University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 0

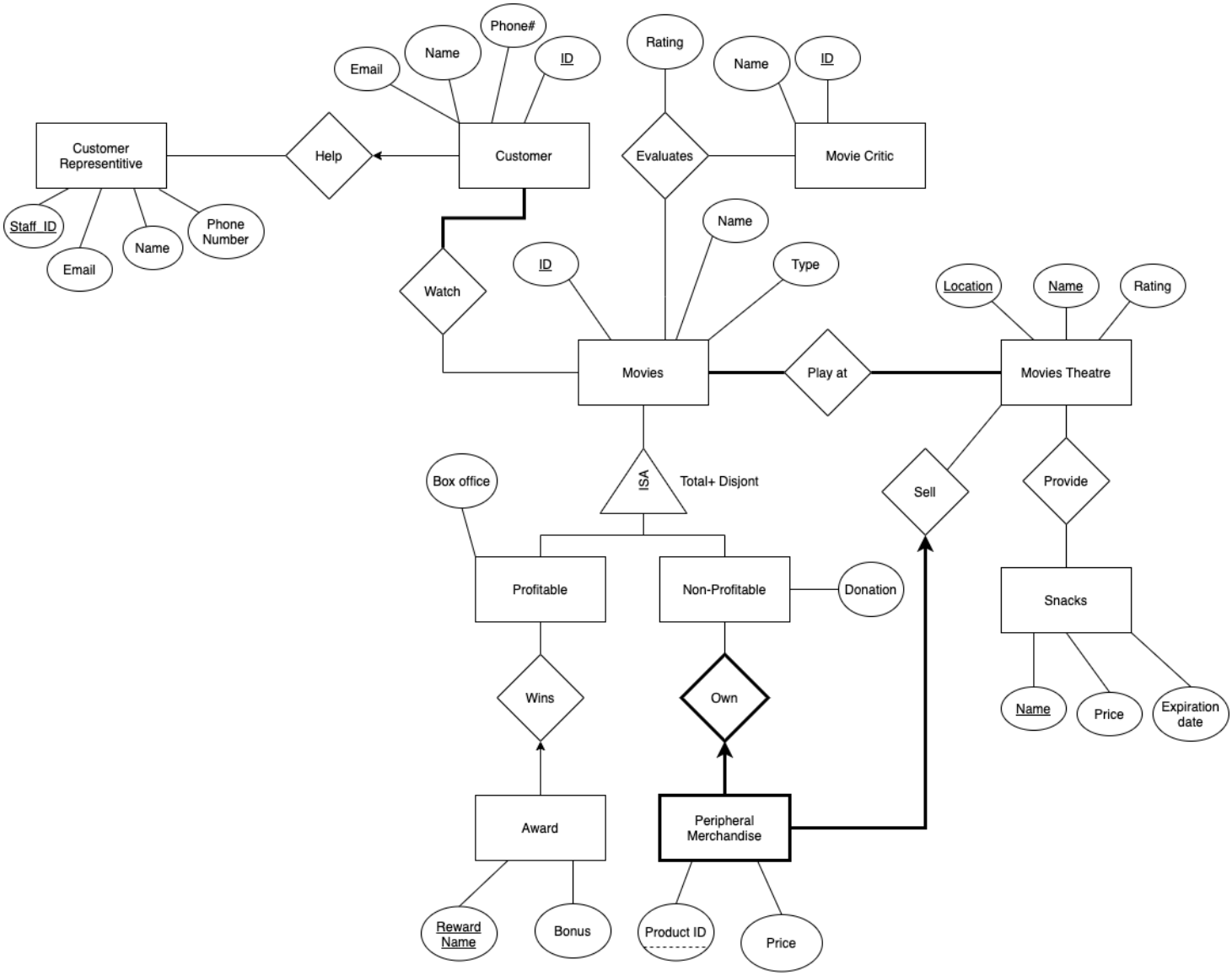
Date: Jan 18

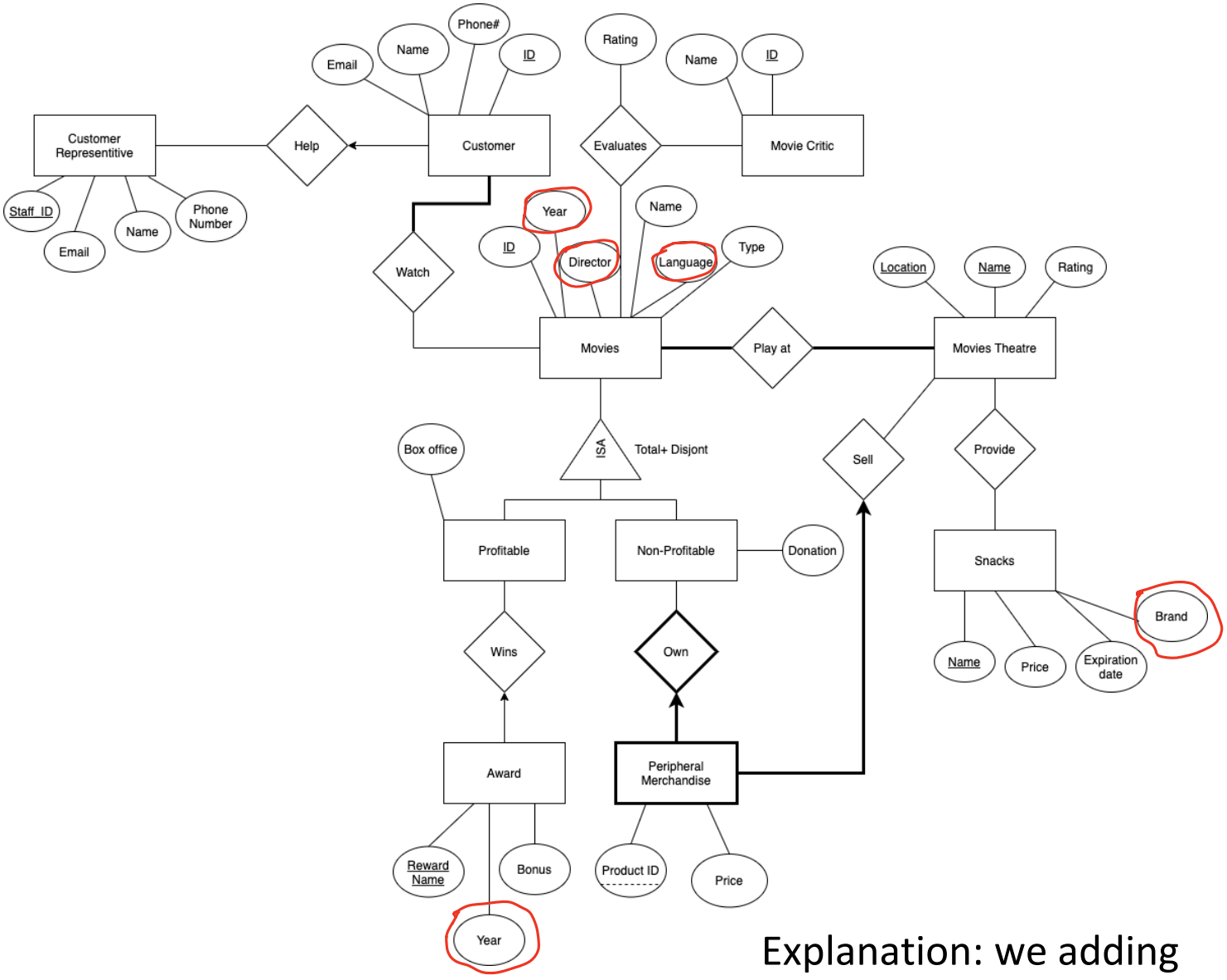
Group Number: 18

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Ruming Zhong	38917878	a7u2b	zhongroy189@gmail.com
Yibo Chen	13390794	h0m3b	jonnec0408@gmail.com
Gengyao Liu	47728829	w9y1b	liugengyao123@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia


Phone#
Name
Email
ID
Rating
Name
ID
Customer Representitive
Help
Customer
Evaluates
Movie Critic
Staff_ID
Email
Name
Phone Number
Watch
ID
Name
Type
Location
Name
Rating
Movies
Play at
Movies Theatre
ISA
Total+ Disjont
Box office
Sell
Provide
Profitable
Non-Profitable
Donation
Snacks
Wins
Own
Name
Price
Expiration date
Award
Peripheral Merchandise
Reward Name
Bonus
Product ID
Price



Explanation: we adding extra attributes in our ERD to do task 4.

Task 3:

Customer_Representitive (Staff_ID:Integer(10), Email:VarChar(20), Name:VarChar(20), Phone_num:Integer(10))
Candidate key: Staff_ID, (Email, Phone_num)

Customer_Help (ID:Integer(10), Email:VarChar(20), Name:VarChar(20), Phone_num:Integer(10), **Staff_ID**:Integer(10))
Candidate key: ID, Email, Phone_num

Movie_Critic (Id:Integer(10), Name:VarChar(20))
Candidate key: ID

Evaluates (Rating: Integer(3), **MCID**: Integer(10), **MID**: Integer(10))
Candidate key: (MCID, MID)

Watch (**CID**: Integer(10), **MID**: Integer(10))
Candidate key:(CID, MID)

Movie(ID:Integer(10), Name:VarChar(20), Type:VarChar(15), Director: VarChar(10), Language: VarChar(10), Year: Integer(4))
Candidate key: ID

Profitable_Movie (**ID**:Integer(10), Box_office: Integer(10))
Candidate key: ID

Non-Profitable_Movie (**ID**:Integer(10), Donation: Integer(10))
Candidate key: ID

Peripheral_Merchandise_Sell_Own (**MID**: Integer(10), Product_ID: Integer(10), Price: Integer(10), **Location:** VarChar(20)**, Name:** VarChar(20))
Location and Name Not Null.
Candidate key: (MID, Product ID)

Award_Win (Reward_Name: VarChar(20), Bonus: Integer(10), **MID**: Integer(10), Year: Integer(4))
Candidate key: Reward_Name

Snacks (Name: VarChar(20), Price: Integer(2), Expiration_date:Date, Brand: VarChar(15))
Candidate key: Name

Provide (**SName:** VarChar(20), **Location:** VarChar(20), **MT_Name:** VarChar(20))
Candidate key: (SName, location, MT_name)

Movies_Theatre (Location: VarChar(20), Name: VarChar(20), Rating: Integer (3))
Candidate key: (Name, Location)

Play_at (**MID**: Integer(10), **Location:** VarChar(20), **Name**: VarChar(20))
Candidate key: (MID, location, Name)

Task 4:

Customer_Representitive:
Staff_ID -> Staff_ID, Email, Name, Phone_num
Email, Phone_num -> Name, Staff_ID, Email, Phone_num
Phone_num -> Name

Customer_Help:
ID -> Email, ID, Name, Phone_num, Staff_ID
Email -> Email, ID, Name, Phone_num, Staff_ID
Phone_num -> Email, ID, Name, Phone_num, Staff_ID

Movie_Critic:
ID -> ID, Name

Evaluates:
MCID, MID -> Rating, MCID, MID

Watch:
CID,MID -> CID, MID

Movie:
ID-> ID, Name, Type, Director, Year, Language
Name, Director -> Type, Language

Profitable_Movie:
ID-> ID, Box_Office

Non-Profitable_Movie:
ID-> ID, Donation

Peripheral_Merchandise_Sell_Own:
MID, ProductID -> MID, ProductID, Price, Location, Name

Award_Win:
Reward_name -> Reward_name, Bonus, MID, Year

Snacks:

Name -> Price, Name, Expiration_date, Brand

Provide:
SName, Location, MT_Name -> SName, Location, MT_Name

Movies_Theatre:
Location, Name -> Rating, Location, Name

Play_at:
MID, Location, Name -> MID, Location, Name

Task 5:

Before normalization:

Customer_Representitive (Staff_ID:Integer(10), Email:VarChar(20), Name:VarChar(20), Phone_num:Integer(10))
Candidate key: Staff_ID, (Email, Phone_num)

Customer_Representitive:
Staff_ID -> Staff_ID, Email, Name, Phone_num
Email, Phone_num -> Name, Staff_ID, Email, Phone_num
Phone_num -> Name

Since the functional dependency Phone_num -> Name violate the BCNF, we need to separate this relation. Therefore, the original schema, Customer_Representitive (Staff_ID, Email, Name, Phone_num, becomes 2 tables, which are Customer_Representitive_1 (Staff_ID, Email, Phone_num and Customer_Representitive_2 (Name, Phone_num)

Movie(ID:Integer(10), Name:VarChar(20), Type:VarChar(15), Director: VarChar(10), Language: VarChar(10), Year: Integer(4))
Candidate key: ID

Movie:
ID-> ID, Name, Type, Director, Year, Language
Name, Director> Type, Language

Since the functional dependency Name, Director -> Type, Language violates the BCNF, we need to separate this relation into two tables. The original schema is, Movie(ID, Name, Type, Director, Language, Year), according to this functional dependency, we separate the table to, Movie_1(ID, Name, Director, Year) and Movie_2(Name, Type, Director, Language).

After normalization to BCNF:

Customer_Representitive_1 (Staff_ID:Integer(10), Email:VarChar(20),Phone_num:Integer(10))
Candidate key: Staff_ID, Email

Customer_Representitive_2 (Name:VarChar(20), Phone_num:Integer(10))
Candidate key: Phone_num

Movie_1(ID:Integer(10), Name:VarChar(20), Director: VarChar(10) Year: Integer(4))
Candidate key: ID

Movie_2(Name:VarChar(20), Type:VarChar(15), Director: VarChar(10), Language: VarChar(10))
Candidate key: (Name, Director)

Task 6:

```
CREATE TABLE Customer_Representitive_1(
Staff_ID Integer(10) PRIMARY KEY,
Email VarChar(20),
Phone_num Integer(10),
Unique(Email, Phone_num)
);

CREATE TABLE Customer_Representitive_2(
Phone_num Integer(10)  PRIMARY KEY,
Name VarChar(20)
);

CREATE TABLE Customer_Help (
ID Integer(10) PRIMARY KEY,
Email VarChar(20) Unique,
Staff_ID Integer(10),
Phone_num Integer(10) Unique,
Name VarChar(20),
FOREIGN KEY(Staff_ID) REFERENCES Customer_Representitive_1 (Staff_ID)
ON DELETE CASCADE
ON UPDATE CASCADE
);


CREATE TABLE Movie_Critic (
ID Integer(10) PRIMARY KEY,
Name VarChar(20)
);

CREATE TABLE Evaluates (
Rating Integer(3),
MCID Integer(10),
MID Integer(10),
PRIMARY KEY(MCID,MID),
FOREIGN KEY(MCID) REFERENCES Movie_Critic (ID)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY(MID) REFERENCES Movie_1 (ID)
ON DELETE CASCADE
ON UPDATE CASCADE
```

```
);

CREATE TABLE Watch (
CID Integer(10),
MID Integer(10),
PRIMARY KEY(CID,MID),
FOREIGN KEY(CID) REFERENCES Customer_Help (ID)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY(MID) REFERENCES Movie_1 (ID)
ON DELETE CASCADE
ON UPDATE CASCADE
);


CREATE TABLE Movie_1 (
ID Integer(10) PRIMARY KEY,
Name VarChar(20),
Year Integer(4),
Director VarChar(10)
);

CREATE TABLE Movie_2 (
Name VarChar(20),
Type VarChar(15),
Director VarChar(10),
Language VarChar(10),
PRIMARY KEY (Name, Director)
);

CREATE TABLE Profitable_Movie (
ID Integer(10) PRIMARY KEY,
Box_office Integer(10),
FOREIGN KEY(ID) REFERENCES Movie_1 (ID)
ON DELETE CASCADE
ON UPDATE CASCADE
);

CREATE TABLE Non_Profitable_Movie (
ID Integer(10) PRIMARY KEY,
Donation  Integer(10),
FOREIGN KEY(ID) REFERENCES Movie_1 (ID)
ON DELETE CASCADE
ON UPDATE CASCADE
```

```
);

CREATE TABLE Peripheral_Merchandise_Sell_Own (
MID Integer(10),
Product_ID Integer(10),
Price Integer (10),
Location VarChar(20) NOT NULL,
Name VarChar(20) NOT NULL,
PRIMARY KEY(MID, Product_ID),
FOREIGN KEY(MID) REFERENCES Movie_1 (ID)
ON DELETE NO ACTION
ON UPDATE CASCADE,
FOREIGN KEY(Name, Location) REFERENCES Movies_Theatre (Name, Location)
ON DELETE NO ACTION
ON UPDATE CASCADE
);


CREATE TABLE Award_Win (
Reward_Name VarChar(20) PRIMARY KEY,
Bonus Integer(10),
MID Integer(10),
Year Integer(4),
FOREIGN KEY (MID) REFERENCES Profitable_Movie (ID)
ON DELETE CASCADE
ON UPDATE CASCADE
);

CREATE TABLE Snacks (
Name VarChar(20) PRIMARY KEY,
Price Integer(2),
Expiration_date Date,
Brand VarChar(15)
);

CREATE TABLE Provide (
SName VarChar(20),
MT_Name VarChar(20),
Location VarChar(20),
PRIMARY KEY(SName, MT_Name, Location),
FOREIGN KEY (SName) REFERENCES Snacks (Name)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY(SName, Location) REFERENCES Movies_Theatre (Name, Location)
```

```
ON DELETE CASCADE
ON UPDATE CASCADE
);


CREATE TABLE Movies_Theatre (
Location VarChar(20), Name VarChar(20),
Rating_Num Integer(3),
PRIMARY KEY(Name, Location)
);


CREATE TABLE Play_at (
Location VarChar(20),
Name VarChar(20),
MID Integer(10),
PRIMARY KEY(Name, Location, MID),
FOREIGN KEY(Name, Location) REFERENCES Movies_Theatre (Name, Location)
ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY(MID) REFERENCES Movie_1 (ID)
ON DELETE CASCADE
ON UPDATE CASCADE
);
```

Task 7:

Table Customer_Representitive_1

Staff_ID	Email	Phone#
1	1@gmail.com	7785110323
2	2@gmail.com	7785431342
3	3@gmail.com	7782442349
4	4@gmail.com	7789426455
5	5@gmail.com	7784625332

Table Customer_Representitive_2

Phone#	Name
7785110323	BigApple
7785431342	BigPen
7782442349	RunningMan
7789426455	HappyCat
7784625332	BigApple

Table Customer_Help

ID	Email	Phone#	Name	**Staff_ID**
1	c1@gmail.com	7784563625	James	1
2	c2@gmail.com	7786449151	Chole	2
3	c3@gmail.com	7784336315	Fandy	3
4	c4@gmail.com	7789445613	Kevin	4
5	c5@gmail.com	7786346354	Delvin	5
6	c6@gmail.com	7786354143	Dustin	6
7	c7@gmail.com	7789635143	Daniel	1
8	c8@gmail.com	7781236359	Jermise	2
9	c9@gmail.com	7785346517	Jimmy	3
10	c10@gmail.com	7788876351	Chole	4

Table Movie_Critic

ID	Name
1	Jeff
2	David
3	Jimmy
4	Susan
5	Sandy

Table Evaluates

MCID	Rating	**MID**
1	9	3
2	8	2
3	9	1
4	7	4
5	8	5

Table Watch

CID	**MID**
1	2
2	3
3	1
4	5
5	4

Table Profitable_Movie

ID	Box_office
1	500000000
2	2000000000
3	300000000
4	200000000
5	300000000

Table Non-Profitable_Movie

ID	Donation
6	50000000
7	70000000
8	55000000
9	65000000
10	35000000

Table Movie_1

ID	Name	Year	Director
1	ManchesterByTheSea	1997	Jeff
2	Léon	1992	David
3	FutureWaterWorld	2007	Micheal
4	TomAndJerry	2017	Zac
5	MonsterHunter	2020	Sam
6	AWonderfulDay	1998	Bob
7	RunningMan	2019	Justin
8	Victoria'sDream	2011	Jim
9	Monkey'sComing	2016	Lebron
10	IronMan	2019	Kim

