Slide27- 1:

Slide27- 2:

Slide27- 3:

Slide27- 4:

Slide27- 5: 빅데이터 핵심 기술의 주요 모듈

빅데이터와 관련된 4가지 주요 단계를 살펴봅시다:

① **데이터 수집(data acquisition)**
② **데이터 저장 및 관리(data storage and management)**
③ **데이터 처리 및 분석(data processing and analysis)**
④ **데이터 프라이버시와 보안(data privacy and security)**

📦 이것은 완전한 데이터 생명주기입니다. 우리는 마치 "데이터 공장"을 짓는 것과 같습니다: 입고, 창고 보관, 가공부터 출하와 보안까지!

Slide27- 1: 그냥 슬라이드 보고 강의

Slide27- 2:

Slide27- 3: 핵심기술 공부

Slide27- 4:

Slide27- 5:大数据关键技术主要模块

我们来看一下大数据涉及的四大关键环节(huán jié)：

① 数据采集（data acquisition）
② 数据存储与管理（data storage and management）
③ 数据处理与分析（data processing and analysis）
④ 数据隐私(**yǐn sī**)与安全（data privacy and security）

📦 这是一个完整的数据生命周期。
我们就像在建一个"数据工厂"：从进货、仓储(Cāngchǔ)、加工，到出货 (Chūhuò)和安全保密！

Slide27- 6: 데이터 수집 도구 - ETL과크롤러

첫 번째 단계는 "데이터 수집"입니다!
어떻게 수집할까요?

- **ETL 도구**는 정규군으로, 구조화된 데이터 소스에서 데이터를 추출하는 데 적합합니다
- **크롤러 도구**는 게릴라군으로, 웹페이지나 API에서 데이터를 "긁어오는" 데 능숙합니다

🕷️ 예를 들어: 타오바오에서 "고양이 화장실"을 검색하면, 곧 다른 플랫폼에서도 고양이 사료를 추천합니다.
그것이 바로 "크롤러"가 당신의 "행동 데이터"를 긁어온 것입니다!

Slide27- 7: ETL 프로세스 개요

ETL의 의미는: **Extract(추출) → Transform(변환) → Load(적재)**

그림에서 보이는 것:

- 다양한 시스템에서 데이터를 추출
- 중간 계층에서 일시적으로 "정제" 및 "통일된 형식"으로 변환
- 마지막으로 이 데이터들을 데이터 웨어하우스나 데이터 마트에 적재

채소를 씻고 요리하는 것과 같습니다:
채소를 깨끗이 씻고 잘라야 하고, 냄비를 달궈야 맛있는 요리가 나옵니다 😋

Slide27- 6:数据采集工具：ETL 与爬虫

我们第一步就是"抓(zhuā)数据"！怎么抓？

1. ETL 工具是正规军 (zhèngguījūn)，适合从结构化数据源抽取(chōuqǔ)数据
2. 爬虫(Páchóng)工具是游击队，擅长(shàncháng)从网页、接口里"扒(bā)"数据

🕷️ 举个例子：你在淘宝搜索"猫砂盆(Māo shā pén)"，很快别的平台也给你推猫粮(māo liáng)。那就是"爬虫"帮你爬来的"行为数据"！

Slide27- 7:ETL流程概览(gài lǎn)

ETL 的含义是：
Extract（提取）→ Transform（转换）→ Load（加载）

图中显示了：

- 把来自不同系统的数据抽取出来
- 暂时放在中间层做"清洗"、"统一格式"
- 最后再把这些数据装载到数据仓库或数据集市(jí shì)中

就像洗菜做饭：菜要洗净切好，锅要烧(shāo)热，饭(fàn)才能香😋

异构: **yì gòu 서로 다른 구조**

Slide27- 8: ETL 처리 프로세스 상세

ETL의 처리 흐름은 다음을 포함합니다:

- **데이터 추출**: 분산 데이터베이스나 로그에서 추출
- **정제/통합/변환**: 복잡한 데이터를 정리(예: null 값, 중복 레코드)
- **적재**: 정제된 데이터를 데이터 웨어하우스로 전송

📊 최종 목표는 분석 가능한 "깨끗한 데이터 레이크"를 형성하는 것입니다.

Slide27- 9: 수집된 데이터는 어디로 가는가?

