1. **Моделі і методи зберігання даних**

Моделі та методи зберігання даних - це способи організації, зберігання та доступу до даних у базі даних. Існує кілька моделей і методів зберігання даних, кожна з яких має свої особливості та сфери застосування. Серед найпоширеніших:

|  |  |  |
| --- | --- | --- |
| Назва | Опис моделі | Основні характеристики |
| Реляційна модель бази даних | Дані організовані в таблиці з рядками і стовпчиками | * Таблиці представляють сутності, а зв'язки встановлюються за допомогою зовнішніх ключів. * Забезпечує цілісність даних за допомогою обмежень (наприклад, первинних ключів, зовнішніх ключів). * Популярні системи керування реляційними базами даних (СКБД) включають MySQL, PostgreSQL та Microsoft SQL Server |
| NoSQL Database Model | Різноманітні моделі (документ, ключ-значення, сімейство стовпців, граф), які відрізняються від традиційної реляційної моделі | * Забезпечує гнучкість для роботи з неструктурованими або напівструктурованими даними. * Масштабується по горизонталі, що робить її придатною для великомасштабних розподілених систем. * Приклади: MongoDB (сховище документів), Cassandra (сховище з широкими стовпцями) і Redis (сховище ключ-значення) |
| Об'єктно-орієнтована модель баз даних | Зберігає дані у вигляді об'єктів, подібно до концепцій об'єктно-орієнтованого програмування | * Об'єкти інкапсулюють дані та поведінку. * Підтримує успадкування, інкапсуляцію та поліморфізм. * Часто використовується в додатках, де об'єкти в мові програмування відображаються безпосередньо на об'єкти бази даних |
| Графова модель бази даних | Дані представлені у вигляді вузлів і ребер у структурі графа | * Підходить для даних зі складними зв'язками та взаємозв'язками. * Ефективний для обходу зв'язків між сутностями. * Neo4j - популярна система управління графовими базами даних |
| Columnar Database Model | Зберігає дані в стовпцях, а не в рядках | * Оптимізована для аналітичних запитів та агрегацій. * Добре підходить для сховищ даних і додатків бізнес-аналітики. * Apache Cassandra та Amazon Redshift є прикладами стовпчикових баз даних |
| Файлова система зберігання | Найпростіша форма зберігання даних, де файли зберігаються в ієрархічній структурі файлової системи | * Підходить для невеликих додатків і простих потреб у зберіганні даних. * Обмежені можливості запитів порівняно з системами баз даних. * Приклади включають традиційні файлові системи, такі як NTFS, ext4, або хмарні системи зберігання даних, такі як Amazon S3 |
| Модель бази даних In-Memory | Дані зберігаються в системній пам'яті, а не на диску для швидшого доступу до них | * Забезпечує високошвидкісне отримання та обробку даних. * Підходить для додатків, що вимагають аналітики в реальному часі або відповідей з низькою затримкою. * Приклади: Redis, Memcached та SAP HANA |

1. **Класифікація інформаційних систем і місце серед них інформаційно-пошукових систем**

Інформаційні системи можна класифікувати на різні типи залежно від їхніх функцій, сфери застосування та характеру інформації, з якою вони працюють.

