


Indian Institute of Technology Madras
ZANZIBAR CAMPUS

Z3003 · DATA MINING: SCALING TO LARGE DATA

# Data Warehousing: Concepts & Architecture

*Lecture 9 · Friday 2 October 2026*

**Dr. Innocent Nyalala** | BS Semester 5 (DSAI) | IIT Madras Zanzibar

# Session Outline

IIT Madras Zanzibar

1. (0–4 min) Recap and roadmap
2. (4–13 min) Why a warehouse? OLTP vs OLAP, with a real query on our farms
3. (13–22 min) Inmon's four properties, each with a concrete example
4. (22–31 min) Architecture: three tiers, enterprise warehouse vs data marts, column storage
5. (31–36 min) Data lakes: schema-on-write vs schema-on-read
6. (36–44 min) ETL, worked on messy district records
7. (44–50 min) Exercise, summary, next week

# Learning Outcomes

IIT Madras Zanzibar

*By the end of this session, you will be able to:*

| | |
|---|---|
| **Define** | a data warehouse using Inmon's four properties, each with an example |
| **Contrast** | operational (OLTP) and analytical (OLAP) systems, and explain why they are kept separate |
| **Sketch** | a three-tier warehouse architecture, and compare Inmon (top-down) with Kimball (bottom-up) |
| **Distinguish** | a data warehouse from a data lake: schema-on-write vs schema-on-read |
| **Trace** | records through Extract–Transform–Load, and map ETL onto the KDD process |

# Recap: From One Dataset to a Whole Organisation

IIT Madras Zanzibar

**Weeks 1–3**

one dataset: clean → integrate → transform → reduce

**Today**

where an ORGANISATION keeps all of it, over years

**Week 4**

querying it fast: data cubes and OLAP

**Today's question:** how do organisations keep years of cleaned data from many systems so that analysts can answer big questions in seconds — without slowing down daily work?

# Data Warehousing: Concepts & Architecture

IIT Madras Zanzibar

→ **Foundations: Why a Warehouse?**

Inmon's Four Defining Properties

Warehouse Architecture

Data Lakes

ETL: Building the Warehouse

In-Class Exercise

# The Problem: A Minister's Simple Question

IIT Madras Zanzibar

**Pemba North office system**

**Pemba South office system**

**North Unguja office system**

**Zanzibar Urban office system**

**South Unguja office system**

→

**?**

'Yield by crop, region and season, 10 years'

1. **Scattered**
   five separate databases
2. **Inconsistent**
   'PN' vs 'Pemba North'; TZS vs USD; kg vs t
3. **No history**
   old values are overwritten
4. **Busy**
   a 10-year query would slow down daily work

**The fix:** copy, clean and integrate the data into ONE separate system designed for analysis — a data warehouse.

# What Is a Data Warehouse?

IIT Madras Zanzibar

**Definition (W. H. Inmon)**

A data warehouse is a subject-oriented, integrated, time-variant and nonvolatile collection of data in support of management's decision-making.

**The shop counter = operational system**

- serves one customer at a time
- fast, safe updates all day
- keeps only the current state