학생 여러분, 우리는 방금 데이터를 어떻게 "수집"하는지 이야기했습니다.
그런데 생각해 보셨나요:
📦 이렇게 많은 데이터를 어디에 둬야 할까요? 책상 위? 하드디스크? (웃음)

이제 빅데이터의 두 번째 단계로 들어갑니다: **데이터 저장 및 관리**!

Slide27- 8:

我们进一步看到 ETL 的处理流程包括：

1. 抽取数据：从分布式数据库或日志中提取
2. 清洗/转换/集成：把乱七八糟(luàn qī bā zāo)的数据清理好（比如 null 值、重复记录）
3. 加载：把清洗好的数据送入数据仓库

📊 最终的目标是形成可以分析的"干净数据湖"。

Slide27- 9:采集后的数据去哪儿？

同学们，我们刚刚说完怎么把数据"抓"下来—但你有没有想过：

📦 这么多数据，应该放哪儿？是放在桌子上，还是装(zhuāng)进硬盘(yìngpán) ？

这就进入我们大数据的第二个环节：数据存储与管理！

Slide27- 10: 어떤 저장 방식이 있나?

우리의 초기 선택은 일반적으로:

- **파일 시스템(File system)**: 예를 들어 컴퓨터에 "데이터 폴더" 만들기
- **데이터베이스 시스템**: 구조화된 저장, 예를 들어 MySQL이나 Oracle

📁 문제는 - 이러한 전통적인 방식으로 빅데이터를 저장할 수 있을까요?
한 마디로: **부족합니다!**

Slide27- 11: 단일 머신 파일 시스템의 문제점

단일 컴퓨터, 단일 하드디스크는 한 사람이 이사하는 것과 같습니다. 건물 전체의 가구를 한 사람에게 옮기라고 하면 무리입니다!

⚠️ 주요 제약사항:

- 저장 용량 부족
- 읽기/쓰기 속도 제한
- 오류 허용 능력 약함

따라서 우리는 "팀으로 일하는" 저장 방식이 필요합니다: **분산 파일 시스템**

Slide27- 10: 存储方式有哪些？

我们最初的选择一般是：

1. 文件系统（File system）：比如你在电脑里建个"数据文件夹"
2. 数据库系统：结构化存储，比如 MySQL 或 Oracle

📁 问题是—这些传统方式能不能装下我们的大数据？
一个字：不够！

Slide27- 11:单机文件系统的问题

单台电脑、单个硬盘就像一个人搬家—
你让他搬一整栋楼的家具，那是要命的！

⚠️ 主要限制(**xiàn zhì**)：

1. 存储容量不够
2. 读写速度受限
3. 容错能力弱

所以我们需要"组团干活"的存储方式：分布式文件系统。

Slide27- 12: 분산 환경

분산 파일 시스템은 마치 - 1TB 비디오를 10개 부분으로 나누어 10개의 하드디스크에 분산 저장하는 것과 같습니다.
각 하드디스크는 일부만 관리하면 됩니다.

동시에:

- **분산 데이터베이스**, 여러 노드가 협력하여 읽기/쓰기 작업
- 오류 허용성이 강하여, 한 대의 기계가 다운되더라도 데이터가 자동으로 복구됩니다!

📌 약간 "다중 백업 시스템" 같아요. 한 대가 고장나도 "대기 중인" 것이 바로 투입됩니다 😂

Slide27- 13: 관계형 데이터베이스의 한계

관계형 데이터베이스 RDBMS(MySQL, Oracle 등)는 오랫동안 사용되어 왔으며 안정적이고 신뢰할 수 있습니다.
하지만! 단점도 명확합니다:

- 테이블 구조가 고정되어 있어 적응성이 떨어짐
- 비구조화 데이터 처리에 약함
- 수직 확장 비용이 높음(하드웨어 업그레이드)

틱톡 비디오, 웨이보 댓글, 이미지 인식 데이터를 저장하라고 하면? 도움을 요청할 겁니다 🆘

Slide27- 12: 分布式环境

分布式(fēn bù shì)文件系统就像是——你把一个 1TB 的视频切成10份， 分发给10个硬盘储存，每个硬盘只需要管一部分。

同时也有：

1. 分布式数据库，多个节点协作读写
2. 容错性强，就算掉了一台机器，数据也能自动恢复(huī fù)!

