A Project Report on:

Database Management System for Video Parlour System

Jawaharlal Nehru University
New Delhi - 110067

Submitted By: Group J

Khushi Pawecha (22/10/JC/065)
Shikha Singh(22/10/JC/049)
Aparna Ghosh(22/10/JC/019)
Rohit Raj(22/10/JC/031)

Abstract

This report describes the implementation of a Database Management System for Video Parlour System. It was developed for Storing, Managing, and Updating the Data used for movie streaming.

This report mainly includes the background and development process of the project. The main steps in the development process are identifying the Entities and the Relationship among them and presenting them in graphical form with an ER Diagram. It also includes some of the SQL queries implemented on Oracle.

ACKNOWLEDGEMENTS

We are very pleased to present the report “Database Management System for Video Parlour System” which has been prepared for the project .

We would like to express our deep gratitude to our professor **Mr. T.V. Vijay Kumar** for his guidance, supervision and continuous support during the period of the project work.

Khushi Pawecha (22/10/JC/065)
Shikha Singh (22/10/JC/049)
Aparna Ghosh (22/10/JC/019)
Rohit Raj (22/10/JC/031)

Jawaharlal Nehru University
MCA 2nd Sem

Table of Contents

Introduction: DBMS for Video Parlour System

This project is about the implementation of a Database Management System in Video parlour System. The project replaces the Traditional method of Storing and Managing Data with a DBMS.

Problem Statement

Maintaining a video parlour system manually can lead to several issues and challenges. Here are some common problems faced when relying on Traditional processes:

Data Inaccuracy and Inconsistency: Manual entry of data can result in errors, typos, or omissions, leading to inaccurate and inconsistent information. This can cause confusion, wrong decisions, and difficulties in managing inventory, customer records, and financial transactions.

Difficulty in Data Retrieval: Finding specific information manually can be time-consuming and prone to errors. Searching through physical records or documents may not be efficient when handling a large video collection or a significant customer base. It can result in delays in serving customers and may lead to a poor customer experience.

Lack of Real-time Updates: Traditional systems require time and effort to update information, such as video availability, or customer records. As a result, real-time updates may not be feasible, leading to outdated data and potential conflicts when managing reservations or rentals.

Limited Scalability: Traditional systems may struggle to handle the increasing volume of data and growing customer base. As the video parlour expands, managing manual records becomes more challenging and time-consuming. It can hinder efficient operations and limit the ability to accommodate business growth.

Uses of the Project

A video parlour system in a database management system (DBMS) can have several uses and benefits. Here are some common uses of a video parlour system:

- Video Inventory Management: A video parlour system helps manage the inventory of videos available in the parlour. It allows tracking and organizing videos based on various criteria such as title, actor, release date, and availability status. This helps in efficient management of video collections and ensures that customers can easily find the desired videos.
- Customer Management: The video parlour system can store and manage customer information, including personal details, contact informationand payment details. This information helps in providing personalized services, targeted marketing, and maintaining customer relationships.
- Reporting and Analytics: A video parlour system can generate reports and provide analytics on various aspects such as popular videos, customer preferences, etc. These insights assist in making informed decisions, optimizing operations, and identifying business opportunities.
- Billing and Financial Management: The system can handle billing processes, generating invoices or receipts and other charges. It can also integrate with payment systems to facilitate secure and efficient financial transactions.

Background

The project uses the following technologies.

DBMS:Oracle

Oracle is a leading relational database management system (RDBMS) developed by Oracle Corporation. It is widely used in enterprise-level applications and is known for its reliability, scalability, and comprehensive feature set. Here are some key points about Oracle:

- Relational Database Management System: Oracle is an RDBMS that organizes data into tables consisting of rows and columns, following the relational model. It provides a structured and efficient way to store, manage, and retrieve data.
- SQL and PL/SQL: Oracle uses SQL (Structured Query Language) as its primary language for interacting with the database. SQL allows users to perform various operations such as querying, inserting, updating, and deleting data. Additionally, Oracle supports PL/SQL (Procedural Language/SQL), which is an extension of SQL that allows users to write procedural logic and create stored procedures, functions, triggers, and packages.

Books

For understanding the theory of DBMS and SQL, we took the following books for reference.

1. Abraham Silberchatz, Henry F. Korth, S.Sudarshan; *Database System Concepts*. McGraw Hill, 6th ed
2. Elmasri,R., Navathe, S.B., Fundamentals of Database Systems, Fourth Edition, Pearson Education

DBMS for Video Parlour System

The development of the **Schema** for RDBMS started with the Entity-Relationship Model(ER Model). Then the ER Model was mapped to the Relational Model.

Entity-Relationship Model

The Entity-Relationship Model (ER Model) is used to describe the structure of the Database. It takes into account the following:

- Entities and Entity Sets
- The Attributes of those Entities
- Relationships among those Entities

ER Diagram

The core of the ER Model is the Entity-Relationship Diagram (ER Diagram), a graphical representation of the ER model. It is like a blueprint of a database.

Addressing the Requirements

The first step in the ER Model is to Identify the Entities of the System that we are interested in.

Entities of Video Parlour System

The **entity** is any real-world object having independent existence. An entity can be a person, place, or concept. The 6 main Entities associated with the Video Parlour System are:

1. Customer
2. Movie
3. Invoice
4. Message
5. Report
6. Authentication

Attributes

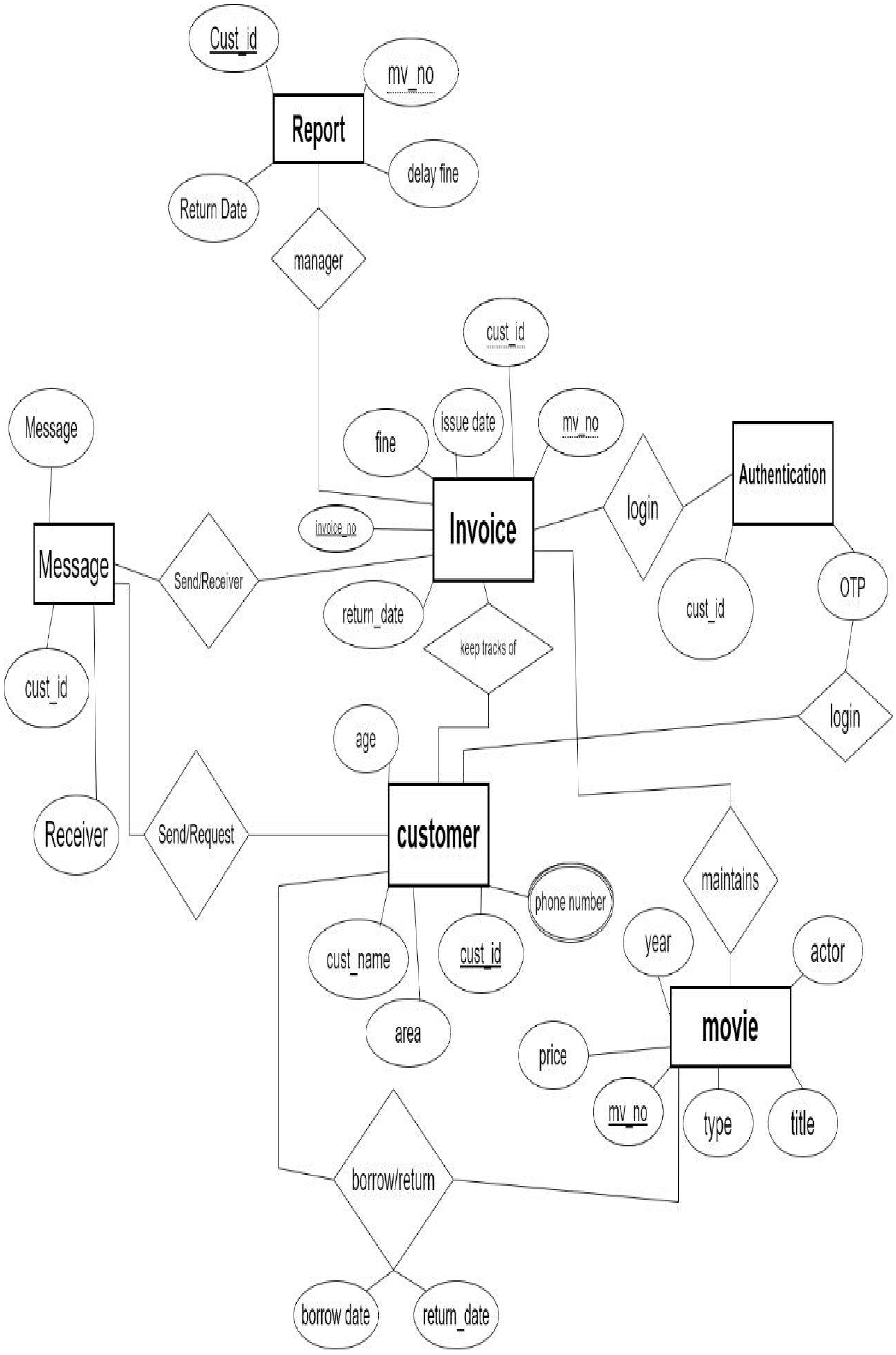
The Properties of an entity are called Attributes. The attribute(s) which can uniquely identify the Entity is called **Primary Key**. The main Attributes of all the above Entities are listed in the table below. The Primary Keys are underlined.

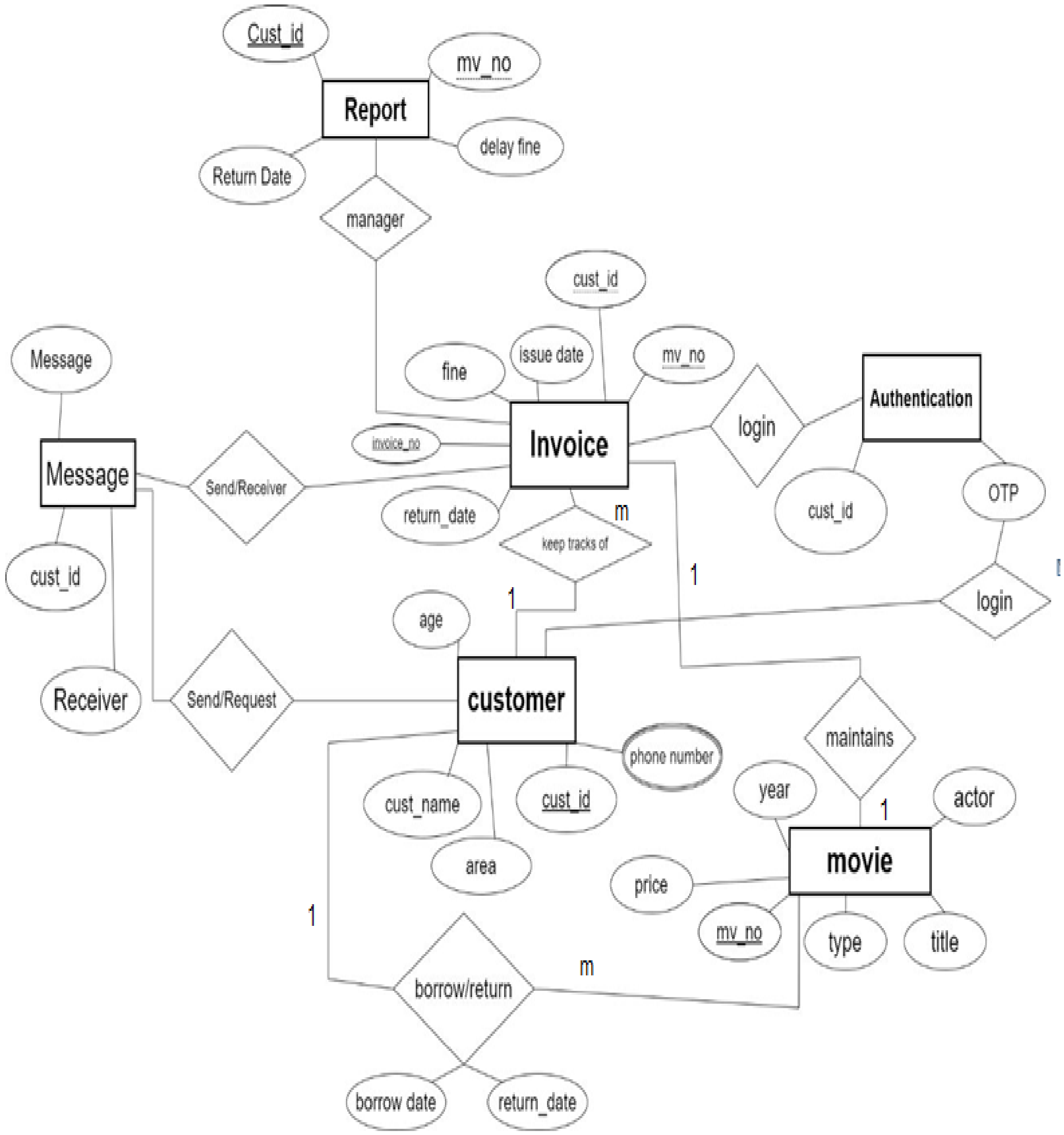
Entity	**Attributes**
Customer	Custid Cname Phone Area age
Movie	Mvno Title Actor Year price
Invoice	Invno Mvno Custid Issuedate Returndate Fine
Message	Custid Receiver message
Report	Custid

	Mvno Returndate delaytime
Authentication	Custid otp

ER Diagram -

An ER (Entity-Relationship) diagram is a visual representation of the relationships between entities in a database. It is commonly used in database design to illustrate the logical structure of a database system. ER diagrams are created using various symbols and notations to depict entities, attributes, relationships, and constraints .


Cust_id
mv_no
Report
delay fine
Return Date
manager
cust_id
fine
issue date
mv_no
Authentication
login
invoice_no
Invoice
Message
Message
Send/Receiver
return_date
cust_id
OTP
keep tracks of
cust_id
login
age
Receiver
Send/Request
customer
maintains
phone number
cust_name
cust_id
year
actor
area
movie
price
mv_no
type
title
borrow/return
borrow date
return_date


Cust_id
mv_no
Report
delay fine
Return Date
manager
cust_id
fine
issue date
mv_no
Message
Invoice
login
Authentication
invoice_no
Message
Send/Receiver
return_date
cust_id
OTP
m
keep tracks of
cust_id
1
1
login
age
Receiver
Send/Request
customer
maintains
phone number
cust_name
cust_id
year
actor
1
area
movie
price
mv_no
type
title
1
m
borrow/return
borrow date
return_date

