Mini project report on
Costume Rental Management System

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

UE21CS351A – DBMS Project

Submitted by:

PRASHIL HIMANSHU JATAKIYA–PES2UG21CS211

Kruthik K – PES2UG21CS243

Under the guidance of

Dr. Mannar Mannan J

Associate Professor

Designation: PES University

AUG - DEC 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

Costume Rental Management System

is a bonafide work carried out by

Prashil Himanshu Jatakiya	PES2UG21CS211
Kruthik K	PES2UG21CS243

In partial fulfilment for the completion of fifth semester DBMS Project (UE21CS351A) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2023 – DEC. 2023. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature
Dr. Mannar Mannan J
Associate Professor

DECLARATION

We hereby declare that the DBMS Project entitled Costume Rental Management System has been carried out by us under the guidance of Dr. Mannar Mannan, Associate Professor and submitted in partial fulfilment of the course requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering of PES University, Bengaluru during the academic semester AUG – DEC 2023.

Prashil Himanshu Jatakiya	PES2UG21CS211
Kruthik K	PES2UG21CS243

ACKNOWLEDGEMENT

I would like to express my gratitude to Dr. Mannar Mannan J, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE21CS351A - DBMS Project.

I take this opportunity to thank Dr. Sandesh B J, C, Professor,ChairPerson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this DBMS Project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

Ready-To-Wear , developed exclusively in Python using the PyMySQL library, serves as an all-in-one Ready-To-Wear Rental Management System for costumes, props, jewellery, and footwear. Leveraging MySQL as the backend database, it offers a seamless interface for adding, deleting, renting, returning, and viewing inventory items. This system efficiently manages inventory tracking, rental transactions, and customer profiles while ensuring data integrity and security. Its Python-based architecture with PyMySQL and MySQL integration provides a robust foundation for automating rental processes and enhancing operational efficiency in the entertainment and event industries.

TABLE OF CONTENTS

1. INTRODUCTION

The Ready-To-Wear Rental Management System for Costumes, Props, Footwear, and Jewellery is a comprehensive database project aimed at enhancing the operational efficiency of rental services, catering to customers seeking diverse items for events, performances, and occasions. This system endeavours to simplify the process for rental providers, offering a user-friendly interface for managing and facilitating access to costumes, props, footwear, and jewellery, ensuring a seamless experience for both renters and customers.

Key Features:

1.Inventory Catalog: This system maintains a detailed inventory catalogue encompassing costumes, props, footwear, and jewellery, allowing rental administrators to input comprehensive item details. Each entry includes essential information such as item name, category, size, condition, and rental availability. Visual references in the form of images aid in easy identification.

Customer-centric Functionality: Customers are provided with an intuitive interface to view available items, select, and rent according to their preferences. The system facilitates easy browsing, enabling customers to explore the inventory and make informed rental decisions.

Rental Operations: Core functionalities such as adding, deleting, viewing, renting, and returning items are seamlessly integrated into the system. Rental providers can efficiently manage the rental process, ensuring smooth transactions from selection to return, tracking rental durations, and associated fees.

Item Management: Admins can easily add new items, update details, and remove items that are no longer available for rental or require maintenance. Each item's status and availability are meticulously tracked to ensure accurate rental information.

The Ready-To-Wear Rental Management System aims to revolutionize the rental process by offering a centralized platform that streamlines the management of costumes, props, footwear, and jewellery, enhancing customer satisfaction while optimizing operations for rental providers. Through its intuitive functionalities, it seeks to elevate the rental experience for all stakeholders involved.

2. PROBLEM DEFINITION

Rental management services catering to costumes, props, footwear, and jewellery encounter multifaceted challenges in effectively overseeing their operations, often hindered by outdated systems or manual processes, leading to inefficiencies and missed opportunities. Key problem areas include:

1.Inadequate Inventory Organization:

- Limited Tracking Systems: Outdated or manual methods for inventory management result in challenges when retrieving specific items promptly and accurately.
- Inefficient Item Identification: Difficulty in quick and precise identification of available costumes, props, footwear, and jewellery due to inadequate cataloging methods.

2.Fragmented Item and Rental Tracking:

- Disconnected Rental History: Absence of a centralized system leads to challenges in establishing a comprehensive rental history for each item, impacting efficient tracking of usage and condition.
- Customer Preferences Management: Without a robust system, tracking customer preferences for specific items or categories becomes a daunting task, hindering personalized service.

3.Operational Hurdles in Rental Services:

- Exhibition and Rental Coordination: Managing multiple rentals for specific events or exhibitions involves intricate logistics, from scheduling to item availability and return coordination.
- Resource Allocation Challenges: Allocating items for rental demands effective planning, and without a streamlined system, it becomes challenging to manage resource distribution effectively.

4.Inventory and Maintenance Challenges:

- Inaccurate Inventory Records: Manual recording methods often lead to discrepancies in inventory status, causing potential issues such as overbooking or item shortage during peak demands.

- Maintenance Oversight: Difficulty in monitoring the condition and maintenance schedules of costumes, props, footwear, and jewellery items, potentially affecting their availability for rental.

5.Lack of Customer Interaction and Insight:

- Limited Customer Engagement: Absence of a centralized database for customer information limits the ability to engage customers, understand preferences, and provide tailored services.
- Data-Driven Insights Absence: Without robust analytics, obtaining insights into customer behaviour, popular rental choices, or trends becomes challenging, inhibiting strategic decision-making.
-

The Rental Management Service faces challenges in organizing inventory, tracking rentals, managing operations efficiently, maintaining inventory accuracy, and engaging customers effectively. Addressing these issues requires a comprehensive, integrated system to streamline operations, enhance customer experiences, and optimize rental services.

3. ER MODEL

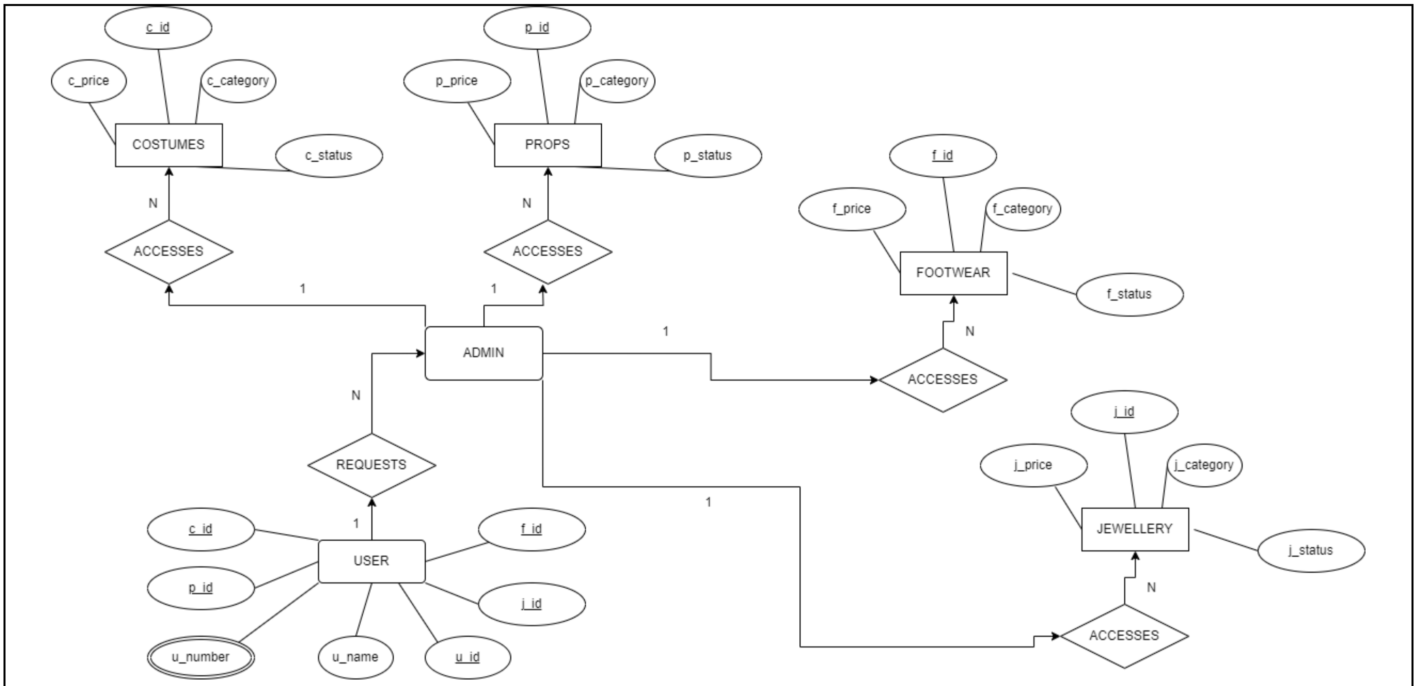


Figure 3.1

4. ER TO RELATIONAL MAPPING

4.1 STEPS OF ALGORITHM FOR CHOOSEN PROBLEM

1.Requirements Analysis:

- Identify system requirements, user roles (administrators, customers), and operational needs (inventory management, rental processes).

