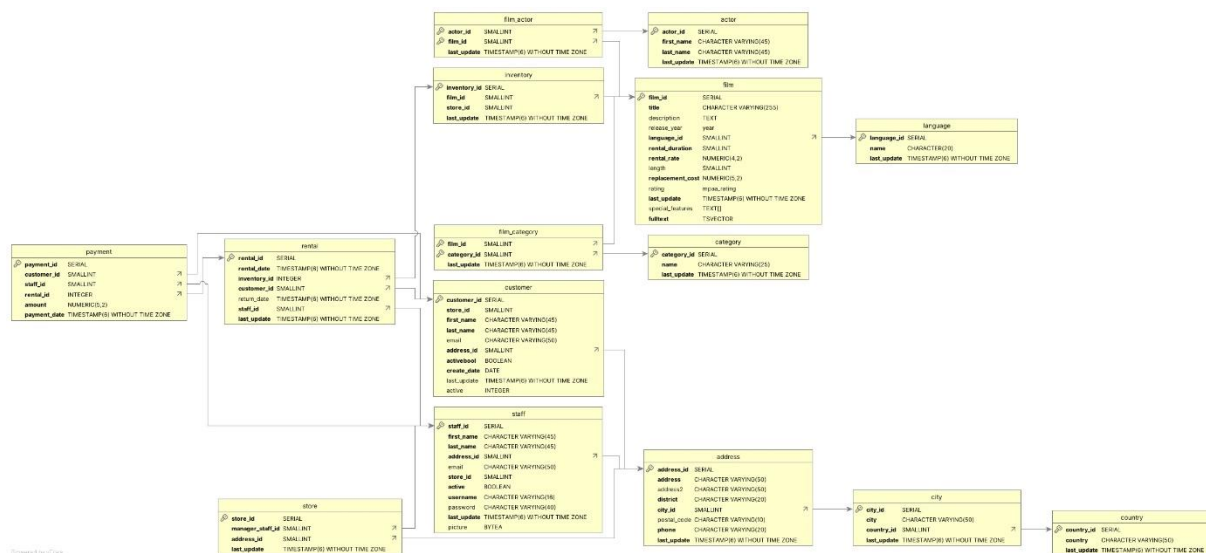# *3.2: Data Storage & Structure*

# Answers 3.2

❖ **Step 2. Extract the ERD**



❖ **Step 3. Create the first draft of a data dictionary**

- Take a moment to examine your ERD. Does the Rockbuster database have a snowflake schema or a star schema? Write a brief explanation for your answer
  - The Rockbuster database uses a **snowflake schema** because: It normalizes data into multiple related dimension tables, fact tables like rental and payment reference these normalized dimension tables through foreign keys.

- List all the fact tables and all the dimension tables in the schema. For each table, list every column and its data type, and write a brief description of the column. To get an idea of what this should look like, check out these example fact and dimension tables.

# Fact Tables

## rental

| Columns | Data Type | Description |
| --- | --- | --- |
| rental_id | SERIAL | Unique identifier for the rental transaction. |
| rental_date | TIMESTAMP | Date and time when the rental started. |
| inventory_id | INTEGER | Identifier for the rented film (from the `inventory` table). |
| customer_id | SMALLINT | Identifier for the customer renting the film. |
| return_date | TIMESTAMP | Date and time when the rental was returned. |
| staff_id | SMALLINT | Identifier for the staff member handling the rental. |
| last_update | TIMESTAMP | Timestamp of the last update to this record. |

## payment

| Columns | Data Type | Description |
| --- | --- | --- |
| payment_id | SERIAL | Unique identifier for the payment transaction. |
| customer_id | SMALLINT | Identifier for the customer making the payment. |
| staff_id | SMALLINT | Identifier for the staff handling the payment. |
| rental_id | INTEGER | Identifier for the related rental transaction. |
| amount | NUMERIC(5,2) | Amount paid for the rental. |
| payment_date | TIMESTAMP | Date and time when the payment was made. |

