TABLE OF CONTENTS

Assignment 01 – The GMG Movie Database Design

Your database development team (comprising of 2-3 members) has been commissioned to design a set of database tables that will be used to track information for GMG, a movie production studio. The database should track the following information:

- The names of the movies.
- The year which a movie was produced.
- The rating for the movie. (e.g. G, PG, PG-13, r, etc.)
- The first and last names of the producer for each movie. (assume that there is only one producer per movie)
- The first and last names of each actor in each movie.
- Keep track of the main and supporting actors in a movie.
- The amount of money each actor was paid for making the movie.
- The names and addresses of the theatres where each movie was shown. (there can be many theatres, possibly thousands, where each movie was shown)
- The number of tickets sold for each movie at each theatre.
- The price per ticket at each theatre - for the purpose of this assignment you should assume that a theatre charges the same amount of money for every ticket that it sells.

There are many different database designs that can be used to store this information. Whatever design you choose, make sure that it is in **3rd Normal Form**. Create the appropriate amount of tables and the optimum number of fields in those tables. Feel free to create extra fields to hold primary key values if you wish.

Tasks:
Model a relational database using a(n):

a. **Entity Relationship Diagram** (50 marks), and
b. **Data Dictionary** (50 marks)
that adequately contains the relevant amount of information needed to help database programmer build tables within the database.

A sample format for a data dictionary is shown below. Feel free to use either this format or any other relevant format as a reference for your own data dictionary.

ORDER_ITEM				
Attribute name	Data type	Size	Constraint	Comment
Ord_no	Number		Not null	FK to Order
Prod_code	Varchar2	10		FK to Product
Qty	Number		>0	
Price	Number		>0	Price of product
Total	Number		Check	total = price * quantity
Qty_filled	Char	1	Y or N default to N	Has the product been delivered?
Ship_id	Number		>0	FK to Cust_Ship

[Total: 100 marks]

Task a: Entity Relationship Diagram

NORMALIZATION PROCESS

- **Unnormalized Normal Form (UNF)**

- The Movie table is in unnormalized form and suffers from anomalies.

Movie (movie_id {PK}, movie_name, year_produced, genre, runtime, rating, producer_id, producer_firstname, producer_lastname, producer_gender, (actor_id, actor_firstname, actor_lastname, actor_gender, role, payment)*, (theatre_id, theatre_name, theatre_status, price_per_ticket, address_unit, street, city, postcode, state, country, quantity_sold, return_sales)*)

- **First Normal Form (1NF)**

- Repeating groups are identified, removed, and appropriately placed into new tables.
- The Composite Key of the new tables are determined.
- The number “1” is appended to the table names to indicate 1NF.

Movie-1 (movie_id {PK}, movie_name, year_produced, genre, runtime, rating, producer_id, producer_firstname, producer_lastname, producer_gender)

CastMember-1 (movie_id {PPK}, actor_id {PPK}, actor_firstname, actor_lastname, actor_gender, role, payment)

TicketSale-1 (movie_id {PPK}, theatre_id {PPK}, theatre_name, theatre_status, price_per_ticket, address_unit, street, city, postcode, state, country, quantity_sold, return_sales)

- **Second Normal Form (2NF)**

- Attributes that are partially functionally dependent on the Composite Keys in the previously added tables are identified, removed, and appropriately placed into new tables.
- The Primary Key of the new tables are determined.
- The number “2” is appended to the table names to indicate 2NF.

Movie-2 (movie_id {PK}, movie_name, year_produced, genre, runtime, rating, producer_id, producer_firstname, producer_lastname, producer_gender)

CastMember-2 (movie_id {PPK}, actor_id {PPK}, role, payment)

Actor-2 (actor_id {PK}, actor_firstname, actor_lastname, actor_gender)

TicketSale-2 (movie_id {PPK}, theatre_id {PPK}, quantity_sold, return_sales)

Theatre-2 (theatre_id {PK}, theatre_name, theatre_status, price_per_ticket, address_unit, street, city, postcode, state, country)

- **Third Normal Form (3NF)**

- The Movie table has transitive dependencies, thus attributes that are transitively functionally dependent on non-key attributes are identified, removed, and placed into a new table.
- The Primary Key of the new table is determined.
- The number “3” is appended to the table names to indicate 3NF.