**The archive room = data warehouse**

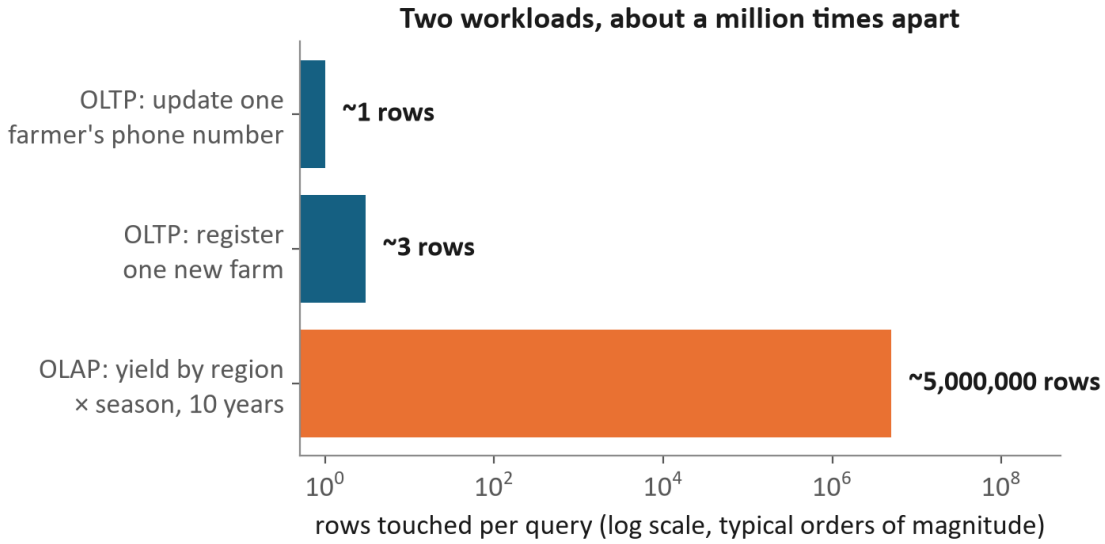
- years of receipts, sorted by subject
- read-mostly: study trends
- never blocks the counter

**In one line:** a separate database of cleaned, integrated, historical data, built for analysis.

# Two Kinds of Work: OLTP vs OLAP

```
-- OLTP: one farmer, one row
UPDATE farmer SET phone='0777…'
WHERE farmer_id = 1042;
```

```
-- OLAP: millions of rows, summarised
SELECT region, season, SUM(yield)
FROM harvest_history
GROUP BY region, season;
```




| | OLTP (operational) | OLAP (warehouse) |
|---|---|---|
| **Purpose** | run the business today | analyse the business over time |
| **Typical query** | update one record | summarise millions of records |
| **Designed for** | fast, safe writes | fast, large reads |

# Why Keep Them in Separate Systems?

IIT Madras Zanzibar

1

**Performance**

a 10-year query on the live system slows or blocks daily transactions

2

**Structure**

many small normalized tables: great for writes, slow for big summaries

3

**History**

live systems overwrite old values; analysis needs every past state

**OLTP system: tuned for fast, safe writes**

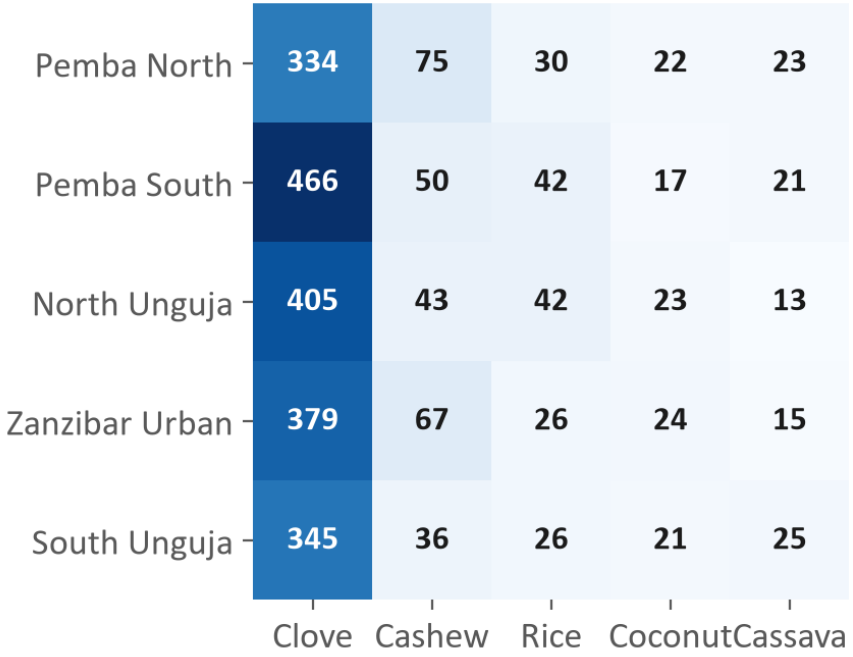
**Warehouse: tuned for fast, large reads**

