

PROJECT REPORT:

Muntaha Munia

CREATE TABLES:

Our database name is – **dvd_rental (information about dvd rent shop) and creating tables queries are used:**

Tables:

Customers

- customer_id INT PRIMARY KEY
- Fname VARCHAR
- Lname VARCHAR
- Email VARCHAR

Payments:

- Rent_cost INT
- Rent_paid VARCHAR
- Film_id VARCHAR FOREIGN KEY
- Customer_id VARCHAR FOREIGN KEY

Film_dvd:

- Film_id INT PRIMARY KEY
- Language VARCHAR
- Title VARCHAR
- Rating INT

```
createTable = "CREATE TABLE IF NOT EXISTS customers(customer_id INT NOT NULL AUTO_INCREMENT,fname VARCHAR(50),lname VARCHAR(50), email VARCHAR(50),
PRIMARY KEY(customer_id))"
databaseRunQuery(createTable)

createTable = "CREATE TABLE IF NOT EXISTS film_dvd(film_id INT NOT NULL AUTO_INCREMENT,language VARCHAR(30), titel VARCHAR(50), rating INT, PRIMARY
KEY(film_id))"
databaseRunQuery(createTable)

createTable = "CREATE TABLE IF NOT EXISTS payments(rent_cost INT NOT NULL, rent_paid VARCHAR(1), film_id INT, customer_id INT, FOREIGN KEY(customer_id)
REFERENCES customers(customer_id) ON DELETE CASCADE, FOREIGN KEY(film_id) REFERENCES film_dvd(film_id) ON DELETE CASCADE)"
databaseRunQuery(createTable)
```

```
mysql> DESC customers;
```

Field	Type	Null	Key	Default	Extra
customer_id	int(11)	NO	PRI	NULL	auto_increment
fname	varchar(50)	YES		NULL	
lname	varchar(50)	YES		NULL	
email	varchar(50)	YES		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> DESC film_dvd;
```

Field	Type	Null	Key	Default	Extra
film_id	int(11)	NO	PRI	NULL	auto_increment
language	varchar(30)	YES		NULL	
titel	varchar(50)	YES		NULL	
rating	int(11)	YES		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> DESC payments;
```

Field	Type	Null	Key	Default	Extra
rent_cost	int(11)	NO		NULL	
rent_paid	varchar(1)	YES		NULL	
film_id	int(11)	YES	MUL	NULL	
customer_id	int(11)	YES	MUL	NULL	

```
4 rows in set (0.00 sec)
```

Details : Here, we have created 3 tables in our database dvd_rental:

1. Customers (all the information about customers)
2. Film_dvd(all the information about film_dvd)
3. Payments(all the information about customers who rent dvd)

- **FUNCTIONS:**

1. **INSERT:** To insert the information in the tables, used queries are:

```
def insert():
    allinone = input("Please enter first name, last name and email , all separated by a space\n")
    splitString = allinone.split(" ")
    #customer_id = splitString[0]
    fname = splitString[0]
    lname = splitString[1]
    email = splitString[2]
    insertquery = "INSERT INTO customers(fname,lname,email) VALUES ('" + fname + "\",\"" + lname + "\",\"" + email + "\")"
    databaseRunQuery(insertquery)

    allinone = input("Please enter language, titel, rating,all separated by a space\n")
    splitString = allinone.split(" ")
    #film_id = splitString[0]
    language = splitString[0]
    titel = splitString[1]
    rating = splitString[2]
    insertquery = "INSERT INTO film_dvd(language,titel,rating) VALUES(\"" + language + "\",\"" + titel + "\",\"" + rating + "\")"
    databaseRunQuery(insertquery)

    allinone = input("Please enter rent_cost, rent_paid(Y/N), film_id, customer_id all, separated by a space\n")
    splitString = allinone.split(" ")
    rent_cost = splitString[0]
    rent_paid = splitString[1]
    film_id = splitString[2]
    customer_id = splitString[3]
    insertquery = "INSERT INTO payments VALUES(\"" + rent_cost + "\",\"" + rent_paid + "\",\"" + film_id + "\",\"" + customer_id + "\")"
    databaseRunQuery(insertquery)
```

Detail: By using insert function , we add values to the attributes of all the tables we have created.

```
My Team Name: Amaryce, Eileen, Muntaha, Shaina
1. Insert
2. Delete
3. Lookup by customer_id
4. Lookup All data
5. Update
Please enter a command, type q or quit to exit.
1
Please enter first name, last name and email , all separated by a space
BB CC bbcc@gmail.com
Please enter language, titel, rating,all separated by a space
English Fun 7
Please enter rent_cost, rent_paid(Y/N), film_id, customer_id all, separated by a space
23 Y 3 3
```