|  |  |  |
| --- | --- | --- |
|  | Функція | Приклади |
| Системи обробки транзакцій (TPS) | Обробляють щоденні транзакції та операції | Системи касових терміналів (POS), системи обробки замовлень |
| Інформаційні системи управління (MIS) | Надає менеджерам узагальнену, структуровану інформацію для прийняття рішень | Інформаційні панелі, звіти та ключові показники ефективності (KPIs) |
| Системи підтримки прийняття рішень (DSS) | Підтримує прийняття рішень, надаючи інтерактивні та аналітичні інструменти | Інструменти аналізу даних, виконавчі інформаційні системи |
| Виконавчі інформаційні системи (EIS) | Надає стратегічну інформацію керівникам вищого рівня | Інформаційні панелі керівників, системи стратегічного планування |
| Системи управління знаннями (KMS) | Збирає, організовує і робить доступними організаційні знання | Вікі, системи управління документами |
| Системи планування ресурсів підприємства (ERP) | Інтегрує та керує основними бізнес-процесами в режимі реального часу | SAP, Oracle ERP |
| Системи управління взаємовідносинами з клієнтами (CRM) | Керує взаємодією з клієнтами та аналізує дані про клієнтів | Salesforce, HubSpot |
| Системи управління ланцюгами поставок (SCM) | Керує потоком продуктів та інформації в ланцюжку поставок | SAP SCM, Oracle SCM |
| Системи бізнес-аналітики (BI) | Збирає, обробляє та аналізує бізнес-дані для підтримки прийняття рішень | Сховища даних, інструменти інтелектуального аналізу даних |
| Географічні інформаційні системи (GIS) | Збирає, аналізує та відображає просторові або географічні дані | ArcGIS, Google Maps |
| Інформаційні системи охорони здоров'я | Керує інформацією, пов'язаною з охороною здоров'я, включаючи записи про пацієнтів | Електронні медичні картки (EHR), лікарняні інформаційні системи (HIS) |
| Освітні інформаційні системи | Підтримує освітні процеси та управління академічною інформацією | Системи управління навчанням (LMS), студентські інформаційні системи (SIS) |

Інформаційно-пошукові системи, як правило, є частиною різних типів інформаційних систем, в тому числі:

* Системи керування базами даних (СКБД): інформаційно-пошукові системи вбудовуються в СКБД для ефективного пошуку та вилучення даних.
* Пошукові системи: У веб-системах пошукові системи діють як інформаційно-пошукові системи, допомагаючи користувачам знаходити потрібну інформацію в Інтернеті.
* Системи керування документами (СКД): ці системи використовують механізми пошуку інформації для ефективної організації та пошуку документів.
* Системи управління знаннями: Пошук інформації має вирішальне значення для доступу до знань, що зберігаються в системі, та їх вилучення.

Таким чином, інформаційно-пошукові системи відіграють життєво важливу роль у різних типах інформаційних систем, дозволяючи користувачам ефективно шукати, знаходити та отримувати доступ до потрібної інформації відповідно до їхніх потреб.

1. **Організація пошуку. Пошукові машини.**

Організація пошуку в пошукових системах передбачає складний процес індексування, ранжування та вилучення інформації з величезних обсягів даних, доступних в Інтернеті. Організація процесу пошуку відбувається наступним чином:

1. Сканування:

* Процес: Пошукові системи використовують автоматизованих ботів, які називаються "сканерами" або "павуками", для систематичного перегляду веб-сторінок і виявлення нового або оновленого контенту.
* Мета: Знаходити та індексувати веб-сторінки, витягуючи інформацію для подальшого пошуку.

1. Індексування:

* Процес: Зібрана інформація з переглянутих сторінок зберігається в індексі з можливістю пошуку, часто організованому за ключовими словами, фразами і метаданими.
* Мета: Створити структуровану базу даних веб-контенту для швидкого та ефективного пошуку.

1. Алгоритм ранжування:

* Процес: Пошукові системи використовують складні алгоритми для ранжування проіндексованих сторінок на основі релевантності запитам користувачів.
* Фактори: Враховується релевантність ключових слів, якість сторінки, зворотні посилання, активність користувачів тощо.
* Мета: Розставити пріоритети в результатах пошуку, щоб надати користувачам найбільш релевантну і цінну інформацію.

1. Запити користувачів:

* Процес: Користувачі вводять запити в пошукову систему, зазвичай використовуючи ключові слова або фрази, пов'язані з їхніми інформаційними потребами.
* Мета: Передати наміри користувача пошуковій системі для отримання релевантних результатів.

