Extract

```
EXTERNAL DATA SOURCES
=====================

┌────────────────────────────────┐    ┌──────────────────────┐
│ IMDb Non-Commercial Datasets   │    │ Box Office Mojo      │
│ https://datasets.imdbws.com/   │    │ (Web Scraping)       │
├────────────────────────────────┤    ├──────────────────────┤
│ • name.basics.tsv.gz (291 MB)  │    │ • Revenue Data       │
│ • title.basics.tsv.gz (212 MB) │    │   (22 MB)            │
│ • title.crew.tsv.gz (78 MB)    │    │                      │
│ • title.episode.tsv.gz (51 MB) │    │ • BoxOfficeMojo IDs  │
│ • title.principals.tsv.gz (736)│    │   Mapping            │
│ • title.ratings.tsv.gz (8 MB)  │    │                      │
│                                │    │                      │
│ Total: 1.37 GB compressed      │    │                      │
└────────────────────────────────┘    └──────────────────────┘
           │                                    │
           │ [Manual Download & Decompress]     │
           │                                    │
           v                                    v
┌─────────────────────────────────────────────────────────────┐
│        LOCAL FILE SYSTEM (MySQL Uploads Directory)          │
│        C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/       │
│                                                             │
│  • name.basics.tsv        • title.principals.tsv            │
│  • title.basics.tsv       • title.ratings.tsv               │
│  • title.crew.tsv         • box_office_revenue.tsv          │
│  • title.episode.tsv                                        │
└─────────────────────────────────────────────────────────────┘
           │
           │ [LOAD DATA INFILE - Bulk Loading]
           │ (High-performance, bypasses SQL parsing)
           v

================================================================================
LAYER 2: STAGING AREA
================================================================================

   { STAGING SCHEMAS }
   ═══════════════════

┌────────────────────────────────┐    ┌──────────────────────────┐
│      SCHEMA: imdb              │    │ SCHEMA: boxofficemojo    │
├────────────────────────────────┤    ├──────────────────────────┤
│ • NameBasics                   │    │ • BoxOfficeRevenue       │
│   (12M+ rows)                  │    │   (daily revenue records)│
│                                │    │                          │
│ • TitleBasics                  │    │ • BoxOfficeMojoIds       │
│   (11.9M rows)                 │    │   (tconst mapping)       │
│                                │    │                          │
│ • TitleCrew                    │    └──────────────────────────┘
│   (director/writer lists)      │
│                                │
│ • TitleEpisode                 │
│   (9.18M rows)                 │
│                                │
│ • TitlePrincipals              │
│   (94M+ rows - largest)        │
│                                │
│ • TitleRatings                 │
│   (1.4M rows)                  │
└────────────────────────────────┘
```

Transform

```
[1] STRING PARSING FOR MULTI-VALUED ATTRIBUTES
──────────────────────────────────────────────

Fixed-Width Attributes (SUBSTRING_INDEX):
  genres = "Action,Drama,Thriller"
    --> genre1 = "Action"
    --> genre2 = "Drama"
    --> genre3 = "Thriller"

  knownForTitles = "tt001,tt002,tt003,tt004"
    --> knownForTitle1 = "tt001"
    --> knownForTitle2 = "tt002"
    --> knownForTitle3 = "tt003"
    --> knownForTitle4 = "tt004"

Variable-Width Attributes (RECURSIVE CTEs):
  directors = "nm0001,nm0002,nm0003,...,nm0150"

    WITH RECURSIVE DirectorSplit AS (
      -- Base case: Extract first director
      -- Recursive case: Extract next director
      -- Repeat until exhausted
    )

    --> Row 1: titleID=1, directorID=nm0001
    --> Row 2: titleID=1, directorID=nm0002
    --> Row 3: titleID=1, directorID=nm0003
    ... (up to 150+ rows for single title)

  ⚠ Bottleneck: 30% of transformation time
  ⚠ Recursion depth: Can exceed 1,000 iterations
  ⚠ Config: cte_max_recursion_depth = 5000


[2] DATA TYPE CONVERSIONS
─────────────────────────

Boolean Flags:
  isAdult = '1'  -->  TRUE (BOOLEAN)
  isAdult = '0'  -->  FALSE (BOOLEAN)

NULL Markers:
  '\N'  -->  NULL (handled by LOAD DATA INFILE)

Numerical Precision:
  averageRating = "7.5"  -->  7.5 (DECIMAL(3,1))
  numVotes = "12345"     -->  12345 (INT)
  revenue = "12345678"   -->  12345678.00 (DECIMAL(12,2))


[3] SURROGATE KEY GENERATION
────────────────────────────

Natural Keys (IMDb IDs):          Surrogate Keys (Performance):
  tconst = "tt0123456"        -->   titleID = 1 (AUTO_INCREMENT)
  nconst = "nm0987654"        -->   personID = 1 (AUTO_INCREMENT)

Benefits:
  • 4-byte integer vs 20-character string (join performance)
  • Immutable (natural keys may change)
  • Sequential allocation (cache-friendly)

Preservation:
  • Natural keys stored as UNIQUE columns
  • Enables source data traceability
  • Supports incremental ETL
```

```
[4] DATE DIMENSION POPULATION (ON-DEMAND STRATEGY)
    ──────────────────────────────────────────────

    INSERT IGNORE INTO DimDate (...)