📌 有点像"多备胎(bèitāi) 系统"，一台挂了还有"热身(rèshēn)"的等着顶上(dǐng shàng) 😂

Slide27- 13: 关系型数据库遇到瓶颈

关系型数据库 RDBMS（如 MySQL、Oracle）用得很久了，稳定可靠。

BUT！它的缺点也很明显：

1. 表结构固定，适应性差
2. 不擅长(shàncháng)处理非结构化数据
3. 垂直 (chuízhí)扩展 (kuòzhǎn)成本高

   （升级 (shēngjí)硬件 (yìngjiàn)）

你让它存抖音视频、微博评论、图像识别数据？
它会喊救命的 🆘

垂直扩展 (Scale-up) vs. 水平扩展 (Scale-out) 비교

구분	**垂直扩展 (수직 확장 - Scale-up)**	**水平扩展 (가로 확장 - Scale-out)**
방식	**단일 서버의 성능**을 향상시킴	**여러 대의 서버**를 추가하여 시스템을 확장함
비유	**한 명의 슈퍼맨을 더 강하게 키우기**	**보통 병사 여러 명을 더 많이 뽑기**
장점	- 소프트웨어 변경 최소화 - 데이터 일관성 유지 쉬움	- **비용 효율적** (범용 하드웨어 사용) - **확장성과 유연성이 높음** - **단일 장애점 제거** (고가용성)
단점	- **비용이 매우 높음** - **성능 향상에 한계**가 있음 - **단일 장애점** 존재	- 소프트웨어 아키텍처 변경 필요 (분산 시스템) - 데이터 일관성 유지가 더 복잡함

“垂直扩展” 与 “水平扩展” 对比

对比维度	**垂直扩展**	**水平扩展**
中文术语	垂直扩展 / 纵向扩展 / **Scale-up**	水平扩展 / 横向扩展 / **Scale-out**
核心方式	提升单个**服务器**的性能（更强CPU、更多**内存**）	增加**更多服务器**的数量，形成集群
形象比喻	把**一个超人**训练得更强大	招募**更多的普通士兵**
主要优点	- 软件架构改动小 - 数据一致性容易保证	- **成本更低**（使用普通商用硬件） - **扩展性近乎无限** - **无单点故障**，可用性高
主要缺点	- **硬件成本非常昂**贵 - 存在**性能上限** - 存在单**点故障**风险	- 软件架构需要为分布式设计 - 数据一致性、网络通信更复杂

Slide27- 14: 새로운 솔루션 - NoSQL + 분산 데이터베이스

그래서 우리는 "비관계형" 길로 나아갔습니다:

✅ NoSQL 데이터베이스가 등장했으며, 다음을 포함합니다:

- **키-값 데이터베이스**(예: Redis)
- **컬럼 패밀리 데이터베이스**(예: HBase)
- **문서 데이터베이스**(예: MongoDB)
- **그래프 데이터베이스**(예: Neo4j)

이러한 데이터베이스는 "변신 요정"처럼 다양한 현대적 데이터를 저장할 수 있으며, 분산 수평 확장도 가능합니다!

Slide27- 15: 데이터 처리 및 분석

학생 여러분, 데이터를 수집하고 저장했습니다. 다음은 무엇을 할까요? 당연히 처리해야죠!
배달 음식을 주문했으면 열어서 먹어야 하는 것처럼! 🍣
그래서 우리는 세 번째 단계로 들어갑니다: **데이터 처리 및 분석**.

📌 키워드: **고성능 컴퓨팅(High-performance Computing)**

Slide27- 14: 新的解决方案：NoSQL + 分布式数据库

于是我们走上了"非关系型"的道路：

✅ NoSQL 数据库应运而生，包括：

1. 键值数据库（如 Redis）
2. 列族数据库（如 HBase）
3. 文档数据库（如 MongoDB）
4. 图数据库（如 Neo4j）

这些数据库就像"百变小精灵"，能存各种乱七八糟的现代数据，还能分布式横向扩展(héngxiàng kuòzhǎn, 水平扩展, Scale-out)！

Slide27- 15: 数据处理与分析

同学们，数据抓回来了、存好了，接下来要干嘛？当然是处理啦！

就像你点了外卖，还要打开来吃！🍣

所以我们进入第三个环节：数据处理与分析。

📌 关键词：高效计算（High-performance Computing）

Slide27- 16: 분산 아키텍처 - MapReduce와 Spark

방대한 데이터에 대해 단일 컴퓨터로 처리? 계산이 끝날 때까지 기다리면 우리 모두 졸업합니다...

