даних та надавати кожному користувачу доступ до даних згідно з його правами.

## 1.2 Вимоги до СУБД

## 1.2 Requirements to DBMS

## 1.2 Требования к СУБД

### General requirements to DBMS:

- Independence of applications and data. Description of data structure is separated from applications code. System should allow independent change of data structure and applications code.
- High-level query language. System should provide data processing tools, independent from any application.
- Integrity. System should not allow data insertion that violate previously defined constraints.
- Consistency. System should not allow data damage as the result of concurrent work of multiple applications.
- Fault tolerance. DBMS should not allow data loss.
- Security and access control. System should not allow unauthorized access to data and should allow access to data to each user according to its privileges.

## 1.2 Вимоги до СУБД

## 1.2 Requirements to DBMS

## 1.2 Требования к СУБД

Основные требования к СУБД:

- Разделение приложений и данных. Описание структуры данных отделено от кода программ. Система должна допускать независимое изменение структуры данных и кода приложения.
- Высокоуровневый язык запросов. Система должна предоставлять средства обработки данных, независимые от любых приложений.
- Целостность. Система должна предотвращать запись данных, которые нарушают заранее определенные ограничения.
- Согласованность. Система должна предотвращать нарушения данных вследствие параллельной работы нескольких приложений.
- Отказоустойчивость. СУБД не должна допускать потери данных.
- Защита и разграничение доступа. Система должна предотвращать несанкционированный доступ к данным и предоставлять каждому пользователю доступ к данным согласно его правам.

## 1.2 Вимоги до СУБД

## 1.2 Requirements to DBMS

## 1.2 Требования к СУБД

- У ранні роки існування СУБД передбачалось, що дані інформаційних систем підприємства будуть зберігатися у єдиній БД. На практиці це ніколи не було реалізовано: зазвичай для кожної програми або групи програм створюється окрема БД.
- In early years of DBMS was assumed that data of information systems of organization will be stored in a single DB. In practice this was never implemented: usually for each application or group of applications a separate DB is created.
- В ранние годы существования СУБД предполагалось, что данные информационных систем предприятия будут храниться в единой БД. На практике это никогда не было реализовано: как правило для каждого приложения или группы приложений создается отдельная БД.

## 1.3 Розділення даних та програм

## 1.3 Independence of data and applications

## 1.3 Разделение данных и приложений

- Кожна СУБД створює деякий рівень абстракції для інших програмних компонент, які використовують її послуги. Необхідно щоб опис структури даних був загальний для усіх програм, що призводить до ідеї відділення опису структури даних від програм. Такий опис зберігається у самій БД і називається схемою бази даних.
- Each DBMS creates some level of abstraction for other software components that use its services. It is required to data structure description is general for all applications, which leads to the idea of separation of data structure from applications. Such description is stored in DB itself and it is called database scheme.
- Каждая СУБД создает некоторый уровень абстракции для других программных компонент, которые используют ее услуги. Необходимо чтобы описание структуры данных было общим для всех приложений, что приводит к идее отделения описания структуры данных от приложений. Такое описание хранится в самой БД и называется схемой базы данных.