Movie-3 (movie_id {PK}, movie_name, year_produced, genre, runtime, rating, producer_id {FK})

Producer-3 (producer_id {PK}, producer_firstname, producer_lastname, producer_gender)

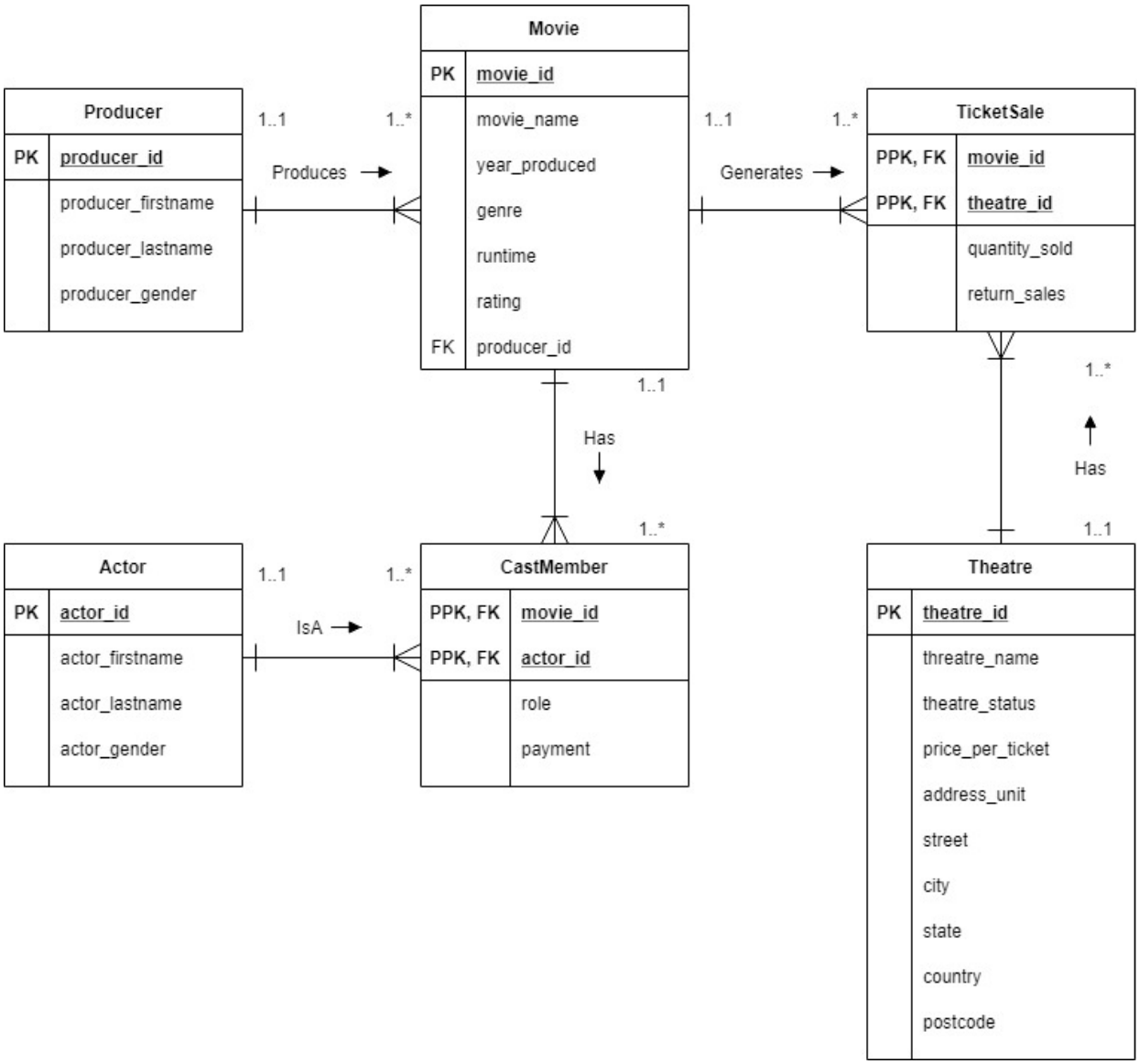
CastMember-3 (movie_id {PPK}, actor_id {PPK}, role, payment)