# Dimension Tables

## customer

| Columns | Data Type | Description |
| --- | --- | --- |
| customer_id | SERIAL | Unique identifier for each customer. |
| store_id | SMALLINT | Identifier for the store serving the customer. |
| first_name | CHARACTER VARYING(45) | Customer's first name. |
| last_name | CHARACTER VARYING(45) | Customer's last name. |
| email | CHARACTER VARYING(50) | Customer's email address. |
| address_id | SMALLINT | Identifier for the customer's address. |
| activebool | BOOLEAN | Indicates if the customer is active (true/false). |
| create_date | DATE | Date the customer record was created. |
| last_update | TIMESTAMP | Timestamp of the last update to the record. |

## film

| Columns | Data Type | Description |
| --- | --- | --- |
| film_id | SERIAL | Unique identifier for each film. |
| title | CHARACTER VARYING(255) | Title of the film. |
| description | TEXT | Short description of the film. |
| release_year | YEAR | Year the film was released. |
| rental_duration | SMALLINT | Number of days for rental duration. |
| rental_rate | NUMERIC(4,2) | Cost of renting the film. |
| length | SMALLINT | Duration of the film in minutes. |
| replacement_cost | NUMERIC(5,2) | Cost to replace the film. |
| rating | mapa_rating | Rating of the film (e.g., PG, R, etc.). |
| last_update | TIMESTAMP | Timestamp of the last update to the record. |

## inventory

| Columns | Data Type | Description |
| --- | --- | --- |
| inventory_id | SERIAL | Unique identifier for each inventory item. |
| film_id | SMALLINT | Identifier for the film in the inventory. |
| store_id | SMALLINT | Identifier for the store where the film is available. |
| last_update | TIMESTAMP | Timestamp of the last update to the record. |

## store

| Columns | Data Type | Description |
| --- | --- | --- |
| store_id | SERIAL | Unique identifier for each store. |
| manager_staff_id | SMALLINT | Identifier for the store's manager. |
| address_id | SMALLINT | Identifier for the store's address. |
| last_update | TIMESTAMP | Timestamp of the last update to the record. |

## staff

| Columns | Data Type | Description |
| --- | --- | --- |
| staff_id | SERIAL | Unique identifier for each staff member. |
| first_name | CHARACTER VARYING(45) | Staff member's first name. |
| last_name | CHARACTER VARYING(45) | Staff member's last name. |
| store_id | SMALLINT | Identifier for the store where the staff works. |
| active | BOOLEAN | Indicates if the staff member is active. |
| username | CHARACTER VARYING(16) | Staff member's system username. |
| last_update | TIMESTAMP | Timestamp of the last update to the record. |

## ❖ Step 4. Find information:

- Which actors brought Rockbuster the most revenue?

| | actor_id [PK] integer | actor_name text | total_revenue numeric |
|---|---|---|---|
| 1 | 107 | Gina Degeneres | 3129.17 |
| 2 | 181 | Matthew Carrey | 2543.78 |
| 3 | 198 | Mary Keitel | 2426.92 |
| 4 | 81 | Scarlett Damon | 2403.81 |
| 5 | 102 | Walter Torn | 2403.18 |
| 6 | 60 | Henry Berry | 2392.36 |
| 7 | 58 | Christian Akroyd | 2378.97 |
| 8 | 144 | Angela Witherspoon | 2357.11 |
| 9 | 111 | Cameron Zellweger | 2322.94 |
| 10 | 28 | Woody Hoffman | 2315.92 |

```sql
SELECT
    a.actor_id,
    a.first_name || ' ' || a.last_name AS actor_name,
    SUM(p.amount) AS total_revenue
FROM
    payment AS p
JOIN
    rental AS r ON p.rental_id = r.rental_id
JOIN
    inventory AS i ON r.inventory_id = i.inventory_id
JOIN
    film_actor AS fa ON i.film_id = fa.film_id
JOIN
    actor AS a ON fa.actor_id = a.actor_id
GROUP BY
    a.actor_id, actor_name
ORDER BY
    total_revenue DESC
LIMIT 10;
```

- What language are the majority of movies in the collection?

| | language<br>character (20) | movie_count<br>bigint |
|---|---|---|
| 1 | English | 1000 |

Query    Query History

```sql
 1  SELECT
 2      l.name AS language,
 3      COUNT(f.film_id) AS movie_count
 4  FROM
 5      film AS f
 6  JOIN
 7      language AS l ON f.language_id = l.language_id
 8  GROUP BY
 9      l.name
10  ORDER BY
11      movie_count DESC
12  LIMIT 1;
13
```