UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

SEMESTER II-2024/2025

BITP1323 DATABASE

PROJECT FINAL REPORT

Course Learning Outcome (CLO): CLO 2 & CLO 3

Group No : **13**

Group Name : **ISG**

Project Title : **Costume Rental System**

Member Names:

No	Matric No	Full Name	Lab Session	Role
1	B032420168	THUNAYAN TURKI OMAR SHAMLAN	S1G2	Project Manager (Leader)
2	B032420175	MUSTAFA AHMED ABDELMAHMOUD AHMED	S1G2	Project Member
3	B032420170	OSAZUWA PHILIP CHUKWUKA	S1G2	Project Member
4	B032420176	NADA AHMED ABDELMAHMOUD AHMED	S1G2	Project Member

For Lecturer's Use Only:

Part A: Marks for all Members

Components			Maximum Mark	Given Mark
Proposal	Introduction		**10**	
	Problem Statements			
	Objectives			
	Scope of project			
	Expected Outcome / Proposed Solution			
Final Report	Chapter 1 Introduction	1.1 Introduction	**10**	
		1.2 Problem Statements		
		1.3 Objectives		
		1.4 Scope of project		
		1.5 Expected Outcome / Proposed Solution		
	Chapter 2 Data Collection	2.1 Introduction 2.2 Fact-finding Description 2.3 Entities Description 2.4 Attributes Description 2.5 Data Entries Description 2.6 INSERT, UPDATE, DELETE Operation	**20**	
	Chapter 3 Conceptual & Logical Design	3.1 Introduction	**2**	
		3.2 Entity Relationships Diagram	**10**	
		3.3 Business Rules	**10**	
	Chapter 4 Physical Design	4.1 Introduction	**2**	
		4.2 Data Definition Language	**10**	
		4.3 Data Manipulation Language	**10**	
		4.4 Structured Query Language (SQL) & Relational Algebra (RA)	**10**	
		4.5 Conclusion	**6**	
		Total Mark	**100**	

Part B: Presentation

Write member names only		Maximum Mark	Given Mark
Presentation	Teamwork Skill (TS) – Recorded video & upload into YouTube. – Submit YouTube link only. Paste in report at appendix page. – Design a poster (must contain **introduction, problem statement, objective, Normalized ERD, scope, expected outcome, example TWO of RA & SQL, TWO of DDL, TWO of DML, group no, group name, members' name**) – Length video duration must less than 10 minutes.	**20**	
Final Report & Poster Format		**10**	

Part C: Peer Evaluation (*will be given later*)

Write member names only		Maximum Mark
Peer Evaluation	Group Leader:	**5**
	Member 2:	
	Member 3:	
	Member 4:	
	Member 5:	
	Total Mark	**5**

Total Marks (Part A + (Part B + format) + Part C)

Member Names	Maximum Mark	Given Mark	Maximum Mark (%)	Given Mark (%)
Member 1:	**135**		**20%**	
Member 2:				
Member 3:				
Member 4:				
Member 5:				

Table of Content

Chapter 1

INTRODUCTION

1.1 Introduction

The Costume Rental System is a proposed database solution designed to support the management of costume rentals for events such as theatrical performances, cultural festivals, school functions, and cosplay. The goal is to replace inefficient manual methods, such as paper-based and unstructured digital records, with a centralized relational database. Small to mid-sized costume rental services often struggle with issues such as duplicate bookings, loss of customer records, and poor inventory tracking. These inefficiencies affect overall service delivery and customer satisfaction. To address these challenges, the project will focus on the design of an Entity-Relationship (ER) model that defines the key entities involved in a costume rental business and their relationships. The ER model will help organize data related to costumes, rentals, staff, and customers, ensuring efficient data management and consistency. The final output will be a well-structured Entity-Relationship Diagram (ERD), which serves as the foundational blueprint for database implementation in the future.

1.2 Problem Statement

- Existing system has limited features such as manual record-keeping and lack of real-time inventory tracking, which has caused booking errors, inefficient service delivery and reduced customer satisfaction.
- There is no automated reporting system, which has made report generation time-consuming and often inaccurate.