## 1.3 Розділення даних та програм

## 1.3 Independence of data and applications

## 1.3 Разделение данных и приложений

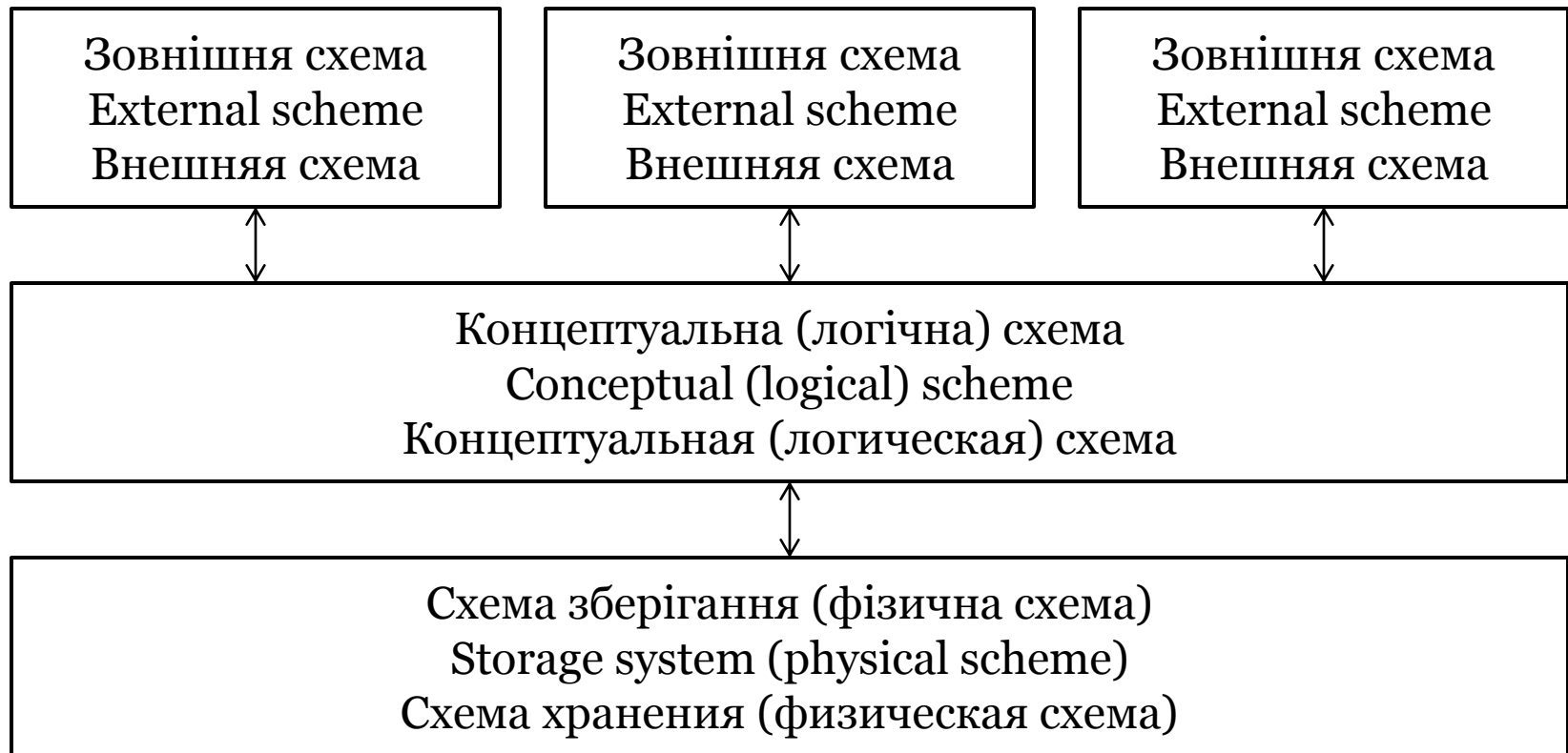
- Для визначення схем використовуються мови опису даних. Для реалізації ідеї відокремлення даних від програм було запропоновано модель мови опису даних ANSI/SPARC.
- To define schemes data description languages are used. In order to implement the idea of independence of data and applications the data description language models ANSI/SPARC was proposed.
- Для определения схем используются языки описания данных. Для реализации идеи отделения данных от приложений была предложена модель языка описания данных ANSI/SPARC.



## 1.3 Розділення даних та програм

## 1.3 Independence of data and applications

## 1.3 Разделение данных и приложений



ANSI/SPARC

## 1.3 Розділення даних та програм

## 1.3 Independence of data and applications

## 1.3 Разделение данных и приложений

Модель ANSI/SPARC включає:

- Зовнішню схему, що містить опис даних у вигляді, у якому вони будуть використовуватися програмами, а також відображення логічної структури даних у зовнішню схему.
- Концептуальну схему, що містить повний опис логічної структури даних, доступний для СУБД (логічна схема БД).
- Схему зберігання, що описує яким чином організоване зберігання логічних структур даних (фізична схема БД).

## 1.3 Розділення даних та програм

## 1.3 Independence of data and applications

## 1.3 Разделение данных и приложений

ANSI/SPARC model includes:

- External scheme, which contains data description in the form in which they will be used by applications, as well as the mapping of logical data structure into external scheme.
- Conceptual scheme, which contains full description of logical data structure, accessible by DBMS (logical DB scheme).
- Storage scheme, which describes how storage of logical data structures is organized (physical DB scheme).

## 1.3 Розділення даних та програм

## 1.3 Independence of data and applications

## 1.3 Разделение данных и приложений

Модель ANSI/SPARC включает:

- Внешнюю схему, содержащую описание данных в виде, в котором они будут использоваться приложениями, а также отображение логической структуры данных во внешнюю схему.
- Концептуальную схему, содержащую полное описание логической структуры данных, доступное для СУБД (логическая схема БД).
- Схему хранения, описывающую каким образом организовано хранение логических структур данных (физическая схема БД).