1. Сторінка результатів пошуку (SERP):

* Процес: Пошукова система генерує SERP, що відображає список впорядкованих результатів на основі запиту користувача.
* Компоненти: Кожен результат містить заголовок, фрагмент (короткий опис), URL-адресу і часто додаткові елементи, такі як зображення, відео або тематичні фрагменти.
* Мета: Створити зручний інтерфейс з релевантною інформацією, щоб допомогти користувачам знайти те, що вони шукають.

1. Релевантність і ранжування:

* Процес: Алгоритм ранжування пошукової системи визначає порядок результатів на основі релевантності та якості.
* Мета: Надавати користувачам найбільш релевантний і авторитетний контент, підвищуючи задоволеність користувачів.

1. Постійне оновлення:

* Процес: Пошукові системи регулярно оновлюють свої індекси, повторно скануючи і переіндексуючи веб-сторінки, щоб відобразити зміни в контенті і поведінці користувачів.
* Мета: Забезпечити, щоб результати пошуку залишалися актуальними та релевантними з плином часу.

1. Персоналізація:

* Процес: Деякі пошукові системи персоналізують результати на основі поведінки користувача, його вподобань і місцезнаходження.
* Мета: Забезпечити індивідуальний досвід пошуку для окремих користувачів.

1. Цикл зворотного зв'язку:

* Процес: Пошукові системи аналізують поведінку користувачів, наприклад, кліки, час, проведений на сторінках, і повторні відвідування, щоб уточнити і поліпшити свої алгоритми.
* Мета: Постійно підвищувати релевантність пошуку та задоволеність користувачів.

Популярні пошукові системи, такі як Google, Bing і Yahoo, використовують різні варіанти цих процесів для організації та оптимізації пошуку для користувачів. Алгоритми і технології, що лежать в їх основі, є складними і постійно розвиваються, щоб відповідати динамічній природі Інтернету і очікуванням користувачів.

1. **Створення і типи індексів**

Індекси відіграють вирішальну роль в оптимізації продуктивності бази даних, прискорюючи операції пошуку даних. Вони забезпечують структурований спосіб ефективного доступу та пошуку певних рядків даних. Типи індексів та короткий огляд того, як вони створюються:

|  |  |  |
| --- | --- | --- |
| Тип | Опис | Створення |
| Однорівневий індекс | Структура індексу, яка безпосередньо вказує на розташування даних | Створюється за допомогою простого списку або масиву, де кожен запис містить значення ключа і вказівник на відповідний блок даних |
| Багаторівневий індекс | Ієрархічна структура індексу з декількома рівнями індексів, що ведуть до кінцевого розташування даних | Перший рівень часто є розрідженим індексом, що вказує на другий рівень, який, у свою чергу, вказує на дані |
| Кластерний індекс | Рядки даних у таблиці фізично впорядковані на основі ключа індексу | Зазвичай створюється автоматично під час визначення первинного ключа в таблиці або явно визначається для унікального обмеження |
| Некластерний індекс | Структура індексу та сховище даних відокремлені. Індекс містить вказівники на реальні рядки даних | Створюється явно за допомогою інструкції CREATE INDEX для одного або декількох стовпців таблиці |
| Унікальний індекс | Гарантує, що індексований стовпець (або стовпці) містить унікальні значення, запобігаючи дублюванню записів | Створюється за допомогою оператора CREATE UNIQUE INDEX |
| Складений індекс | Індекс для кількох стовпців, а не для одного | Створюється за допомогою інструкції CREATE INDEX для комбінації стовпців |
| Растровий індекс | Являє собою декілька рядків з бітовим масивом для кожного окремого значення, де кожен біт вказує на наявність або відсутність значення | Часто використовується у сховищах даних для стовпців з низькою кардинальністю |
| Індекс покриття | Індекс, який включає всі стовпці, необхідні для задоволення запиту, усуваючи необхідність доступу до самої таблиці | Створюється шляхом включення всіх необхідних стовпців до некластерного індекс |
| Просторовий індекс | Оптимізує пошук просторових даних на основі геометричних фігур або географічних координат | Створюється для стовпців, що містять просторові типи даних |
| Повнотекстовий індекс | Оптимізує пошук великих обсягів текстових даних, надаючи можливість розширеного текстового пошуку | Створюється для стовпців з текстовими даними за допомогою оператора CREATE FULLTEXT INDEX |
| Хеш-індекс | Використовує хеш-функцію для зіставлення клавіш з розташуванням індексу, що полегшує швидкий доступ до даних | Створюється на основі одного або кількох стовпців за допомогою інструкції CREATE HASH INDEX |
| Функціональний індекс | Індекс, створений на основі результату функції, застосованої до одного або декількох стовпців | Створюється за допомогою інструкції CREATE INDEX з функціональним виразом |