2.Database Design:

- Develop a normalized database schema encompassing items (costumes, props, footwear, jewellery), rental history, customer profiles, and transaction records.

3.Item Cataloging & Addition:

- Validate and add new items to the inventory database, ensuring comprehensive details, such as item type, condition, availability, and rental price.

4.Rental Assignment & Tracking

- Associate rented items with customers, update rental status, and track rental history for each item and customer, maintaining accurate records.

5.Item Update & Maintenance:

- Validate and update item details, ensuring accuracy in inventory records, maintenance schedules, and item availability for rental.

6.Item Removal & Availability:

- Confirm item ID for deletion, manage item removal from inventory while ensuring accurate availability status for rentals.

7.Inventory Display & Management:

- Fetch and display items based on criteria (type, availability), enabling efficient inventory management and customer browsing.

8.Error Handling & Notification:

- Implement error management to handle exceptions gracefully, providing informative messages for users and administrators.

9.Performance Optimization:

- Optimize database queries and indexes to enhance system performance, ensuring quicker retrieval of items and transactional efficiency.

10.Testing & Validation:

- Thoroughly test system functionalities, rental processes, and database operations to validate system accuracy and performance.

11.Documentation & Reference:

- Document system algorithms, database schema, functionalities, and operational procedures for future reference and maintenance.

4.2 COMPLETE DIAGRAM OF RELATIONAL MAPPING

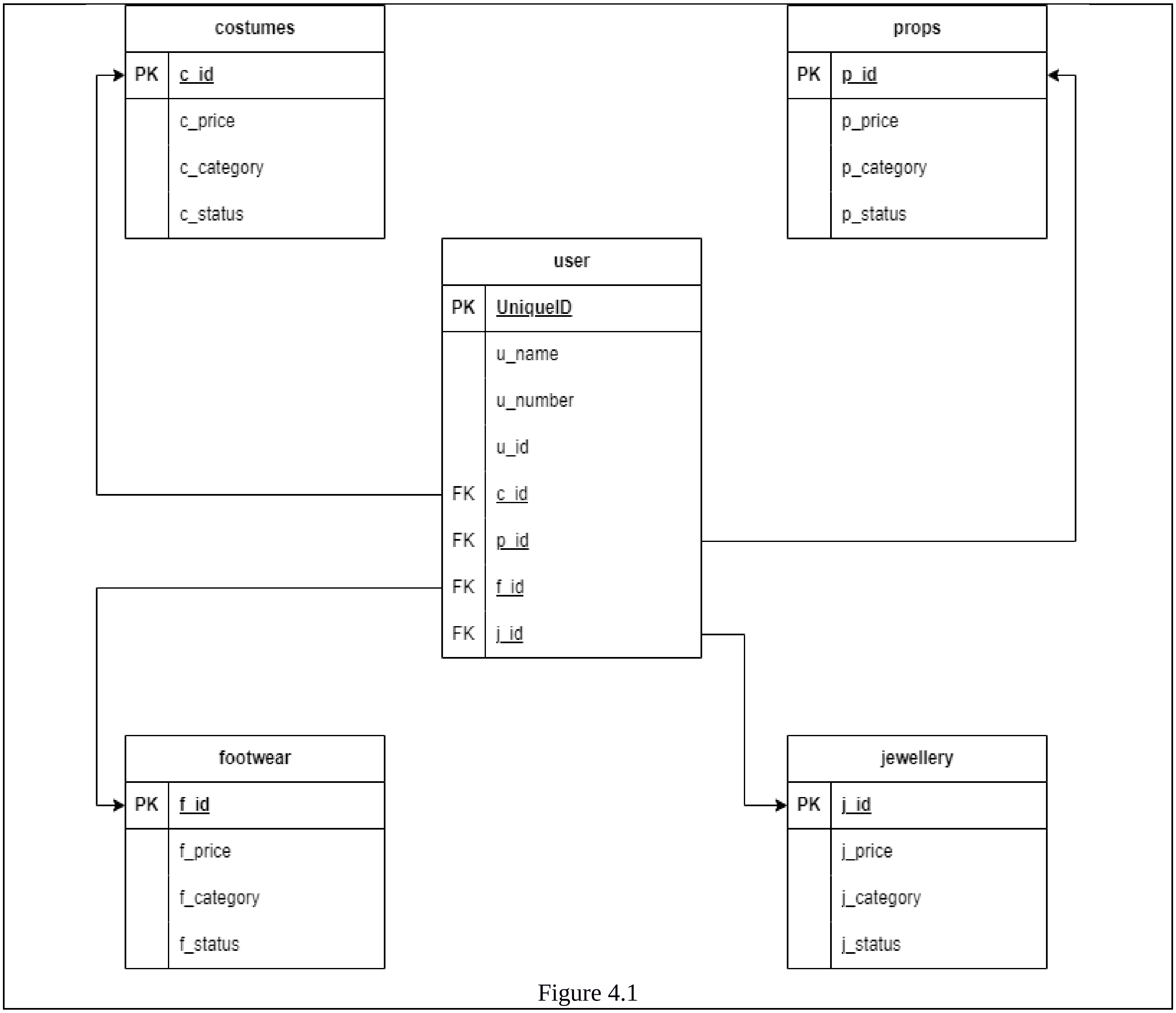
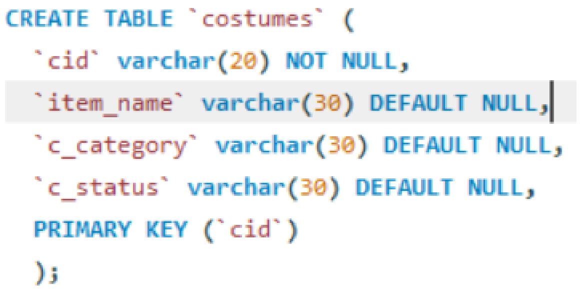


Figure 4.1

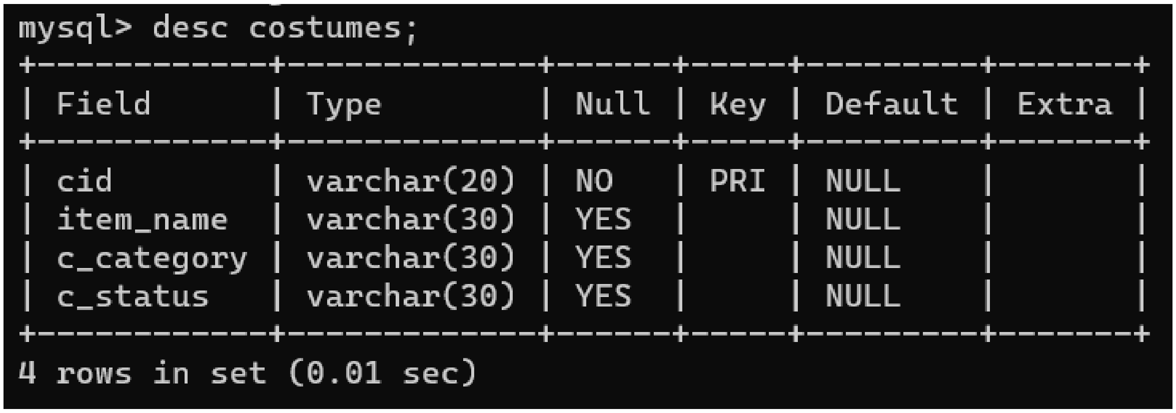
5. DDL STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE CREATION

Costumes table:

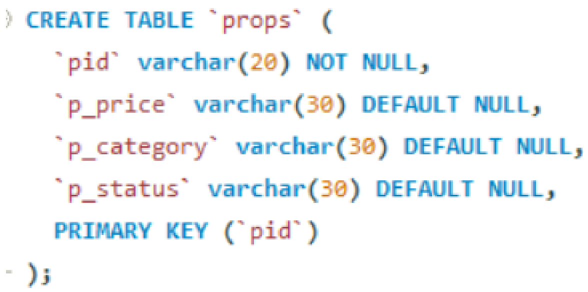


```
CREATE TABLE `costumes` (
  `cid` varchar(20) NOT NULL,
  `item_name` varchar(30) DEFAULT NULL,
  `c_category` varchar(30) DEFAULT NULL,
  `c_status` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`cid`)
  );
```