## 1.3 Розділення даних та програм

## 1.3 Independence of data and applications

## 1.3 Разделение данных и приложений

- В ідеалі така модель забезпечує еволюцію програм та системи. При появі нових програм достатньо визначити нову зовнішню схему. В результаті впровадження нової програми не вплине на роботу інших програм. Нові структури, додані до концептуальної схеми, не вплинуть на роботу інших програм, оскільки їх зовнішні схеми не будуть містити нових елементів даних.
- Ideally such model provides evolution of applications and a system. When new programs appeared it is sufficient to define new external scheme. As the result, implementation of new application will not affect other applications. New structures, added to the conceptual scheme, will not affect other applications, since their external schemes do not contain new data elements.
- В идеале такая модель обеспечивает эволюцию приложений и системы. При появлении новых приложений достаточно определить новую внешнюю схему. В результате внедрение нового приложения не повлияет на работу других приложений. Новые структуры, добавленные в концептуальную схему, не повлияют на работу других приложений, поскольку их внешние схемы не будут содержать новые элементы данных.

## 1.3 Розділення даних та програм

## 1.3 Independence of data and applications

## 1.3 Разделение данных и приложений

- У реальності еволюційні можливості можуть бути реалізовані лише у випадку, коли розробники БД та програм їх ретельно враховують. Еволюція БД має сенс лише коли цінність накопичених даних є високою. У інших випадках створення нової БД є більш виправданим рішенням.
- In reality evolution capabilities could be implemented only in case if DB and applications developers carefully follow these capabilities. DB evolution is reasonable only in case of stored data is highly valuable. In other cases development of new DB is more reasonable solution.
- В реальности эволюционные возможности могут быть реализованы только в случае, когда разработчики БД и приложений их тщательно учитывают. Эволюция БД имеет смысл только когда ценность накопленных данных является высокой. В других случаях создание новой БД является более оправданным решением.

## 1.3 Розділення даних та програм

## 1.3 Independence of data and applications

## 1.3 Разделение данных и приложений

- У останні роки багато методологій розробки програм передбачають генерацію схеми БД на основі об'єктної моделі програми для швидкої розробки прототипів. При цьому не зникає складність та необхідність вирішувати задачі проектування БД.
- Recently many software development methodologies assume generation of DB scheme based on application object model for rapid prototypes developments. At the same time complexity is not reduced and it is still required to solve DB design problems.
- В последние годы много методологий разработки приложений предусматривают генерацию схемы БД на основе объектной модели приложения для быстрой разработки прототипов. При этом не исчезает сложность и необходимость решать задачи проектирования БД.

## 1.3 Розділення даних та програм

## 1.3 Independence of data and applications

## 1.3 Разделение данных и приложений

- Особливості проектування схеми БД суттєво залежать від моделі даних, що застосовується. Складність проектування може по-різному розподілятися між БД та програмою: чим бідніше модель даних, тим більше необхідно зробити на рівні програми.
- Features of DB scheme design depend on data model, which is used. Design complexity is differently distributed between DB and application: if the data model is poor, the more work need to be done on the application layer.
- Особенности проектирования схемы БД существенно зависят от применяемой модели данных. Сложность проектирования может по-разному распределяться между БД и приложением: чем беднее модель данных, тем больше необходимо сделать на уровне приложения.



## 1.4 Мови запитів

## 1.4 Query languages

## 1.4 Языки запросов

- Наявність опису логічної структури даних дозволяє виконувати достатньо складні операції маніпулювання даними у СУБД. Такі операції записуються на мові запитів. Мови сучасних СУБД є декларативними – дозволяють визначити необхідний результат, але не спосіб виконання запиту. СУБД обирає найбільш ефективні (за деяким критерієм) алгоритми отримання результату.
- Existence of logical data structure allows to execute complex enough operations of data manipulation in DBMS. These operations are written in a query language. Languages of modern DBMS are declarative – allow to define required result, but not the way to execute query. DBMS chooses the most effective (by some criteria) algorithms to obtain the result.
- Наличие описания логической структуры данных позволяет выполнять достаточно сложные операции манипулирования данными в СУБД. Такие операции записываются на языке запросов. Языки современных СУБД являются декларативными – позволяют определить необходимый результат, но не способ выполнения запроса. СУБД выбирает наиболее эффективные (по некоторому критерию) алгоритмы получения результата.

## 1.4 Мови запитів

## 1.4 Query languages

## 1.4 Языки запросов

- Практика розробки програм без використання мов запитів призвела до появи так званих NoSQL систем. При використанні таких систем частина функцій СУБД переноситься у програму, що призводить до підвищення складності програми та вартості її розробки або до зниження якості, що у деяких випадках є припустимим.
- Practice of application development without using query languages led to occurrence of so-called NoSQL systems. When using these systems some DBMS functions are transferred to application, which led to increasing of applications complexity and development cost, or to quality decreasing, which is acceptable in some cases.
- Практика разработки приложений без использования языков запросов привела к появлению так называемых NoSQL систем. При использовании таких систем часть функций СУБД переносится в приложение, что приводит к повышению сложности приложения и стоимости его разработки или к снижению качества, что в некоторых случаях является допустимым.