Створення індексу:

*SQL Syntax (для некластерного індексу)*

|  |
| --- |
| CREATE INDEX index\_name  ON table\_name (column1, column2, ...); |

*Приклад:*

|  |
| --- |
| CREATE INDEX idx\_employee\_name  ON employees (last\_name, first\_name); |

Синтаксис створення індексів може відрізнятися залежно від використовуваної системи керування базами даних (СКБД), оскільки різні СКБД мають власний синтаксис і можливості створення індексів.

1. **Проблеми індексування**

Хоча індексування має вирішальне значення для оптимізації продуктивності бази даних, воно пов'язане з власним набором викликів і потенційних проблем. Серед найпоширеніших:

1. Надмірна індексація:

* Проблема: Створення занадто великої кількості індексів у таблиці може призвести до збільшення вимог до сховища та витрат на обслуговування.
* Вплив: Уповільнення операцій модифікації даних (вставки, оновлення та видалення), оскільки індекси повинні оновлюватися разом з реальними даними.

1. Підрозділ "Індексування":

* Проблема: недостатнє використання індексів або нестворення індексів для стовпців, які часто використовуються в операторах WHERE.
* Вплив: Повільніше виконання запитів, особливо для великих наборів даних, оскільки механізму бази даних доводиться виконувати повне сканування таблиці.

1. Застаріла статистика:

* Проблема: Статистика, пов'язана з індексами (наприклад, кардинальність), може з часом застаріти через модифікацію даних.
* Вплив: База даних може використовувати неоптимальні плани запитів, що призводить до зниження продуктивності.

1. Невідповідний тип індексу:

* Проблема: Вибір неправильного типу індексу для конкретного випадку використання (наприклад, використання кластерного індексу, коли некластерний індекс є більш придатним).
* Вплив: Неоптимальна продуктивність запитів і підвищені вимоги до сховища.

1. Фрагментація індексу:

* Проблема: З часом, коли дані додаються, оновлюються або видаляються, індекси можуть стати фрагментованими, що призводить до неефективного зберігання і повільного пошуку.
* Вплив: Зниження продуктивності через додаткові операції вводу/виводу, необхідні для доступу до фрагментованих даних.

1. Стовпці високої кардинальності:

* Проблема: Створення індексів для стовпців з високою кардинальністю (багато унікальних значень) може не дати значних переваг і збільшити вимоги до сховища.
* Вплив: Збільшення витрат на зберігання та обслуговування без пропорційного покращення продуктивності запитів.

1. Проблеми паралелізму:

* Проблема: паралельні транзакції, що включають операції модифікації даних в індексованих стовпцях, можуть призвести до боротьби за сторінки індексу.
* Вплив: Зменшення паралелізму і потенційне зниження продуктивності під час одночасної модифікації даних.

1. Індексування неселективних стовпців:

* Проблема: Індексування стовпців з низькою вибірковістю (коли багато рядків мають однакові значення) може не призвести до значного підвищення продуктивності запиту.
* Вплив: Збільшення витрат на зберігання та обслуговування без відповідного підвищення продуктивності.