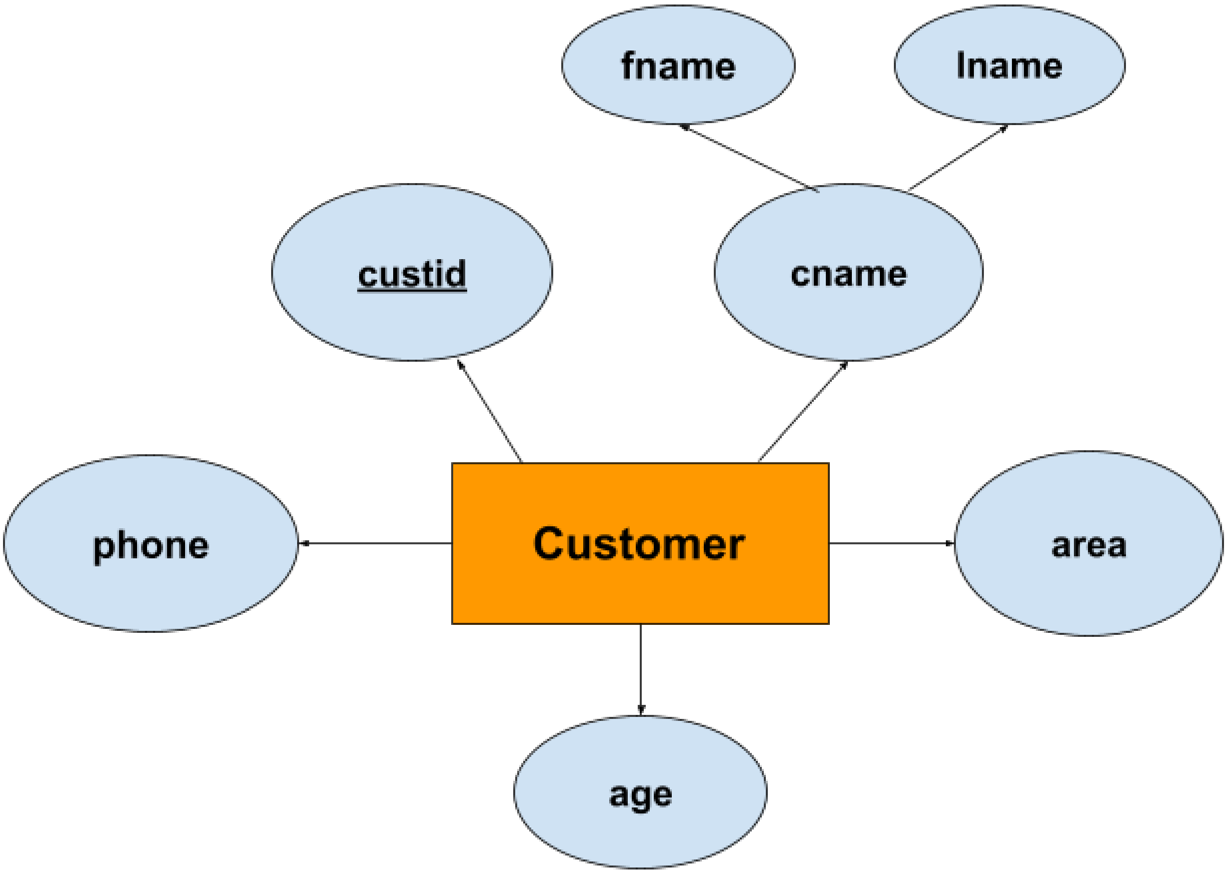
Mapping ER Model to Relational Model (Tables)

Step 1 : Customer

Customer is an entity which have attributes -

Custid , Cname , Phone , Area , Age.

We have used Custid as a primary key in the table.



Customer

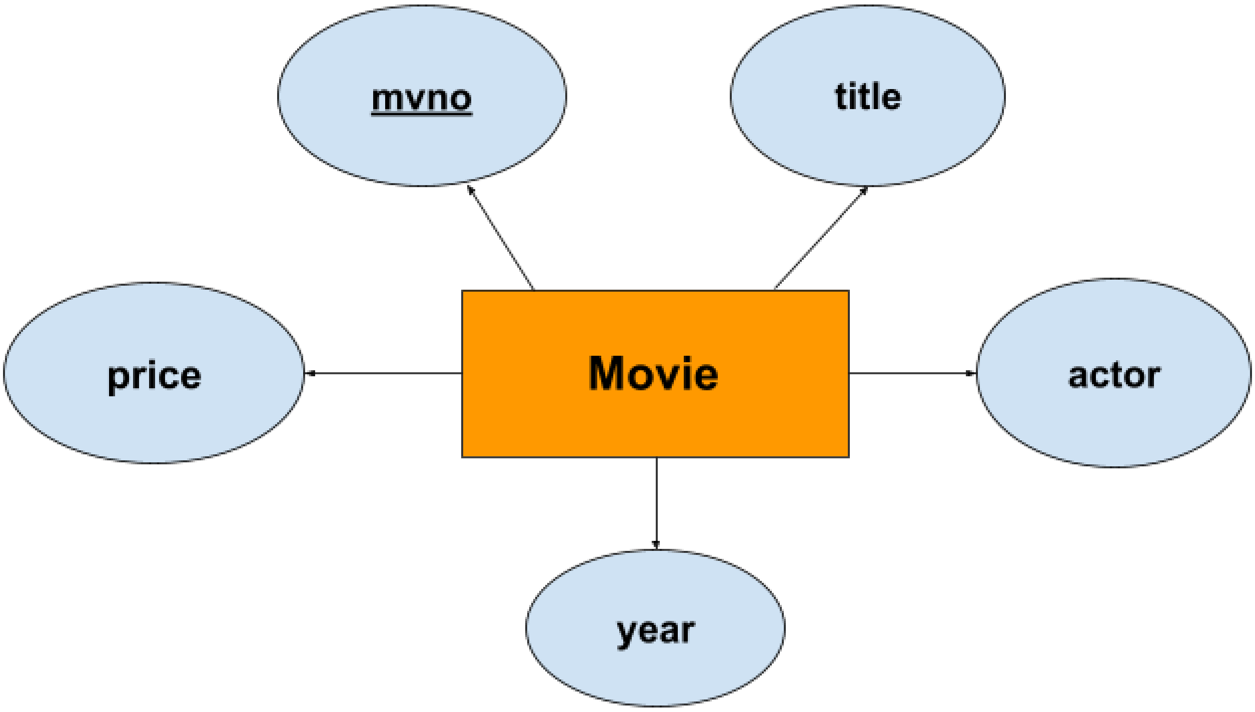
<u>custid</u>	fname	lname	age	area	phone

Step 2 : Movie

Movie is an entity with attributes –

mvno, title, type, actor, price, year.

Here, we use mvno as primary key.

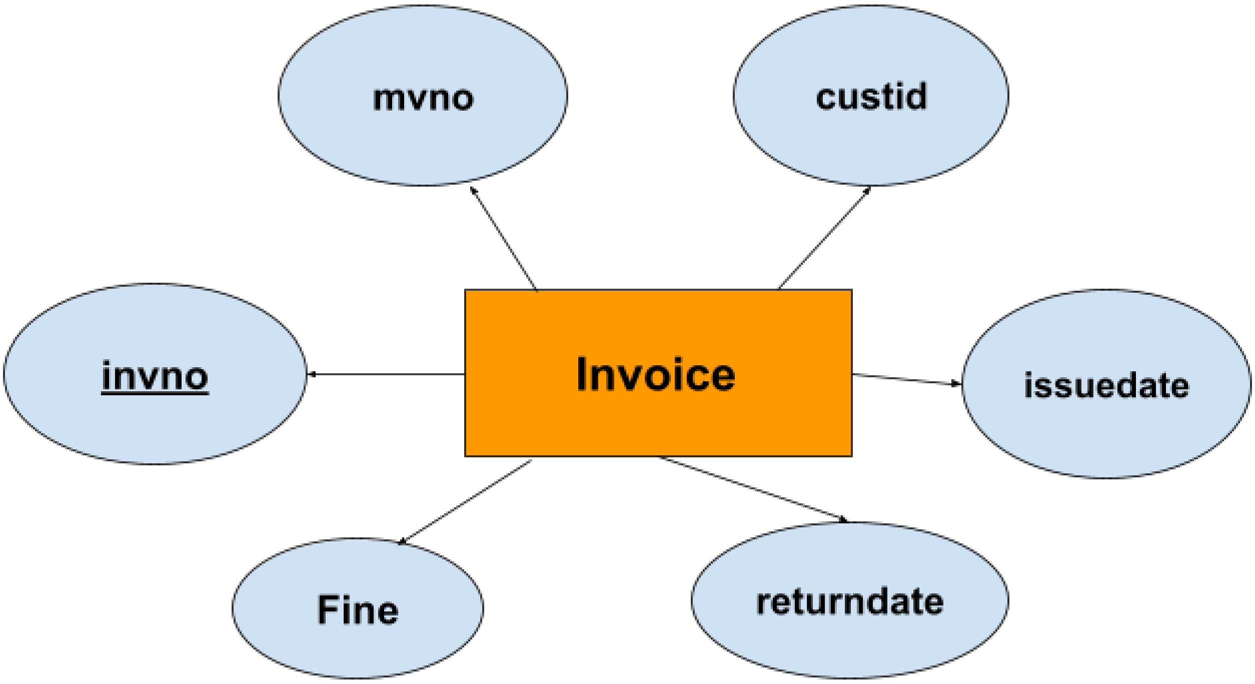


Movie

mvno	title	actor	year	price

Step 3 : Invoice

Invoice is an entity with attributes –
invno, mvno, custid, issuedate, returndate, fine.
Here, we use invno as primary key, mvno and custid as foreign key.



Invoice

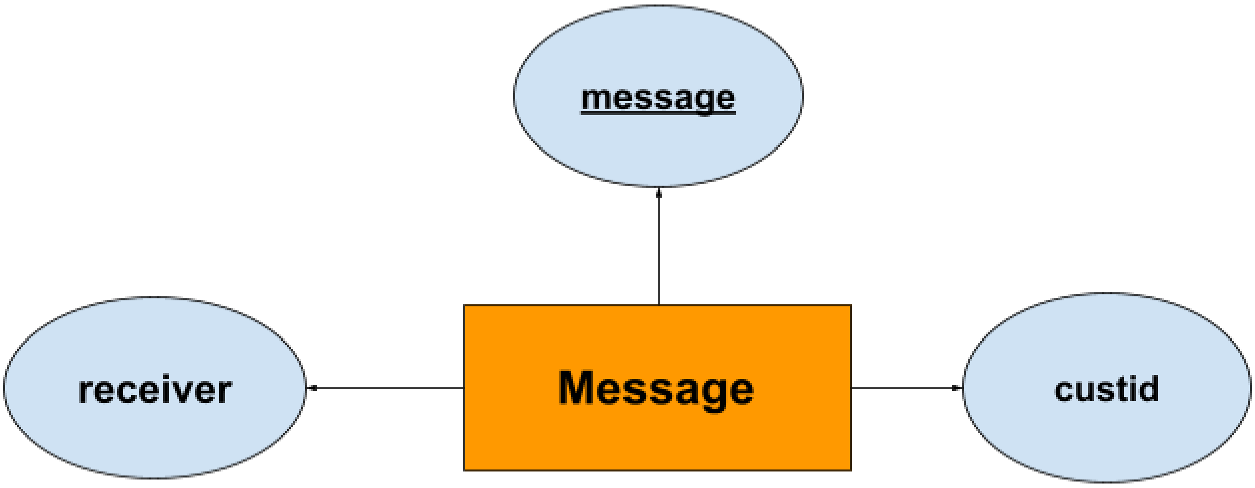
invno	mvno	custid	issuedate	returndate	Fine

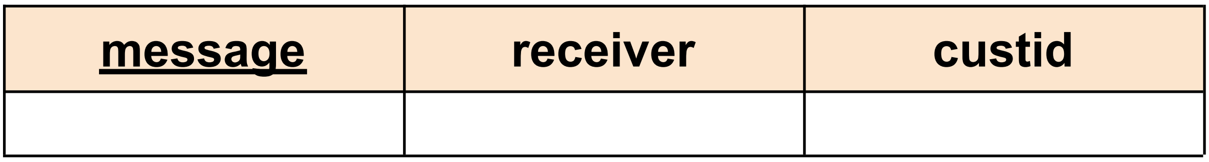
Step 4 : Message

Message is an entity which have attributes -

message, custid , receiver .

We have used message as a primary key in the table.





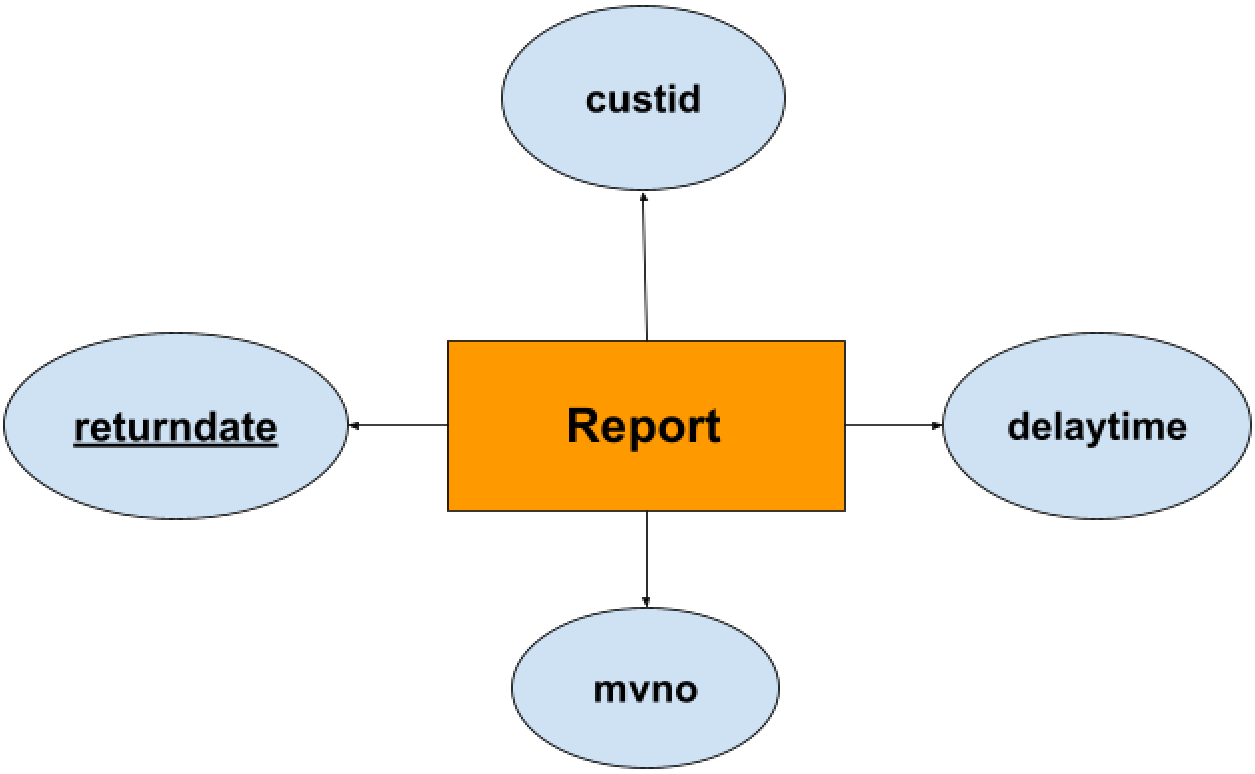
message	receiver	custid

Step 5 :Report

Report is an entity which have attributes -

custid ,mvno ,returndate ,delaytime .

We have used returndate as a primary key in the table.



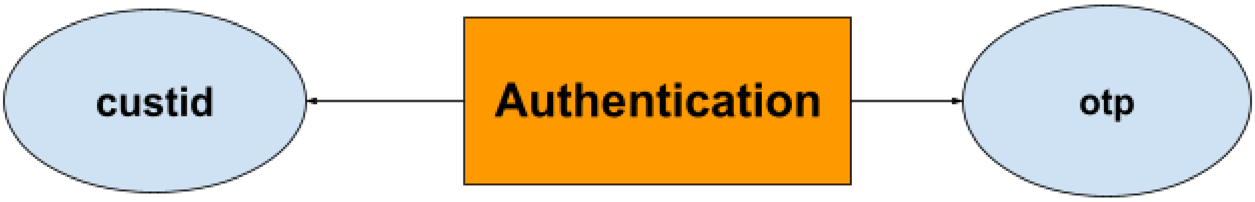
custid	mvno	returndate	delaytime

Step 6 :Authentication

Authentication is an entity which have attributes -

custid , otp .

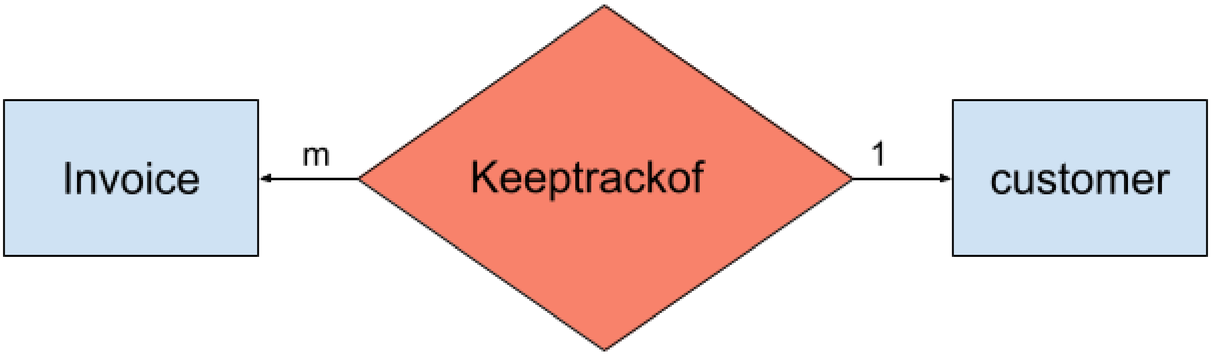
We have used custid as a primary key in the table



<u>**custid**</u>	**otp**

Step 7 : Keeptrackof :

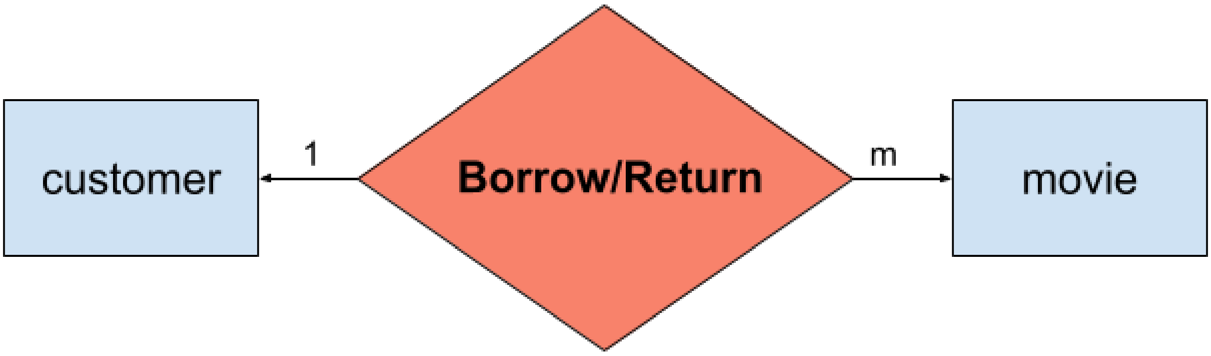
Keeptrackof is Many-to-one relationship between Invoice and customer



invno	mvno	Fine	issuedate	returndate	custid

Step 8 : Borrow / Return :

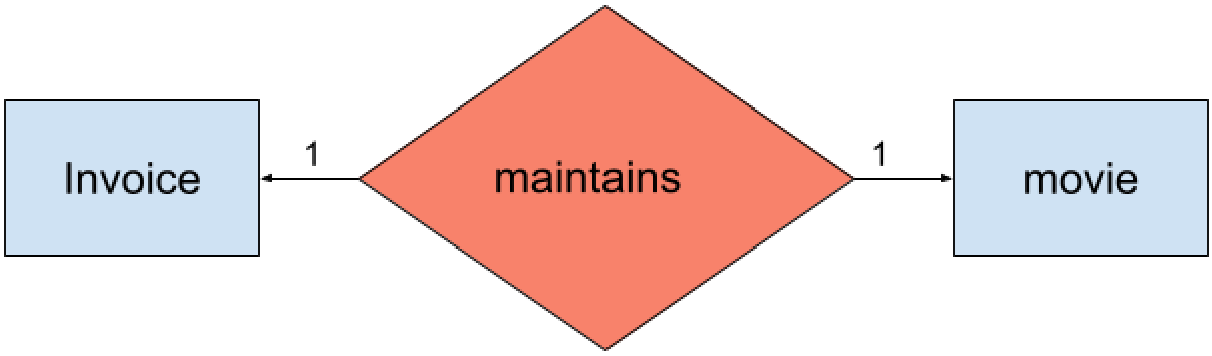
Borrow/Return is a One-to-Many relationship between customer and movie .



custid	mvno	title	type	actor	price	year

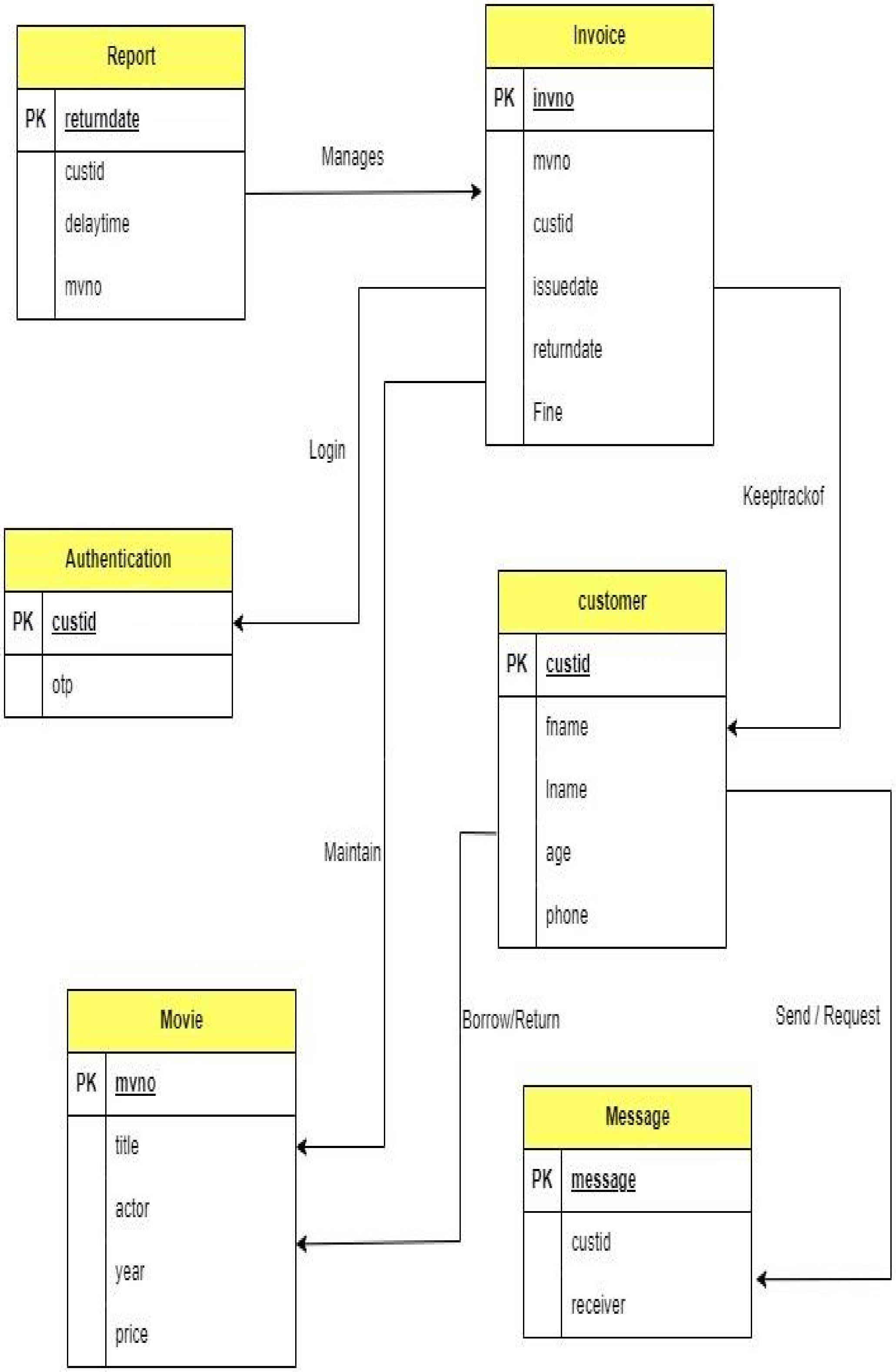
Step 9 : Maintains :