1.3 Objectives

- This Entity-Relationship (ER) model is to propose a structured design and relational database for a Costume Rental System that efficiently manages core operations such as customer registration, costume inventory, rental transactions, staff activity, and category classification.
- To implement defining key entities such as Customer, Costume, Category, Rental, and Staff along with their relationships, the model ensures accurate tracking of rentals, real-time costume availability, and secure management of customer and staff information.
- To ensure the ER model provides a reliable structure for developing a future database system that supports efficient operations and accurate data retrieval.

1.4 Scope of Project

a. Target Users:

- **Admin (System Administrator):**The Admin is responsible for managing the entire Costume Rental System, ensuring smooth operations, data accuracy, and system security.
- **Customers:** Customers are the primary users of the system, renting costumes for personal or organizational use.

b. Proposed Modules

Proposed Modules refer to the individual functional components or sections of a system that are planned for development to achieve the objectives of the project. Each module is designed to handle specific tasks or operations within the system such as:

- Costume Inventory Management Module
- Customer Management Module
- Rental Management Module
- Payment and Billing Module

1.5 Expected Outcome / Proposed Solution

The outcome of this project will be an ERD MODEL designed to simplify the process of renting theatrical and costume wear. It will include:

- An easy-to-use user interface for browsing costumes, searching, and booking.
- A backend management system for processing orders and tracking inventory.
- A database containing details of costumes, users, and rentals.
- A flexible and attractive design that ensures a smooth user experience.

Thus, the result will be a fully functional online costume rental system, ready for real-world application and further development.

Chapter 2

DATA COLLECTION

2.1 Introduction

When creating our Costume Rental System, we had to first reflect on real rental businesses to analyse how they functioned. We started collecting research by physically visiting a shop, we also had a group brainstorming to generate ideas, and online research of rental business websites. In doing so we were able to identify what our system should add, such as the customers profile, costume information, rentals, payments, etc. We were then able to breakdown the major entities including their attributes and data routes for our database

2.2 Fact-finding Description

a. Observation

- We visited a local costume rental shop and quietly watched how things worked.
- Noticed delays because staff had to check costume availability manually.
- All records were on paper or Excel — no proper digital system.
- Customers were mostly students, teachers, and cosplay fans.

b. Brainstorming

- We held a group session where everyone shared ideas one by one.
- No judging — just open discussion.
- Great ideas came up like auto-checking costume stock, tracking rentals, and offering student discounts.

c. Website Research

- We looked at other costume rental websites to learn what works.
- Found helpful features like filters by size or theme, online booking, and photo previews.
- Read user reviews to understand what people liked or disliked.

2.3 Entities Description

- **CUSTOMER:** Manages customer registration and stores customer details for rental tracking.
- **COSTUME:** Represents individual costumes in the inventory, including details for availability and rental pricing.
- **RENTAL:** Records transaction details for each costume rental, including customer and rental timelines.
- **RENTAL_DETAILS:** Provides detailed information about each item in a rental transaction.
- **PAYMENT:** Manages financial transactions related to rental payments, ensuring accurate tracking of amounts, methods, and statuses.

2.4 Attributes Description

- CUSTOMER:

Customer_ID(PK), First_Name, Last_Name, Email, Phone_Number, Address.

- COSTUME:

Costume_ID (PK), Name, Costume_Size, Price_Per_Day, Costume_Status.

- RENTAL:

Rental_ID (PK), Rental_Date, Return_Date, Total_Price, Customer_ID (FK).

- RENTAL_DETAILS:

Customer_ID (PK, FK2), Rental_ID (PK, FK1), Quantity.

- PAYMENT:

Payment_ID (FK), Payment_Date, Payment_Method, Payment_Status, Rental_ID (FK).

2.5 Data Entries Description

a. CUSTOMER:

Attributes:

- **Customer_ID (PK):** The format must start with “C” and the max size length of the character will be 10, Ex: C000000001.
- **First_Name:**
- **Last_Name:**
- **Email:** valid email format. Ex: Name@email.com.
- **Phone_Number:** the format must start with the country's code the max size length of character will be 10, Ex: +60 111111111.
- **Address:**

b. COSTUME:

Attributes**:**

- **Costume_ID (PK):** The format must start with “CS” max size length 10-character, Ex: CO00000001.
- **Name:**
- **Costume_Size:** format should include with “size” , Ex: S,M,L.
- **Price_Per_Day:**
- **Costume_Status:** Format must include Status, Ex: (available, rented, cleaning).

c. RENTAL:

Attributes**:**

- **Rental_ID (PK):** Format must start with ”R” mix size length 10 characters. Ex: R000000001.
- **Rental_Date:** format should start with day/month/year. Ex:1/05/2025.
- **Return_Date:** format should start with day/month/year. Ex:1/05/2025.
- **Total_Price:**
- **Customer_ID (FK):** referencing Customer(Customer_ID).

d. RENTAL_DETAILS:

Attributes:

- **Costume_ID (PK, FK2).**Composite primary key & foreign key from (Costume)
- **Rental_ID (PK, FK1).**Composite primary key & foreign key from (Rental)
- **Quantity:**Represents how many of each costume is rented (e.g., 1, 2, 3…)

e. PAYMENT:

Attributes**:**

- **PaymentID (Primary Key):** The Format must start with” P” mix size length 10 characters. Ex: R000000001.
- **PaymentDate:** format should start with day/month/year. Ex:1/05/2025.
- **PaymentMethod:** the format is the method of payment used. Ex: (cash, credit card). **PaymentStatus:** the format must include Status, Ex (completed, pending, failed).
- **RentalID (Foreign Key):** referencing Rental(Rental_ID).

2.6 INSERT, UPDATE, DELETE Operation

a. CUSTOMER

- **INSERT:** Add new customer records during registration.
- **UPDATE:** Modify customer details such as contact information, address, or membership status.
- **DELETE:** Remove a customer record if they are no longer active.

b. COSTUME

- **INSERT**: Add new costumes to the inventory.
- **UPDATE**: Update details such as condition status, stock quantity, or rental price.
- **DELETE**: Remove costumes from the inventory.

c. RENTAL

- **INSERT**: Create new rental transactions when a customer rents a costume.
- **UPDATE**: Update rental details such as return date, total price, or status.
- **DELETE**: Remove rental records

d. RENTAL_DETAILS

- **INSERT**: Add details of costumes included in a rental.
- **UPDATE**: change the quantity or return status of a specific costume in a rental.
- **DELETE**: Remove a specific costume from a rental record if it was cancelled or entered by mistake.

e. PAYMENT

- **INSERT**: Record new payment transactions linked to rentals.
- **UPDATE**: Modify payment details such as payment status.
- **DELETE**: Remove payment records.