Table Movie_2

Name	Director	Type	Languague
ManchesterByTheSea	Jeff	RomanticFilm	English
Léon	David	RomanticFilm	French
FutureWaterWorld	Micheal	AdventureFilm	English
TomAndJerry	Zac	ComedyFilm	Japanese
MonsterHunter	Sam	ActionFilm	English
AWonderfulDay	Bob	RomanticFilm	English
RunningMan	Justin	ComedyFilm	Krean
Victoria'sDream	Jim	RomanticFilm	French
Monkey'sComing	Lebron	ComedyFilm	Chinese
IronMan	Kim	ActionFilm	English

Table Peripheral_Merchandise_Sell_Own

MID	Product_ID	Price	**Location**	**Name**
6	1	20	2333 Robinson Rd	WonderfulPoster
7	2	50	1032 Champlain St	RunningDog
8	3	80	1928 Steve St	Victoria'sPillow
9	4	150	1545 Melform Rd	Monkey'sFamily
10	5	400	1211 New York St	IronMan

Table Award_Win

Reward_Name	Bonus	**MID**	Year
BestMovies	1000000	3	2008
GoldenMovie	2000000	2	1993
ValuableMovie	1200000	5	2021
ChampionMovie	1500000	1	1999
RecommendedMovie	900000	4	2019

Table Snacks

Name	Price	Expiration_date	Brand
PopCorn	5	2021-3-10	NiceDay
Coke	3	2021-3-11	Pepsi
Crisps	3	2021-3-12	McDonalds
FrenchFries	4	2021-3-9	McDonalds
IceCreams	4	2021-3-15	Häagen-Dazs

Table Provide

SName	**Location**	**MT_Name**
PopCorn	2333 Robinson Rd	Ocean Theatre
PopCorn	1032 Champlain St	Champlain Theatre
PopCorn	1928 Steve St	Shelton Theatre
Coke	1545 Melform Rd	Champlain Theatre
Coke	1211 New York St	Shelton Theatre
Coke	1211 New York St	Ocean Theatre
FrenchFries	1032 Champlain St	Champlain Theatre
FrenchFries	1928 Steve St	Shelton Theatre
Crisps	1545 Melform Rd	Champlain Theatre
Crisps	1211 New York St	Shelton Theatre
IceCreams	2333 Robinson Rd	Ocean Theatre
IceCreams	1211 New York St	Ocean Theatre

Table Movies_Theatre

Location	Name	Rating
2333 Robinson Rd	Ocean Theatre	10
1032 Champlain St	Champlain Theatre	9
1928 Steve St	Shelton Theatre	9
1545 Melform Rd	Champlain Theatre	10
1211 New York St	Shelton Theatre	9
1211 New York St	Ocean Theatre	9

Table Play_at

MID	**Location**	**Name**
1	2333 Robinson Rd	Ocean Theatre
2	1032 Champlain St	Champlain Theatre
3	1928 Steve St	Shelton Theatre
4	1545 Melform Rd	Champlain Theatre
5	1211 New York St	Shelton Theatre
6	1211 New York St	Ocean Theatre
7	2333 Robinson Rd	Ocean Theatre
8	1032 Champlain St	Champlain Theatre
9	1928 Steve St	Shelton Theatre
10	1545 Melform Rd	Champlain Theatre

Task 8:

Insertion: We want to add new staff to our customer_representitive_1 with his staff id of 1, email 1@gmail.com and phone with 7785110323.

Deletion: We want to delete the movie id of 4, the name “Tom And Jerry”, the Year 2017, and the director Zac from our movie_1 list.

Update: We wants to update customer information in customer_help list where id is 3, email is c3@gmail.com, the phone number is 7784336315, the name is Fandy, and corresponding staff id is 3 to the new email cuq5@gmail.com, and new phone 7788876971

Selection: We want to see which movies ID have box office that more than 300000000 on our profitable movie list.

Projection: We want to see what kinds of Award is store on our Award list.

Join: we want to join the movie_1 table and profitable_Movie table, by looking that movie IDs are equal, to find the name and the box office of the movies that have more than 350000000 box office.

Division: find the locations that provide all kind of snacks