**Principle:** separate the workloads so each system can be optimised for its own job.

# A Real Analytical Query on Our 500 Farms

IIT Madras Zanzibar

**Total revenue ($ thousands) by region × crop — all 500 farms**




```
df.pivot_table(
  index='region', columns='crop',
  values='revenue_usd', aggfunc='sum')
```

1 **Cloves dominate**
≈ 75% of all revenue, in every region

2 **Top cell**
Pemba South × Clove ≈ $466k

**Scale it up:** 10 years × 50,000 farms × daily updates — this is why the summary lives in a warehouse, and why Week 4 precomputes it as a data cube.

# Data Warehousing: Concepts & Architecture

IIT Madras Zanzibar

Foundations: Why a Warehouse?

→ **Inmon's Four Defining Properties**

Warehouse Architecture

Data Lakes

ETL: Building the Warehouse

In-Class Exercise

# Inmon's Four Properties at a Glance

IIT Madras Zanzibar

**S** **Subject-oriented**

organised around business subjects, not applications

**e.g. a 'Crop yield' area — not a 'login form' table**

**I** **Integrated**

data from many sources made consistent

**e.g. 'PN' and 'Pemba North' → one name; TZS → USD**

**T** **Time-variant**

every record says WHEN it was true

**e.g. yield by season for 10 years, not just 'current yield'**

**N** **Nonvolatile**

loaded data is never edited or deleted

**e.g. a correction is a NEW record, not an overwrite**

**Memory aid:** S-I-T-N — Subject, Integrated, Time-variant, Nonvolatile.

# Subject-Oriented: Organised by What the Business Studies

IIT Madras Zanzibar

**Operational: organised by application**

**Registration app**
**farmer_form, id_upload, sessions**

**Payments app**
**invoices, receipts, refunds**

**Login system**
**users, passwords, tokens**



**Warehouse: organised by subject**

- **Farmer**
- **Crop**
- **Harvest**
- **Sales**

**Rule:** bring together everything about one subject from every application; leave out purely technical data (logins, sessions) that no analyst needs.

# Integrated: One Consistent Version of the Truth

IIT Madras Zanzibar

## Before: two sources disagree

| source | region | revenue | yield |
|---|---|---|---|
| A (Pemba) | PN | 4,602,500 TZS | 1,450 kg |
| B (Unguja) | North Unguja | 2,526 USD | 2.1 t |

## After: integrated in the warehouse

| region | revenue (USD) | yield (t) |
|---|---|---|
| Pemba North | 1,841 | 1.45 |
| North Unguja | 2,526 | 2.10 |

**1** **Names**
'PN' → 'Pemba North' (one code list)

**2** **Currency**
4,602,500 TZS ÷ 2,500 (illustrative rate) = 1,841 USD

**3** **Units**
1,450 kg ÷ 1,000 = 1.45 t

**This is Week 2's integration —** done once, centrally, so every analyst gets the same answer instead of each cleaning the data differently.

# Time-Variant and Nonvolatile: Keep the History

IIT Madras Zanzibar

### Operational system: overwrite

| farmer_id | region |
|---|---|
| 1042 | South Unguja (was Pemba North — erased) |

*The 2024 value is lost. Last year's report by region silently changes.*

### Warehouse: keep every version

| farmer_id | region | valid_from | valid_to |
|---|---|---|---|
| 1042 | Pemba North | 2019-03-01 | 2025-06-30 |
| 1042 | South Unguja | 2025-07-01 | (current) |

**T** **Time-variant**
every row says WHEN it was true (valid_from, valid_to)

**N** **Nonvolatile**
rows are only appended — a correction is a NEW row

**Consequence:** warehouses only grow. Keeping every version is why storage, compression and column stores matter at scale.