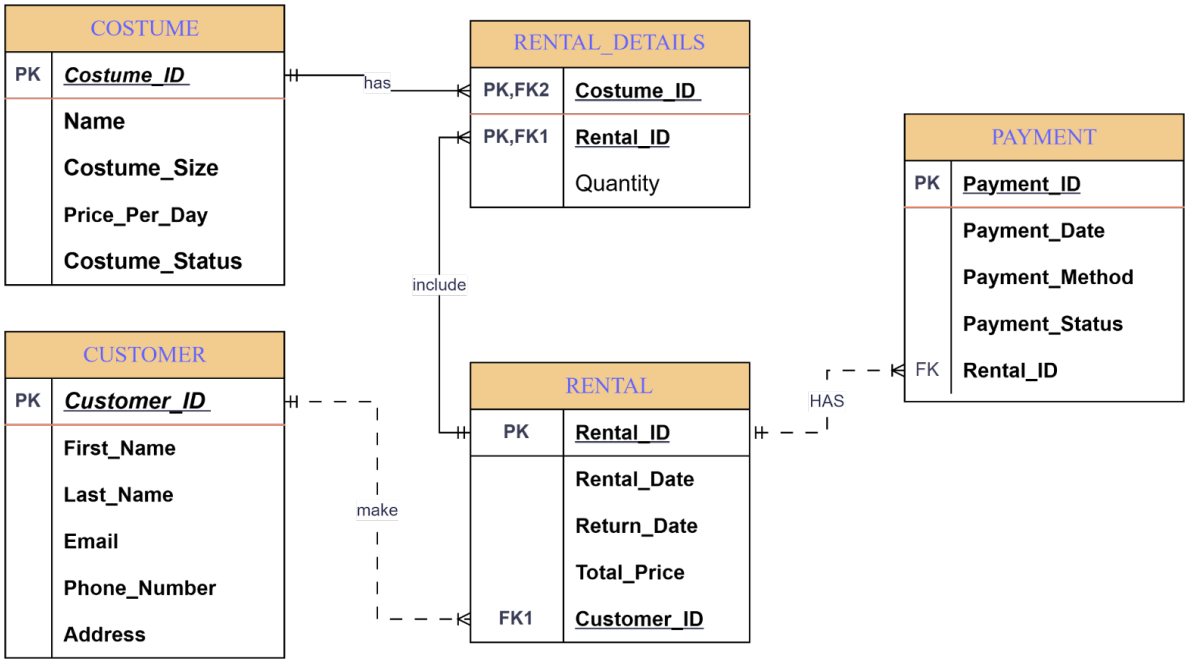
Chapter 3

CONCEPTUAL & LOGICAL DESIGN

3.1 Introduction

In this chapter we focus on the design of the database system by introducing the Entity-Relationship Diagram (ERD), which outlines the relationships between key entities such as customers, rentals, costumes, and payments. It also defines the business rules that govern data interactions, ensuring data integrity and consistency. This chapter provides a clear framework for building a reliable and scalable database system.

3.2 Entity Relationships Diagram



3.3 Business Rules

- Each customer can make one or more rentals, and each rental can be rented by only one customer.
- Each rental can include one or more costumes, and each costume can be rented multiple times, their connection is tracked in the Rental Details table, which must not contain duplicate entries of the same costume in a single rental.
- One rental can have one payment, and each payment belongs to only one rental.

Chapter 4

PHYSICAL DESIGN

4.1 Introduction

we focused on implementing the database structure using SQL. We created tables using Data Definition Language (DDL) to define the structure of entities such as Customer, Costume, Rental, Renta_Details, and Payment. Then, we used Data Manipulation Language (DML) to insert sample data into these tables. We also performed SQL queries and Relational Algebra (RA) operations to retrieve useful information, such as customer rental statistics and costume availability. This progress demonstrates how our database can store, manage, and retrieve data efficiently for real-world rental operations.

4.2 Data Definition Language

a. CUSTOMER Table

```
CREATE TABLE CUSTOMER
( Customer_ID    VARCHAR(10) PRIMARY KEY,
   First_Name  VARCHAR(15) NOT NULL,
   Last_Name   VARCHAR(15) NOT NULL,
   Email       VARCHAR(50)  NOT NULL,
   Phone_Number    NUMBER(15)  NOT NULL,
   Address         VARCHAR(100)    NOT NULL );
```

b. COSTUME Table

```
CREATE TABLE COSTUME
(  Costume_ID      VARCHAR(10)     PRIMARY KEY,
   Name         VARCHAR(20)     NOT NULL,
   Costume_Size       VARCHAR(10)     NOT NULL,
   Costume_Status   VARCHAR(20) NOT NULL,
 Price_Per_Day     NUMBER(20)  NOT NULL   );
```

c. RENTAL Table

```
CREATE TABLE RENTAL
(
   Rental_ID       VARCHAR(10)     PRIMARY KEY,
   Rental_Date     DATE        NOT NULL,
   Return_Date      DATE        NOT NULL,
   Total_Price     NUMBER(15)  NOT NULL,
   Customer_ID     REFERENCES CUSTOMER (Customer_ID)
);
```

d. RENTAL_DETAILS Table

```
CREATE  TABLE  RENTAL_DETAILS
(
  Customer_ID     REFERENCES CUSTOMER (Customer_ID),
   Rental_ID     REFERENCES RENTAL (Rental_ID),
   Quantity        NUMBER (10)      NOT NULL
);
```

e. PAYMENT TABLE

```
CREATE TABLE PAYMENT
(
   Payment_ID      VARCHAR(10)     PRIMARY KEY,
   Payment_Date    DATE        NOT NULL,
   Payment_Method  VARCHAR(20) NOT NULL,
   Payment_Status  VARCHAR(20) NOT NULL,
   Rental_ID       REFERENCES  RENTAL (Rental_
);
```