그래서 우리는 "분산 컴퓨팅 아키텍처"에 의존해야 합니다!

오늘 우리는 두 명의 스타를 소개합니다:

- 🧱 **MapReduce**: 원로, 안정적, 배치 처리에 적합
- ⚡ **Spark**: 신예, 빠른 속도, 메모리 컴퓨팅 지원, 마치 "분노의 질주"!

Slide27- 17: 왜 분산 컴퓨팅 프레임워크가 필요한가?

다시 한번 검토해봅시다:

- 💾 데이터가 점점 더 많아짐
- 🕰️ 시간은 점점 더 촉박
- 🧠 분석은 점점 더 복잡

그래서 우리는 여러 대의 기계가 동시에 작업하도록 해야 합니다!

📊 시험 때 소그룹으로 나누어 보고서를 작성하면, 각자가 한 부분을 담당하여 속도가 배로 빨라지는 것과 같습니다!

이것이 MapReduce, Spark 등 분산 컴퓨팅 프레임워크가 존재하는 의미입니다!

Slide27- 16: 分布式架构：MapReduce 和 Spark

面对海量数据，靠单台计算机处理？
等它 算完我们都毕业了...

所以我们要靠"分布式计算架构"！

今天我们要认识两位大明星：

1. 🧱 MapReduce：老牌、稳定、适合批量处理
2. ⚡ Spark：新秀(xīn xiù)、速度快、支持内存计算，简直像"速度与激情(jī qíng)"！

Slide27- 17:为什么需要分布式计算框架？

让我们来回顾一下：

💾 数据越来越多
🕰️ 时间越来越紧(**jǐn)**
🧠 分析越来越复杂

于是 我们需要让多台机器同时工作！

📊 就像考试的时候分小组写报告， 每人负责(fù zé)一块，速度就能翻倍(fān bèi)！

这就是 MapReduce、Spark 等分布式计算框架存在的意义！

Slide27- 18: 분산 컴퓨팅 솔루션

📦 MapReduce와 Spark는 어떤 능력이 있을까요?

✅ 세 가지 주요 문제를 해결할 수 있습니다:

- **데이터 분산 저장**(어디에?)
- **병렬 작업 할당**(누가 할까?)
- **효율적인 결과 통합**(어떻게 합칠까?)

"초대형 공장 조립 라인"으로 이해할 수 있습니다: 원료 투입 → 구역별 처리 → 통합 출하 → 완전 자동화, 효율 극대화!

Slide27- 18: 分布式计算解决方案

📦 MapReduce 和 Spark 有哪些本事呢？

✅ 它们能帮我们解决三大问题：

1. 数据分布式存储（在哪？）
2. 并行任务分配（谁做？）
3. 高效结果整合（怎么合并？）

你可以理解为一个"超大型工厂流水线"：
原料进来 → 分区处理 → 汇总出货 → 全自动，效率爆表(Bào biǎo)！

崛起： **jué qǐ**

Slide27- 19: 데이터 프라이버시와 보안

학생 여러분, 데이터를 아무리 잘 처리해도 프라이버시가 유출되면 "끝장"입니다 ❄️

이제 빅데이터의 네 번째 핵심 기술 포인트를 강의하겠습니다: **데이터 프라이버시와 보안**.

📢 왜 중요할까요? 예를 들어보겠습니다: 앱에서 탈모 광고를 한 번 봤더니, 엄마의 친구 모멘트가 모두 모발 이식 병원 광고로 가득 - 무섭지 않나요? 😂
따라서 우리는 완벽한 것이 필요합니다:

- **프라이버시 보호 체계**(보지 말아야 할 것은 남이 보지 못하게)
- **보안 체계**(훔쳐가지 말아야 할 것은 훔쳐가지 못하게)