## 1.5 Цілісність та узгодженість

## 1.5 Integrity and consistency

## 1.5 Целостность и согласованность

- Окрім опису структур даних до логічної схеми БД можуть бути включені додаткові умови – обмеження цілісності. СУБД перевіряє обмеження цілісності при виконанні будь-яких змін даних та не допускає порушень цих обмежень. Це дозволяє значно спростити розробку програм та підвищити їх якість.
- Besides data structured descriptions, DB scheme can contain additional conditions – integrity constraints. DBMS checks integrity constraints when executing any data changes and does not allow violations of these constraints. This allows significantly simplify applications development and increase their quality.
- Кроме описания структур данных в логическую схему БД могут быть включены дополнительные условия – ограничения целостности. СУБД проверяет ограничения целостности при выполнении любых изменений данных и не допускает нарушений этих ограничений. Это позволяет значительно упростить разработку приложений и повысить их качество.

## 1.5 Цілісність та узгодженість

## 1.5 Integrity and consistency

## 1.5 Целостность и согласованность

- Стан БД, у якому виконуються умови предметної області, називається узгодженим. Кінцевий набір операцій, що переводить БД з одного узгодженого стану до іншого, називається транзакцією. Однією з важливих функцій СУБД є запобігання порушенням узгодженості при одночасній роботі декількох програм (або користувачів).
- DB state in which domain conditions are satisfied is called consistent state. Final set of operations that transfer DB from one to another consistent state is called transaction. One of the main DBMS functions is prevention of consistency violations when concurrent applications (or users) are working.
- Состояние БД, в котором выполняются условия предметной области, называется согласованным. Конечный набор операций, переводящий БД из одного согласованного состояния в другое, называется транзакцией. Одной из наиболее важных функций СУБД является предотвращение нарушения согласованности при одновременной работе нескольких приложений (или пользователей).

## 1.6 Відмовостійкість

## 1.6 Fault tolerance

## 1.6 Отказоустойчивость

- Відмови систем призводять до зупинки бізнес-процесів, а втрати даних призводять до катастрофічних подій (не тільки для функцій підприємства, а і для життя людей та стану навколишнього середовища). Сучасні СУБД можуть гарантувати повну збереженість даних та відновлення після відмови в узгодженому стані за доли секунди.
- System faults lead to stopping of business processes, while data losses lead to catastrophic events (not only for enterprise functions, but also for human lives and environment condition). Modern DBMS can guarantee full safety of data and recovery in consistent state in milliseconds after faults.
- Отказы систем приводят к остановке бизнес-процессов, а потери данных приводят к катастрофическим событиям (не только для функций предприятия, а и для жизни людей и состояния окружающей среды). Современные СУБД могут гарантировать полную сохранность данных и восстановление после отказа в согласованном состоянии за доли секунды.

## 1.6 Відмовостійкість

## 1.6 Fault tolerance

## 1.6 Отказоустойчивость

- Висока вартість пов'язана з необхідністю багаторазового дублювання засобів на усіх рівнях, починаючи з обладнання. При використанні хмарних сервісів необхідно мати додаткові ресурси, здатні забезпечити виживання при відмові. Тому при проектуванні системи необхідно обирати рівень відмовостійкості, що є дійсно необхідним.
- High cost is related to the requirement of presence of multiple duplicates on all layers, starting from hardware. When using cloud services, it is required to have additional resources, that could provide survival in case of faults. Therefore, when designing system it is necessary to choose really required level of fault tolerance.
- Высокая стоимость связана с необходимостью многократного дублирования средств на всех уровнях, начиная с оборудования. При использовании облачных сервисов необходимо иметь дополнительные ресурсы, способные обеспечить выживание при отказе. Поэтому при проектировании системы необходимо выбирать действительно необходимый уровень отказоустойчивости.

## 1.7 Безпека та розмеження доступу

## 1.7 Security and access control

## 1.7 Безопасность и разграничение доступа

- Дані потребують захисту не тільки від відмов, а й від несанкціонованого доступу. Усі сучасні СУБД надають засоби перешкоджання та розмеження доступу. В деяких системах засоби захисту СУБД взагалі не використовуються, але по-справжньому надійний захист повинен бути багаторівневим, а деякі види захисту неможливо реалізувати без СУБД.
- Data should be protected not only from faults, but also from unauthorized access. All modern DBMS provide access prevention and control tools. In some systems security features of DBMS are not used at all, but really reliable protection should be multi-layered, while some protection types are not possible without DBMS.
- Данные требуют защиты не только от отказов, а и от несанкционированного доступа. Все современные СУБД предоставляют средства предотвращения и разграничения доступа. В некоторых системах средства защиты СУБД вообще не используются, но по-настоящему надежная защита должна быть многоуровневой, а некоторые виды защиты невозможно реализовать без СУБД.

