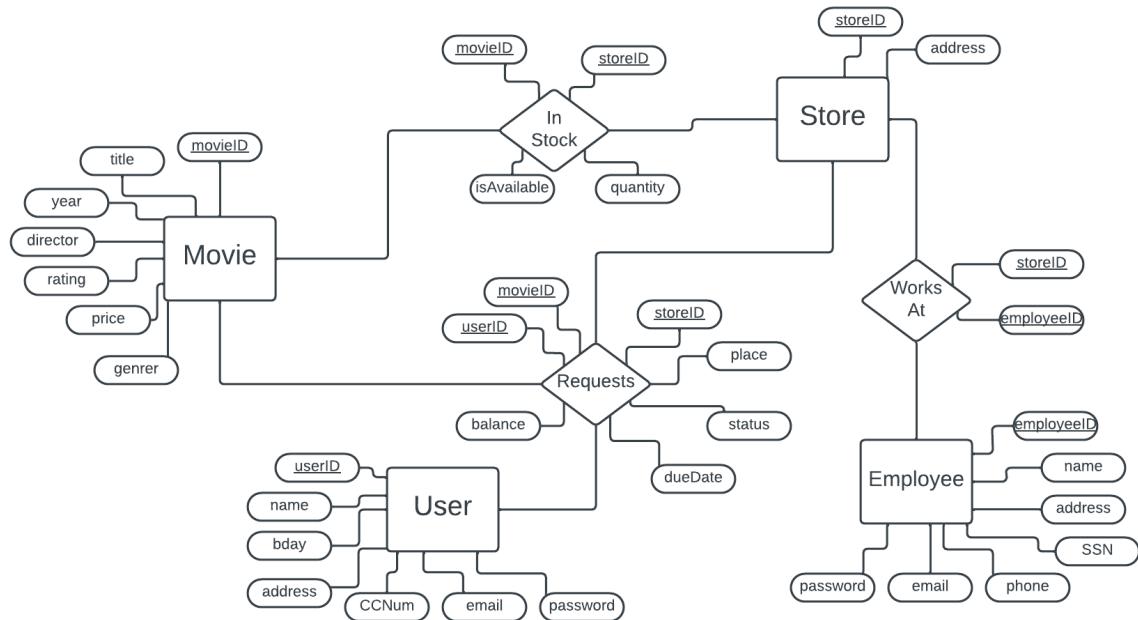
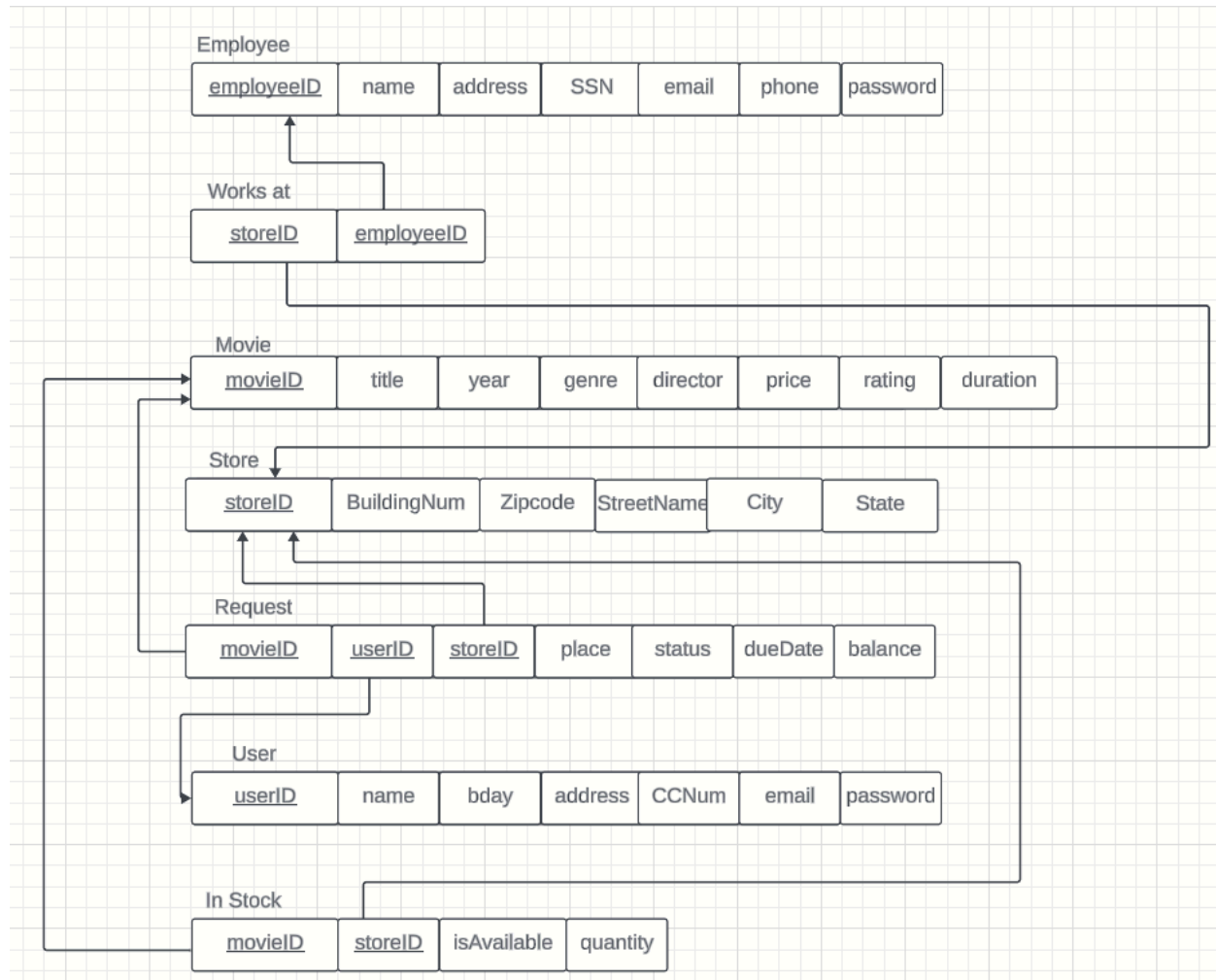