Slide27- 19: 数据隐私与安全

同学们，数据处理得再好，如果隐私 (yǐnsī): 泄露 (xièlòu) 了，那就“凉凉 (liáng liáng)”了❄️

现在我们来讲大数据的第四个关键技术点：数据隐私与安全。

📢 为什么重要？
举个例子：你在App上看了一下脱发(tuōfà)广告，结果你妈的朋友圈 (péngyouquān) 全是植发(zhífà)医院—是不是很可怕？😂
所以我们要有完善的：

1. 隐私保护体系（不该看的别让别人看到）
2. 安全体系（不该被盗(dào)的别被盗）

Slide27- 20: 전체 구조 복습

✅ **데이터 저장 및 관리**: 분산 파일 시스템(예: HDFS)
✅ **데이터 처리 및 분석**: MapReduce, Spark, Flink 등 빅데이터 처리 프레임워크

이 둘을 합치면 빅데이터의 "엔진"입니다!
🔧

데이터가 어디에 저장되고,
어떻게 분할되고,
어떻게 작업을 실행하고,
어떻게 결과를 병합하는지,
모두 완벽하게 조화를 이루어야 합니다!

Slide27- 20: 整体结构再复习

我们来做一次整合回顾：
✅ 数据存储与管理：分布式文件系统（如 HDFS）
✅ 数据处理与分析：MapReduce、Spark、Flink 等大数据处理框架

这两者加在一起，就是大数据的"发动机(**fā dòng jī**)"！
🔧

数据在哪里存，
怎么分片，
怎么跑(pǎo)任务，
怎么合并结果，
全部要配合得天衣无缝！

Slide27- 21: 빅데이터 핵심 기술 요약

이 페이지의 그림은 매우 중요합니다!
두 가지 핵심 질문을 요약합니다:

- 📦 **어떻게 저장할까**(분산 저장) → GFS, HDFS
- ⚙️ **어떻게 계산할까**(분산 처리) → MapReduce, Spark, Flink

빅데이터 기술의 궁극적인 목표는 다음을 해결하는 것입니다:

📌 "데이터가 너무 커서 저장할 수 없고 + 너무 많아서 계산할 수 없는" 문제!

학생A: 선생님, 그럼 나중에 우리 모두 Spark를 배워야 하나요?
선생님: 당연하죠! 나중에 이력서에 Spark를 쓸 줄 모르면 대기업에 지원하기 부끄럽습니다 🏃

Slide27- 21: 大数据的核心技术概括(gài kuò)

这一页图非常关键！
它总结了两大核心问题：

- 📦 怎么存（分布式存储）→ GFS、HDFS
- ⚙️ 怎么算（分布式处理）→ MapReduce、 Spark、 Flink

我们大数据技术的终极目标就是解决：

📌 “数据太大放不下 + 太多算不动”的问题！

学生A：老师，那以后我们是不是都得 (děi) 学Spark？
老师：当然！以后写简历不会写Spark都不好意思投大厂(tóu dà chǎng)🏃

Slide27- 22: 분산 저장 시스템 - GFS와 HDFS

두 개의 "초강력 저장 관리자"를 살펴봅시다:

- 🧊 **GFS**: Google File System, 구글이 개발, "빅데이터 저장계"의 시조
- 🔥 **HDFS**: Hadoop Distributed File System, GFS에서 영감을 받아 만든 오픈소스계의 중추

그들의 목표는 –

📦 데이터를 분할하고 파티션을 나누어 다른 기계에 분산, 마치 백과사전 전체를 100페이지로 나누어 각 페이지를 다른 도서관에 두는 것과 같습니다.

📢 한 페이지가 사라져도 다른 복사본으로 복원할 수 있습니다!

Slide27- 22: 分布式存储系统：GFS 与 HDFS

我们来看两个"超强存储大管家"：

- 🧊 GFS：Google File System，由谷歌开发，是"大数据存储界"的鼻祖(bí zǔ)
- 🔥 HDFS：Hadoop Distributed File System， 受GFS 启发(qǐfā)，是开源界的中坚(zhōngjiān)力量

它们的目标就是——

- 📦 把数据分片、分区、分布在不同的机器上，就像把一整本百科全书拆成(chāi chéng)100页，每页放在不同图书馆。
- 📢 即使掉一页，也能用其他副本(fù běn)还原！

Slide27- 23: 단일 머신 VS 분산 - 그림으로 비교

이 그림의 대조는 매우 형상적입니다:

- **왼쪽**: 단일 기계가 혼자 분투, 읽기/쓰기 느림, 다운되기 쉬움
- **오른쪽**: 분산 저장, 여러 노드가 협력, 효율 높고 오류 허용성 강함

📌 택배 회사로 생각할 수 있습니다:

한 사람이 택배를 배달하면 하루에
10건만 가능하지만,
10명이 함께 배달하면
100건을 처리할 수 있습니다!