## 1.8 Продуктивність

## 1.8 Performance

## 1.8 Производительность

- Найбільш важливими метриками є пропускна здатність та час відгуку системи. Для СУБД це запити або інші дії різної складності. Пропускна здатність – середня кількість подібних дій за одиницю часу. Час відгуку – середній час виконання дії. Покращення однієї з характеристик не обов'язково призведе до покращення іншої.
- The most important metrics are throughput and response time. For DBMS there are queries or other actions of different complexity. Throughput is an average number of similar actions at a time. Response time is an average time of execution. Improvement of one of these metrics will not necessary lead to improvement of another.
- Наиболее важными метриками являются пропускная способность и время отклика системы. Для СУБД это запросы или другие действия разной сложности. Пропускная способность – среднее количество подобных действий за единицу времени. Время отклика – среднее время выполнения действия. Улучшение одной характеристики не обязательно приведет к ухудшению другой характеристики.



## 1.8 Продуктивність

## 1.8 Performance

## 1.8 Производительность

- Важливо розрізняти вимір на стороні клієнта або сервера. Час виконання на сервері може бути меншим за час відправлення запиту та повернення відповіді. Тому для веб-застосунків найбільш важливим є час генерації HTML сторінки у відповідь на запит користувача.
- It is important to differentiate between the measure on client or server side. Execution time on server could be less than time required to send request and receive response. Then for web-applications generation time of HTML pages in response to users request is more important.
- Важно различать измерение на стороне клиента или сервера. Время выполнения на сервере может быть меньше времени отправки запроса и возврата ответа. Поэтому, для веб-приложений наиболее важным является время генерации HTML страницы в ответ на запрос пользователя.

## 1.8 Продуктивність

## 1.8 Performance

## 1.8 Производительность

- Для паралельних систем найбільш важливою характеристикою є масштабованість (пропускної здатності або часу відгуку). Необхідно оцінити характеристику з певним обсягом даних, навантаженням та одним обчислювачем, а також ту ж саму характеристику з обсягом даних, навантаженням та обчислювачами, збільшеними у  $N$  разів.
- For concurrent systems the most important metric is scalability (of throughput or response time). It is required to measure some metric with some data amount, workload, and one computation instance, and then the same metric with data amount, workload, and computation instances increased  $N$  times.
- Для параллельных систем наиболее важной характеристикой является масштабируемость (пропускной способности или времени отклика). Необходимо оценить характеристику с определенным объемом данных, нагрузкой и одним вычислителем, а также ту же самую характеристику с объемом данных, нагрузкой и вычислителями, увеличенными в  $N$  раз.