Maintains is a One-to-One relationship between invoice and movie .



invno	mvno

Schema


Report
PK returndate
custid
delaytime
mvno
Manages
Invoice
PK invno
mvno
custid
issuedate
returndate
Fine
Login
Keeptrackof
Authentication
PK custid
otp
customer
PK custid
fname
lname
age
phone
Maintain
Borrow/Return
Send / Request
Movie
PK mvno
title
actor
year
price
Message
PK message
custid
receiver

Normalization :

1. Customer :

<u>custid</u>	fname	lname	age	area	phone

1NF –

customer

<u>custid</u>	fname	lname	age	area

Phone

<u>custid</u>	phone

2NF –

Custid → fname, lname, age, area

There exists no partial dependencies.

So, this relation is already in 2NF.

3NF –

There exists no transitive dependencies .

So, this relation is already in 3NF.

BCNF –

There exists no non-trivial dependencies .

So, this relation is already in BCNF.

2. Movie:

mvno	title	actor	year	price

1NF –

Here, each attribute contains atomic value.

So, this relation is in 1NF**.**

2NF –

Mvno → title, type, actor, price, year

There exists no partial dependencies.

So, this relation is already in 2NF.

3NF –

There exists no transitive dependencies.

So, this relation is already in 3NF.

BCNF –

There exists no non-trivial dependencies.

So, this relation is already in BCNF.

3. Invoice:

invno	mvno	custid	issuedate	returndate	Fine

1NF –

Here, each attribute contains atomic value.

So, this relation is in 1NF.

2NF –

invno → issuedate, returndate, fine

invno → mvno

invno → custid

There exists no partial dependencies.

So, this relation is already in 2NF.

3NF –

There exists no transitive dependencies.

So, this relation is already in 3NF.

BCNF –

There exists no non-trivial dependencies.

So, this relation is already in BCNF.

Structured Query Language(SQL)

The main SQL commands that we used are:

DDL - Data Definition Language

In DDL, we have used CREATE, ALTER, TRUNCATE, DROP and RENAME commands.

DML - Data Manipulation Language

In DDL, we have used INSERT, UPDATE, DELETE, and SELECT commands.

Table Creation

Movie Table

```
create table Movie(
mvno varchar2(5) primary key,
title varchar2(25),
type varchar2(20),
actor varchar2(20),
price number(4),
year number(4));
```

Customer Table

```
create Table Customer(
custid varchar2(6) primary key,
cname varchar2(15),
area varchar2(15),
phone number(10),
age number(2));
```

Invoice Table

```
create table Invoice(
invno varchar2(5) primary key,
mvno varchar2(5),
custid varchar2(6),
issuedate Date,
returndate Date,
fine number(3),
foreign key (mvno) references Movie(mvno) on delete cascade,
foreign key (custid) references customer(custid ) on delete cascade);
```

Data Insertion

Inserting in Movie Table

```
insert into Movie(mvno, title, type, actor, price, year)
values('M0100','Welcome','Comedy','Akshay Kumar',350,2007);

insert into Movie(mvno, title, type, actor, price, year)
values('M0045','Dhoom 3','Action','Amir Khan',500,2013);

insert into Movie(mvno, title, type, actor, price, year)
values('M0002','The Martian','Sci Fi','Matt Damon',1000,2016);

insert into Movie(mvno, title, type, actor, price, year)
values('M0072','Kung Fu Panda','Animated','Jack
Black',398,2008);

insert into Movie(mvno, title, type, actor, price, year)
values('M0132','Raja Hindusthani','Romance','Amir
Khan',220,1996);

insert into Movie(mvno, title, type, actor, price, year)
values('M0087','Chak De India','Sports Drama','Shah Rukh
Khan',411,2007);

insert into Movie(mvno, title, type, actor, price, year)
values('M0022','The Theory ','Biographical Drama','Eddie
Redmayne',732,2014);
```

```
insert into Movie(mvno, title, type, actor, price, year)
values('M0041','Special 26','Thriller','Akshay Kumar',480,2013);

insert into Movie(mvno, title, type, actor, price, year)
values('M0112','A Beautiful Mind','Biographical Drama','Russel
Crowe',515,2001);

insert into Movie(mvno, title, type, actor, price, year)
values('M0050','Interstellar','Sci Fi','Matthew
McConaughey',800,2014);
```

Show Table :

```
SQL> select *from Movie;
```

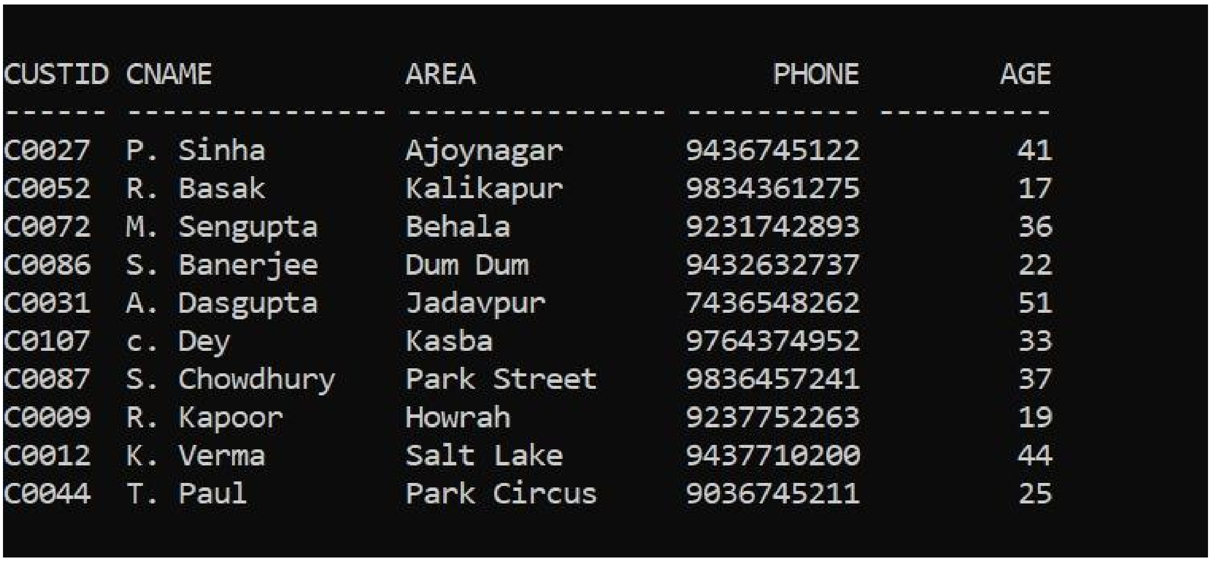
Inserting in Customer Table

```
insert into Customer(custid,cname,area,phone,age)
values('C0027','P. Sinha','Ajoynagar',9436745122,41);

insert into Customer(custid,cname,area,phone,age)
values('C0052','R. Basak','Kalikapur',9834361275,17);

insert into Customer(custid,cname,area,phone,age)
values('C0072','M. Sengupta','Behala',9231742893,36);
```

```
insert into Customer(custid,cname,area,phone,age)
values('C0086','S. Banerjee','Dum Dum',9432632737,22);

insert into Customer(custid,cname,area,phone,age)
values('C0031','A. Dasgupta','Jadavpur',7436548262,51);

insert into Customer(custid,cname,area,phone,age)
values('C0107','c. Dey','Kasba',9764374952,33);

insert into Customer(custid,cname,area,phone,age)
values('C0087','S. Chowdhury','Park Street',9836457241,37);

insert into Customer(custid,cname,area,phone,age)
values('C0009','R. Kapoor','Howrah',9237752263,19);

insert into Customer(custid,cname,area,phone,age)
values('C0012','K. Verma','Salt Lake',9437710200,44);

insert into Customer(custid,cname,area,phone,age)
values('C0044','T. Paul','Park Circus',9036745211,25);
```

Show Table :

```
SQL> select *from Customer;
```



```
CUSTID CNAME           AREA                 PHONE        AGE
------ --------------- --------------- ---------- ----------
C0027  P. Sinha        Ajoynagar       9436745122         41
C0052  R. Basak        Kalikapur       9834361275         17
C0072  M. Sengupta     Behala          9231742893         36
C0086  S. Banerjee     Dum Dum         9432632737         22
C0031  A. Dasgupta     Jadavpur        7436548262         51
C0107  c. Dey          Kasba           9764374952         33
C0087  S. Chowdhury    Park Street     9836457241         37
C0009  R. Kapoor       Howrah          9237752263         19
C0012  K. Verma        Salt Lake       9437710200         44
C0044  T. Paul         Park Circus     9036745211         25
```

Inserting in Invoice Table

```
insert into
Invoice(invno,mvno,custid,issuedate,returndate,fine)
values('I0002','M0045','C0012','28-APR-2015','03-MAY-2015',
0);
```

```
insert into
Invoice(invno,mvno,custid,issuedate,returndate,fine)
values('I0043','M0087','C0086','15-MAR-2012','22-MAR-2012'
,100);
```

```
insert into
Invoice(invno,mvno,custid,issuedate,returndate,fine)
values('I0078','M0050','C0044','18-JUN-2014','26-JUN-2014',0
);
```

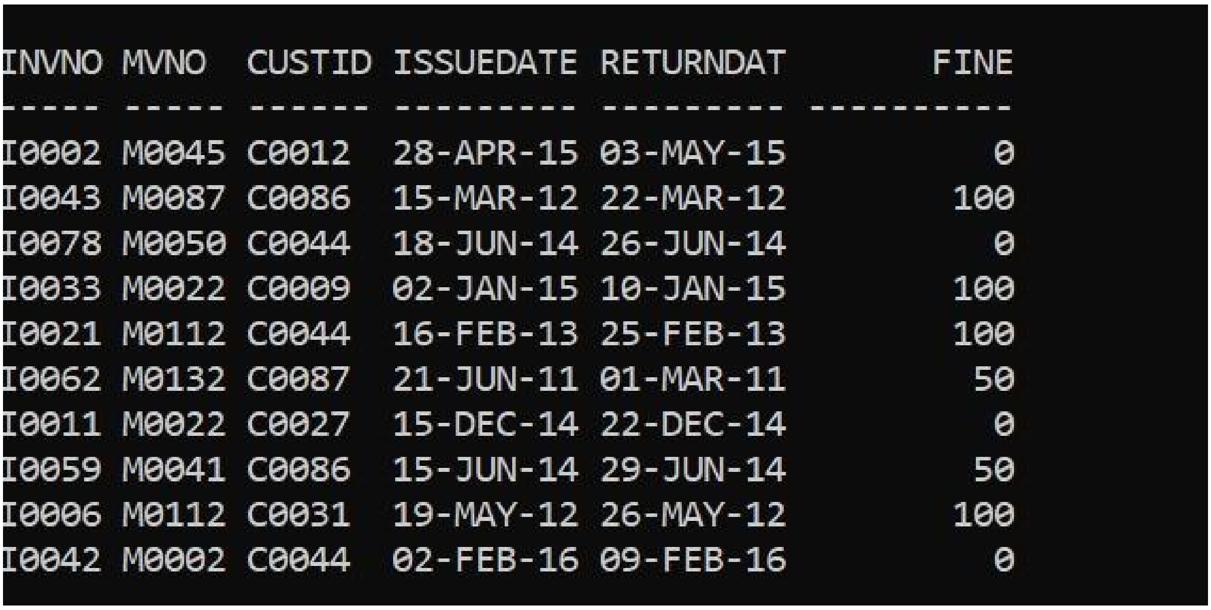
```
insert into
Invoice(invno,mvno,custid,issuedate,returndate,fine)
values('I0033','M0022','C0009','02-JAN-2015','10-JAN-2015',1
00);

insert into
Invoice(invno,mvno,custid,issuedate,returndate,fine)
values('I0021','M0112','C0044','16-FEB-2013','25-FEB-2013',1
00);

insert into
Invoice(invno,mvno,custid,issuedate,returndate,fine)
values('I0062','M0132','C0087','21-JUN-2011','01-MAR-2011',
50);

insert into
Invoice(invno,mvno,custid,issuedate,returndate,fine)
values('I0011','M0022','C0027','15-DEC-2014','22-DEC-2014',
0);

insert into
Invoice(invno,mvno,custid,issuedate,returndate,fine)
values('I0059','M0041','C0086','15-JUN-2014','29-JUN-2014',5
0);
```

insert into
Invoice(invno,mvno,custid,issuedate,returndate,fine)
values('I0006','M0112','C0031','19-MAY-2012','26-MAY-2012',
100);

insert into
Invoice(invno,mvno,custid,issuedate,returndate,fine)
values('I0042','M0002','C0044','02-FEB-2016','09-FEB-2016',0
);

Show Table :

SQL> select *from Invoice;



```
INVNO MVNO  CUSTID ISSUEDATE RETURNDAT       FINE
----- ----- ------ --------- --------- ----------
I0002 M0045 C0012  28-APR-15 03-MAY-15          0
I0043 M0087 C0086  15-MAR-12 22-MAR-12        100
I0078 M0050 C0044  18-JUN-14 26-JUN-14          0
I0033 M0022 C0009  02-JAN-15 10-JAN-15        100
I0021 M0112 C0044  16-FEB-13 25-FEB-13        100
I0062 M0132 C0087  21-JUN-11 01-MAR-11         50
I0011 M0022 C0027  15-DEC-14 22-DEC-14          0
I0059 M0041 C0086  15-JUN-14 29-JUN-14         50
I0006 M0112 C0031  19-MAY-12 26-MAY-12        100
I0042 M0002 C0044  02-FEB-16 09-FEB-16          0
```

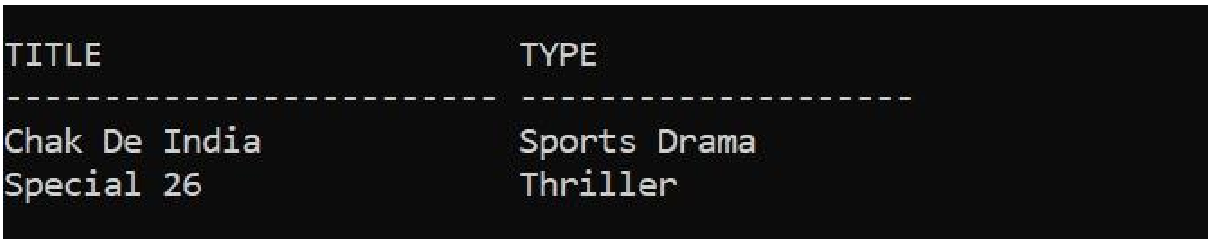
SQL QUERIES :-

(1) Find out the title, type of the movies that has been issued to S. Banerjee

Input :

SQL> select title,type from Movie where mvno in(select mvno from Invoice where custid in(select custid from Customer where cname='S. Banerjee'));

Output:



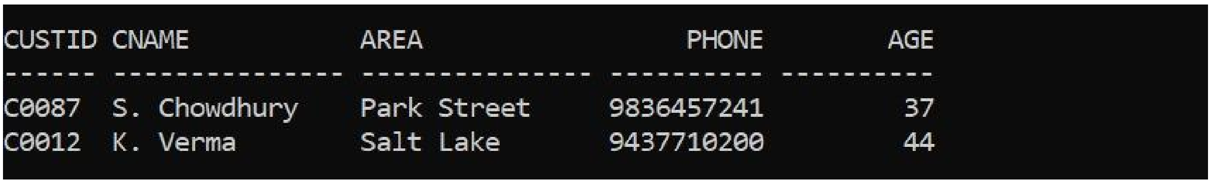
```
TITLE                     TYPE
------------------------- --------------------
Chak De India             Sports Drama
Special 26                Thriller
```

(2) Find out the customer details that have seen movies whose actor is Amir Khan

Input :

SQL> select *from Customer where custid in(select custid from Invoice where mvno in(select mvno from Movie where actor='Amir Khan'));

Output:



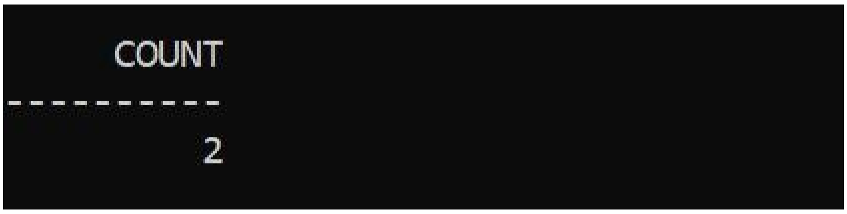
```
CUSTID CNAME           AREA                 PHONE        AGE
------ --------------- --------------- ---------- ----------
C0087  S. Chowdhury    Park Street     9836457241         37
C0012  K. Verma        Salt Lake       9437710200         44
```

(3) find Out the total number of movies that have been issued betvieen 1Sc' lune to 30" June, 2014

Input :

SQL> select count(mvno) count from Invoice where issuedate between '15-JUN-2014' and '30-JUN-2014';

Output:



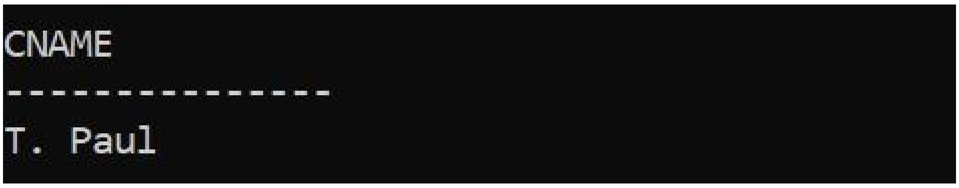
```
     COUNT
----------
         2
```

(4) Find the name of the customer who borrowed maximum movie,

Input :

SQL> select cname from Customer where custid in(select custid from Invoice group by custid having count(mvno) in(select max(count(mvno)) from Invoice group by custid));

Output:



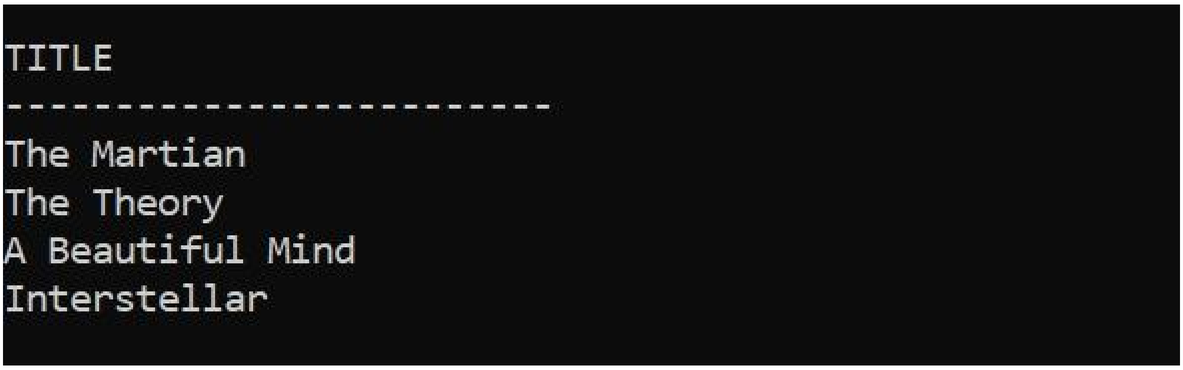
```
CNAME
---------------
T. Paul
```

(5) list all movie title which has price>500.

Input :

SQL> select title from Movie where price>500;

Output:



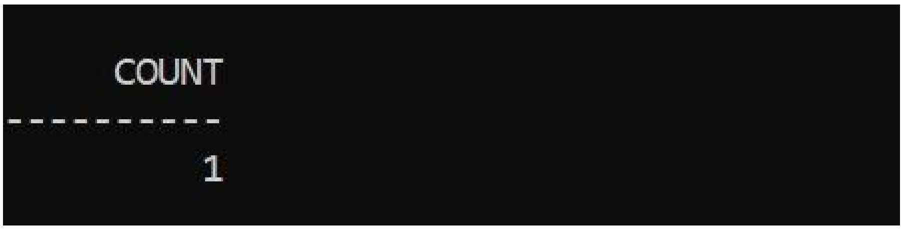
```
TITLE
-------------------------
The Martian
The Theory
A Beautiful Mind
Interstellar
```

(6) Find Out the total number of movies borrowed by Mr, S. Chowdhury.

Input :

SQL> select count(mvno) count from Invoice where custid in(select custid from Customer where cname='S. Chowdhury');

Output:



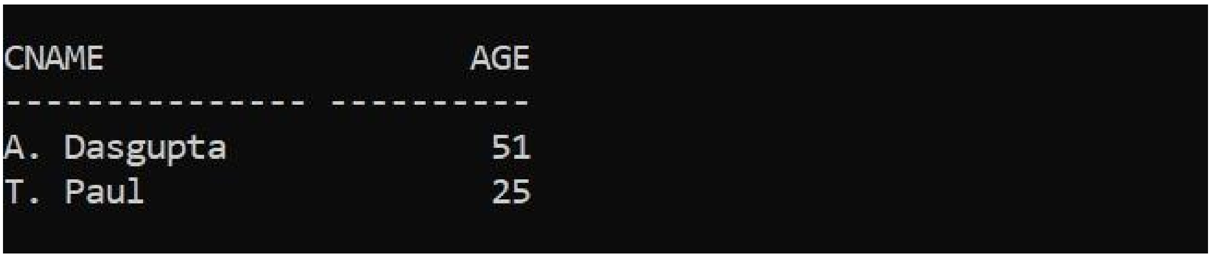
```
     COUNT
----------
         1
```

7) Find name and age of all customers who have borrowed a movie title 'A Beautiful Mind'

Input :

SQL> select cname,age from Customer where custid in(select custid from Invoice w here mvno in(select mvno from Movie where title='A Beautiful Mind'));

Output:



```
CNAME                  AGE
--------------- ----------
A. Dasgupta             51
T. Paul                 25
```

Summary

We developed a Database Management System for Video Parlour System. After identifying the main Entities and the Relationship among them, we created an ER Model with the help of an ER Diagram. Then we mapped the ER Model to the Relational Model. Then we normalized the relations till BCNF. Then we had implemented the relation as tables under RDBMS.

Eventually, we used the technologies like Oracle Databse , SQL Developer and SQL plus etc to implement the DBMS.

Now the Video Parlour System can use the DBMS for Storing, Managing, and Updating their data on a regular basis. This will help them get rid of the Data Duplication, Inconsistency and many other issues they were facing due to the use of traditional Methods.

Overall, a video parlour system in a DBMS streamlines operations, enhances customer experience, improves efficiency, and provides valuable insights for effective management and growth of the video parlour business.