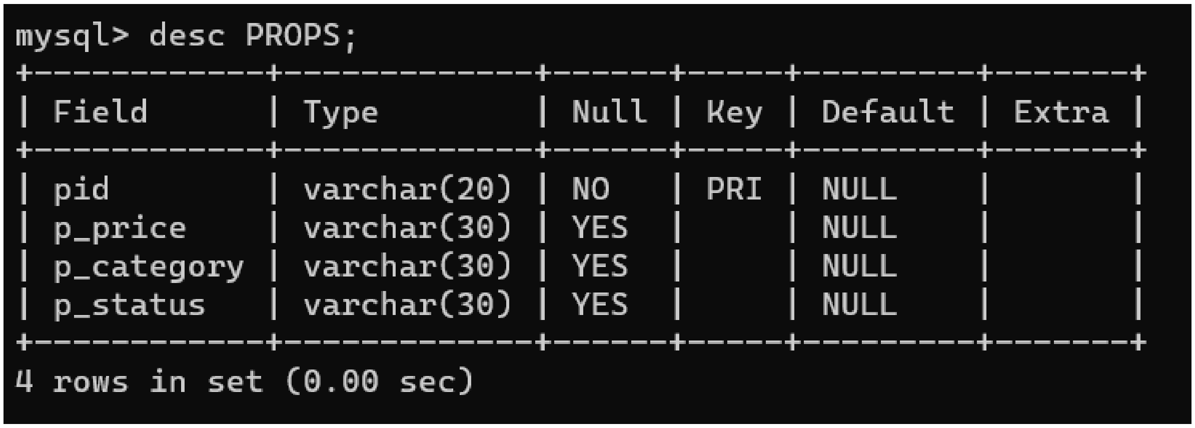


```
mysql> desc costumes;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| cid        | varchar(20) | NO   | PRI | NULL    |       |
| item_name  | varchar(30) | YES  |     | NULL    |       |
| c_category | varchar(30) | YES  |     | NULL    |       |
| c_status   | varchar(30) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

Props table:

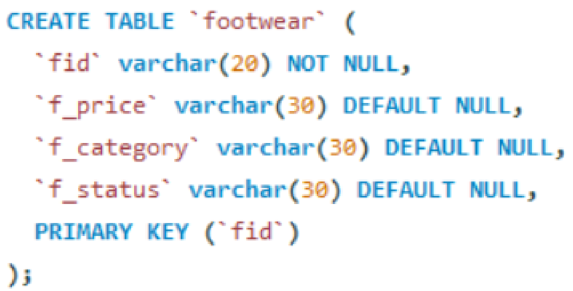


```
CREATE TABLE `props` (
  `pid` varchar(20) NOT NULL,
  `p_price` varchar(30) DEFAULT NULL,
  `p_category` varchar(30) DEFAULT NULL,
  `p_status` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`pid`)
);
```

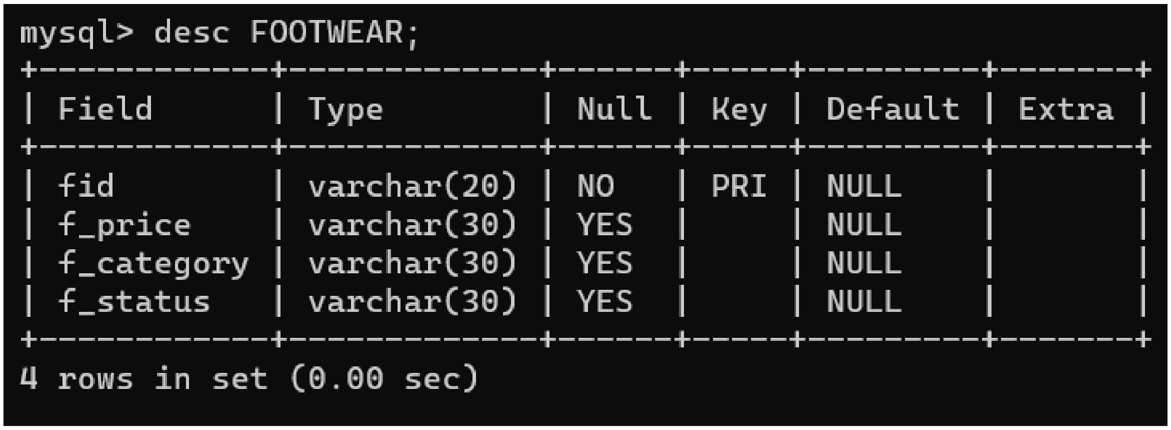


```
mysql> desc PROPS;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| pid        | varchar(20) | NO   | PRI | NULL    |       |
| p_price    | varchar(30) | YES  |     | NULL    |       |
| p_category | varchar(30) | YES  |     | NULL    |       |
| p_status   | varchar(30) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

FootwearTable:

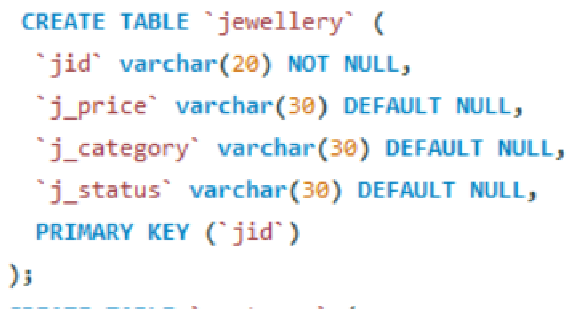


```
CREATE TABLE `footwear` (
  `fid` varchar(20) NOT NULL,
  `f_price` varchar(30) DEFAULT NULL,
  `f_category` varchar(30) DEFAULT NULL,
  `f_status` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`fid`)
);
```

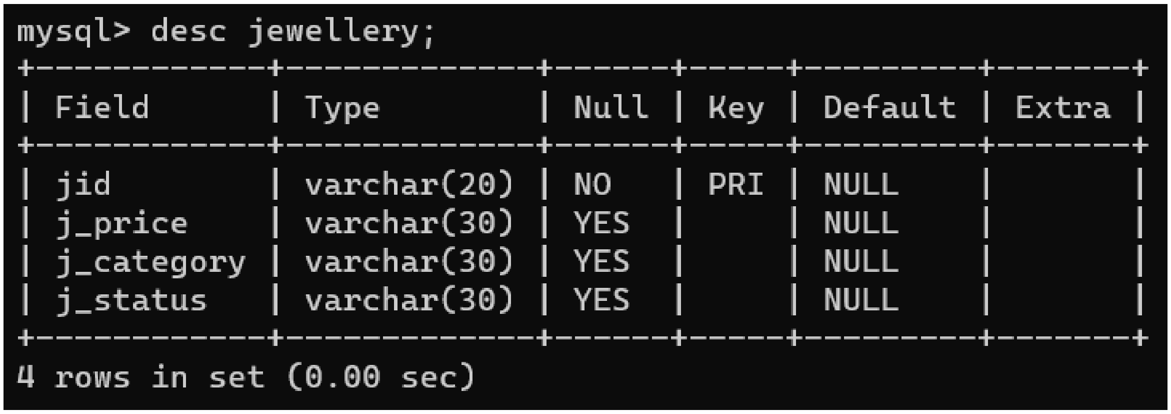


```
mysql> desc FOOTWEAR;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| fid        | varchar(20) | NO   | PRI | NULL    |       |
| f_price    | varchar(30) | YES  |     | NULL    |       |
| f_category | varchar(30) | YES  |     | NULL    |       |
| f_status   | varchar(30) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Jewellery Table:

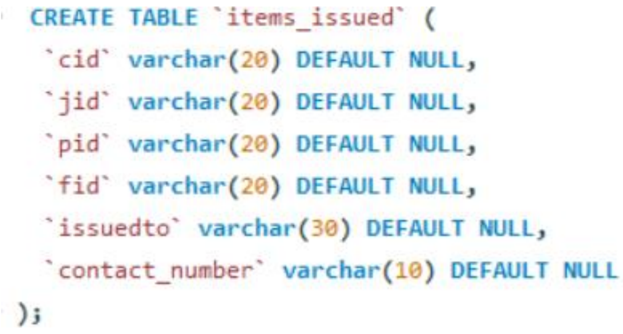


```
CREATE TABLE `jewellery` (
  `jid` varchar(20) NOT NULL,
  `j_price` varchar(30) DEFAULT NULL,
  `j_category` varchar(30) DEFAULT NULL,
  `j_status` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`jid`)
);
```

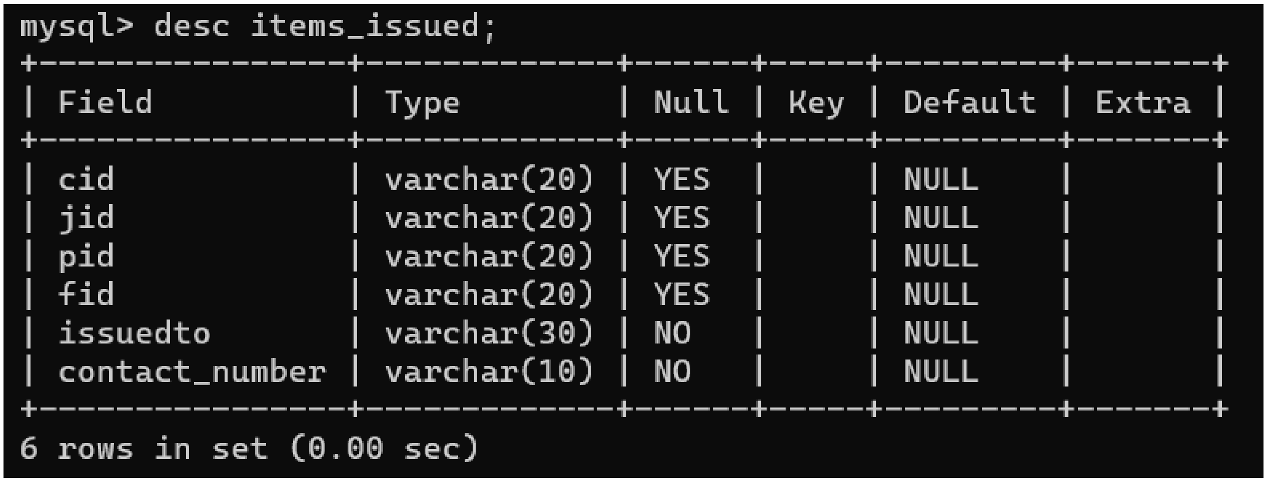