    SELECT DISTINCT revenue_date FROM BoxOfficeRevenue
    UNION
    SELECT CURDATE()
    UNION
    SELECT CURDATE() + INTERVAL 1 DAY;

    Approach: Only materialize dates referenced in fact tables
    Trade-off: Smaller storage vs. ETL complexity


[5] DATA VALIDATION & FILTERING
───────────────────────────────

Defensive WHERE Clauses:
    • Filter NULL references before insertion
    • Validate foreign key relationships
    • Remove orphaned records (e.g., episodes without parents)

Temporary Constraint Disabling:
    SET FOREIGN_KEY_CHECKS = 0;
    SET UNIQUE_CHECKS = 0;
    -- Bulk insert operations --
    SET FOREIGN_KEY_CHECKS = 1;
    SET UNIQUE_CHECKS = 1;
```

Load

```
LOADING ORDER (Dependency-Aware):
─────────────────────────────────

STEP 1: Independent Dimensions
┌─────────────────────────┐    ┌─────────────────────────┐
│      DimPerson          │    │        DimTitle         │
├─────────────────────────┤    ├─────────────────────────┤
│ • 12M+ people           │    │ • 11.9M titles          │
│ • Professions parsed    │    │ • Genres parsed         │
│ • AUTO_INCREMENT keys   │    │ • Type categorization   │
└─────────────────────────┘    └─────────────────────────┘
          │                                  │
          └─────────────────┬────────────────┘
                            │
                            v

STEP 2: Bridge Tables (Many-to-Many Relationships)
┌─────────────────────────────────────────────────────────────┐
│              BridgeTitlePerson (Bridge Table)               │
├─────────────────────────────────────────────────────────────┤
│ • Links titles to cast/crew members                         │
│ • category = "actor", "actress", "director", etc.           │
│ • Resolves many-to-many between DimTitle and DimPerson      │
└─────────────────────────────────────────────────────────────┘
                            │
                            v
┌─────────────────────────────────────────────────────────────┐
│                    DimCrew (Snowflake)                      │
├─────────────────────────────────────────────────────────────┤
│ • Normalized crew roles (directors, writers)                │
│ • Populated via RECURSIVE CTEs                              │
│ • titleID --> personID mapping                              │
└─────────────────────────────────────────────────────────────┘
                            │
                            v

STEP 3: Hierarchical Dimensions
┌─────────────────────────────────────────────────────────────┐
│                        DimEpisode                           │
├─────────────────────────────────────────────────────────────┤
│ • 9.18M TV episodes                                         │
│ • Self-referencing: parentTitleID --> DimTitle              │
│ • episodeTitleID --> DimTitle                               │
│ • Season/Episode hierarchy                                  │
└─────────────────────────────────────────────────────────────┘
                            │
                            v

STEP 4: Time Dimension
┌─────────────────────────────────────────────────────────────┐
│                          DimDate                            │
├─────────────────────────────────────────────────────────────┤
│ • On-demand population                                      │
│ • Year, Quarter, Month, Day hierarchy                       │
│ • Supports temporal roll-up/drill-down                      │
└─────────────────────────────────────────────────────────────┘
                            │
                            v

STEP 5: Fact Tables (Transaction Grain)
┌──────────────────────────────┐  ┌──────────────────────────┐
│   FactRatingSnapshot         │  │ FactBoxOfficeRevenue     │
├──────────────────────────────┤  ├──────────────────────────┤
│ • Periodic Snapshot          │  │ • Transaction Fact Table │
│ • Grain: One row per title   │  │ • Grain: One row per title│
│   per snapshot date          │  │   per revenue date       │
│                              │  │                          │
│ Foreign Keys:                │  │ Foreign Keys:            │
│  - titleID --> DimTitle      │  │  - titleID --> DimTitle  │
│  - snapshotDateID --> DimDate│  │  - dateID --> DimDate    │
│                              │  │                          │
│ Measures:                    │  │ Measures:                │
│  - averageRating (DECIMAL)   │  │  - revenue (DECIMAL)     │
│  - numVotes (INT)            │  │  - rank (INT)            │
└──────────────────────────────┘  └──────────────────────────┘
```