## 1.8 Продуктивність

## 1.8 Performance

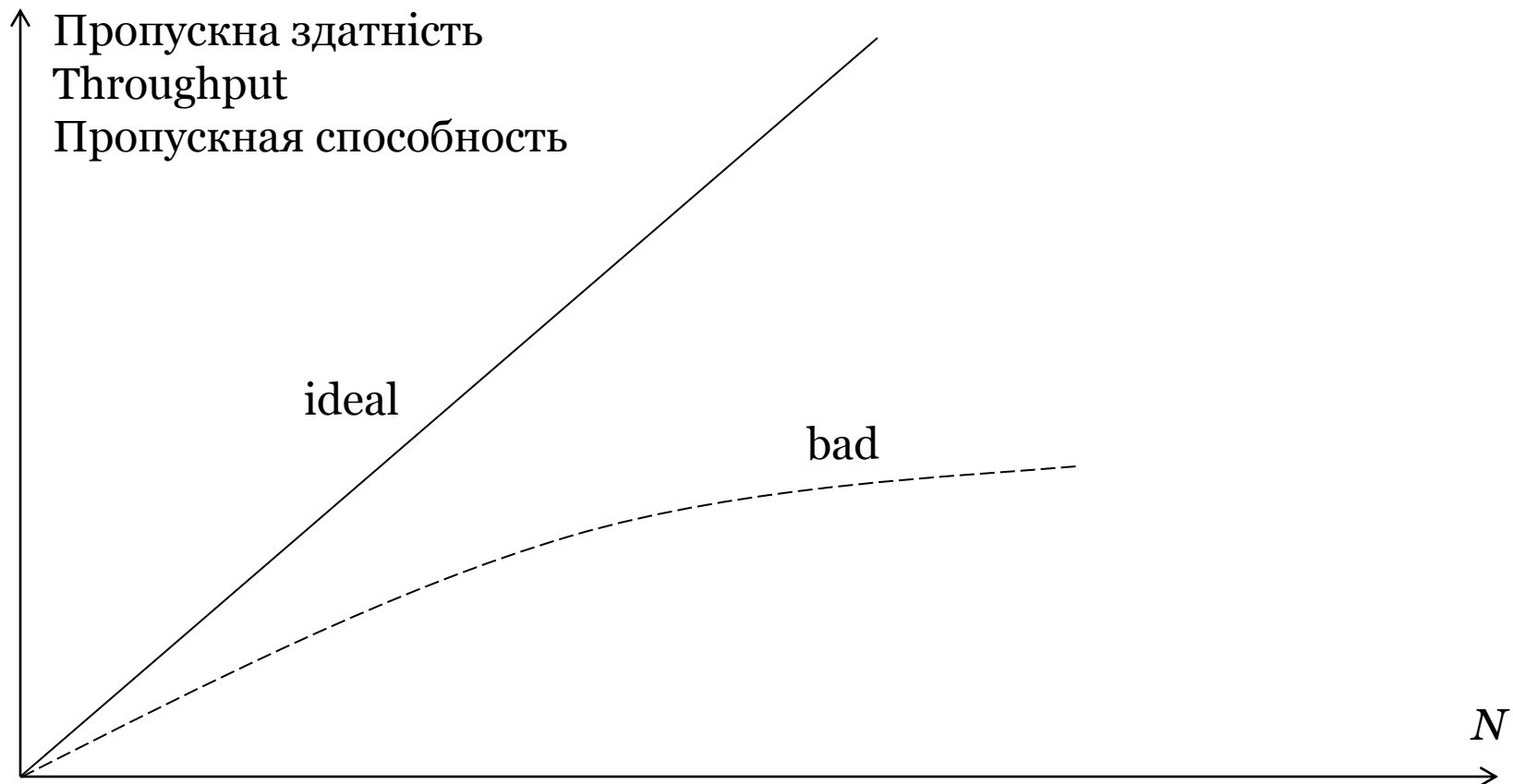
## 1.8 Производительность

- Ідеальна масштабованість за пропускнуою здатністю може бути представлена лінійною залежністю: система, що містить у  $N$  разів більше обладнання і даних може обробляти у  $N$  разів більше запитів. В реальності така масштабованість недосяжна через витрату ресурсів на синхронізацію паралельних обчислень.
- Ideal scalability of throughput could be described using linear dependency: system that contains  $N$  times more hardware and data could process  $N$  times more queries. In reality such scalability is not reachable because of resources usage to synchronize parallel computations.
- Идеальная масштабируемость по пропускной способности может быть представлена линейной зависимостью: система, содержащая в  $N$  раз больше оборудования и данных может обрабатывать в  $N$  раз больше запросов. В реальности такая масштабируемость недостижима из-за использования ресурсов для синхронизации параллельных вычислений.