Actor-3 (actor_id {PK}, actor_firstname, actor_lastname, actor_gender)

TicketSale-3 (movie_id {PPK}, theatre_id {PPK}, quantity_sold, return_sales)

Theatre-3 (theatre_id {PK}, theatre_name, theatre_status, price_per_ticket, address_unit, street, city, postcode, state, country)

ENTITY RELATIONSHIP DIAGRAM



Task b: Data Dictionary

Movie				
Attribute name	Data type	Size	Constraint	Comment
movie_id	VARCHAR	6	Primary key	PK of Movie
movie_name	VARCHAR	100	NOT NULL	Full name of movie
year_produced	NUMBER	4	>0	Year of movie production
genre	VARCHAR	15		Main genre of movie in 1 word
runtime	NUMBER	3	NOT NULL	Full length of movie in minutes
rating	VARCHAR	5	G, PG, PG-13, R, NC-18, U, P13 or 18, default to U	Rating for the movie
producer_id	NUMBER	6	Foreign key	FK to Producer, ID of the producer of the movie

Producer				
Attribute name	Data type	Size	Constraint	Comment
producer_id	VARCHAR	6	Primary key	PK of Producer
producer_firstname	VARCHAR	50	NOT NULL	First name of producer
producer_lastname	VARCHAR	50	NOT NULL	Last name of producer
producer_gender	CHAR	1	M, F, or O, default to O	Gender of producer; M for Male, F for Female, O for Other

Actor				
Attribute name	Data type	Size	Constraint	Comment
actor_id	VARCHAR	6	Primary key	PK of Actor
actor_firstname	VARCHAR	50	NOT NULL	First name of actor
actor_lastname	VARCHAR	50	NOT NULL	Last name of actor
actor_gender	CHAR	1	M, F, or O, default to O	Gender of actor; M for Male, F for Female, O for Other

CastMember				
Attribute name	Data type	Size	Constraint	Comment
actor_id	VARCHAR	6	Primary key, Foreign Key	PPK of CastMember, FK to Actor
movie_id	VARCHAR	6	Primary key, Foreign Key	PPK of CastMember, FK to Movie
role	CHAR	1	M or S, default to S	Role of the actor in the movie; M for Main, S for Supporting
payment	NUMBER	9	>0	Money paid to actor for their role in the movie

TicketSale				
Attribute name	Data type	Size	Constraint	Comment
movie_id	VARCHAR	6	Primary key, Foreign Key	PPK of TicketSale, FK to Movie
theatre_id	VARCHAR	6	Primary key, Foreign Key	PPK of TicketSale, FK to Theatre
quantity_sold	NUMBER		NOT NULL	Total quantity of tickets sold
return_sales	NUMBER		NOT NULL	Total amount of money earned from ticket sales

Theatre				
Attribute name	Data type	Size	Constraint	Comment
theatre_id	VARCHAR	6	Primary key	PK of Theatre
theatre_name	VARCHAR	30	NOT NULL	Full name of theatre
theatre_status	CHAR	1	Y, N, or U default to U	Is the theatre still operating? Y for Yes, N for No, U for Unknown
price_per_ticket	NUMBER		>0	Price of each ticket sold at the theatre
address_unit	VARCHAR	10	NOT NULL	Unit number of address
street	VARCHAR	50	NOT NULL	Street address of theatre
city	VARCHAR	30	NOT NULL	City of which the theatre is located
postcode	VARCHAR	10	NOT NULL	Postcode of the theatre address
state	VARCHAR	30	NOT NULL	State of which the theatre is located
country	VARCHAR	30	NOT NULL	Country of which the theatre is located