📦📦📦📦📦📦📦📦📦📦

Slide27- 23:单台 VS 分布式：图解比较

这张图的对比很形象(**xíngxiàng)**：

1. 左边：单台机器孤军奋战 (gū jūn fèn zhàn)，读写慢、容易宕机 (dàngjī)
2. 右边：分布式存储，多个节点协作，效率高、容错性强

📌 你可以想象成一个快递公司：

一个人送快递，
一天只能送10单；
但10个人一起送，
就能搞定100单！

📦📦📦📦📦📦📦📦📦📦

Slide27- 24: MapReduce 작업 흐름도

이 그림은 MapReduce의 3단계 프로세스를 보여줍니다:

1. **Map 단계**: 분할 처리(각 기계가 데이터의 작은 부분을 처리)
2. **Shuffle 단계**: 동일한 key의 데이터를 집계
3. **Reduce 단계**: 계산 결과를 병합

예를 들어:
🍔 각 소그룹이 먼저 자기 그룹이 햄버거를 몇 개 먹었는지 센다(map)
그런 다음 모든 "햄버거 수량"을 집계(shuffle)
마지막으로 전체 반이 총 몇 개를 먹었는지 통계(reduce)!

Slide27- 24: MapReduce 的工作流程图

我们来看大名鼎鼎的 (dàmíngdǐngdǐng) MapReduce！
这张图展示了 MapReduce 的三步走流程：

1. Map 阶段：分片处理（每台机器处理一小段数据）
2. Shuffle 阶段：将相同 key 的数据聚合
3. Reduce 阶段：合并计算结果

举个例子：
🍔 各小组先自己数自己组吃了多少个汉堡（map）
然后所有"汉堡数量"聚合（shuffle）

最后统计出全班一共吃了多少（reduce）！

Slide27- 25: MapReduce + HDFS – 천생연분

- 📦 저장은 HDFS
- ⚙️ 처리는 MapReduce

이 커플은 빅데이터 처리계의 "황금 파트너"입니다!

함께 사용할 때:

- MapReduce는 HDFS에서 데이터를 읽음
- 계산 후 다시 HDFS에 기록
- "배치 처리 + 자동 오류 허용 + 확장성" 실현

학생B: 선생님, 그러면 MapReduce는 "육체 노동자"이고 HDFS는 "창고 관리자"인가요?

선생님(웃음): 재능이 있네요! 바로 그 의미입니다!

Slide27- 25:MapReduce + HDFS：天作之合

- 📦 存储用 HDFS
- ⚙️ 处理用 MapReduce

这对CP就是大数据处理界的"黄金搭档(dādàng)"！

它们配合使用时：

1. MapReduce 会从 HDFS 中读取数据
2. 计算完再写回 HDFS
3. 实现"批处理+自动容错+可扩展性"

学生B：老师，这么说 MapReduce 是"体力劳动者"，HDFS 是"仓库(cāng kù)管理员"？

老师（笑）：你太有才了！就是这个意思！

Slide27- 26: 구글이 주도한 빅데이터 기술

여러분은 모르실 수 있지만, 사실 오늘 우리가 강의한 빅데이터 기술의 많은 부분이 구글의 독창적인 설계에서 왔습니다!

예를 들어:

- 📦 **GFS** → 분산 파일 시스템의 "조상"
- 🧠 **MapReduce** → 초기 분산 컴퓨팅 프레임워크
- 📊 **BigTable** → 컬럼형 데이터베이스, HBase의 전신

말하자면: 구글이 없었다면 오늘날의 빅데이터 생태계도 없었을 것입니다!

학생A: 선생님, 그러면 우리는 매일 "구글의 사상으로 수업을 듣는" 건가요?

선생님: 맞습니다. 그리고 "무료 오픈소스 사용"이죠. 인터넷에 감사합니다!