```
mysql> desc jewellery;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| jid        | varchar(20) | NO   | PRI | NULL    |       |
| j_price    | varchar(30) | YES  |     | NULL    |       |
| j_category | varchar(30) | YES  |     | NULL    |       |
| j_status   | varchar(30) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Items_issued Table:



```
CREATE TABLE `items_issued` (
  `cid` varchar(20) DEFAULT NULL,
  `jid` varchar(20) DEFAULT NULL,
  `pid` varchar(20) DEFAULT NULL,
  `fid` varchar(20) DEFAULT NULL,
  `issuedto` varchar(30) DEFAULT NULL,
  `contact_number` varchar(10) DEFAULT NULL
);
```

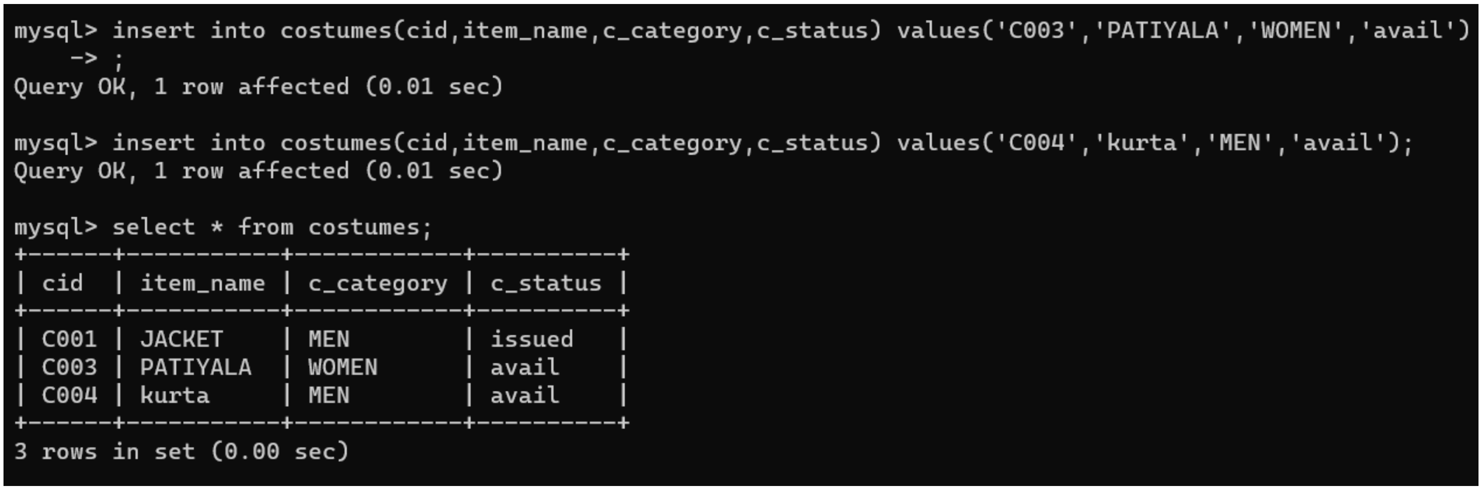


```
mysql> desc items_issued;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| cid            | varchar(20) | YES  |     | NULL    |       |
| jid            | varchar(20) | YES  |     | NULL    |       |
| pid            | varchar(20) | YES  |     | NULL    |       |
| fid            | varchar(20) | YES  |     | NULL    |       |
| issuedto       | varchar(30) | NO   |     | NULL    |       |
| contact_number | varchar(10) | NO   |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

6. DML STATEMENTS

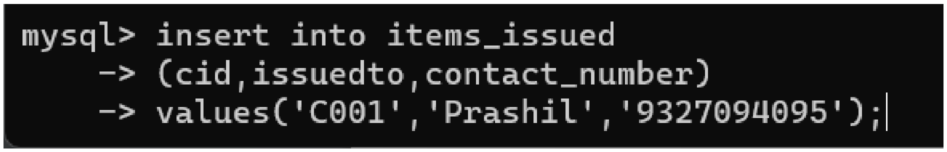
STATEMENTS WITH SCREEN SHOTS OF THE TABLE WITH INSERTED VALUES

1. Inserting values in costumes table

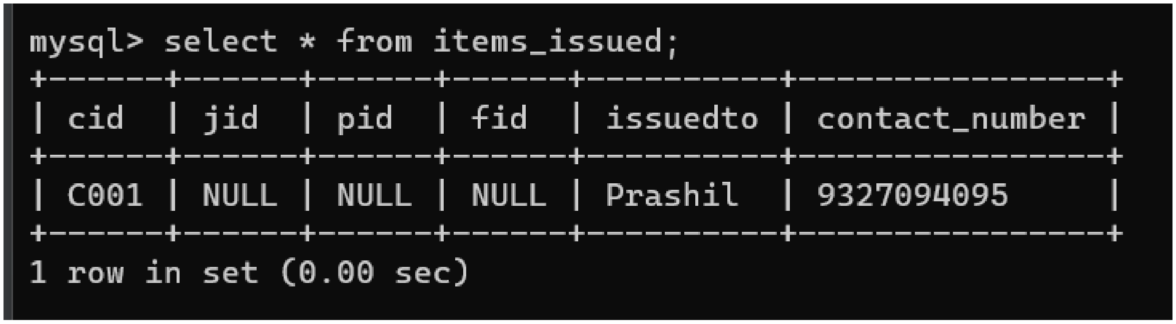


```
mysql> insert into costumes(cid,item_name,c_category,c_status) values('C003','PATIYALA','WOMEN','avail')
    -> ;
Query OK, 1 row affected (0.01 sec)

mysql> insert into costumes(cid,item_name,c_category,c_status) values('C004','kurta','MEN','avail');
Query OK, 1 row affected (0.01 sec)

mysql> select * from costumes;
+------+-----------+------------+----------+
| cid  | item_name | c_category | c_status |
+------+-----------+------------+----------+
| C001 | JACKET    | MEN        | issued   |
| C003 | PATIYALA  | WOMEN      | avail    |
| C004 | kurta     | MEN        | avail    |
+------+-----------+------------+----------+
3 rows in set (0.00 sec)
```

2. Insert into items_issued

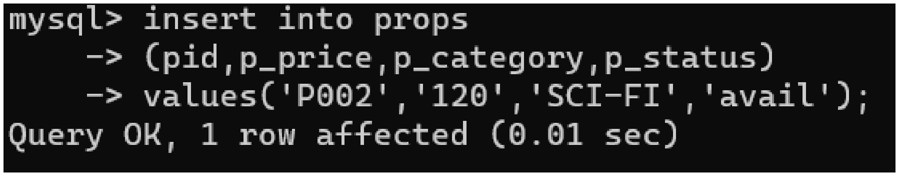


```
mysql> insert into items_issued
    -> (cid,issuedto,contact_number)
    -> values('C001','Prashil','9327094095');
```

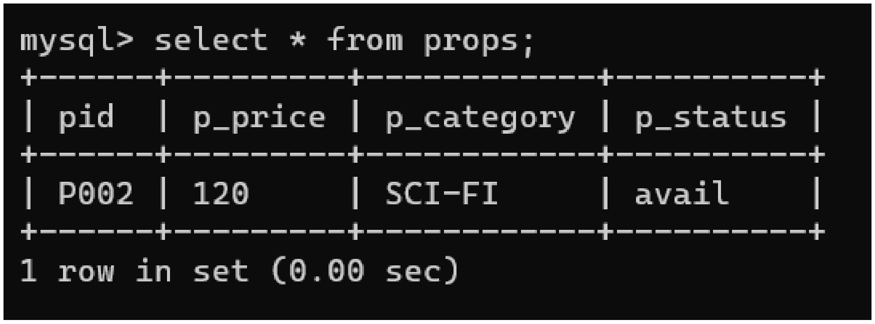