1. Індексування для рідкісних запитів:

* Проблема: Створення індексів для запитів, які рідко виконуються, може бути економічно неефективним.
* Вплив: Непотрібні накладні витрати на зберігання та обслуговування індексів, які використовуються нечасто.

1. Індексування на стовпцях з частими оновленнями:

* Проблема: Створення індексів для стовпців, які часто оновлюються, може призвести до високих витрат на обслуговування.
* Вплив: Уповільнення операцій оновлення і збільшення конкуренції за ресурси.

1. Складні плани запитів:

* Проблема: Оптимізатор бази даних може вибирати складні плани запитів за наявності декількох індексів, що призводить до неоптимальної продуктивності.
* Вплив: Збільшення часу обробки запитів і потенційна конкуренція за ресурси.

Щоб усунути ці проблеми, адміністратори баз даних повинні ретельно планувати і контролювати використання індексів, регулярно оновлювати статистику і аналізувати продуктивність запитів, щоб приймати обґрунтовані рішення щодо створення і підтримки індексів. Кожна система баз даних може мати специфічні інструменти і методи для керування індексами, тому важливо звернутися до документації до конкретної бази даних, яку ви використовуєте.

1. **Запити до пошукових машин**

Пошукові системи призначені для пошуку інформації на основі запитів користувачів. Користувачі зазвичай вводять запити, використовуючи ключові слова або фрази, щоб знайти відповідний вміст. Види запитів:

|  |  |  |
| --- | --- | --- |
| Вид запиту | Приклад | Результат |
| Простий пошук за ключовим словом | погода | Знаходить загальну інформацію про погоду |
| Пошук по фразі | "машинне навчання" | Знаходить сторінки, що містять точну фразу "машинне навчання" |
| Булевий пошук | data AND science | Знаходить сторінки, що містять слова "дані" і "наука" |
| Виключення термінів | програмування -java | Знаходить сторінки, пов'язані з програмуванням, але виключає ті, в яких згадується Java |
| Пошук за спеціальними символами | artific\* intelligence | Знаходить сторінки, що містять варіації на кшталт "штучний інтелект", "штучно розумний" тощо |
| Пошук за сайтом | site:wikipedia.org освоєння космосу | Знаходить сторінки про освоєння космосу саме з Вікіпедії |
| Пошук за типом файлу | filetype:pdf квантова фізика | Знаходить документи у форматі PDF, пов'язані з квантовою фізикою |
| Пошук за числовим діапазоном | smartphone $500..$1000 | Знаходить сторінки про смартфони у вказаному ціновому діапазоні |
| Пов'язаний пошук | related:nytimes.com | Повертає сторінки, пов'язані з вказаним веб-сайтом (в даному випадку nytimes.com) |
| Прогноз погоди | прогноз погоди Київ | Отримує поточні або майбутні погодні умови для Києва |
| Функція калькулятора | sqrt(25) + 10 | Повертає результат математичного виразу |
| Конвертація одиниць виміру | 100 USD в EUR | Повертає значення конвертованої валюти |
| Перетворення часового поясу | час у Токіо | Повертає поточний час у Токіо |
| Котирування акцій | AAPL stock | Отримує біржові котирування та інформацію для вказаної акції (у цьому випадку Apple) |
| Пошук місцевого бізнесу | ресторани поблизу мене | Знаходить місцеві ресторани на основі місцезнаходження користувача |
| Розклад кіносеансів | розклад сеансів фільму Барбі | Отримує розклад сеансів певного фільму |

1. **Якість роботи пошукачів**

Якість роботи пошукової системи має вирішальне значення для надання користувачам точної, релевантної та своєчасної інформації. На загальну якість роботи пошукової системи впливають наступні фактори:

|  |  |  |
| --- | --- | --- |
|  | Критерії | Вплив |
| Релевантність результатів | Здатність пошукової системи видавати результати, які відповідають намірам користувача і його запиту | Користувачі повинні знаходити найбільш релевантну та корисну інформацію у верхній частині результатів пошуку |
| Точність | Точність інтерпретації запитів користувачів і надання точної інформації | Користувачі покладаються на те, що пошукові системи надають надійні та фактично правильні результати |
| Швидкість | Швидкість, з якою результати пошуку отримуються та відображаються користувачеві | Швидший час відгуку сприяє позитивному користувацькому досвіду |
| Користувацький досвід | Загальна зручність і дружелюбність інтерфейсу пошуку | Інтуїтивно зрозумілий дизайн і проста навігація покращують користувацький досвід |
| Охоплення | Повнота індексу пошукової системи, що забезпечує включення широкого спектру контенту | Користувачі очікують, що пошукові системи охоплюють широкий спектр інформації, доступної в Інтернеті |
| Свіжість результатів | Своєчасність оновлень пошукової системи для відображення найновішого контенту | Користувачі шукають актуальну інформацію, особливо новини, події та дані в режимі реального часу |
| Репутація та надійність | Надійність і достовірність джерел, включених до результатів пошуку | Користувачі довіряють пошуковим системам, які постійно надають надійну інформацію з авторитетних джерел |
| Персоналізація | Налаштування результатів пошуку на основі вподобань користувача, його місцезнаходження та історії пошуку | Персоналізовані результати підвищують задоволеність користувачів і релевантність |
| Мобільна адаптивність | Адаптивність інтерфейсу пошукової системи до різних пристроїв, особливо мобільних | Зі збільшенням використання мобільних пристроїв, адаптивний дизайн має вирішальне значення для доступності для користувачів |
| Безпека та конфіденційність | Заходи, вжиті для забезпечення конфіденційності та безпеки користувачів, особливо для функцій персоналізованого пошуку | Користувачі з більшою ймовірністю довіряють і використовують пошукові системи, які надають пріоритет їхній приватності |
| Розуміння запитів | Здатність пошукової системи розуміти складні запити, включаючи обробку природної мови | Покращене розуміння призводить до більш точних і релевантних результатів пошуку |
| Релевантність оголошень | Релевантність і ненав'язливість реклами, що відображається в результатах пошуку | Користувачі цінують рекламу, яка релевантна їхнім запитам і не порушує процес пошуку |
| Доступність | Забезпечення доступності пошукової системи для користувачів з обмеженими можливостями | Доступність підвищує інклюзивність і забезпечує ширшу базу користувачів |

1. **Посилальне ранжування (Page Rank)**

PageRank - це алгоритм, який використовується пошуковою системою Google для ранжування веб-сторінок у результатах пошуку. Він був розроблений співзасновниками Google Ларрі Пейджем і Сергієм Бріном і є ключовим фактором у визначенні порядку результатів пошуку за певним запитом.

Як працює PageRank:

1. Аналіз посилань:

* PageRank базується на аналізі структури посилань в Інтернеті. Він розглядає посилання як голоси або схвалення важливості сторінки.
* Чим більше посилань отримує сторінка, тим більш важливою і авторитетною вона вважається.

1. Важливість вхідних посилань:

* Вхідні посилання (посилання, що вказують на сторінку з інших сторінок) мають вирішальне значення. Не всі посилання рівноцінні; важливість сторінки, що посилається на іншу, також враховується.
* Посилання з високорейтингової сторінки є більш цінним, ніж посилання з менш впливової сторінки.

1. Демпферний фактор:

* PageRank вводить коефіцієнт демпфування (зазвичай встановлюється на 0,85) для вирішення проблеми нескінченних петель і тупиків у графі посилань.
* Коефіцієнт демпфування відображає ймовірність того, що користувач буде продовжувати переходити за посиланнями. Він запобігає нескінченному накопиченню рейтингу.