Slide27- 26:谷歌引领的大数据技术

大家可能不知道，其实我们今天讲的大数据技术，很多都来自谷歌的原创设计！

比如：

- 📦 GFS → 分布式文件系统的"祖宗zǔ zōng"
- 🧠 MapReduce → 早期分布式计算框架
- 📊 BigTable → 列式数据库，HBase的前身

可以说：没有谷歌，就没有今天的大数据生态圈！

学生A：老师，那我们是不是每天都在"用谷歌的思想上课"？

老师：是的，而且是"免费开源使用"哦，感谢互联网！

Slide27- 27: 빅데이터 기술 생태계 지도

이 그림은 "천서" 수준의 지식 지도라고 할 수 있습니다. 📸

구역별로 분석해봅시다:

📦 **분산 저장**:

- GFS, HDFS: 데이터를 보관하는 "분산 창고 시스템"

🗃️ **분산 데이터베이스**:

- BigTable, HBase, NoSQL, NewSQL: 상품 리뷰, 소셜 콘텐츠 등 구조가 유연한 데이터 저장에 적합

⚙️ **분산 처리**:

- MapReduce: 원로 배치 처리 엔진
- Spark: 메모리 컴퓨팅 지원, 반복 작업에 적합
- Flink: 스트림 처리의 왕, 실시간 데이터 분석에 적합

📌 이것이 "빅데이터 도구 풀 패키지"입니다! 각 도구마다 전문 분야가 있습니다.

Slide27- 27:大数据技术生态图谱

这张图可以说是“天书”级别(jíbié)的知识地图 📸

我们来分☒解析一下：

📦 分布式存储：

GFS、HDFS：数据存放的“分仓系统”

🗃️ 分布式数据库：

BigTable、HBase、NoSQL、NewSQL：适合存储结构灵活的数据，如商品评论、社交内容

⚙️ 分布式处理：

1. MapReduce：老牌批处理引擎
2. Spark：支持内存计算，适合迭代性任务
3. Flink：流式处理之王，适合实时数据分析

📌 这就是“大数据工具全家桶(tǒng)”！

每一个工具都有它的专长(zhuān cháng)。

Slide27- 28: 퀴즈 ✅ 정답: C

Slide27- 29: 퀴즈 ✅ 정답: B

Slide27- 30: 퀴즈 ✅ 정답: C

Slide27- 31: 퀴즈 ✅ 정답: D

Slide27- 32: 퀴즈 ✅ 정답: C

Slide27- 33: 퀴즈 ✅ 정답: D

Slide27- 34: 퀴즈 ✅ 정답: C

Slide27- 35: 퀴즈 ✅ 정답: C

Slide27- 36: 퀴즈 ✅ 정답: C

Slide27- 37: 퀴즈 ✅ 정답: B

Slide27- 38: 강의 마무리

좋습니다, 이번 수업은 여기서 마치겠습니다!
오늘 우리는 "빅데이터 시대"의 전말을 알아보았습니다. 정보화 물결, 3대 지원 기술부터 데이터가 어떻게 생성되고 어떻게 발전하는지까지 살펴보았습니다.
다음 수업에서는 계속해서 "빅데이터의 핵심 기술"을 탐구하겠습니다.

마지막 질문 하나: 오늘 여러분도 매일 "데이터를 생산"하고 있다는 것을 발견하셨나요? 😉

감사합니다, 수업 끝! 👏👏👏

Slide1- 28:✅ 正确答案：C

Slide2- 29:✅ 正确答案：B

Slide3- 30:✅ 正确答案：C

Slide4- 31:✅ 正确答案：D

Slide5- 32:✅ 正确答案：C

Slide6- 33:✅ 正确答案：D

Slide7- 34: ✅ 正确答案：C

Slide8- 35:✅ 正确答案：C

Slide9- 36:✅ 正确答案：C

Slide10- 37:✅ 正确答案：B

Slide27- 38:

好了，我们本节课到这里就结束啦！

今天我们认识了"大数据时代"的来龙去脉，从信息化浪潮、三大支撑技术，到数据如何产生、如何演变。

下节课我们会继续探讨"大数据的关键技术"。

最后提问一下：你今天有没有发现，自己每天其实也在"生产数据"呢？😉

谢谢大家，下课！👏👏👏