# Data Warehousing: Concepts & Architecture

IIT Madras Zanzibar

Foundations: Why a Warehouse?

Inmon's Four Defining Properties

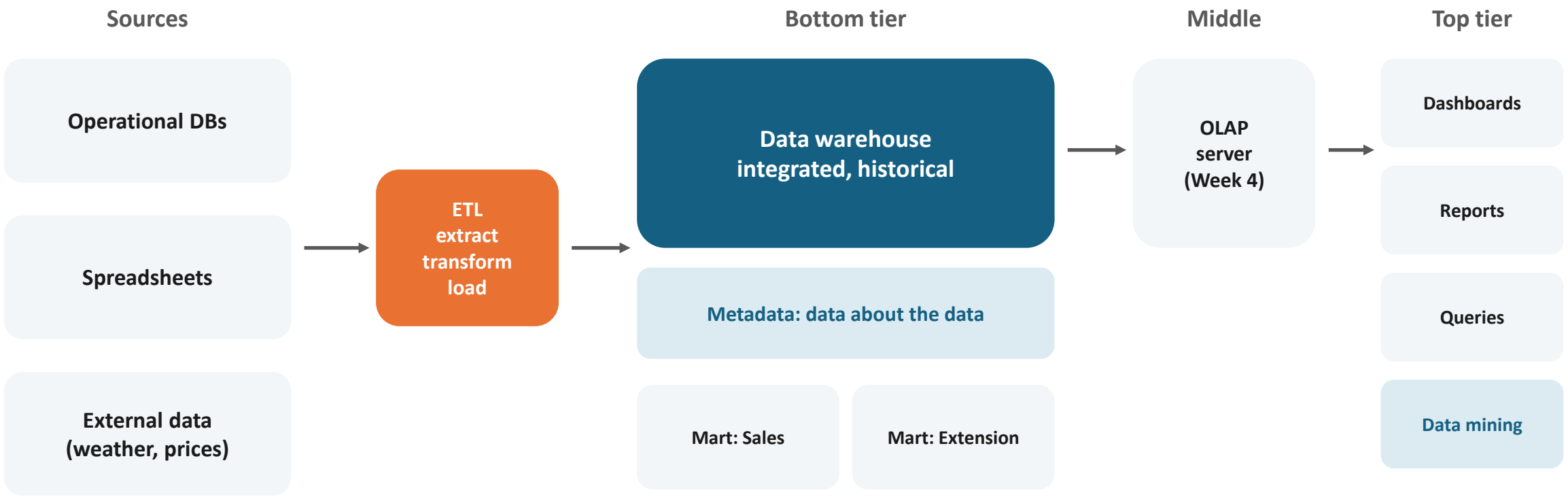
→ **Warehouse Architecture**

Data Lakes

ETL: Building the Warehouse

In-Class Exercise

# The Three-Tier Warehouse Architecture




**Where this course lives:** Weeks 2–3 = the ETL box. Week 4 = the middle tier (OLAP). Weeks 5–9 = data mining in the top tier.

# Enterprise Warehouse vs Data Marts: Inmon vs Kimball

IIT Madras Zanzibar

## Inmon: top-down

*enterprise first, marts later*

**Enterprise warehouse**

**→ Sales mart**

**→ Extension mart**

**+** one consistent truth
**−** slow and costly to start

## Kimball: bottom-up

*marts first, joined later*

**Sales mart**

**+ Extension mart**

**→ conformed enterprise view**

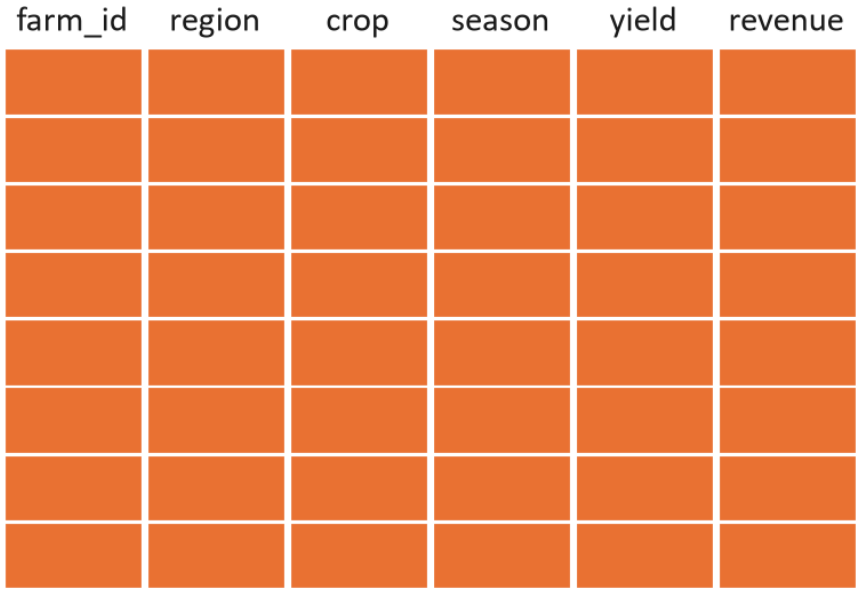
**+** fast, early results
**−** definitions can drift

**In practice:** most organisations mix both — start with a high-value mart, but agree shared definitions from day one.

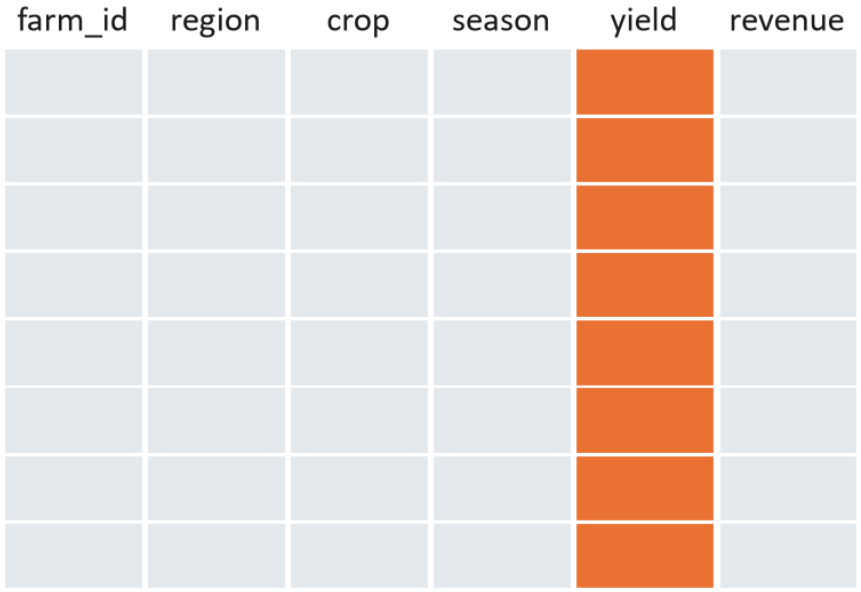
# How Warehouses Store Data: Rows vs Columns

IIT Madras Zanzibar

**Row store: SUM(yield) must read every cell (orange)**




**Column store: SUM(yield) reads one column**




1. **Row store** — best for 'fetch or update one record' (OLTP)
2. **Column store** — best for 'summarise one attribute' (OLAP)
3. **Bonus** — similar neighbours compress well; e.g. Redshift, BigQuery, Snowflake

# Data Warehousing: Concepts & Architecture

IIT Madras Zanzibar

Foundations: Why a Warehouse?

Inmon's Four Defining Properties

Warehouse Architecture

→ **Data Lakes**

ETL: Building the Warehouse

In-Class Exercise

# Data Lakes: Schema-on-Write vs Schema-on-Read

IIT Madras Zanzibar

## Data warehouse: schema-on-WRITE

*like bottled water: purified before the shelf*

- Clean and structure BEFORE storing
- Tables with fixed columns and units
- Ready for fast, trusted queries
- Slower and costlier to load

## Data lake: schema-on-READ

*like a lake: filter a glass when you drink*

- Store raw data as-is, any format
- Tables, JSON, logs, images, PDFs
- Cheap and fast to fill
- Every analysis must clean it again

**Risk:** a lake without a catalogue becomes a 'data swamp' — nobody knows what is in it or whether it is trustworthy.

**Modern 'lakehouse' platforms aim to combine the lake's flexibility with the warehouse's reliability.**

# Data Warehousing: Concepts & Architecture

IIT Madras Zanzibar

Foundations: Why a Warehouse?

Inmon's Four Defining Properties

Warehouse Architecture

Data Lakes

→ **ETL: Building the Warehouse**

In-Class Exercise

# ETL: Extract, Transform, Load




**Connection:** every technique from Weeks 2–3 — missing values, integration, normalization, discretization, sampling — runs inside the T of industrial ETL pipelines.

# Worked Example: One Night of ETL

IIT Madras Zanzibar

## ① Extract: raw rows from two district systems

| src | farm | region | crop | revenue | yield (t) | rain (mm) | date |
|---|---|---|---|---|---|---|---|
| A | 1 | PN | Cassava | 4,602,500 TZS | (missing) | 1,781 | 2026-03-14 |
| B | 2 | North Unguja | Rice | 2,526 USD | 2.10 | 1,766 | 2026-03-14 |
| B | 2 | North Unguja | Rice | 2,526 USD | 2.10 | 1,766 | 2026-03-14 |

**1** **Names**
PN → Pemba North

**2** **Currency**
÷ 2,500 → $1,841

**3** **Missing**
crop median yield (Week 2)

**4** **Duplicates**
drop the repeated row

**5** **Derive**
season; rain → High (Wed)

## ③ Load: clean rows appended to the warehouse

| farm | region | crop | revenue ($) | yield (t) | rain class | season | loaded |
|---|---|---|---|---|---|---|---|
| 1 | Pemba North | Cassava | 1,841 | 1.40* | High | Masika 2026 | 2026-03-15 |
| 2 | North Unguja | Rice | 2,526 | 2.10 | High | Masika 2026 | 2026-03-15 |

**illustrative imputed value. Masika = the long-rains season. Exchange rate and rainfall class boundaries are illustrative.*

# ETL vs ELT: The Cloud-Era Twist

**ETL (classic)**

transform before storing — storage was expensive




**ELT (cloud era)**

store raw first, transform inside the warehouse with SQL




**Either way:** the Transform step is our Weeks 2–3 toolkit; only its position in the pipeline changes.

# Data Warehousing in the Real World

IIT Madras Zanzibar

**1 Health (e.g. DHIS2)**

facility reports aggregated into national dashboards

**2 Mobile money & telecoms**

years of transactions analysed for fraud and planning

**3 Retail chains**

every till's sales loaded nightly for stock and pricing

**4 National statistics**

surveys and censuses integrated over decades

**5 Agriculture ministries**

farm registrations + weather + market prices

**6 Banking**

full transaction history for risk and regulators

**Common thread:** keep history, integrate many sources, and analyse without slowing the live system.

# Choosing an Architecture

IIT Madras Zanzibar

| Scenario | Recommended | Why |
|---|---|---|
| Small organisation, one clear analytical need | a single data mart | fast and cheap to build |
| Large organisation, many departments, long-term consistency | enterprise warehouse (Inmon) | one integrated truth |
| Need results quickly, refine structure later | marts first (Kimball) | early value, conformed later |
| Huge variety of raw data: text, images, sensor logs | data lake (or lakehouse) | store first, structure on read |

**Remember:** whatever the architecture, keep the analytical workload separate from the operational one.

# Data Warehousing: Concepts & Architecture

IIT Madras Zanzibar

Foundations: Why a Warehouse?

Inmon's Four Defining Properties

Warehouse Architecture

Data Lakes

ETL: Building the Warehouse

→ **In-Class Exercise**

# In-Class Exercise (6 minutes, individually)

IIT Madras Zanzibar

**IN-CLASS TASK**

The Zanzibar Ministry of Agriculture wants crop-yield trends across all districts for the past ten years, without slowing down the district offices' daily farmer-registration systems.

1. Which of Inmon's four properties is MOST critical for this request? Why?
2. Name two integration problems you expect across district systems, and how ETL fixes each.
3. Enterprise warehouse or a single data mart to start? Justify in two sentences.
4. Why must the ten-year query NOT run on the registration system?

# Solution

IIT Madras Zanzibar

1. **Q1: Time-variant**
   ten-year trends need every past value kept and dated; an overwrite system cannot answer it

2. **Q2: Integration**
   region codes vs names → one code list; TZS vs USD, kg vs t → convert in the T of ETL

3. **Q3: Start with a mart**
   one clear need (yield trends) → an Agriculture mart delivers fast; use conformed definitions so it can join an enterprise warehouse later

4. **Q4: Separate workloads**
   a scan of ten years of records would lock and slow the live system that clerks rely on (OLTP vs OLAP)

# Summary & Exam Checkpoints

IIT Madras Zanzibar

**Warehouse = SITN**

Subject-oriented, Integrated, Time-variant, Nonvolatile — know an example of each.

**OLTP ≠ OLAP**

Few rows, fast writes vs millions of rows, big reads — so keep them in separate systems.

**Architecture**

Sources → ETL → warehouse (+ marts, metadata) → OLAP → tools. Inmon top-down vs Kimball bottom-up.

**Lake vs warehouse**

Schema-on-read vs schema-on-write. ETL's Transform = our Weeks 2–3 toolkit.

# Common Questions

IIT Madras Zanzibar

**Q** **Is a data warehouse just a big database?**

**A** It is a database, but one designed differently: integrated, historical, read-optimised and separate from operations.

**Q** **Is a data lake better than a warehouse?**

**A** Neither — they solve different problems. Many organisations use a lake for raw data and a warehouse for trusted reporting.

**Q** **Why not just query the live system at night?**

**A** Data still lacks history and integration, and global businesses have no 'night'. Separation solves all three.

**Q** **Where does data mining fit?**

**A** In the top tier: mining algorithms read the clean, integrated history stored in the warehouse.

# What's Next: Week 4

IIT Madras Zanzibar

**Wed** **Multidimensional data model**
data cubes; star, snowflake and fact-constellation schemas

**Thu** **OLAP operations**
roll-up, drill-down, slice, dice; indexing for fast aggregation

**Tut** **Design a star schema**
for a case-study dataset

**Fri** **Cube computation at scale**
what to precompute when the full cube is too large

**Mini-project:** the brief was released in Tutorial 3 — form your team and choose your dataset before Week 4's tutorial.

# References & Further Reading


- Han, J., Pei, J. & Tong, H. (2023). Data Mining: Concepts and Techniques, 4th ed. Morgan Kaufmann — Chapter 3 (Data warehousing and OLAP).
- Inmon, W. H. (2005). Building the Data Warehouse, 4th ed. Wiley.
- Kimball, R. & Ross, M. (2013). The Data Warehouse Toolkit, 3rd ed. Wiley.
- Stonebraker, M. et al. (2005). C-Store: a column-oriented DBMS. Proc. VLDB, 553–564.
- Armbrust, M. et al. (2021). Lakehouse: a new generation of open platforms. Proc. CIDR.