Database Design Document - CubeBuster

ER Diagram



Schema



Data Dictionary

Movie Table

Primary Key: MovieID

Attributes:

- **MovieID**: This is a unique identifying ID number for each movie in the database. A new code should be generated for every movie that is put into the system. Even if there are multiple of the same movie in the system, they will receive their own unique ID. Each code should be generated sequentially from the last one. If a movie is removed from the database, the ID number will not be replaced unless there is a system change later in the process. The first movie added to the system should take the number 00000001.
 - Data Type: int
 - Default: 00000000
 - Not Null
 - On Delete: Set Default
- **Title**: The name of the movie in the system should help with identification on the user level considering users/ employees are unlikely to memorize the MovieID number. There

can be multiple identical names in the system, either between copies of the same movie or movies that have the same name.

- Data Type: varchar(40)
- Not Null
- movieYear: Integer representation of the year in which the movie was released. Can help the user find the correct movie if there are multiple with the same name.
 - Data Type: int
 - Not Null
- Genre: The category the movie falls under. Some examples include: RomCom, Horror, Historical, etc. Can help the user find the correct movie if there are multiple with the same name. If there are multiple genres, listing the most prominent first would suffice.
 - Data Type: varchar(15)
 - Not Null
- Director: The name of the person who directed the movie. Will appear in the movie description, and would help users find the correct movie if there are multiple with the same name. If there are multiple directors, listing each of their last names would suffice.
 - Data Type: varchar(20)
 - Not Null
- Price: The unit cost of the item used during the rental transaction that the user must pay. The price should be displayed for every movie prior to rental.
 - Data Type: Decimal (3,2)
 - Not Null
- Rating: An alphanumeric representation of the rating of the movie. For example rated R, PG-13, etc. Should be displayed when users are looking at the movie, and certain users below age limits should not be able to check out movies.
 - Data Type: varchar(5)
 - Not Null
- Duration: A representation of the movie time in minutes
 - Data Type: Int
 - Not Null

Employee Table

Primary Key: EmployeeID

Attributes:

- EmployeeID: Unique numerical identification number for each employee. A new ID number should be generated sequentially for each employee that is added to the system. The first employee added to the system should take the ID number 000001.
 - Data Type: int
 - Default: 000000
 - Not Null
 - Unique
- empName: The name of the employee. Part of the basic personal identifiable information collected for legal and logistical reasons.
 - Data Type: varchar(20)

- Not Null
- Address: An alphanumeric representation of the address at which the employee resides. The employee should provide this information during the hiring process, and could be used for various reasons later.
 - Data Type: varchar(30)
 - Not Null
- SSN: The social security of the employee that should be provided upon their hiring. This 9 digit number should be unique to each employee. Used for legal and or tax purposes.
 - Data Type: char (9)
 - Not Null
 - Unique
- Email: Alphanumeric identification for the employee's personal email. Should be unique from other employees. Must be a valid email: include an @, and end in a '.edu/ .com/ .net' etc.
 - Data Type: varchar(25)
 - Not Null
 - Unique
- Phone: Numerical representation of the employee phone number for logistic reasons. Must be 10 digits long with no other types allowed.
 - Data Type: char (10)
 - Not Null
- empPassword: An alphanumeric string that allows the employee access to their work account when they login. This does not have to be unique, but should be private with only the employee knowing what it is.
 - Data Type: varchar(20)
 - Not Null

Customer Table

Primary Key: cID

Attributes:

- UserID: A unique numerical representation of the user's identification. A new ID will be generated sequentially for each user that is added to the system. The first employee added to the database should take on the ID number 0000001.
 - Data Type: int
 - Default: 00000000
 - Not Null
 - Unique
- username: The name of the user in the system. This does not need to be unique to each user, but will be used for basic identification purposes. Billing might also use the name on the account.
 - Data Type: varchar(20)
 - Not Null