```
mysql> select * from items_issued;
+------+------+------+------+----------+----------------+
| cid  | jid  | pid  | fid  | issuedto | contact_number |
+------+------+------+------+----------+----------------+
| C001 | NULL | NULL | NULL | Prashil  | 9327094095     |
+------+------+------+------+----------+----------------+
1 row in set (0.00 sec)
```

3.Insert into Props



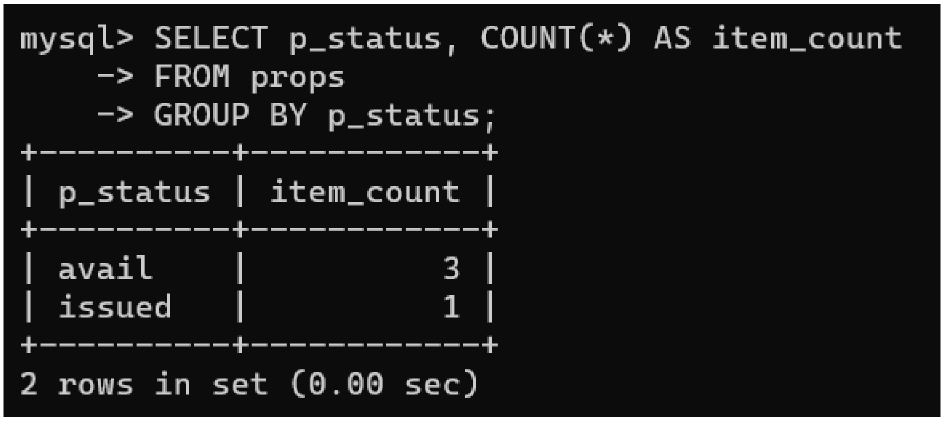
```
mysql> insert into props
    -> (pid,p_price,p_category,p_status)
    -> values('P002','120','SCI-FI','avail');
Query OK, 1 row affected (0.01 sec)
```



```
mysql> select * from props;
+------+---------+------------+----------+
| pid  | p_price | p_category | p_status |
+------+---------+------------+----------+
| P002 | 120     | SCI-FI     | avail    |
+------+---------+------------+----------+
1 row in set (0.00 sec)
```

7. QUERIES

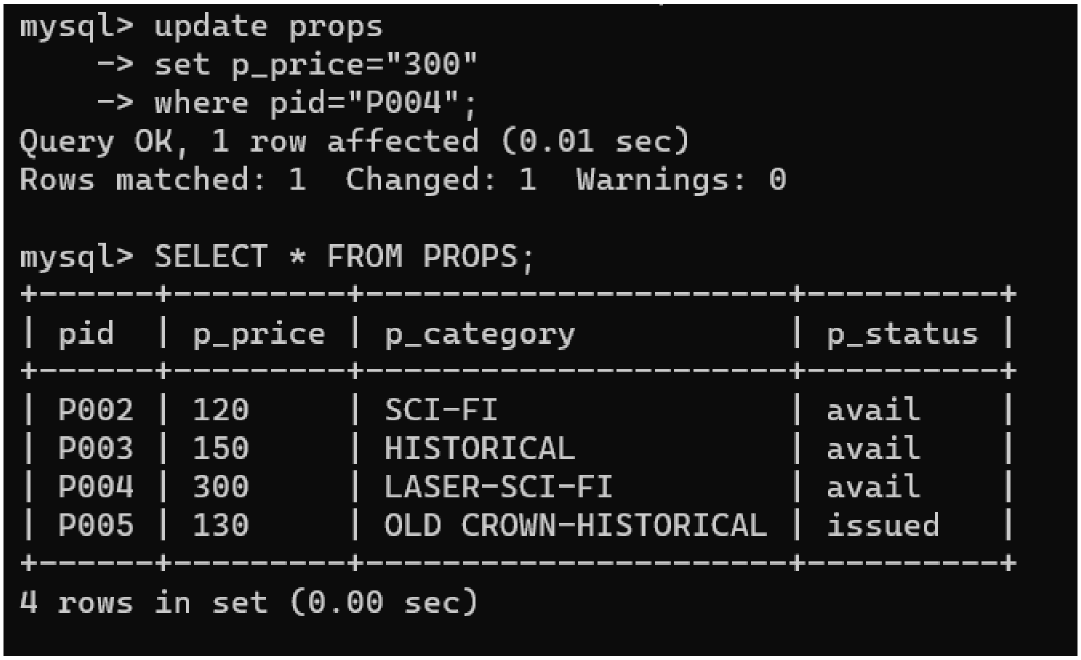
7.1 SIMPLE QUERY WITH GROUP BY



```
mysql> SELECT p_status, COUNT(*) AS item_count
    -> FROM props
    -> GROUP BY p_status;
+----------+------------+
| p_status | item_count |
+----------+------------+
| avail    |          3 |
| issued   |          1 |
+----------+------------+
2 rows in set (0.00 sec)
```

7.2 UPDATE OPERATION

-- Example: UPDATING PRICE TO 300 OF ITEM WITH PID=”P004”



```
mysql> update props
    -> set p_price="300"
    -> where pid="P004";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM PROPS;
+------+---------+----------------------+----------+
| pid  | p_price | p_category           | p_status |
+------+---------+----------------------+----------+
| P002 | 120     | SCI-FI               | avail    |
| P003 | 150     | HISTORICAL           | avail    |
| P004 | 300     | LASER-SCI-FI         | avail    |
| P005 | 130     | OLD CROWN-HISTORICAL | issued   |
+------+---------+----------------------+----------+
4 rows in set (0.00 sec)
```

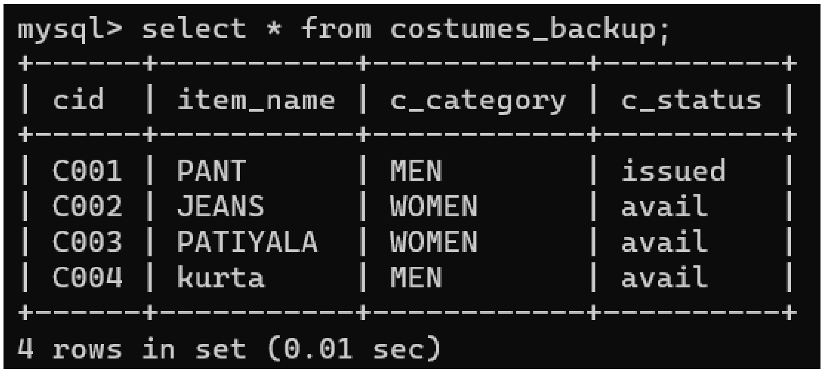
7.3 TRIGGER DELIMITER

This trigger is created for accidental deletion of any costumes from the costume table. If anything gets deleted from the costumes table, it gets backed up in costumes_backup table. The below trigger is written for the same functionality.

```
 DELIMITER //
CREATE TRIGGER backup_deleted_costumes
AFTER DELETE ON costumes
FOR EACH ROW
BEGIN
    INSERT INTO costumes_backup (cid, item_name, c_category, c_status)
    VALUES (OLD.cid, OLD.item_name, OLD.c_category, OLD.c_status);
END;
```

//

DELIMITER ;



```
mysql> select * from costumes_backup;
+------+-----------+------------+----------+
| cid  | item_name | c_category | c_status |
+------+-----------+------------+----------+
| C001 | PANT      | MEN        | issued   |
| C002 | JEANS     | WOMEN      | avail    |
| C003 | PATIYALA  | WOMEN      | avail    |
| C004 | kurta     | MEN        | avail    |
+------+-----------+------------+----------+
4 rows in set (0.01 sec)
```

7.4 CORRELATED QUERY

-- Example: Delete props with a price lower than the average price of props with prop_id = “P006”

INITIAL TABLE:

```
+------+---------+----------------------+----------+
| pid  | p_price | p_category           | p_status |
+------+---------+----------------------+----------+
| P002 | 120     | SCI-FI               | avail    |
| P003 | 150     | HISTORICAL           | avail    |
| P004 | 300     | LASER-SCI-FI         | avail    |
| P005 | 130     | OLD CROWN-HISTORICAL | issued   |
| P006 | 200     | SWORD-HISTORICAL     | avail    |
+------+---------+----------------------+----------+
```

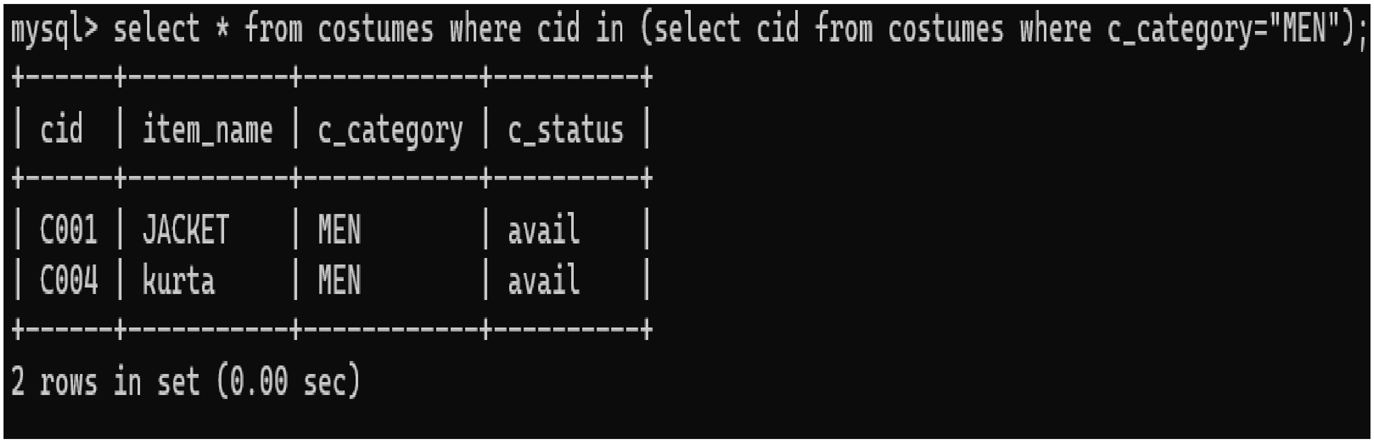
QUERY:

```
DELETE FROM props
WHERE p_price < (SELECT AVG(p_price) FROM props WHERE pid = 'P006');
```

Resulting table:

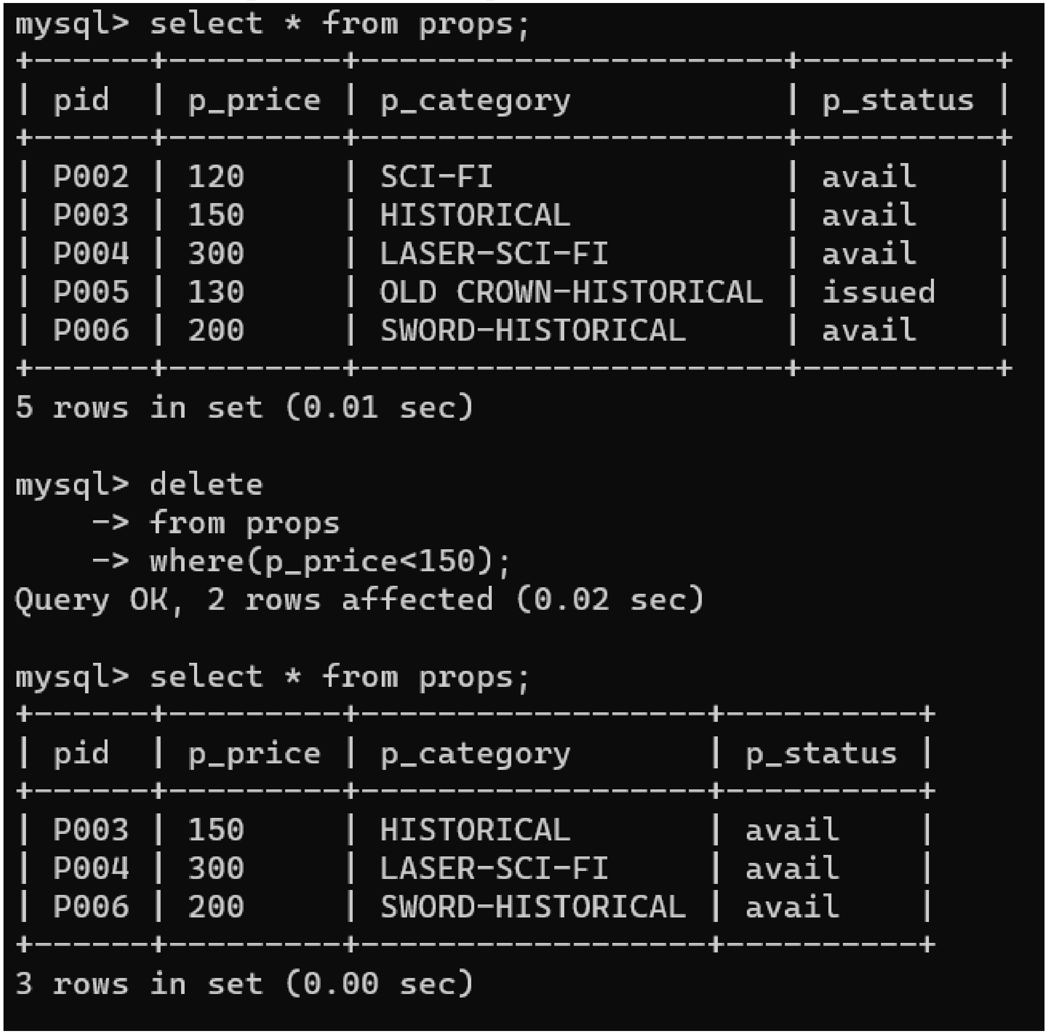
```
+------+---------+----------------------+----------+
| pid  | p_price | p_category           | p_status |
+------+---------+----------------------+----------+
| P004 | 300     | LASER-SCI-FI         | avail    |
| P006 | 200     | SWORD-HISTORICAL     | avail    |
+------+---------+----------------------+----------+
```

7.5 NESTED QUERY



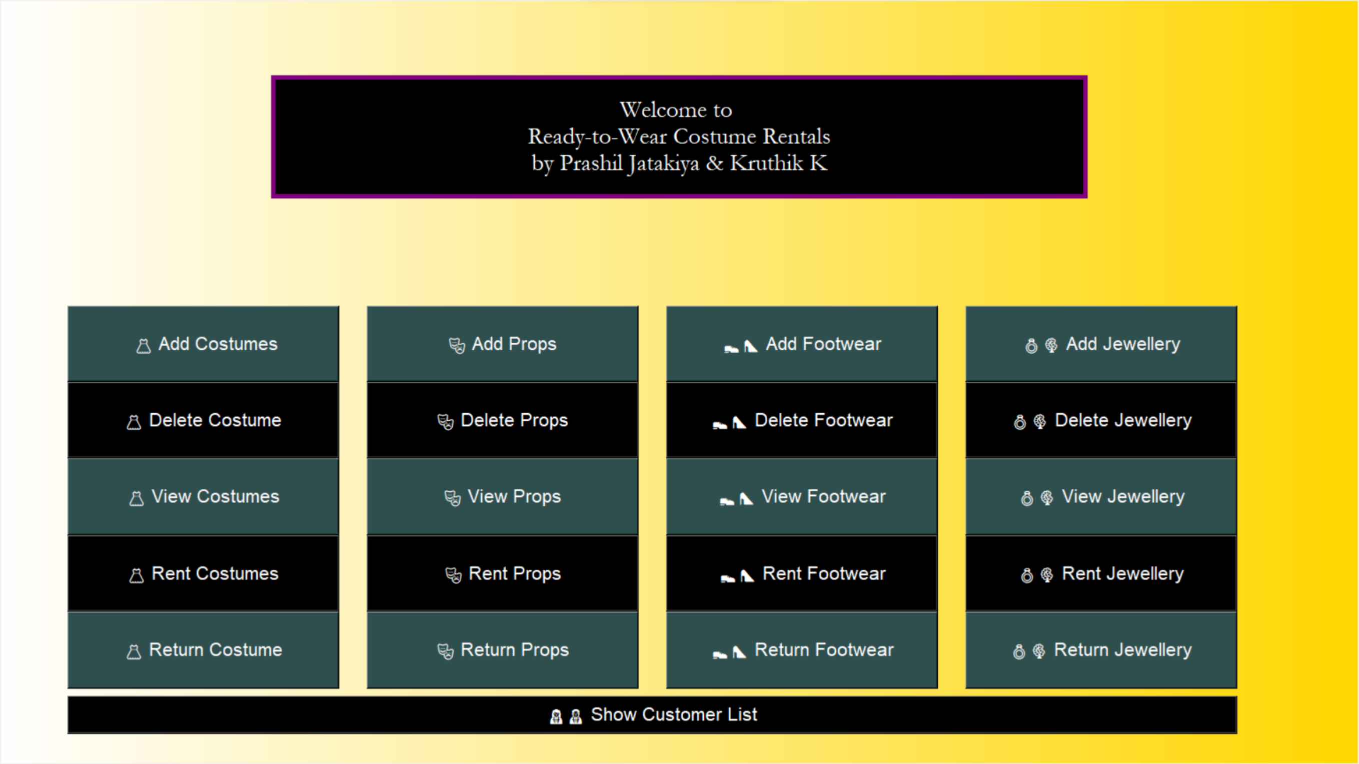
```
mysql> select * from costumes where cid in (select cid from costumes where c_category="MEN");
+------+-----------+------------+----------+
| cid  | item_name | c_category | c_status |
+------+-----------+------------+----------+
| C001 | JACKET    | MEN        | avail    |
| C004 | kurta     | MEN        | avail    |
+------+-----------+------------+----------+
2 rows in set (0.00 sec)
```

7.6 DELETE Operation On Props

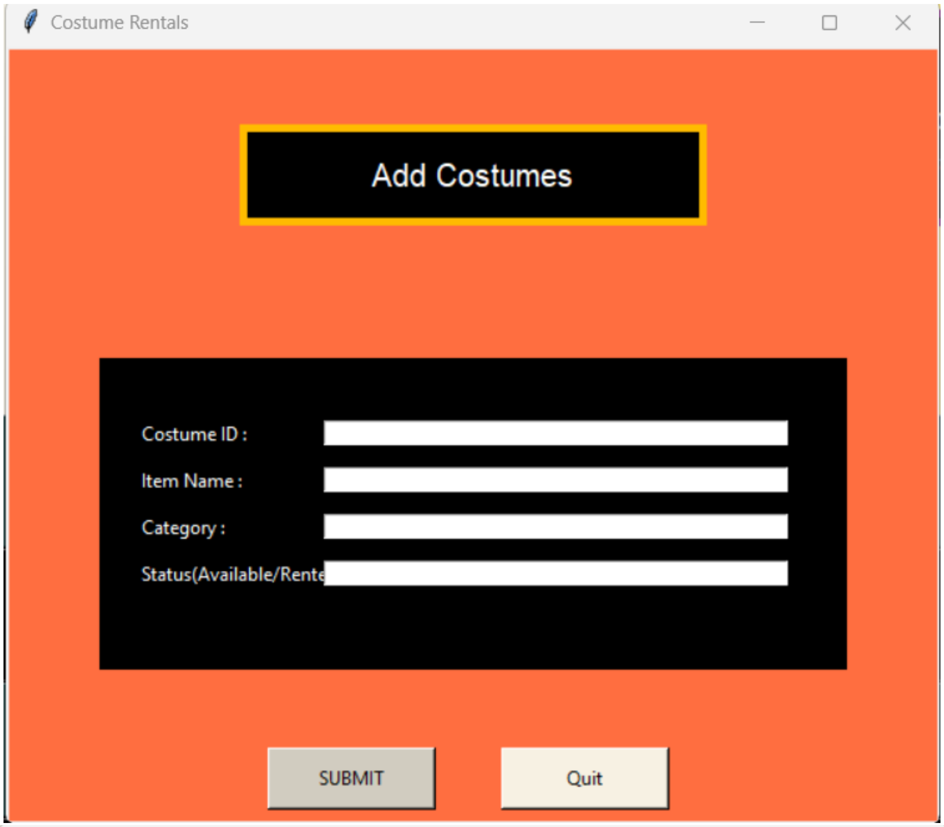


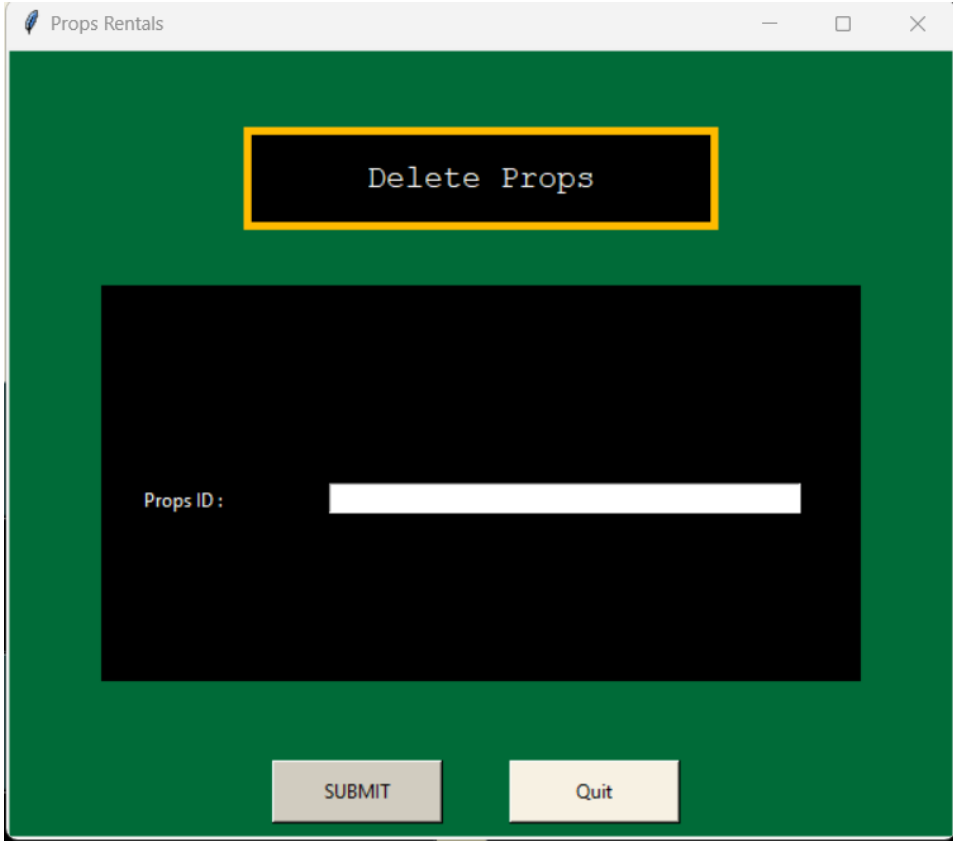
```
mysql> select * from props;
+------+---------+----------------------+----------+
| pid  | p_price | p_category           | p_status |
+------+---------+----------------------+----------+
| P002 | 120     | SCI-FI               | avail    |
| P003 | 150     | HISTORICAL           | avail    |
| P004 | 300     | LASER-SCI-FI         | avail    |
| P005 | 130     | OLD CROWN-HISTORICAL | issued   |
| P006 | 200     | SWORD-HISTORICAL     | avail    |
+------+---------+----------------------+----------+
5 rows in set (0.01 sec)

