Kyle Purves
Kajaram Kanthasamy
Ofir Sela

TV Show and Movies Database - A Scalable Solution for Entertainment

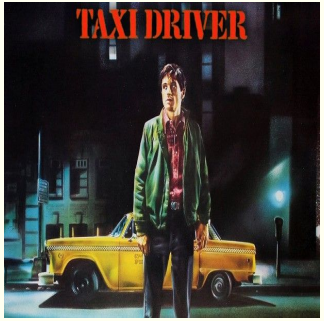
CCPS510: Database Systems 1

Instructor: Ersan Cam

What is the issue we solve?

- Lack of centralized, structured data for managing:
  - Production companies
  - Directors and actors
  - Movies, TV shows, and reviews

- Challenges:
  - Handling complex relationships (e.g., actors in multiple roles)
  - Ensuring data accuracy and scalability



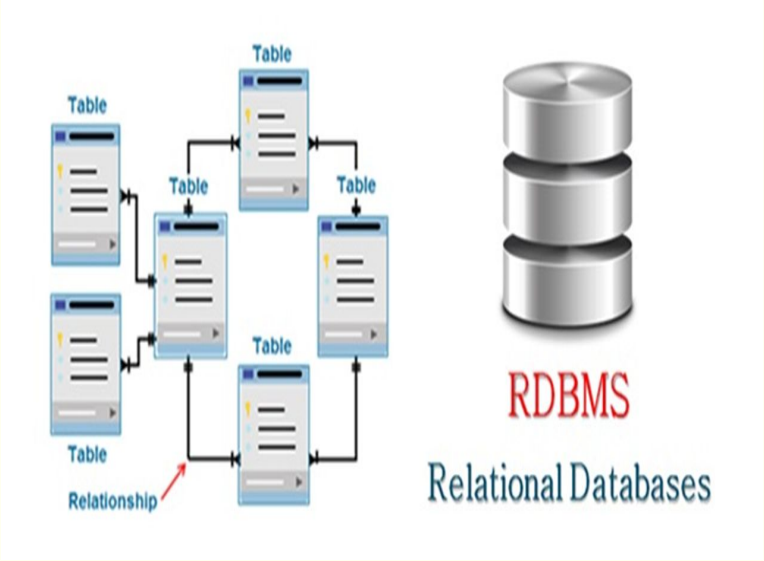
Our solution…

- Relational Database Management System
- Structured, centralized approach for data management
- Key Features:

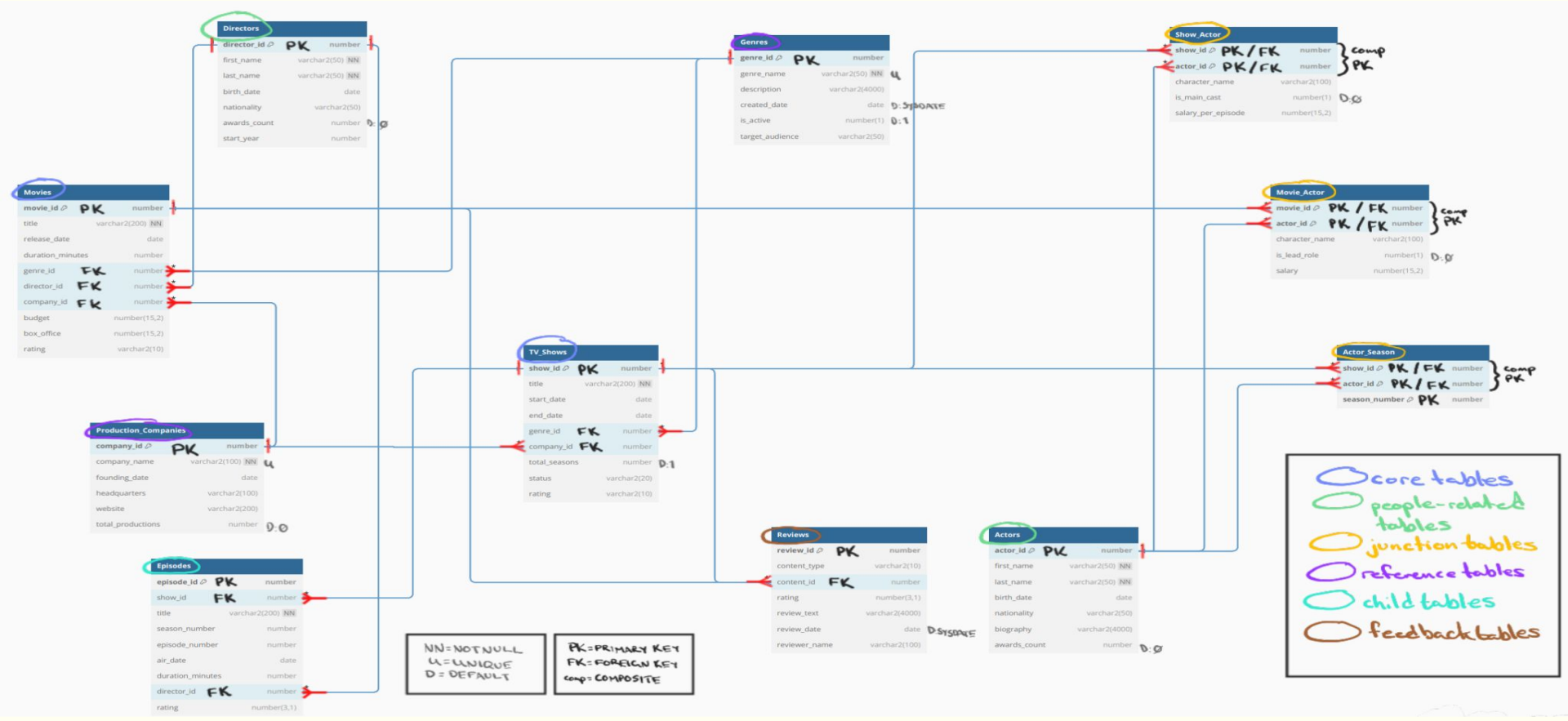
  - Referential integrity (foreign keys, cascading updates)

