Goals:

* Translate ERR to Relational Schema
* Create tables based on Relational Schema
* A group of black and white lines

  AI-generated content may be incorrect.Populate tables by injecting values with SQL

Illustrate:

* All entities (Member, Store, Store\_Object, Movie, Disk, and Player) becomes a table
* Ternary relationship Transaction is M:N:P and has its own attributes so it becomes its own table with foreign keys “Store\_ID”, ‘Object\_ID”, and “UserID”
* Binary relationship Owns is 1:N so a foreign key “Store\_ID” is added to Store\_Object
* Binary relationship is\_on is 1:N so foreign key is “Movie\_ID” is added to Disk
* Superclass/subclass relationship between parent class Store\_Object and child classes Movie and Player mapped using Option C
* Each Table gets an ID as a primary key

Constraints:

* 10% discount is applied to weekday rental
* Members can rent up to 10 disks and 1 player
* Store\_Object must have either a Movie\_ID or a Player\_ID
* There must be at least one disk for each movie
* Players come in 3 generations, higher generations have better features
* Rental period is always at least one day, even if object is rented and returned in the same day
* Reservations are cancelled if they are not picked up after 3 days
* If a transaction is completed then it must have an end date
* The total price of a transaction is calculated after it is completed based on the price per day of the object rented and the time between the start and end dates

Difficulties:

* Relating Disk and Player to Store\_Object during table creation
* Calculating the total price of a transaction since start and end dates are in a different table from price per day
* Started with an overcomplicated schema that we ended up simplifying, removed separate tables from director and producer, and changed actor from a multivalued attribute