- Address: An alphanumeric representation of the address at which the user resides. Can be used for billing purposes for the user. User does not have to enter this information until they are purchasing a rental.
 - Data Type: varchar(30)
 - Can be Null
- CCNum: The credit card number of the user saved in the database for use when they are purchasing a rental. The number should be 16 digits long. Does not need to be provided until a rental is in the process of being purchased.
 - Data Type: char(16)
 - Can be Null
- Bday: The Date representation of the user's birthday. In the format YYYY-MM-DD. Each user should input when they create an account, and are not allowed to check out a movie without an account. Used to check if the user can check out the movie based on the rating, for example a 15 year old cannot check out a rated R movie.
 - Data Type: Date
 - Not Null
- Email: Alphanumeric identification for the user's personal email. Should be unique from other users. Must be a valid email: include an @, and end in a '.edu/ .com/ .net' etc. Can be used as part of the user's login information, as well as for notification and communication purposes.
 - Data Type: varchar(25)
 - Not Null
 - Unique
- Userpwd: An alphanumeric string that allows the user access to their personal account when they log in to the rental service. This does not have to be unique, but should be private with only the user knowing what it is.
 - Data Type: varchar(20)
 - Not Null

Store Table

Primary Key: StoreID

Attributes:

- StoreID: A uniquely identifying numerical representation of the store number. Every new store should receive a new number sequentially as they are added. The store ID can be used by employees and users to understand which store they are most closely associated with. The first store should take on the ID number 0001.
 - Data Type: int
 - Not Null
 - Unique
 - Default: 0000
- Building Number: A numerical representation of the building number of the store
 - Data Type: int
 - Not Null
- Street: An alphabetical representation of the street name of the store

- Data type: varchar(20)
- Not NULL
- City: An alphabetical representation of the store city
 - Data type: varchar(20)
 - Not NULL
- Zipcode: An integer representation of the zip code of the store. Int(5) is depreciated on MySQL, so there is no int(limiter)
 - Data type: int
 - Not Null
- State: An alphabetical representation of a state based off of its abbreviation
 - Data type: varchar(2)
 - Not Null

Request Table

Primary Key: MovieID, UserID, StoreID

Foreign Key: MovieID, UserID, StoreID

Attributes:

- MovieID: Uniquely identifying numerical code to represent the movie that is being requested during this transaction. Foreign key to the Movie Table.
 - Data Type: int
 - On Delete: Set Default
 - On Cascade: Update
- UserID: Uniquely identifying numerical code to represent the user that is requesting the movie during this transaction. Foreign key to the User Table.
 - Data Type: int
 - On Delete: Set Default
 - On Cascade: Update
- StoreID: Uniquely identifying numerical code to represent the store the user is requesting the movie from during this transaction. Foreign key to the Store Table.
 - Data Type: int
 - On Delete: Set Default
 - On Cascade: Update
- Place: Numerical representation of where in the queue the user falls for the movie they are trying to rent. If the movie is available then they should be 0 in the queue. If the movie is being rented then they should be 1+ depending on how many other users placed a hold before them.
 - Data Type: char(2)
 - Not Null
- reqStatus: A textual representation of the status of the request. The request can be: waiting, ready, active, overdue, or returned. The user and employees should be able to see the status of any requests that are made.
 - Data Type: char(8)
 - Not Null

- dueDate: Date representation of a rental is due back to the store from the user. In Date format. This attribute is allowed to be Null until the status of the requests moves to active. Then dueDate is calculated as two weeks out from the date that the status moves to active because the user has picked up the rental.
 - Data type: Date
 - Null allowed
- balance: Representation of the balance due from the user for each request. This can include any fees that are added because of late returns. The balance can remain as 0 until that the movie is picked up by the user. If the status becomes overdue then the balance should increase by the overdue fee.
 - Data Type: Float
 - Not Null

Works At

Primary Key: StoreID, EmployeeID

Foreign Key: StoreID, EmployeeID

Attributes:

- StoreID: The unique numerical representation of the store in which the employee works at. Foreign key to the Store Table.
 - Data Type: int
 - On Delete: Set Default
 - On Cascade: Update
- EmployeeID: The unique identifying representation of the employee. Used to understand where the employees work, and which employees work at which stores. Foreign key to the Employee Table.
 - Data Type: int
 - On Delete: Set Default
 - On Cascade: Update

In Stock

Primary Key: MovieID, StoreID

Foreign Key: MovieID, StoreID

Attributes:

- MovieID: Uniquely identifying numerical code to represent the movie and understanding how many of each movie is currently at each store. Foreign key to the Movie Table.
 - Data Type: int
 - On Delete: Set Default
 - On Cascade: Update
- StoreID: Uniquely identifying numerical code to represent the store that the movie is currently available at. Foreign key to the Store Table.
 - Data Type: int
 - On Delete: Set Default
 - On Cascade: Update

- isAvailable: Boolean value to determine whether a movie is currently available at a certain store. If the movie is not currently being rented then True, but if there is none of a movie at the store then False.
 - Data Type: bool
 - Not Null
- Quantity: Numerical representation of the amount of a specific movie that is related to a specific store. The number represents the total number of the movies that are related to each store. If there is none, then set to 0.
 - Data Type: int
 - Not Null