

**Introduction to Database (CCCS215 Project)**

**Cinema management system**

By

Arjwan alqarni 2110253

Aryam AL-Sulami 2110099

Noura Ahmed Aljehani 2111335

Wrood Fawaz Allibdi 2112158

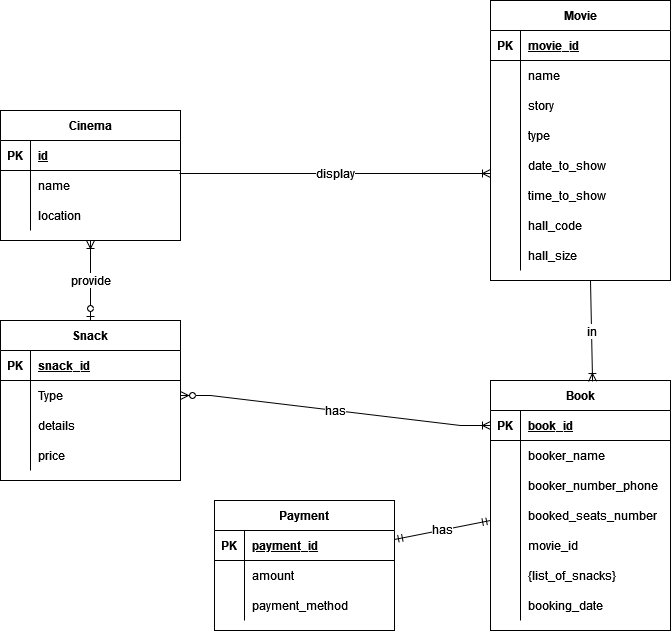
# Problem description

Our idea about cinema. We can solve the problem of security of cinemas applications by improving the security and services. We made many options to users to book any movie and enjoy it. We also have snacks like popcorn and drinks.

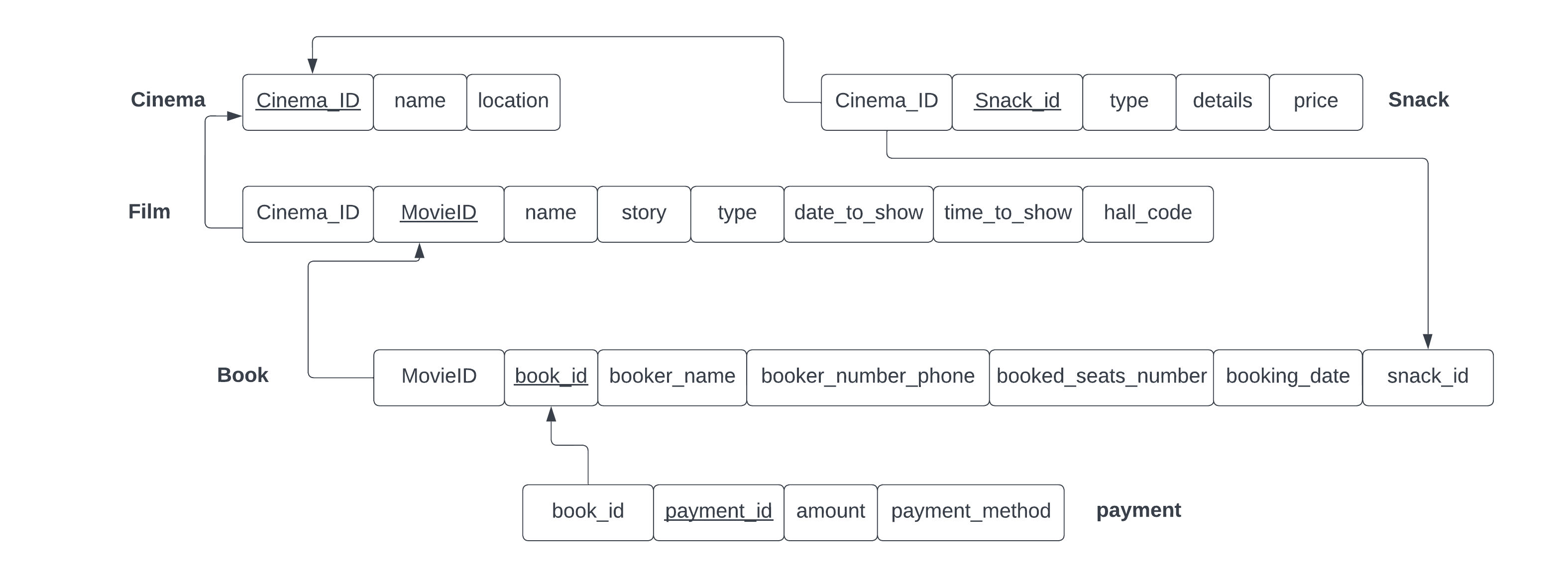
# Business rule

* Cinema displays many Movie
* A movie can be booked by many bookers
* Cinema has many book records that holds needed information
* Cinema provides many different type of snacks
* Snacks can be ordered in book record
* Each reservation should be paid. Each payment is for one reservation

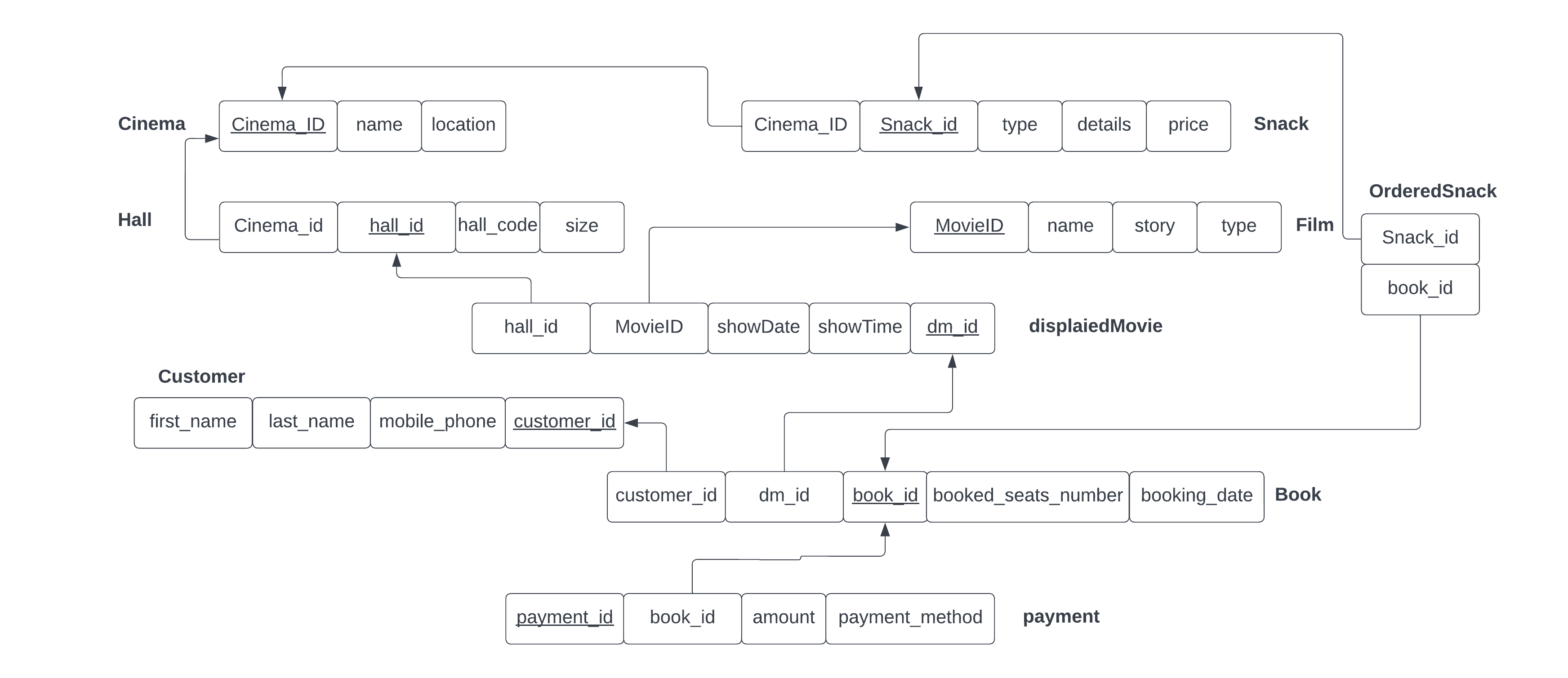
# ER diagram



# Relations



# Normalization



Now we have relations that are in the **3NF** as you can see:

* there is *no repeating group*
* all of them are full dependency
* no transitive

# Normalized Tables

|  |  |  |
| --- | --- | --- |
| Cinema | | |
| Column name | Data type Size | Constraints |
| Cinema\_id | Int | Primary key |
| Name | Varchar(50) | Not null |
| Location | Varchar(256) | Not null |

|  |  |  |
| --- | --- | --- |
| Hall | | |
| Column name | Data type Size | Constraints |
| Hall\_id | Int | Primary key |
| Hall\_code | Varchar(10) | Not null |
| Hall\_size | int | Not null |
| Cinema\_id | Int | Foreign key on Cinema |

|  |  |  |
| --- | --- | --- |
| Movie | | |
| Column name | Data type Size | Constraints |
| Movie\_id | Int | Primary key |
| name | Varchar(50) | Not null |
| story | Varchar(256) | Not null |
| type | Varchar(25) | Not null |

|  |  |  |
| --- | --- | --- |
| DisplaiedMovie | | |
| Column name | Data type Size | Constraints |
| Dm\_id | Int | Primary key |
| Date\_to\_show | Date | Not null |
| Time\_to\_show | Varchar(10) | Not null |
| Movie\_id | Int | Foreign key on Movie |
| Hall\_id | Int | Foreign key on Hall |

|  |  |  |
| --- | --- | --- |
| Customer | | |
| Column name | Data type Size | Constraints |
| Customer\_id | Int | Primary key |
| First\_name | Varchar(25) | Not null |
| Last\_name | Varchar(25) | Not null |
| Mobile\_phone | Varchar(25) | Not null |

|  |  |  |
| --- | --- | --- |
| Snack | | |
| Column name | Data type Size | Constraints |
| Snack\_id | Int | Primary key |
| Type | Varchar(25) | Not null |
| Details | Varchar(256) | Not null |
| Price | Float | Not null |
| Cinema\_id | Int | Foreign key on Cinema |

|  |  |  |
| --- | --- | --- |
| Book | | |
| Column name | Data type Size | Constraints |
| Book\_id | Int | Primary key |
| Booked\_seats\_number | Int | Not null |
| Booking\_date | Date | Not null |
| customer\_id | Int | Foreign key on Customer |
| dm\_id | Int | Foreign key on DisplaiedMovie |

|  |  |  |
| --- | --- | --- |
| OrderedSnack | | |
| Column name | Data type Size | Constraints |
| Book\_id | Int | Foreign key on book |
| Snack\_id | Int | Foreign key on snack |

|  |  |  |
| --- | --- | --- |
| Payment | | |
| Column name | Data type Size | Constraints |
| Payment\_id | Int | Primary key |
| Amount | Float | Not null |
| Payment\_method | Varchar(50) | Not null |
| Book\_id | Int | Foreign key on Book |

# Functional Dependencies

* Cinema\_id 🡪 Name, Location
* Hall\_id 🡪 Hall\_code, Hall\_size, Cinema\_id
* Movie\_id 🡪 name, story, type
* Dm\_id 🡪 Date\_to\_show, Time\_to\_show, Movie\_id, Hall\_id
* Customer\_id 🡪 First\_name, Last\_name, Mobile\_phone
* Snack\_id 🡪 Type, Details, Price, Cinema\_id
* Book\_id 🡪 Booked\_seats\_number, Booking\_date, customer\_id, dm\_id
* OrderedSnack 🡪 (Book\_id, Snack\_id)
* Payment\_id 🡪 Amount, Payment\_method, Book\_id

# Creating tables

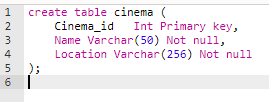
create table cinema (

Cinema\_id Int Primary key,

Name Varchar(50) Not null,

Location Varchar(256) Not null

);





create table hall (

Hall\_id Int Primary key,

Hall\_code Varchar(10) Not null,

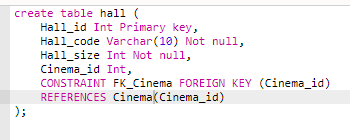
Hall\_size Int Not null,

Cinema\_id Int,

CONSTRAINT FK\_Cinema FOREIGN KEY (Cinema\_id)

REFERENCES Cinema(Cinema\_id)

);





create table movie (

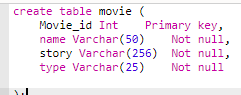
Movie\_id Int Primary key,

name Varchar(50) Not null,

story Varchar(256) Not null,

type Varchar(25) Not null,

);





create table DisplaiedMovie (

Dm\_id Int Primary key,

Date\_to\_show Date Not null,

Time\_to\_show Varchar(10) Not null,

Movie\_id Int,

Hall\_id Int,

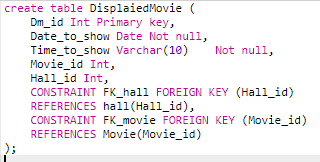
CONSTRAINT FK\_hall FOREIGN KEY (Hall\_id)

REFERENCES hall(Hall\_id),

CONSTRAINT FK\_movie FOREIGN KEY (Movie\_id)

REFERENCES Movie(Movie\_id)

);





create table Snack (

Snack\_id Int Primary key,

Type Varchar(25) Not null,

Details Varchar(256) Not null,

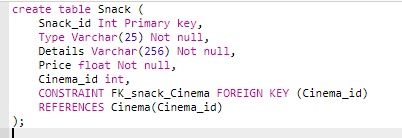
Price float Not null,

Cinema\_id int,

CONSTRAINT FK\_snack\_Cinema FOREIGN KEY (Cinema\_id)

REFERENCES Cinema(Cinema\_id)

);





create table Customer (

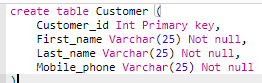
Customer\_id Int Primary key,

First\_name Varchar(25) Not null,

Last\_name Varchar(25) Not null,

Mobile\_phone Varchar(25) Not null

)





create table Book (

Book\_id Int Primary key,

Booked\_seats\_number Int Not null,

Booking\_date Date Not null,

customer\_id Int,

dm\_id Int,

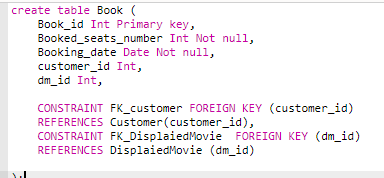
CONSTRAINT FK\_customer FOREIGN KEY (customer\_id)

REFERENCES Customer(customer\_id),

CONSTRAINT FK\_DisplaiedMovie FOREIGN KEY (dm\_id)

REFERENCES DisplaiedMovie (dm\_id)

);





create table OrderedSnack (

Snack\_id Int,

Book\_id Int,

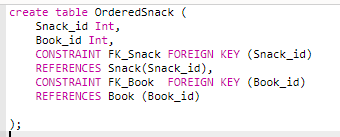
CONSTRAINT FK\_Snack FOREIGN KEY (Snack\_id)

REFERENCES Snack(Snack\_id),

CONSTRAINT FK\_Book FOREIGN KEY (Book\_id)

REFERENCES Book (Book\_id)

);





create table Payment (

Payment\_id Int Primary key,

Amount float Not null,

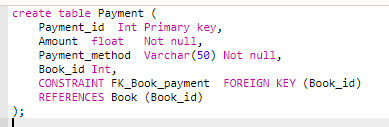
Payment\_method Varchar(50) Not null,

Book\_id Int,

CONSTRAINT FK\_Book\_payment FOREIGN KEY (Book\_id)

REFERENCES Book (Book\_id)

);



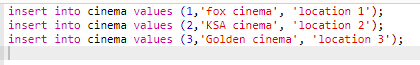


# Rows insert

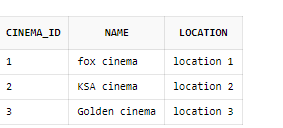
insert into cinema values (1,'fox cinema', 'location 1');

insert into cinema values (2,'KSA cinema', 'location 2');

insert into cinema values (3,'Golden cinema', 'location 3');







insert into hall values (1,'h1-1',50,1);

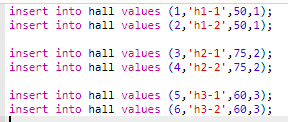
insert into hall values (2,'h1-2',50,1);

insert into hall values (3,'h2-1',75,2);

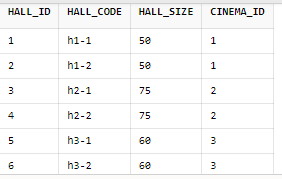
insert into hall values (4,'h2-2',75,2);

insert into hall values (5,'h3-1',60,3);

insert into hall values (6,'h3-2',60,3);





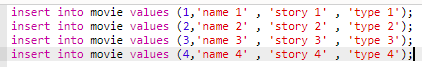


insert into movie values (1,'name 1' , 'story 1' , 'type 1');

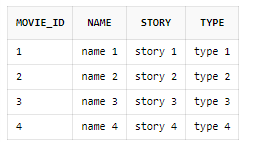
insert into movie values (2,'name 2' , 'story 2' , 'type 2');

insert into movie values (3,'name 3' , 'story 3' , 'type 3');

insert into movie values (4,'name 4' , 'story 4' , 'type 4');







insert into DisplaiedMovie values (1,to\_date('2022-11-01', 'yyyy-mm-dd'), '13:30',1,1);

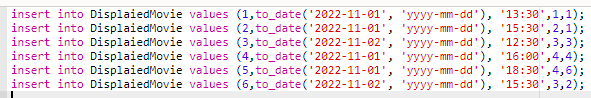
insert into DisplaiedMovie values (2,to\_date('2022-11-01', 'yyyy-mm-dd'), '15:30',2,1);

insert into DisplaiedMovie values (3,to\_date('2022-11-02', 'yyyy-mm-dd'), '12:30',3,3);

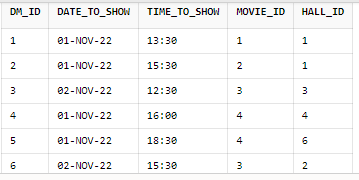
insert into DisplaiedMovie values (4,to\_date('2022-11-01', 'yyyy-mm-dd'), '16:00',4,4);

insert into DisplaiedMovie values (5,to\_date('2022-11-01', 'yyyy-mm-dd'), '18:30',4,6);

insert into DisplaiedMovie values (6,to\_date('2022-11-02', 'yyyy-mm-dd'), '15:30',3,2);







insert into customer values (1, 'Mohammed', 'Saber', '123456789');

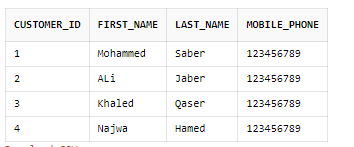
insert into customer values (2, 'ALi', 'Jaber', '123456789');

insert into customer values (3, 'Khaled', 'Qaser', '123456789');

insert into customer values (4, 'Najwa', 'Hamed', '123456789');





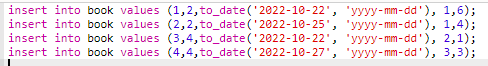


insert into book values (1,2,to\_date('2022-10-22', 'yyyy-mm-dd'), 1,6);

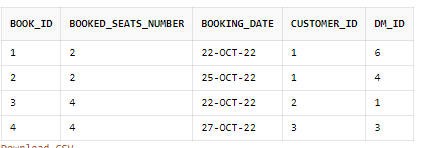
insert into book values (2,2,to\_date('2022-10-25', 'yyyy-mm-dd'), 1,4);

insert into book values (3,4,to\_date('2022-10-22', 'yyyy-mm-dd'), 2,1);

insert into book values (4,4,to\_date('2022-10-27', 'yyyy-mm-dd'), 3,3);





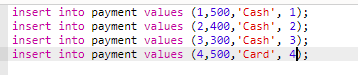


insert into payment values (1,500,'Cash', 1);

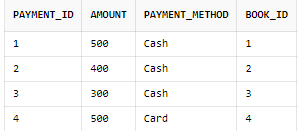
insert into payment values (2,400,'Cash', 2);

insert into payment values (3,300,'Cash', 3);

insert into payment values (4,500,'Card', 4);







# Data Retrieval Queries

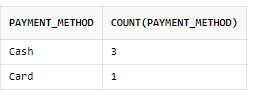
## Get payment method and number of records that contains same payment method.

select PAYMENT\_METHOD, count(PAYMENT\_METHOD)

from payment

group by PAYMENT\_METHOD;





## Get name of film and name of cinema that showing it.

select movie.name as movie, cinema.name as cinema

from movie, cinema, hall, DisplaiedMovie

where

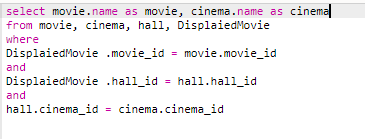
DisplaiedMovie .movie\_id = movie.movie\_id

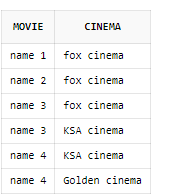
and

DisplaiedMovie .hall\_id = hall.hall\_id

and

hall.cinema\_id = cinema.cinema\_id





## Get name of film and number of its books.

select movie.name , count (movie.name)

from movie join displaiedmovie on

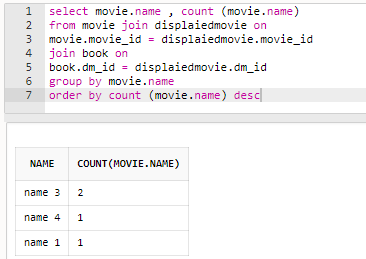
movie.movie\_id = displaiedmovie.movie\_id

join book on

book.dm\_id = displaiedmovie.dm\_id

group by movie.name

order by count (movie.name) desc



## Get name of films that does not have any book.

select movie.name

from movie

where movie.movie\_id not in

(

select movie.movie\_id

from movie, displaiedmovie, book

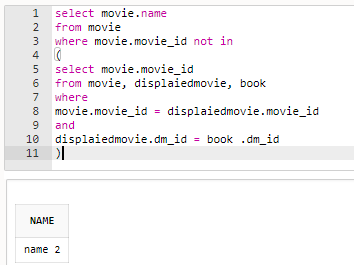
where

movie.movie\_id = displaiedmovie.movie\_id

and

displaiedmovie.dm\_id = book .dm\_id

)



## Get name of films that booked by specific customer

select movie.name as movie, concat(customer.First\_name ,Concat(' ',customer.Last\_name)) as customer

from movie, DisplaiedMovie , book, customer

where

DisplaiedMovie .movie\_id = movie.movie\_id

and

DisplaiedMovie .dm\_id = book.dm\_id

and

book.customer\_id = customer.customer\_id

and

customer.customer\_id = 1