1.8 Продуктивність

1.8 Performance

1.8 Производительность



## 1.8 Продуктивність

## 1.8 Performance

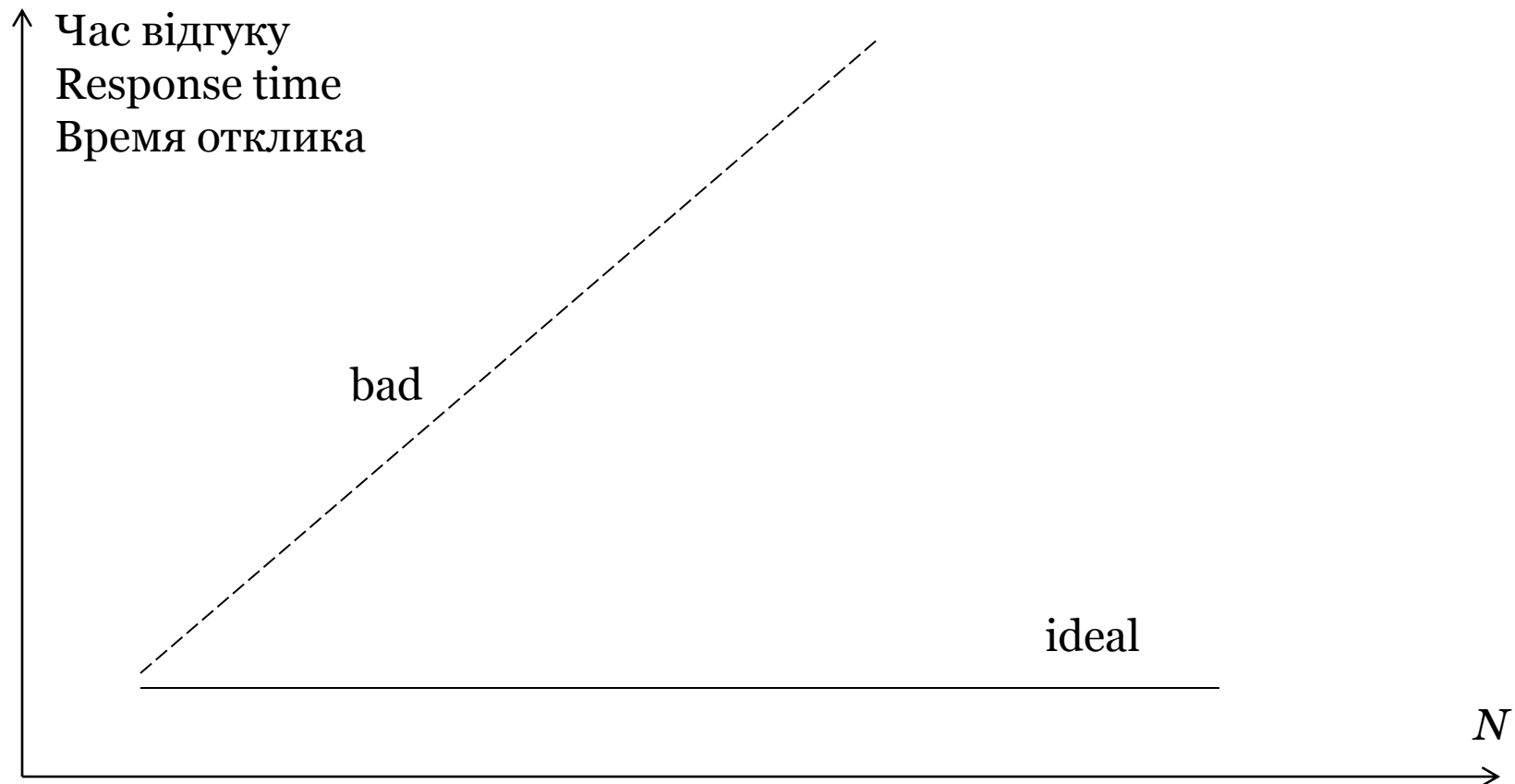
## 1.8 Производительность

- Для масштабованості за часом відгуку ідеальна залежність представляється константою: при збільшенні кількості обчислювачів, обсягу даних та потоку запитів час відгуку не зростає. Ідеальна масштабованість за часом відгуку практично недосяжна.
- For scalability of response time ideal dependency is described using the constant: when increasing number of hardware, data amount, and queries flow response time is not increasing. Ideal scalability of response time is almost not reachable.
- Для масштабируемости по времени отклика идеальная зависимость представляется константой: при увеличении количества вычислителей, объема данных и потока запросов время отклика не возрастает. Идеальная масштабируемость по времени отклика практически недостижима.

1.8 Продуктивність

1.8 Performance

1.8 Производительность



## 1.8 Продуктивність

## 1.8 Performance

## 1.8 Производительность

- Прискорення = час відгуку на системі з одним обчислювачам / час відгуку на системі з  $N$  обчислювачами. Доступність = час нормальної роботи системи / інтервал часу виміру доступності. Усі інтегральні метрики корисні для оцінки роботи системи в цілому, але не для окремих операцій або запитів програм.
- Acceleration = response time for system with one computing instance / response time for system with  $N$  computing instances. Availability = time of normal work of a system / time range used to measure availability. All general metrics are useful to estimate system work in general but not for a single operations or applications queries.
- Ускорение = время отклика на системе с одним вычислителем / время отклика на системе с  $N$  вычислителями. Доступность = время нормальной работы системы / интервал времени измерения доступности. Все интегральные метрики полезны для оценки работы системы в целом, но не для отдельных операций или запросов приложений.

## 1.9 Створення програм, працюючих з БД

## 1.9 Development of applications to work with DB

## 1.9 Создание приложений, работающих с БД

- Сучасні СУБД поставляються з інструментами адміністрування та підтримки БД (створення та зміна таблиць, редагування записів, управління доступом та резервними копіями). Але такий інтерфейс не підходить для бізнес-застосувань. Система повинна бути зрозумілою для масового користувача, який без усякого навчання міг би почати з нею працювати.
- Modern DBMS are provided with administrating and maintenance tools for DB (creating and editing tables, editing records, managing access and backups). But such interface does not suite for business applications. Systems should be understandable for mass user that could start work with it without any training.
- Современные СУБД поставляются с инструментами администрирования и поддержки БД (создание и изменение таблиц, редактирование записей, управление доступом и резервными копиями). Но такой интерфейс не подходит для бизнес-приложений. Система должна быть понятной для массового пользователя, который без всякого обучения мог бы начать с ней работать.



## 1.9 Створення програм, працюючих з БД

## 1.9 Development of applications to work with DB

## 1.9 Создание приложений, работающих с БД

- Для створення бізнес-застосунків використовується клієнт-серверна архітектура. СУБД працює на сервері, а програма – на клієнті. Бізнес-логіка може бути реалізована як на сервері у вигляді збережених процедур, так і на клієнті за допомогою мови програмування.
- Client-server architecture is used to create business applications. DBMS works on the server, while the application works on the client side. Business logic could be implements on the server in the form of stored procedures, as well as on the client using the programming language.
- Для создания бизнес-приложений используется клиент-серверная архитектура. СУБД работает на сервере, а программа – на клиенте. Бизнес-логика может быть реализована как на сервере в виде хранимых процедур, так и на клиенте при помощи языка программирования.