  - Support for complex queries (e.g., trends analysis, rating averages)

  • API integration for seamless front-end access



System Architecture



Business Rules - Examples

- Each movie belongs to **one** genre
- Actor salaries tracked **differently** for movies (total) vs. TV shows (per episode)
- Episode numbers **must** be unique within each season of a TV show.
- A production company **must** have a unique name and keep track of the total number of productions they have completed.
- Directors **must** have a start year for productions
- TV shows **must** have **unique** episode numbers within seasons
- Genres **cannot** be deleted if referenced
- An actor **must** have a **unique** combination of first name, last name, and birth date.

Database Design

Movie_ Show_ID	Title	Type	Genre	Director_ Info	Production_Info	Actor_ List	Rating	Reviews
1	Inception	Movie	Action, Sci-Fi	C.Nolan, 11 awards, start 1998	Warner Bros., Burbank CS, 900	DiCaprio as Cobb ($20M), Tom Hardy as Eames ($7M)	14A	MovieCritic123:9.0, FilmBuff: 8.8
2	The Crown	Show	Drama, History	Various Directors	Netflix, Los Gatos CA, 500	C. Foy as Queen (250K/ep), M. Smith as Philip (200K)	14+	TVFan: 8.5, SeriesLover: 9.0

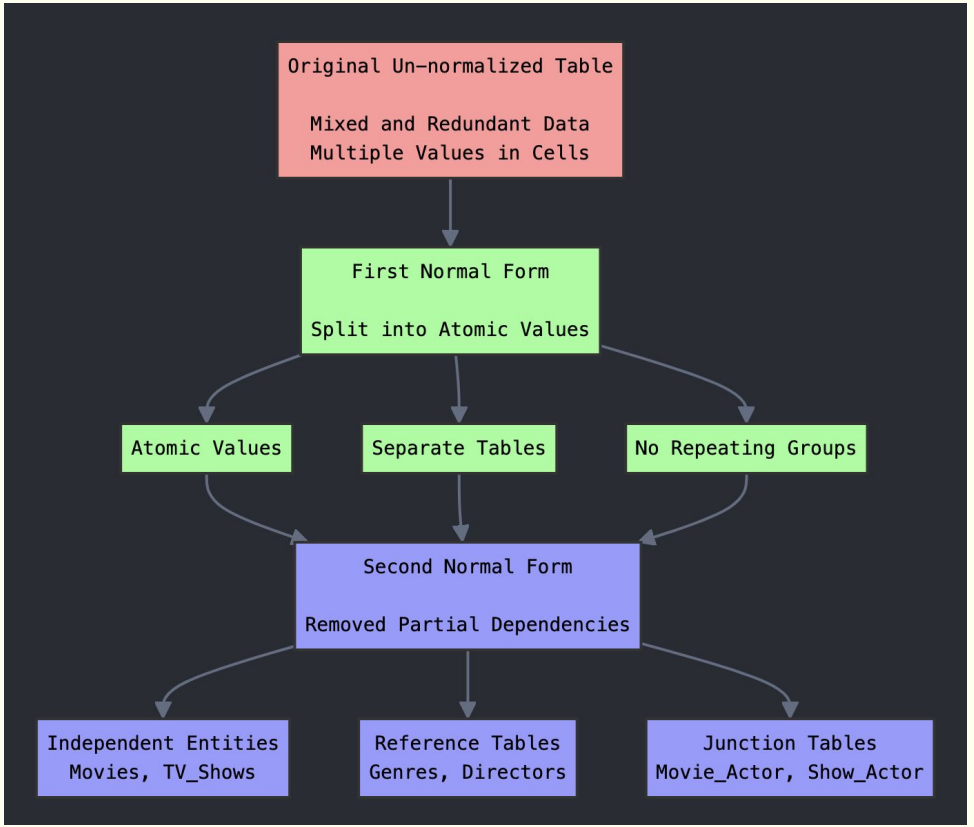
Issues:

- Multiple values in single columns as strings
- Mixed data types (ex, salary could be total or per episode)
- Redundancy (ex, Production_Info repeated for each movie
- No relationships to ensure data integrity

Normalization Process

Original Un-normalized Table:

- Had mixed data (like Director_Info combining name, nationality, awards)
- Multiple values in cells (like multiple genres, multiple actors per movie)



Normalization Process

Steps Taken:

1. **First Normal Form (1NF)**:
   - Split into atomic values (separated Director_Info into individual fields)
   - Created separate tables (Movies, TV_Shows, etc.)
   - Removed repeating groups (Actor lists became separate Actor table)

2. **Second Normal Form (2NF)**:
   - Created independent entities and linked them with foreign keys and bridge tables (Movies, TV_Shows as core tables)
   - Made reference tables (Genres, Directors, Production_Companies)
   - Created bridge tables (Movie_Actor, Show_Actor, Actor_Season).

3. **Benefits**:
   - Reduced redundancy
   - Data integrity and scalability

Movies/Shows

ID	Title	Type	Rating
1	Inception	Movie	14A
2	The Crown	Show	14+

Genres_Applied

Movie_ID	Genre
1	Action
1	Sci-Fi
2	Drama
2	History

Director_Details

Movie_ID	Director_name	Nationality	Awards	Start_Year
1	C. Nolan	British	11	1998

Production_Details

Movie_ID	Company_Name	Location	Total_Prodcutions
1	Warner Bros	Burbank, CA	900
2	Netflix	Los Gatos, CA	500

Genres (reference table for genres)

Genre_ID	Genre_Name	Description
1	Action	High-energy films
2	Sci-Fi	Futuristic stories

Directors (independent entity for directors)

Director_ID	First_name	Last_Name	Nationality	Awards_Count	Start_Year
1	Christopher	Nolan	British	11	1998

Production_Companies (independent entity for companies)

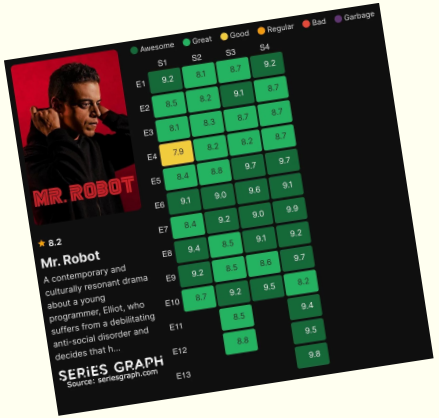
Company_ID	Company_Name	Location	Total_Prodcutions
1	Warner Bros	Burbank, CA	900
2	Netflix	Los Gatos, CA	500

SQL Implementation

- **Sample Tables:**
  - Episodes
  - Movies
  - TV Shows
  - Reviews
- **Constraints:**
  - Foreign keys
  - NOT NULL and CHECK constraints

```sql
-- Create Episodes table
CREATE TABLE Episodes (
    episode_id INT PRIMARY KEY,
    show_id INT REFERENCES TV_Shows(show_id) ON DELETE CASCADE,
    title VARCHAR(200) NOT NULL,
    season_number INT CHECK (season_number > 0),
    episode_number INT CHECK (episode_number > 0),
    air_date DATE,
    duration_minutes INT CHECK (duration_minutes > 0),
    director_id INT REFERENCES Directors(director_id) ON DELETE SET NULL,
    rating DECIMAL(3,1) CHECK (rating >= 0 AND rating <= 10),
    UNIQUE(show_id, season_number, episode_number)
);
```

```sql
-- Create Reviews table
CREATE TABLE Reviews (
    review_id INT PRIMARY KEY,
    content_type VARCHAR2(10) CHECK (content_type IN ('Movie', 'TV Show')),
    content_id INT,
    rating NUMBER(3,1) CHECK (rating >= 0 AND rating <= 10),
    review_text VARCHAR2(4000),
    review_date DATE DEFAULT SYSDATE,
    reviewer_name VARCHAR2(100)
);
```



Lets us show you a quick Demo

Why? What Next?

- Benefits:
  - Streamlined data management
  - Supports entertainment industry analytics
  - Scalable and flexible solution

- Future Directions:
  - Real-time data analytics integration
  - The Database could evolve to include streaming metrics, global audience data
    or real-time social media feedback. (ex. building a database for a site like Letterboxd)