2. **DELETE:**

```
def delete():
    # delete from customer to delete account and all information
    # delete from film after returning
    option = input("Enter 'a' to delete account or 'f' to delete a film\n")
    if (option == 'a'):
        todelete = input("Enter your ID to delete your account\n")
        tuple = databaseRunQuery("DELETE from customers WHERE customer_id = " + todelete + " AND customer_id NOT IN (SELECT customer_id FROM payments WHERE rent_paid REGEXP 'N')")
        #Check to see if we get a result back from the database, None means no rows were returned.
```

Details: By using delete function, we can delete customer information or film dvd information based on our customers need. For instance, if any customer didn't pay, if any one wants to delete that customers info , cant delete it . And also if any one wants to delete any film info and if its rented , it wont be deleted.

3. LOOK UP BY customer_id:

```
def lookupOne():
    table = input("Which table do you want to look up?\n")
    while(table != "customers" and table != "film_dvd" and table != "payments"):
        table = input("Invalid table. Which table do you want to look up?\n")
    id = input("Which customer_id do you want to look up?\n")
    tuple = databaseFetchOne("SELECT * FROM " + table + " WHERE customer_id = " + id ) #Provide a select query here

    #Check to see if we get a result back from the database, None means no rows were returned.
    while(tuple == None):
        id = input("Invalid id. Which customer_id do you want to look up?\n")
        tuple = databaseFetchOne("SELECT * FROM " + table + " WHERE customer_id = " + id )

    if(table == 'customers'):
        print("customer_id: " + str(tuple[0]) + "\t" + "fname: " + tuple[1] + "\t" + "lname: " + tuple[2] + "\t" + "email: " + tuple[3])
    elif(table == 'film_dvd'):
        print("film_id: " + str(tuple[0]) + "\t" + "language: " + tuple[1] + "\t" + "titel: " + tuple[2] + "\t" + "customer_id: " + str(tuple[3]))
    elif(table == 'payments'):
        print("rent_cost: " + str(tuple[0]) + "\t" + "rent_paid: " + tuple[1] + "\t" + "film_id: " + str(tuple[2]) + "\t" + "customer_id: " + str(tuple[3]))
```

Details: By using this function, based on customer id, we can get all the information from

```
Please enter a command, type q or quit to exit.
2
My Team Name: Amaryce, Eileen, Muntaha, Shaina
1. Insert
2. Delete
3. Lookup by customer_id
4. Lookup All data
5. Update
Please enter a command, type q or quit to exit.
3
Which table do you want to look up?(customers or payments)
customers
Which customer_id do you want to look up?
3
customer_id: 3  fname: BB          lname: CC          email: bbcc@gmail.com
```

customers and filmdvd tables.

4. LOOK UP BY ALL:

```
def lookupAll():
    table = input("Which table do you want to search from: customers, film_dvd, or payments?\n")
    if(table == 'customers'):
        print("fname, lname, or email")
    elif(table == 'film_dvd'):
        print("language, titel, or rating")
    elif(table == 'payments'):
        print("rent_cost or rent_paid")
    else:
        print("Table not found\n")
        return

    attribute = input("Which attribute above do you want to look from?\n")
    while(attribute != 'fname' and attribute != 'lname' and attribute != 'email' and attribute != 'language' and attribute != 'title' and attribute != 'rating'):
        attribute = input("Invalid attribute. Which attribute above do you want to look from?\n")
    if(attribute == 'rating' or attribute == 'rent_cost'):
        amount = input("What " + attribute + "would you like to search from. Add <, >, or = before the number separated by a space: ")
        tuples = databaseFetchAll("SELECT * FROM " + table + " WHERE " + attribute + amount)
    else:
        direction = input("From which part of the word do you want to look up: beginning or ending\n")
        while(direction != 'beginning' and direction != 'ending'):
            direction = input("Invalid direction. Try again\n")

        expression = input("What do you want it to start/end with, enter one character?\n")

        if(direction == "beginning"):
            tuples = databaseFetchAll("SELECT * FROM " + table + " WHERE " + attribute + " REGEXP '^" + expression + "$")
        elif(direction == "ending"):
            tuples = databaseFetchAll("SELECT * FROM " + table + " WHERE " + attribute + " REGEXP '" + expression + "$")

    if(table == 'customers'):
        for tuple in tuples:
            print("customer_id: " + str(tuple[0]) + "\t" + "fname: " + tuple[1] + "\t" + "lname: " + tuple[2] + "\t" + "email: " + tuple[3] + "\n")
    elif(table == 'film_dvd'):
        for tuple in tuples:
            print("film_id: " + str(tuple[0]) + "\t" + "language: " + tuple[1] + "\t" + "titel: " + tuple[2] + "\t" + "rating: " + str(tuple[3]) + "\n")
    elif(table == 'payments'):
        for tuple in tuples:
            print("rent_cost: " + str(tuple[0]) + "\t" + "rent_paid: " + tuple[1] + "\t" + "film_id: " + str(tuple[2]) + "\t" + "customer_id: " + str(tuple[3]) + "\n")
```

Details: In this function, we use 3 ways.

- Based on any attributes we can get any information from any tables.
- Based regexp by using only fname “first letter”, we can get all the matches
- Based on rating(ranking) attribute , we can get certain movies which rating is above/below given rating range

To find this information we used regex(beginning or start letter)

```
My Team Name: Amaryce, Eileen, Muntaha, Shaina
1. Insert
2. Delete
3. Lookup by customer_id
4. Lookup All data
5. Update
Please enter a command, type q or quit to exit.
4
Which table do you want to search from: customers, film_dvd, or payments?
customers
fname, lname, or email
Which attribute above do you want to look from?
fname
From which part of the word do you want to look up: beginning or ending
beginning
What do you want it to start/end with, enter one character?
A
customer_id: 1  fname: Aaa      lname: Baa      email: aaa@gmail.com
customer_id: 2  fname: Abb      lname: Bcc      email: abb@gmail.com
```

This information about ranking like if someone wants to rent a film_dvd which rating is above 5:

```
Please enter a command, type q or quit to exit.
4
Which table do you want to search from: customers, film_dvd, or payments?
film_dvd
language, titel, or rating
Which attribute above do you want to look from?
rating
What rating would you like to search from. Add <, >, or = before the number separated by a space: >5
film_id: 1      language: English      titel: Hello      rating: 6
film_id: 2      language: Spanish      titel: Hola       rating: 7
film_id: 3      language: English      titel: Fun        rating: 7
```

5. Update:

```
def update():
    table_to_update = input("What table do you want to update?(customers,film_dvd,payments)\n")
    while (table_to_update != "customers" and table_to_update != "payments" and table_to_update != "film_dvd"):
        table_to_update = input("What table do you want to update?\n")
    lookfor = input("What value would you like to update?\n")
    while (lookfor != "customer_id" and lookfor != "fname" and lookfor != "lname" and lookfor != "email" and lookfor != "film_id" and lookfor !=
"language" and lookfor != "titel" and lookfor != "rent_cost" and lookfor != "rent_paid"):
        lookfor = input("What value would you like to update?\n")
    ids = input("what is your customer id?\n")

    exists = databaseFetchOne("SELECT * FROM " + table_to_update + " WHERE customer_id = " + ids )
    updateto = input("What value would you like it to update to?\n")

    if(table_to_update == "customers"):
        if(lookfor != "customer_id"):
            result = databaseRunQuery("UPDATE customers SET " + lookfor + "= \"" + updateto + "\" WHERE customer_id = " + ids )
            print("Altered " + str(result) + " rows!")
    if(table_to_update == "payments"):
        if(lookfor != "film_id" or lookfor != "customer_id" or lookfor != "rent_cost"):
            result = databaseRunQuery("UPDATE payments SET " + lookfor + "= \"" + updateto + "\" WHERE customer_id = " + ids )
            print("Altered " + str(result) + " rows!")
    if(table_to_update == "film_dvd"):
        if(lookfor != "film_id"):
            result = databaseRunQuery("UPDATE film_dvd SET " + lookfor + "= \"" + updateto + "\" WHERE customer_id = " + ids )
            print("Altered " + str(result) + " rows!")
```

Details: By using this function, we can update any values of attributes. For instance, from the payments table under rent_paid attribute , we change N(No) to Y(yes) when someone paid their dvd rent.

```
Please enter a command, type q or quit to exit.
5
What table do you want to update?(customers,film_dvd,payments)
payments
What value would you like to update?
rent_paid
what is your customer id?3
What value would you like it to update to?
N
Altered 1 rows!
```