## 1.9 Створення програм, працюючих з БД

## 1.9 Development of applications to work with DB

## 1.9 Создание приложений, работающих с БД

- Проблема невідповідності об'єктно-орієнтованих мов програмування та реляційних баз даних вирішуються за допомогою фреймворків об'єктно-реляційного відображення. Розробник працює зі звичними об'єктними моделями, які автоматично перетворюються у таблиці та навпаки.
- Problem of mismatch between object-oriented programming languages and relational databases are solved using the object-relational mapping (ORM) frameworks. Developer works with usual object models, that automatically transformed into tables and vice versa.
- Проблема несоответствия объектно-ориентированных языков программирования и реляционных баз данных решаются при помощи фреймворков объектно-реляционного отображения. Разработчик работает с привычными объектными моделями, которые автоматический превращаются в таблицы и наоборот.

# 1.9 Створення програм, працюючих з БД

## 1.9 Development of applications to work with DB

### 1.9 Создание приложений, работающих с БД

Framework  
takes an object

Framework  
maps an object  
to tables

Framework  
creates SQL  
query to insert  
a record

Операція збереження  
Storing operation  
Операція сохранения

Framework  
takes an object

Framework  
creates SQL  
query to  
retrieve data

Framework  
maps retrieved  
records to  
objects

Операція читання  
Reading operation  
Операція чтения

## 1.9 Створення програм, працюючих з БД

## 1.9 Development of applications to work with DB

## 1.9 Создание приложений, работающих с БД

- Використання фреймворків прискорює розробку, оскільки програмісту не потрібно глибоко знати SQL або реляційну теорію. Однак фреймворки не дозволяють тонкого налаштування запитів. Хорошим рішенням є використання фреймворків для стандартних операцій та чистого SQL для складних запитів.
- Using frameworks accelerates development, since developer does not need to have deep knowledge of SQL or relational theory. But frameworks do not allow tuning of queries. Good practice is to use frameworks for standard operations and pure SQL for complex queries.
- Использование фреймворков ускоряет разработку, поскольку программисту не нужно глубоко знать SQL или реляционную теорию. Однако фреймворки не позволяют тонко настраивать запросы. Хорошим решением является использование фреймворков для стандартных операций и чистого SQL для сложных запросов.

## 1.9 Створення програм, працюючих з БД

## 1.9 Development of applications to work with DB

## 1.9 Создание приложений, работающих с БД

- Спроба вирішити проблему невідповідності призвела до створення баз даних NoSQL, які представляють собою альтернативу реляційним СУБД. Такі БД не мають структурованої схеми та працюють напряму з об'єктами.
- An attempt to solve the mismatch problem led to occurrence of NoSQL databases, which are alternatives to relational DBMS. Such DB do not have structured scheme and operate directly to objects.
- Попытка решить проблему несоответствия привела к созданию баз данных NoSQL, которые представляют собой альтернативу реляционным СУБД. Такие БД не имеют структурированной схемы и работают напрямую с объектами.

## Контрольні питання

1. Основні вимоги до СУБД.
2. Основні компоненти моделі ANSI/SPARC.
3. Основні особливості мов запитів у сучасних СУБД.
4. Поняття незалежності даних.
5. Переваги використання незалежності даних.
6. Поняття обмежень цілісності та узгодженості даних у СУБД.
7. Поняття безпеки та розмежування доступу у сучасних СУБД.
8. Основні метрики оцінки продуктивності.
9. Архітектура клієнт-сервер.
10. Об'єктно-реляційна втрата відповідності.

## Assessment questions

1. Main requirements to DBMS.
2. Main components of ANSI/SPARC model.
3. Main features of query languages in modern DBMS.
4. Data independence.
5. Advantages of data independence.
6. Integrity constraints and data consistency in DBMS.
7. Security and access control in modern DBMS.
8. Main performance metrics.
9. Client-server architecture.
10. Object-relational mismatch.

## Контрольные вопросы

1. Основные требования к СУБД.
2. Основные компоненты модели ANSI/SPARC.
3. Основные особенности языков запросов в современных СУБД.
4. Понятие независимости данных.
5. Преимущества использования независимости данных.
6. Понятие ограничений целостности и согласованности данных в СУБД.
7. Понятия безопасности и разграничения доступа в современных СУБД.
8. Основные метрики оценки производительности.
9. Архитектура клиент-сервер.
10. Объектно-реляционная потеря соответствия.