mysql> delete
    -> from props
    -> where(p_price<150);
Query OK, 2 rows affected (0.02 sec)

mysql> select * from props;
+------+---------+------------------+----------+
| pid  | p_price | p_category       | p_status |
+------+---------+------------------+----------+
| P003 | 150     | HISTORICAL       | avail    |
| P004 | 300     | LASER-SCI-FI     | avail    |
| P006 | 200     | SWORD-HISTORICAL | avail    |
+------+---------+------------------+----------+
3 rows in set (0.00 sec)
```

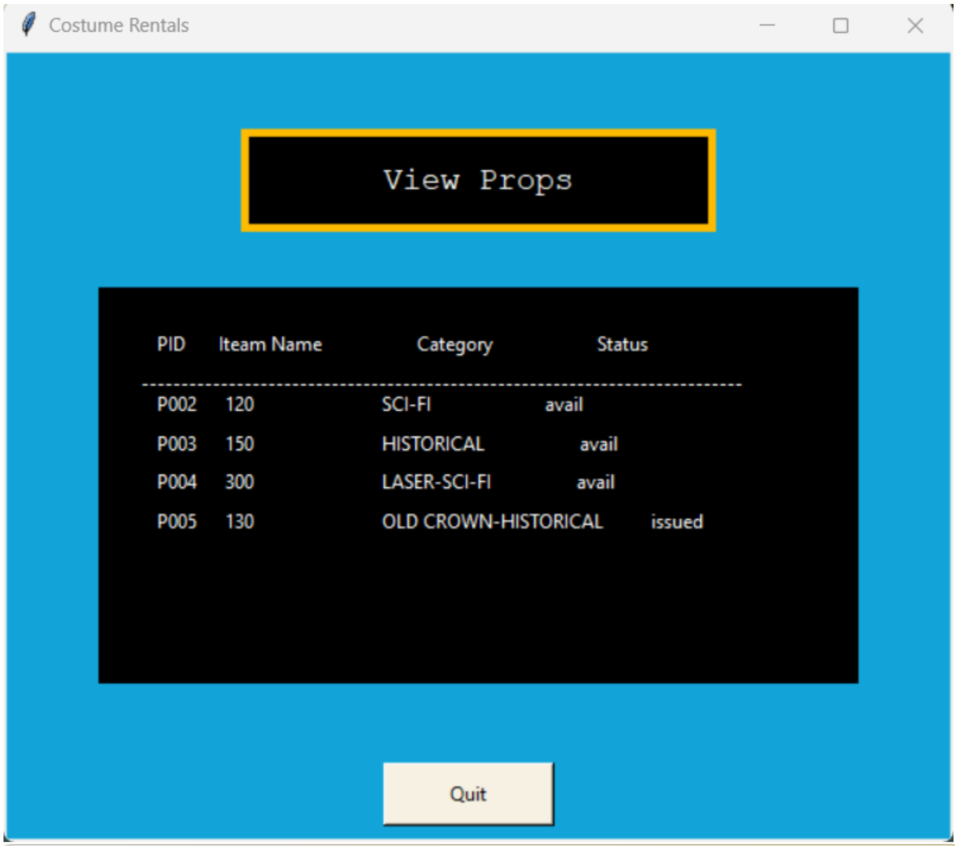
9. FRONT END DEVELOPEMNT



Welcome to
Ready-to-Wear Costume Rentals
by Prashil Jatakiya & Kruthik K

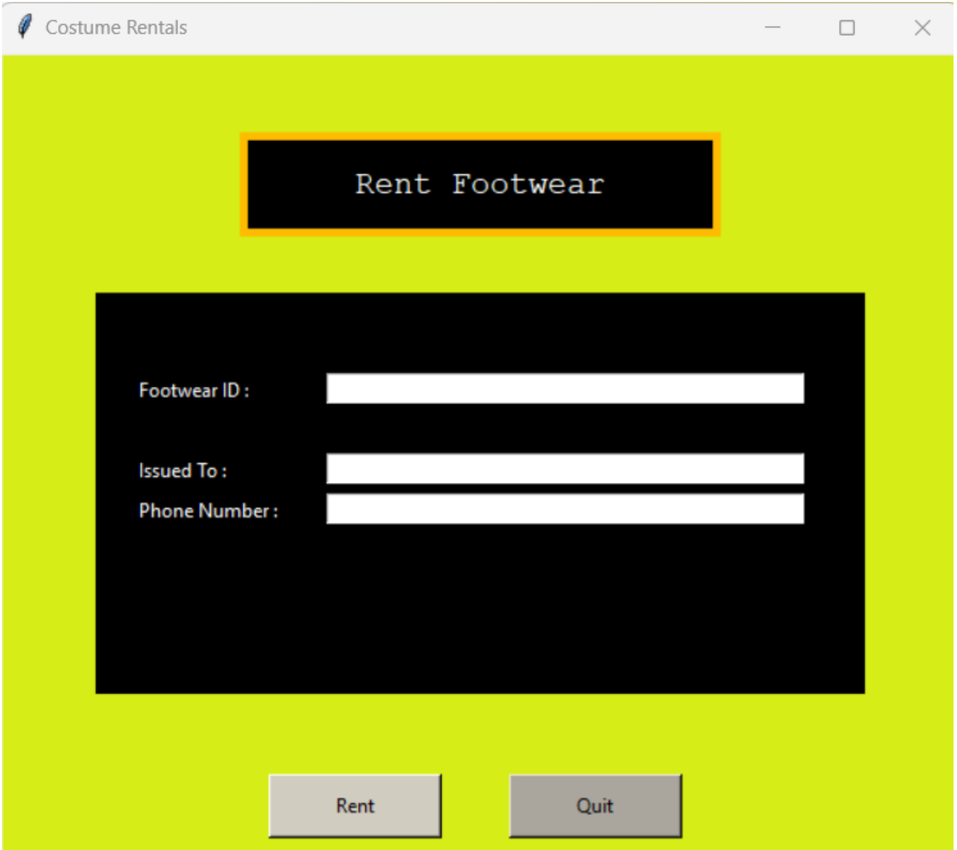

Costume Rentals
Add Costumes
Costume ID :
Item Name :
Category :
Status(Available/Rente
SUBMIT
Quit


Props Rentals
Delete Props
Props ID :
SUBMIT
Quit



View Props

