DDS - Decisions support system (AKA EIS - Executive Information Systems) - Provides info for data driven decisions. Flexible to change of environment and approaches, Used by Non-IT professionals, and less structured queries.

Motivations: In a normal database...

1. **Aggregation reports take a long time**
2. **Must hold lock on all resource for a long time during large scale aggregations**
3. **Competition for computing resources.**

OLTP - *Transaction processing*	OLAP - *Analytical Processing*
Large number of short duration transactions **Frequent Updates, Deletes, and Inserts**	**Retrieval of large amounts of data** **Read only**

	RDBMS	Data Warehouses
Data models	Entity-Relationship	Multidimensional
Schema design	Normalized	Deformalized
Queries	OLTP	OLAP
Derived data & aggregates	Rare	Common
Joins	Many	Some
Updates	Many small ones	Periodical, bulk, ETL
Data access per operation	A few records	Many records (millions)
Workload	Pre-defined	Ad hoc
Indexes	Few	Many
Historical data	None or short term	Long term (mths, yrs)

View Materialization - The Advantage is pre calculating expensive joins because **pre aggregation is essential for interactive response time**. So it will **speed up OLAP Queries**.

Disadvantages are there's an **increased storage cost,** the **content of the view must be maintained** when the underlying detail tables are modified, and **needs a strategy that trades off b/w query performance and accessing up to date data.**

+ View Materialization Problem

Recall:

- A fact table T has d dimension attributes $A_1,\ldots, A_d$, and a numeric attribute B
- A group-by query is parameterized by a subset G of $\{A_1,\ldots, A_d\}$. Its result, denoted as T_G, is referred to as a *cuboid*
- The data cube of T is the set of results of all the 2^d group-by queries (cuboids)

- We will use the following fact table as our running example – 8 possible group by queries

Example 1.

product	location	time	sales
tv	HK	Jan	5
tv	NY	Jan	6
tv	HK	Feb	4
tv	SH	Feb	8
tv	SH	Mar	2
dvd	NY	Jan	3
dvd	SH	Jan	7
dvd	SH	Feb	1
laptop	NY	Feb	4
laptop	SH	Feb	9

The View Materialization Problem:

Find a set of S of k cuboids with the largest benefit(S).

This problem however is known to be NP-hard (we cannot get the exact answer, only optimal).

- Approximation Ratio: Benefit(S)/Benefit(S*)= 1 - 1/e = 0.632.
  S* -> most optimal benefit.

Greedy Algorithm - Greedy(k) /*return a set of k>= 1 cuboids*/

1. initialize a set S with only one cuboid (The fact table T)
2. while |S| < k
3. Tx -> a cuboid Tg maximizing benefit (S U {Tg}) among all the cuboids Tg not belonging to S.
4. add Tx to S.
5. Return S.

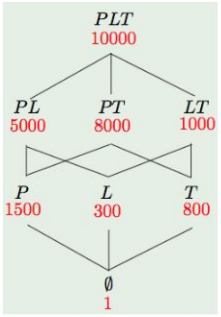
algorithm Greedy(k)
/* return a set of $k \geq 1$ cuboids */
1. initialize a set S with only one cuboid: the fact table T
2. while $|S| < k$
3. $T_X \leftarrow$ a cuboid T_G maximizing $benefit(S \cup \{T_G\})$ among all the cuboids $T_G \notin S$
4. add T_x to S
5. return S

The benefit of selecting a view Tg depends on both:

- the already selected views and
- the views that can be derived from Tg.

- Let us run the algorithm with k = 2 on the lattice shown on the right. Initially, $S = \{T_{\{PLT\}}\}$

G	$benefit(S \cup \{T_G\})$
$\{PL\}$	20000
$\{PT\}$	8000
$\{LT\}$	36000
$\{P\}$	17000
$\{L\}$	19400
$\{T\}$	18400
$\emptyset$	9999



- Therefore, the algorithm adds $T_{\{LT\}}$ to S

Oracle Database Guide (manual on real database systems)
https://learn.uq.edu.au/bbcswebdav/pid-2697062-dt-content-rid-12892545_1/courses/INFS3200S_6720_22239/OracleDWG11.pdf

---

Data Integration

When is data integration required?

- companies merging
- Analyze data from different sources
- Combine data from different websites.
- To get legacy databases to talk to each other.

For example: Telstra has 1K information systems. Health Connect integrates health data to have a global view of the state of the healthcare. Supply Chain management integrates retailers, wholesalers etc.

Distributed Databases - no matter where the data's located, a single organisation has access to full database. We just want to improve the query performance of the system (so fragmentation, replication and subtransaction tasks are made on engineering considerations).

Whitebox Engineering **Problem** - centralized DBS and DW can be considered as data integration.

A different scenario is that the DBs are controlled by different organisations. There's tech, organisational, and political boundaries, which makes these organisations autonomous.

Black Box Problem or possibly ("gray box"- some participants may reveal information).

Database Integration - building a virtual database system that acts as a front end to multiple local DBs. Basically a global system. This is different from DW (physical and loosely coupled with local dbs)

This system provides full database functionality, while interacting with the local systems at their external user interface.

- Local Systems **still maintain their autonomy.**
- Global System provides some means of **resolving the differences in the data representations.**
- Global User can **access info from multiple sources** with a single relatively simple request, as if they were accessing a single centralized database.

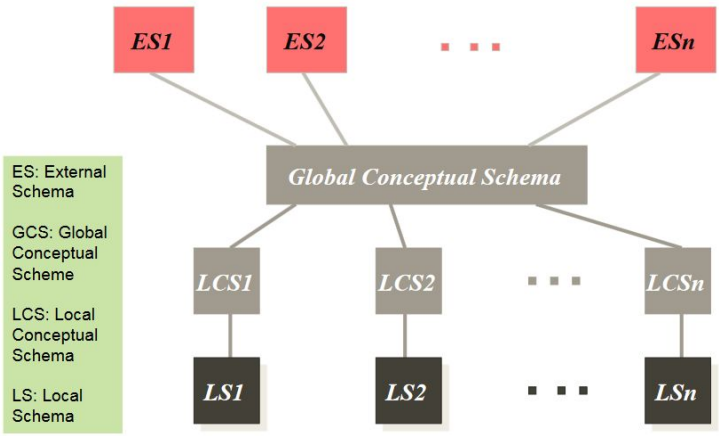
Challenges in DB Integration:

Each DB could be in **different type of DBMS** with different data model, query language etc (relational, semi-structured, NoSQL)
Schema Heterogeneity ▪ S1: Employee(ID, name, address, position, salary) ▪ S2: Worker(EID, name, address) Position(PID, salary, from, until) Storing Employee info on one table vs two tables in different companies.
Data type heterogeneity - Employee ID could be a string or an integer.
Value Heterogeneity - The cashier position can be called "cashier" or "associate"
Semantic Heterogeneity - Salary could be Before Tax value or After tax value.

Data Federation	Interoperable Systems
Build **a virtual global view** of integrated data, without integrating into a centralized database (what DW does). **Federated Database** | **Multi-Database**	No Global Virtual View , only providing mechanisms to communicate w/ different databases

- **semi autonomous** DBs, a global view is provided	-autonomous database systems, limited or no global view is provided.	.

+ Federated Databases



3 Steps for DB Integration

1. Schema Mapping - mapping structures
2. Data Mapping - Match based on content
3. Data Fusion - reconcile mismatched content.

+ Multidatabases