Task 3.2 Data Storage & Structure

By Lee Heng Chuah

You want to understand all the relationships that exist within the Rockbuster database and need to get an overview of all the tables. Your supervisor hasn’t given you an entity relationship diagram (ERD), so you decide to extract one using DBVisualiser or Lucidchart. You will need the diagram to locate all the necessary information quickly and efficiently.

Directions:

Step 1

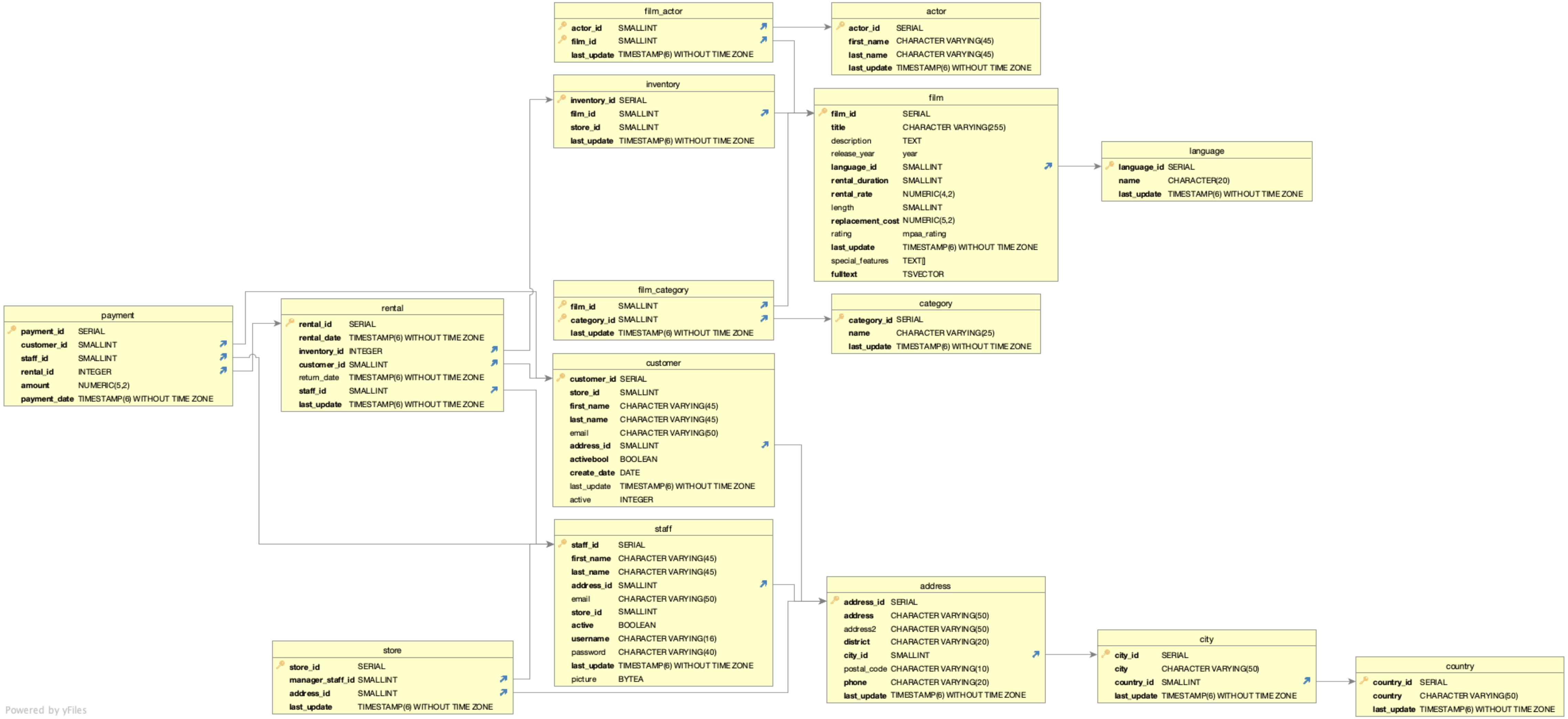
Create a new text document and call it “Answers 3.2”. You will save a copy of your ERD, data dictionary and written answers in this document. – Completed.

Step 2

- Download and install DBVisualiser and Lucidchart (If you haven’t already done so). – Completed.
- Extract the ERD from the Rockbuster database and save it as an image (PNG or JPEG) using the instructions in the Exercise.
- Copy-paste the ERD into your answers document.

  Tip: If the ERD image you save is blurry, try zooming in on the ERD in DBVisualiser or Lucidchart, then download it again.