```
PID     Iteam Name          Category            Status
------------------------------------------------------------------
P002    120                 SCI-FI              avail
P003    150                 HISTORICAL              avail
P004    300                 LASER-SCI-FI            avail
P005    130                 OLD CROWN-HISTORICAL        issued
```

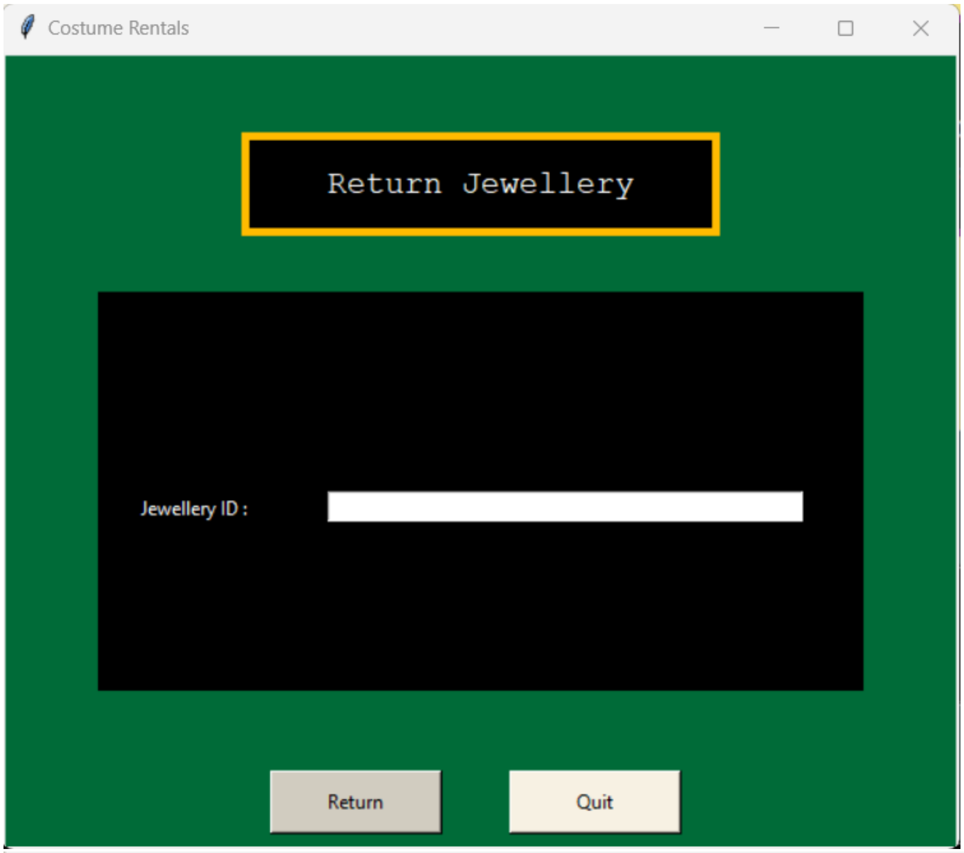


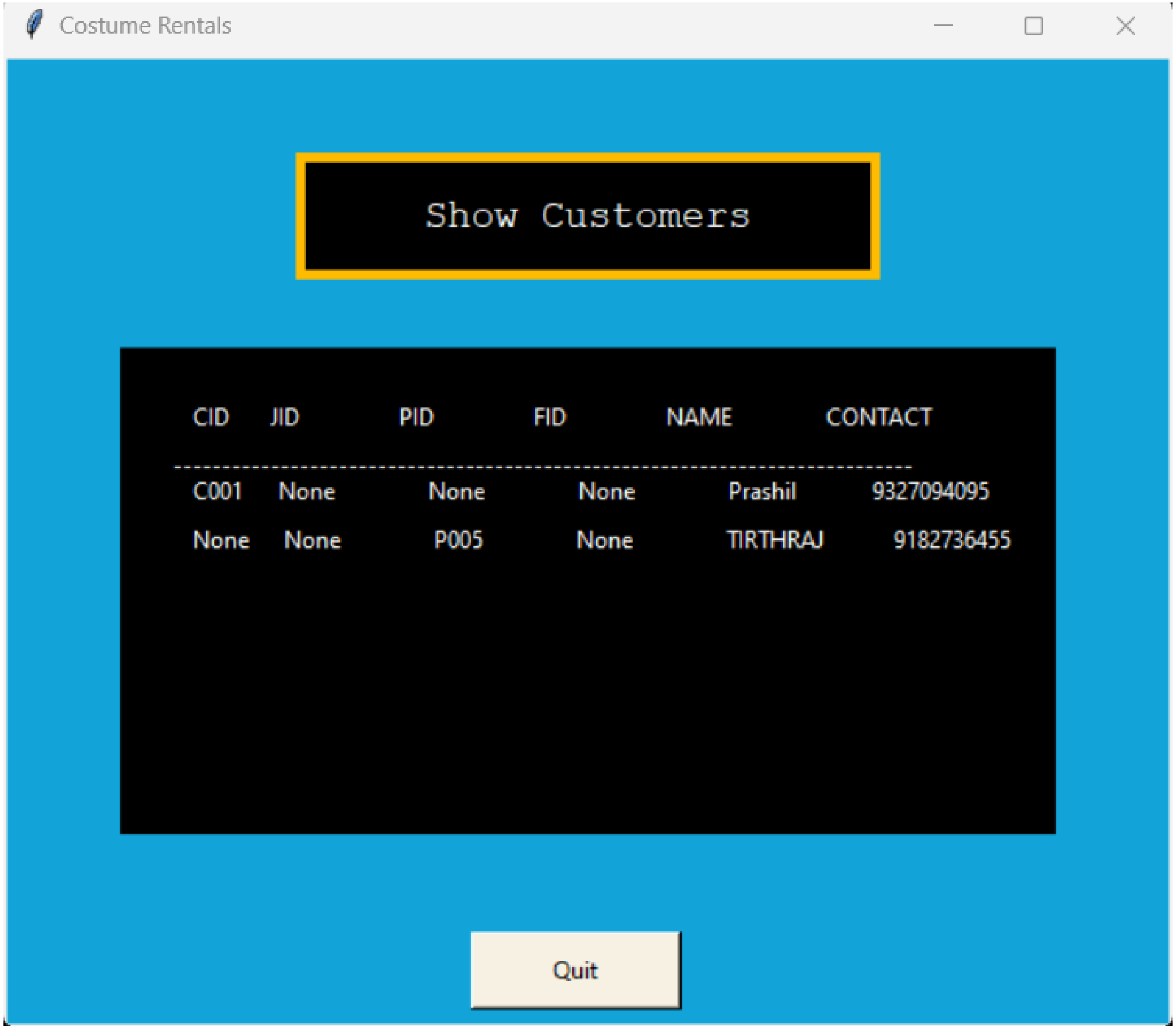
Rent Footwear

Footwear ID :

Issued To :

Phone Number :





```
CID    JID        PID        FID        NAME        CONTACT
-------------------------------------------------------------------------
C001   None       None       None       Prashil     9327094095
None   None       P005       None       TIRTHRAJ    9182736455
```

REFERENCES:

1.Youtube
https://www.youtube.com/watch?v=GODYXAnBDvc&list=PLMnmAHlVrrJJyiVdYwHPp5XQASl24_KBN&index=2

2.Geeks For Geeks
https://www.geeksforgeeks.org/python-mysql/

3. Github