4.3 Data Manipulation Language

a. CUSTOMER Table

```
begin

INSERT INTO CUSTOMER

VALUES ('CU001', 'M,ustafa', 'Ahmed', 'mustafa.ahmed@gmail.com', 1234567890, '123 Nile Street');

INSERT INTO CUSTOMER

VALUES ('CU002', 'Nada', 'Ahmed', 'nada.ahmed@gmail.com', 9876543210, '456 Jasmine Road');

INSERT INTO CUSTOMER

VALUES ('CU003', 'Thunyan', 'Turki', 'thunyan.turki@gmail.com', 5555555555, '789 Desert Lane');

INSERT INTO CUSTOMER

VALUES ('CU004', 'Philip', 'Smith', 'philip.smith@gmail.com', 4444444444, '101 Elm Street');

end
```

b. COSTUME Table

```
begin

INSERT INTO COSTUME

VALUES ('C001', 'Superman Suit', 'M', 'Available', 50);

INSERT INTO COSTUME

VALUES ('C002', 'Batman Suit', 'L', 'Rented', 60);

INSERT INTO COSTUME

VALUES ('C003', 'Spiderman Suit', 'S', 'Available', 40);

end
```

c. RENTAL Table

```
begin
INSERT INTO RENTAL
VALUES ('R001', TO_DATE('2025-05-01', 'YYYY-MM-DD'), TO_DATE('2025-05-05', 'YYYY-MM-DD'), 200, 'CU001');
INSERT INTO RENTAL
VALUES ('R002', TO_DATE('2025-05-03', 'YYYY-MM-DD'), TO_DATE('2025-05-06', 'YYYY-MM-DD'), 180, 'CU002');
end
```

d. RENTAL_DETAILS Table

```
begin
INSERT INTO RENTAL_DETAILS
VALUES ('CU001', 'R001', 2);
INSERT INTO RENTAL_DETAILS
VALUES ('CU002', 'R002', 3);
end
```

e. PAYMENT TABLE

```
begin
INSERT INTO PAYMENT
VALUES ('P001', TO_DATE('2025-05-02', 'YYYY-MM-DD'), 'Credit Card', 'Completed', 'R001');
INSERT INTO PAYMENT
VALUES ('P002', TO_DATE('2025-05-04', 'YYYY-MM-DD'), 'Cash', 'Pending', 'R002');
end
```

USING UPDATE AND DELETE

- A costume with ID CT100 is now available for rent. Write an SQL statement to update its status to 'Available' in the COSTUME table

SQL

```
UPDATE COSTUME
SET Costume_Status = 'Available'
WHERE Costume_ID = 'CT100';
```

- A rental record with ID RENT202 is no longer needed. Write an SQL statement to delete it from the RENTAL table

SQL

```
DELETE FROM RENTAL
WHERE Rental_ID = 'RENT202';
```

4.4 Structured Query Language (SQL) & Relational Algebra (RA)

a) Write an SQL query to display first names, last names, and email addresses of all customers.

SQL

```
SELECT FIRST_NAME, LAST_NAME, EMAIL
FROM CUSTOMER;
```

RA

```
π(FIRST_NAME, LAST_NAME, EMAIL)(CUSTOMER)
```

b) Count the number of customers sharing each last name.

SQL

```
SELECT LAST_NAME, COUNT(*) AS Customer_Count
FROM CUSTOMER
GROUP BY LAST_NAME;
```

RA

```
ρR  ( " Customer_Count ") (π Last_Name)
Last_Name  ℑ  COUNT (*)  (CUSTOMER)
```

c) Display all customer IDs along with their corresponding rental IDs and quantities (assuming we need to join with a CUSTOMER table that contains customer details).

SQL

```
SELECT C. CUSTOMER_ID, R. RENTAL_ID, R. QUANTITY
FROM CUSTOMER C
JOIN RENTAL_DETAILS R
ON C.CUSTOMER_ID = R.CUSTOMER_ID;
```

RA

π C.CUSTOMER_ID **(CUSTOMER)** **⋈** C.CUSTOMER_ID = R.CUSTOMER_ID
(π R.RENTAL_ID, R.QUANTITY) **(RENTAL_DETAILS)**

4.5 Conclusion

Our project, the Costume Rental System, allowed us to apply what we learned in database design to a real-world situation. We started by collecting data about how costume rentals work and identified the problems in manual systems. Based on that, we designed an ERD with important entities like Customer, Costume, Rental, and Payment. After the design, we created the database using DDL statements and added sample data using DML. We also practiced writing SQL queries to get meaningful information from the system, such as rental summaries and costume availability. Overall, this project helped us understand how to design and build a functional database. It was a valuable experience that taught us not just technical skills, but also how to think critically, solve problems, and work as a team.

YouTube Link:

https://youtu.be/oCl16hCbJWs

Poster: