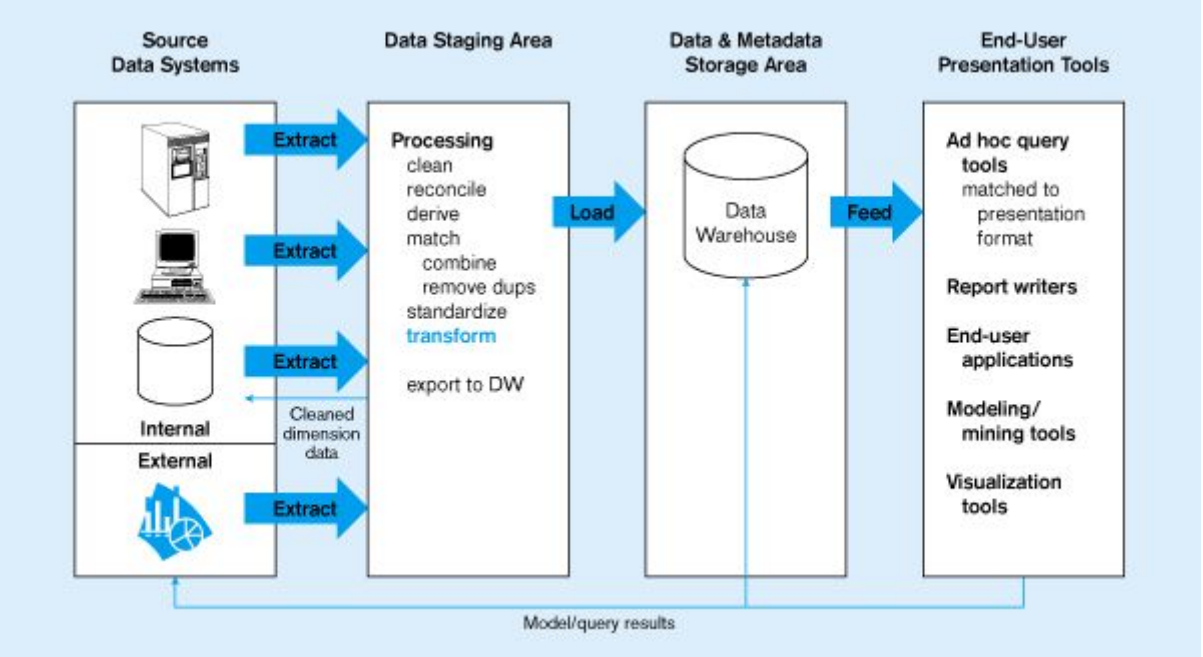
DB zh 3 összefoglaló

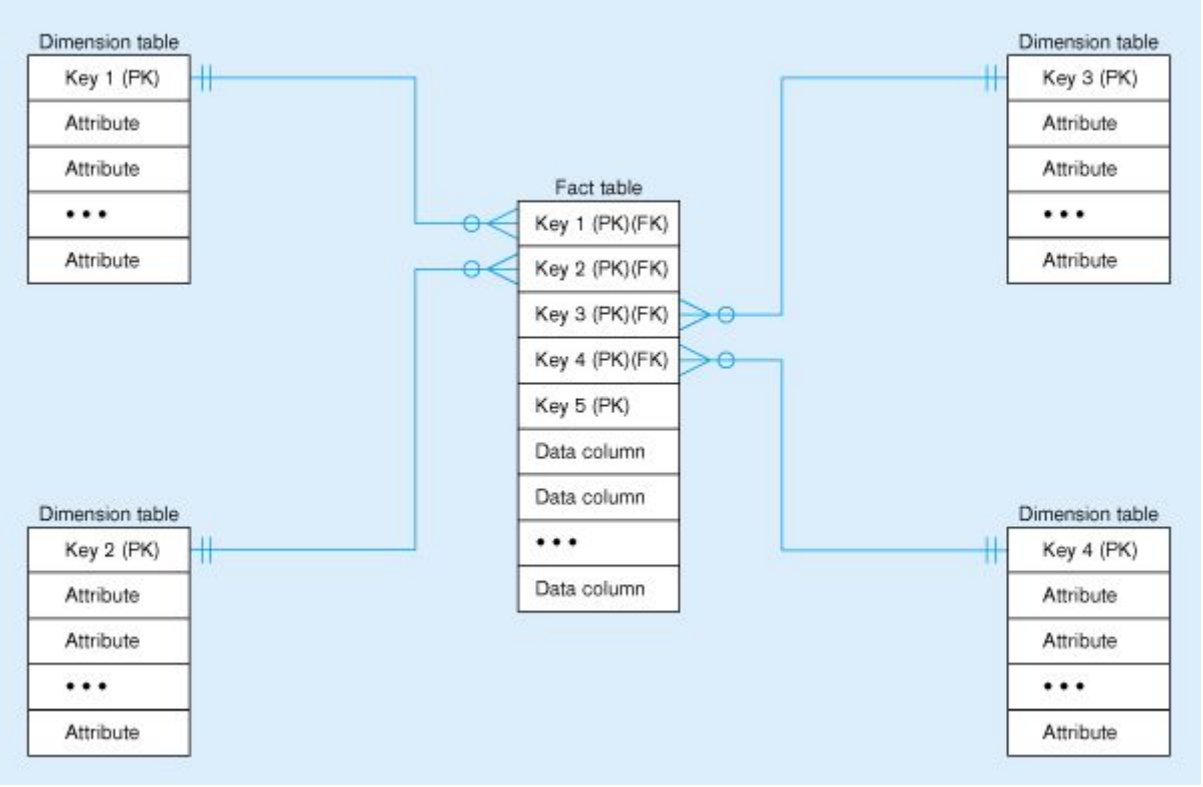
Adattárházak

Rövid definíció: A copy of transaction data, specifically arranged, grouped from operational data sources, to support **business decisions**, reporting etc.
Szokásos architektúra:



Az ETL betűszó jelentése: Extract, Transform, Load

Az adattárházak szokásos sémamegoldása az ún. **csillagséma**, mely **fact** táblákból és **dimenzió** táblákból áll. Ez az adatbázis séma jól támogatja az ad-hoc, interaktív lekérdezéseket. Ezt a sémamegoldást ill. ezeket a lekérdezéseket támogatják jól a **bitmap-indexek**.



A BI[1] eszközök interaktív adatkockájának 3 fő művelete:

- Slice
- Drill down
- Roll up

Five components of a data warehouse are:

- production data sources
- data extraction and conversion
- the data warehouse DBMS
- data warehouse administration
- business intelligence (BI) tools

Subject oriented
A data warehouse can be used to analyte a particular subject area. For examples, “sales” can be a particular subject.

Integrated
A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.

OLAP
OLAP is the use of a set of graphical tools that provides users with multidimensional views of their data and allows them to analyze the data using simple windowing techniques.
A data warehouse is based on a multidimensional data model which views data in the form of a data cube. A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions.

MOLAP operations:

- **Roll up**: summarize data
- **Drill down**: reverse of roll up
- **Slice and dice**: project and select

B+ fa index korlátai

Nevezzen meg két olyan alkalmazási területet, amelyet a szokásos B+ fa alapú index nem támogat kellőképpen!

- Földrajzi lekérdezések > az adatok nem rendezhetőek
- DWM (?) sok rekordot érintő (nem szelektív) lekérdezések

Milyen más megoldás van ezekre a feladatokra?

- R-fa például kifejezetten földrajzi lekérdezéshez jó
- Bitmap-index, max. view

Big data

Eventual consistency: Eventual consistency is a consistency model used in distributed computing to achieve high availability that informally guarantees that, if no new updates are made to a given data item, eventually all accesses to that item will return the last updated value.
Eventually-consistent services are often classified as providing BASE

[1] Business intelligence

- **(B)asically (A)vailable**: basic reading and writing operations are available as much as possible (using all nodes of a database cluster), but without any kind of consistency guarantees (the write may not persist after conflicts are reconciled, the read may not get the latest write)
- **(S)oft state**: without consistency guarantees, after some amount of time, we only have some probability of knowing the state, since it may not yet have converged
- **(E)ventually consistent**: If the system is functioning and we wait long enough after any given set of inputs, we will eventually be able to know what the state of the database is, and so any further reads will be consistent with our expectations

CAP theorem[2]

The CAP theorem or Brewer's theorem states that it is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees:

- **Consistency**: Clients should read the same data. There are many levels of consistency. Levels: strict (RDBMS), tunable (Cassandra), eventual (Amazon Dynamo)
- **Availability**: Data to be available
- **Partition tolerance**: Data to be partitioned across network segments due to network failures.

The CAP theorem implies that in the presence of a network partition, one has to choose between consistency and availability.

Google Spanner

Spanner is Google's highly available, global SQL database. It manages replicated data at great scale, both in terms of size of data and volume of transactions. It assigns globally consistent real-time timestamps to every datum written to it, and clients can do globally consistent reads across the entire database without locking.

In terms of CAP, Spanner claims to be **both consistent and highly available** despite operating over a wide area.
Partitions can happen and in fact have happened at Google, and during some partitions, Spanner chooses C and forfeits A so it is technically a CP system.

However, no system provides 100% availability, so the pragmatic question is whether or not Spanner delivers **availability that is so high that most users don't worry about its outages.**

But how Spanner achieves this high availability?

- Spanner runs on Google's private network. Google controls the entire network and thus can ensure redundancy of hardware and path, upgrades and operations.

https://cloud.google.com/blog/products/gcp/inside-cloud-spanner-and-the-cap-theorem

Key-value stores

A key-value database, or key-value store, is a data storage paradigm designed for storing, retrieving, and managing associative arrays, a data structure more commonly known today as a dictionary or hash table.

Data model: (key, value) pairs
Operations:

- Insert(key, value)
- Fetch(key)
- Update(key, value)
- Delete(key)

Implementation: efficiency, scalability, fault-tolerance

[2] https://www.youtube.com/watch?v=k-Yaq8AHlFA

- Records distributed to nodes based on key
- Replication
- Single-record transactions “eventual consistency”

Document stores
Like key-value stores except value is document.

- XML, YAML, JSON, BSON, binary forms (PDF, docx etc.)

```
{
    "FirstName": "Bob",
    "Address": "5 Oak St.",
    "Hobby": "sailing"
}
```

```
<contact>
  <firstname>Bob</firstname>
  <lastname>Smith</lastname>
  <phone type="Cell">(123) 555-0178</phone>
  <phone type="Work">(890) 555-0133</phone>
  <address>
    <type>Home</type>
    <street1>123 Back St.</street1>
    <city>Boys</city>
    <state>AR</state>
    <zip>32225</zip>
    <country>US</country>
  </address>
</contact>
```

Pandas

What is python pandas?
pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Advantages and distadvantages
Előnyök:

- Könnyebb fejlesztés és debuggolás
- kód olvashatóság
- úgy a python összes előnye

Hátrányok

- Memória
- ACID
- cost based query optimazitation

Segmentation

- mean, group-by, sum, count etc.

Igaz-hamis (Ebből sok nem lesz)

A. táblázatban jelölje, igaz vagy hamis az állítás! (12 pont)

1. A PL/SQL az SQL olyan procedurális kiterjesztése, amely jól szabványosított.
2. A PL/SQL programkód az adatbáziskezelő szerveren fut le, azzal jól integrált (ebből következően pl. tranzakciókat és a jogosultságkezelést is támogatja)
3. Az objektumorientált adatbáziskezelő rendszerek mára rendkívül elterjetté váltak.
4. Sok relációs adatbáziskezelőként ismert rendszer (pl. Oracle, PostgreSQL) tartalmaznak objektumrelációs kiterjesztéseket, és így valójában objektumrelációs adatbáziskezelők.
5. A B+ fa alapú indexek jól használhatóak földrajzi adatokhoz.
6. Az objektumrelációs adatbáziskezelés jól szabványosított, valamennyi gyártó megvalósítása egységes.
7. Célszerű a objektumrelációs lehetőségeket akár egyszerű üzleti adatokhoz is használni.
8. A Google által használt földrajzi vonatkoztatási rendszer, a World Geodetic System (SRID=4326) közvetlenül és egyszerűen használható távolságmérésre.
9. Egy adattárház tulajdonképpen tranzakciós adatok másolata, kifejezetten lekérdezési, elemzési célra.
10. A B+ fa alapú index jól támogatja az adattárházak átfogó lekérdezéseit.
11. Egy materializált nézet lényegében bármilyen SQL kifejezést tartalmazhat.
12. Az in-memory adatbáziskezelőkben feleslegesség vált az SQL procedurális kiterjesztése, az SAP HANA rendszere is csak (deklaratív) SQL-t támogat.

	Igaz	Hamis
1	X	
2	X	
3		X
4	X	
5		X
6		X
7		X
8	X	
9	X	
10		X
11		X
12		X