  What to do if DBVisualiser doesn’t work? In some cases there is a bug in DBVisualiser where the connection between it and PgAdmin breaks and DBVisualiser can’t access the database. Since no solution for this bug is known for now, you can use Lucidchart as an alternative tool.



film_actor

actor_id	SMALLINT
film_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

actor

actor_id	SERIAL
first_name	CHARACTER VARYING(45)
last_name	CHARACTER VARYING(45)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

inventory

inventory_id	SERIAL
film_id	SMALLINT
store_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

film

film_id	SERIAL
title	CHARACTER VARYING(255)
description	TEXT
release_year	year
language_id	SMALLINT
rental_duration	SMALLINT
rental_rate	NUMERIC(4,2)
length	SMALLINT
replacement_cost	NUMERIC(5,2)
rating	mpaa_rating
last_update	TIMESTAMP(6) WITHOUT TIME ZONE
special_features	TEXT[]
fulltext	TSVECTOR

language

language_id	SERIAL
name	CHARACTER(20)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

film_category

film_id	SMALLINT
category_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

category

category_id	SERIAL
name	CHARACTER VARYING(25)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

payment

payment_id	SERIAL
customer_id	SMALLINT
staff_id	SMALLINT
rental_id	INTEGER
amount	NUMERIC(5,2)
payment_date	TIMESTAMP(6) WITHOUT TIME ZONE

rental

rental_id	SERIAL
rental_date	TIMESTAMP(6) WITHOUT TIME ZONE
inventory_id	INTEGER
customer_id	SMALLINT
return_date	TIMESTAMP(6) WITHOUT TIME ZONE
staff_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

customer

customer_id	SERIAL
store_id	SMALLINT
first_name	CHARACTER VARYING(45)
last_name	CHARACTER VARYING(45)
email	CHARACTER VARYING(50)
address_id	SMALLINT
activebool	BOOLEAN
create_date	DATE
last_update	TIMESTAMP(6) WITHOUT TIME ZONE
active	INTEGER

staff

staff_id	SERIAL
first_name	CHARACTER VARYING(45)
last_name	CHARACTER VARYING(45)
address_id	SMALLINT
email	CHARACTER VARYING(50)
store_id	SMALLINT
active	BOOLEAN
username	CHARACTER VARYING(16)
password	CHARACTER VARYING(40)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE
picture	BYTEA

address

address_id	SERIAL
address	CHARACTER VARYING(50)
address2	CHARACTER VARYING(50)
district	CHARACTER VARYING(20)
city_id	SMALLINT
postal_code	CHARACTER VARYING(10)
phone	CHARACTER VARYING(20)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

city

city_id	SERIAL
city	CHARACTER VARYING(50)
country_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

country

country_id	SERIAL
country	CHARACTER VARYING(50)
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

store

store_id	SERIAL
manager_staff_id	SMALLINT
address_id	SMALLINT
last_update	TIMESTAMP(6) WITHOUT TIME ZONE

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 start schema? Write a brief explanation for your answer. → Snowflake schema. The fact tables are in the centre surrounded by multi-dimension tables, and sub-dimensions tables.
- 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

rental

Columns	Data Type	Description
rental_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for rental
rental_date	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for rental but without time zone
inventory_id	INTEGER	4-byte integer that has range from -2,147,483,648 to 2,147,483,647, store the rental inventory id
customer_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store customer id.
return_date	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the rental-return but without time zone
staff_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store staff id.
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the rental information, e.g., rental date, inventory id, customer id, return date and staff id but without time zone

DIMENSIONS

payment

Columns	Data Type	Description
payment_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for payment
customer_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store customer id.
staff_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store staff id.
rental_id	INTEGER	4-byte integer that has range from -2,147,483,648 to 2,147,483,647, store the rental id
amount	NUMERIC(5,2)	Real/Exact number with 5 digits with 2 number after the decimal point – store the exact payment amount
payment_date	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the payment information (data entry) e.g., customer id, staff id, rental id, and amount of payment but without time zone

film_actor

Columns	Data Type	Description
actor_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store actor id.
film_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store film id.
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the film id and actor id but without time zone

inventory

Columns	Data Type	Description
inventory_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for inventory
film_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store film id.
store_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, state store id.
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the inventory information e.g., film id and store id but without time zone

film_category

Columns	Data Type	Description
film_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store film id.
category_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, state category id.
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the film category but without time zone

customer

Columns	Data Type	Description
customer_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for customer
store_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, state store id.
first_name	CHARACTER VARYING (45)	Variable length character string store up to 45 character – customer first name
last_name	CHARACTER VARYING (45)	Variable length character string store up to 45 character – customer last name
email	CHARACTER VARYING (50)	Variable length character string store up to 50 character – customer email address
address_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store address id.
activebool	BOOLEAN	Hold one of three possible values: true, false or null.
create_date	DATE	Store the date of creation customer information.
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the customer information but without time zone
active	INTEGER	4-byte integer that has range from -2,147,483,648 to 2,147,483,647, store customer active or non-active

staff

Columns	Data Type	Description
staff_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for staff
first_name	CHARACTER VARYING (45)	Variable length character string store up to 45 character – customer first name
last_name	CHARACTER VARYING (45)	Variable length character string store up to 45 character – customer last name

address_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store address id.
email	CHARACTER VARYING (50)	Variable length character string store up to 50 character – customer email address
store_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, state store id.
active	BOOLEAN	Hold one of three possible values: true, false or null.
username	CHARACTER VARYING (16)	Variable length character string store up to 16 character – username
password	CHARACTER VARYING (40)	Variable length character string store up to 40 character – password
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the staff information but without time zone
picture	BYTEA	Picture of employee

actor

Columns	Data Type	Description
actor_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for actor
first_name	CHARACTER VARYING (45)	Variable length character string store up to 45 character – customer first name
last_name	CHARACTER VARYING (45)	Variable length character string store up to 45 character – customer last name
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the actor information but without time zone

film

Columns	Data Type	Description
film_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for film
title	CHARACTER VARYING (255)	Variable length character string store up to 255 character – to describe the title of film
description	TEXT	Variable-length character string, with unlimited length to describe the film
release_year	Year	Store the release year
language_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store language id.
rental_duration	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store rental duration

rental_rate	NUMERIC(4,2)	Real/Exact number with 4 digits with 2 number after the decimal point – store the exact rental rate
length	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, state length
replacement_cost	NUMERIC(5,2)	Real/Exact number with 5 digits with 2 number after the decimal point – store the exact replacement cost
rating	mpaa_rating	Store the film rating
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the film information but without time zone
special_features	TEXT[]	Variable-length character string, with unlimited length to describe the film
fulltext	TSVECTOR	A full text describes the film

category

Columns	Data Type	Description
category_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for category
name	CHARACTER VARYING (25)	Variable length character string store up to 25 character – to describe the name of category
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the category information but without time zone

address

Columns	Data Type	Description
address_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for address
address	CHARACTER VARYING (50)	Variable length character string store up to 50 character – to store the address
address2	CHARACTER VARYING (50)	Variable length character string store up to 50 character – to store the address
district	CHARACTER VARYING (20)	Variable length character string store up to 20 character – to store the district
city_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store city id.
postal_code	CHARACTER VARYING (10)	Variable length character string store up to 10 character – to store the postal code
phone	CHARACTER VARYING (20)	Variable length character string store up to 20 character – to store the phone number

last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the address information but without time zone

language

Columns	Data Type	Description
language_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for language
name	CHARACTER(20)	Fixed length character with space padded, store the name of language
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the language name but without time zone

city

Columns	Data Type	Description
city_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for city
city	CHARACTER VARYING (50)	Variable length character string store up to 50 character – to store the city name
country_id	SMALLINT	2 byte signed integer that has a range from -32,768 to 32,767, store country id.
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the city name but without time zone

country

Columns	Data Type	Description
country_id	SERIAL	Same as integer, PostgreSQL automatically general populate value into serial column for country
country	CHARACTER VARYING (50)	Variable length character string store up to 50 character – to store the country name
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	Store both date and time values for the latest update (data entry) on the country name but without time zone

- If a column name doesn't tell you enough to write a description, you can also view the tables in PgAdmin 4. The SQL syntax for selecting table is SELECT * FROM table_name. So SELECT * FROM film would return the film table, for example.

Step 4 – Find Information

Now that your data dictionary and ERD are ready to use, your manager has given you a list of business questions to answer. Use your data dictionary to figure out which tables you'd need to answer the questions below:

- Which actor brought Rockbuster the most revenue? – Table required: actor, film_actor, payment (would know the total payment for the rental), inventory (would know the total remain in store, and how many rent out)
- What language are the majority of movies in the collection? – Table required: language, film table

Step 5 – Save and upload your file

- Save your “Answers 3.2” document as a PDF and upload it here for your tutor to review. Make sure to keep a copy for yourself. You will be finalising your data dictionary at the end of this Achievement.