1. Модель випадкового серфера:

* PageRank заснований на моделі випадкового серфера, в якій серфер випадковим чином натискає на посилання, і ймовірність потрапляння на певну сторінку пропорційна її PageRank.
* Алгоритм імітує поведінку серфера, який переміщується в Інтернеті, натискаючи на посилання.

1. Ітерації та збіжність:

* Розрахунок PageRank включає в себе ітеративні алгоритми. Спочатку кожній сторінці присвоюється однакове значення PageRank.
* Алгоритм застосовується ітеративно, поки значення PageRank не зійдуться до стабільних значень.

1. Формула:

* Формула PageRank заснована на принципі суми PageRank сторінок, що посилаються на певну сторінку. Математично її можна виразити як:

PR(A) = (1 - d) + d \* (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn)),

де:

PR(A) - PageRank сторінки A.

d - коефіцієнт демпфування.

PR(T1), ..., PR(Tn) - PageRank сторінок, що посилаються на сторінку A.

C(T1), ..., C(Tn) - кількість вихідних посилань зі сторінок з T1 по Tn.

1. Панель інструментів PageRank:

* Раніше Google надавав публічний доступ до спрощеної версії PageRank через панель інструментів Google, яка показувала важливість сторінки за шкалою від 0 до 10. Однак у 2016 році Google офіційно вилучив функцію PageRank з панелі інструментів.

Хоча PageRank є важливим фактором в алгоритмі ранжування Google, він є лише одним з багатьох факторів, що враховуються. Google використовує комбінацію алгоритмів, щоб надати користувачам релевантні та якісні результати пошуку. З часом Google впроваджував різні вдосконалення в свої алгоритми, і конкретні деталі того, як обчислюється PageRank, також змінюються.

1. **Поняття інформації як категорії, дані і знання**

Поняття "інформація", "дані" та "знання" взаємопов'язані між собою, але представляють різні етапи обробки та розуміння інформації.

|  |  |  |
| --- | --- | --- |
|  | Визначення | Характеристики |
| Дані | Необроблені факти, цифри або символи, які самі по собі не несуть жодного конкретного значення | * Зазвичай представлені у структурованому форматі. * Бракує контексту чи інтерпретації. * Прикладами є цифри, текст або символи |
| Інформація | Дані, які були оброблені, організовані або структуровані для надання їм значення, контексту та релевантності | * Має контекст і значення. * Передає сенс і може бути зрозумілою одержувачу. * Представляє оброблені або організовані дані. * Прикладами є звіти, резюме або описи, отримані на основі даних |
| Знання | Виходять за рамки інформації і являють собою засвоєне розуміння, уявлення та досвід, отримані на основі інформації | * Включає синтез і застосування інформації. * Представляє глибший рівень розуміння. * Включає контекст, досвід і знання. * Дозволяє приймати обґрунтовані рішення та вирішувати проблеми. * Часто є неявним і емпіричним. * Приклади включають досвід, навички та здатність ефективно застосовувати інформацію |

Підсумовуючи:

* Дані - це необроблені, необроблені факти і цифри.
* Інформація - це оброблені та організовані дані, які несуть в собі значення та контекст.
* Знання - це глибше розуміння і застосування інформації, часто набуте через досвід і знання.

Перехід від даних до інформації та знань передбачає послідовну обробку та інтерпретацію. Дані обробляються для створення інформації, а інформація далі синтезується для створення знань. Ця ієрархія відображає зростаючі рівні складності та абстракції в міру того, як ми переходимо від базових фактів до осмислених уявлень і розуміння, яке можна застосувати на практиці.

1. **Програмне та апаратне забезпечення для організації пошуку інформації в мережі інтернет**

Організація пошуку інформації в Інтернеті передбачає поєднання програмних і апаратних компонентів. Пошукові системи, які відіграють центральну роль у пошуку інформації в Інтернеті, покладаються на складні програмні алгоритми та потужну апаратну інфраструктуру.

Програмні компоненти

1. Програмне забезпечення пошукової системи:

* Опис: Основне програмне забезпечення, яке виконує сканування, індексацію та ранжування веб-сторінок. Приклади включають алгоритм PageRank від Google, алгоритми Bing та інші.
* Ключові особливості: Алгоритми пошуку інформації, обробка природної мови, ранжування релевантності та методи індексування.

1. Веб-сканери (павуки або боти):

* Опис: Автоматизовані програми, які систематично переглядають веб-сторінки, виявляють новий контент і збирають інформацію для індексування.
* Основні функції: Виявлення URL-адрес, вилучення даних і дотримання директив robots.txt.

1. Програмне забезпечення для індексування:

* Опис: Програмне забезпечення, що відповідає за створення та підтримку індексу веб-сторінок. Впорядковує інформацію для швидшого пошуку.
* Основні характеристики: Структури зберігання даних, ефективні алгоритми пошуку та механізми оновлення індексу.

1. Алгоритми ранжування:

* Опис: Алгоритми, які визначають релевантність і порядок результатів пошуку на основі різних факторів, таких як вміст сторінки, поведінка користувачів і авторитетність.
* Ключові особливості: Машинне навчання, аналіз даних і такі фактори, як релевантність, авторитетність і свіжість.

1. Обробка природної мови (NLP):

* Опис: Методи, які дозволяють пошуковим системам розуміти та інтерпретувати значення запитів користувачів і веб-контенту.
* Основні характеристики: Аналіз синтаксису, семантичне розуміння та моделювання мови.

1. Програмне забезпечення для обробки запитів:

* Опис: Програмне забезпечення, яке інтерпретує та обробляє запити користувачів, перетворюючи їх на дієві пошукові команди.
* Основні можливості: Синтаксичний аналіз, розширення запитів і ранжування релевантності.

1. Інтерфейс користувача (UI):

* Опис: Інтерфейсне програмне забезпечення, яке дозволяє користувачам взаємодіяти з пошуковою системою. Він включає рядок пошуку, фільтри та відображення результатів.
* Основні характеристики: Дизайн інтерфейсу користувача, чуйність і доступність.

Апаратні компоненти

1. Сервери:

* Опис: Потужні комп'ютерні системи, на яких розміщується і працює програмне забезпечення пошукової системи, бази даних та інші компоненти.
* Основні характеристики: Обчислювальна потужність, пам'ять і ємність для зберігання даних.

1. Центри обробки даних:

* Опис: Об'єкти, в яких розміщено кілька серверів і мережева інфраструктура для підтримки великого обсягу пошукових запитів і зберігання даних.
* Основні характеристики: Резервування, масштабованість і високошвидкісне підключення.

1. Системи зберігання даних:

* Опис: Апаратне забезпечення для ефективного зберігання індексу, сканованих даних та іншої інформації.
* Основні характеристики: Сховище великої ємності, резервування даних і швидкий пошук.

1. Мережева інфраструктура:

* Опис: Апаратні компоненти, які полегшують зв'язок між серверами, центрами обробки даних та інтернетом.
* Основні характеристики: Пропускна здатність, низька затримка та надійність.

1. Балансувальники навантаження:

* Опис: Пристрої, які розподіляють вхідні пошукові запити між кількома серверами для забезпечення оптимального використання ресурсів.
* Основні характеристики: Масштабованість, відмовостійкість та ефективний розподіл трафіку.

1. Системи резервного копіювання:

* Опис: Системи та обладнання для створення та підтримки резервних копій критично важливих даних для запобігання їх втрати.
* Основні характеристики: Регулярне резервне копіювання, резервування та можливості відновлення даних.

Успішна організація пошуку інформації в Інтернеті вимагає добре скоординованої взаємодії передових програмних алгоритмів і надійної апаратної інфраструктури, здатної впоратися з величезними масштабами і складністю мережі. Обидва аспекти потребують постійних оновлень і вдосконалень, щоб не відставати від динамічної природи онлайнової інформації.