**Tutorial 3**

**Capture the screenshots of your solution, using snipping tool and put them into your document.**

**Data Modeling with Oracle SQL Developer**

Use this tutorial to learn how to create ER model using Oracle SQL Developer. Once you have the understanding use the tool for the lab tasks given in a separate document.

Oracle SQL Developer Data Modeler is a free graphical tool that enhances productivity and simplifies data modeling tasks. Using Oracle SQL Developer Data Modeler users can create, browse and edit, logical, relational, physical, multi-dimensional, and data type models. The Data Modeler provides forward and reverse engineering capabilities and supports collaborative development through integrated source code control. The Data Modeler can be used in both traditional and in Cloud environments.

Download Modeller from the below link:

<https://www.oracle.com/database/technologies/appdev/datamodeler.html>

Unzip and run datamodeler.exe to create ER Models:

Note: You may need to create an account to download. Please follow the on-screen steps to create an account.

# **Data Modeler Tutorial: Modeling for a Small Database**

Follow the below link to learn more about data modeling in Oracle SQL Developer.

<https://docs.oracle.com/cd/E39885_01/doc.40/e48205/tut_data_modeling.htm#DMDUG36166>

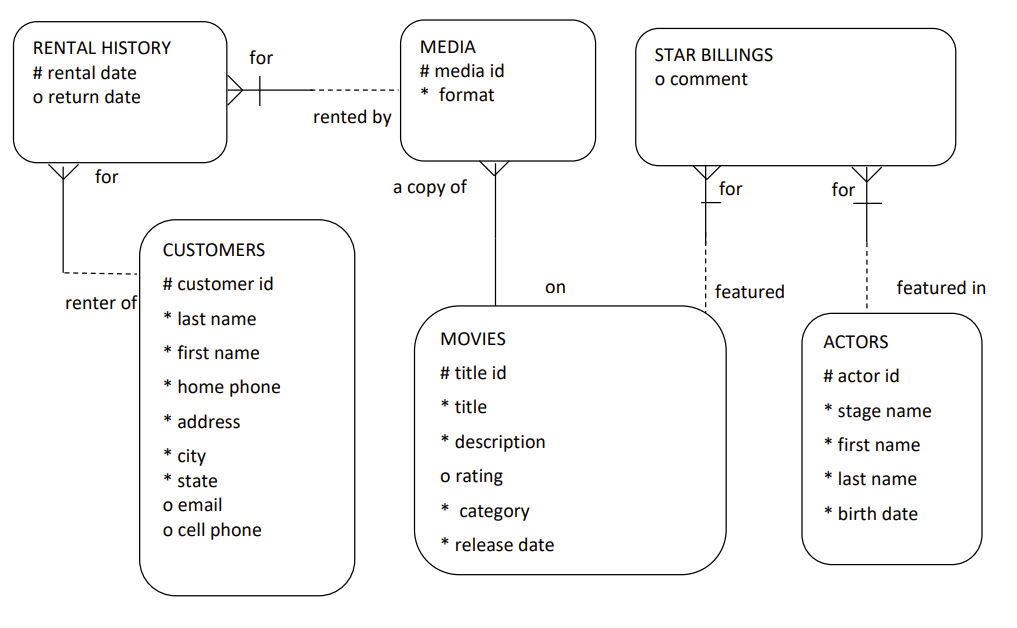
**Project ERD** - **OracleFlix Online Media Rentals**

You will perform the following major steps:

1. [Develop the Logical Model](https://docs.oracle.com/cd/E39885_01/doc.40/e48205/tut_data_modeling.htm#CBAHGHGF).
2. [Develop the Relational Model](https://docs.oracle.com/cd/E39885_01/doc.40/e48205/tut_data_modeling.htm#BABDHGDH).
3. [Generate DDL](https://docs.oracle.com/cd/E39885_01/doc.40/e48205/tut_data_modeling.htm#BABFJJBC).
4. [Save the Design](https://docs.oracle.com/cd/E39885_01/doc.40/e48205/tut_data_modeling.htm#BABEBICH).

# [Develop the Logical Model](https://docs.oracle.com/cd/E39885_01/doc.40/e48205/tut_data_modeling.htm#CBAHGHGF)

To create ER model we will use the following ERD:

The logical model for the database includes following entities:

The model has the following entities:

CUSTOMERS, MOVIES, MEDIA, RENTAL\_HOSTORY, ACTORS, STAR\_BILLINGS

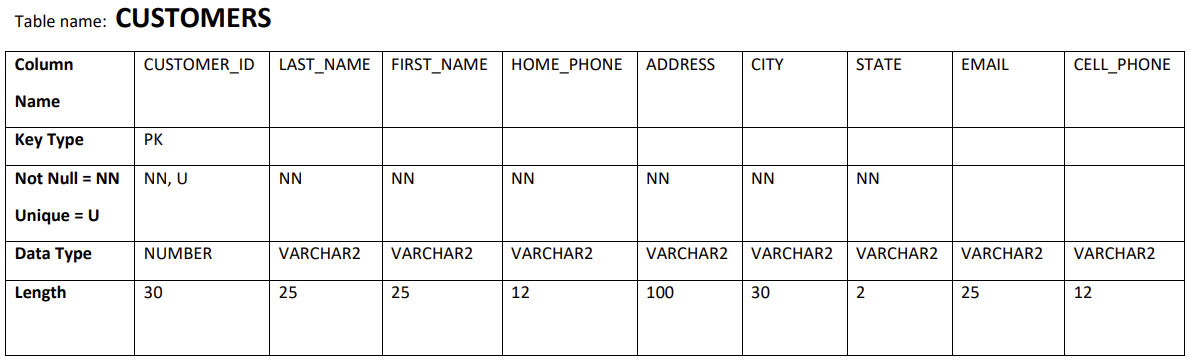
However, before you create the entities, create some domains that will make the entity creation (and later DDL generation) more meaningful and specific.

You can skip the part of adding domain if you choose Logical data type for attributes while creating entities

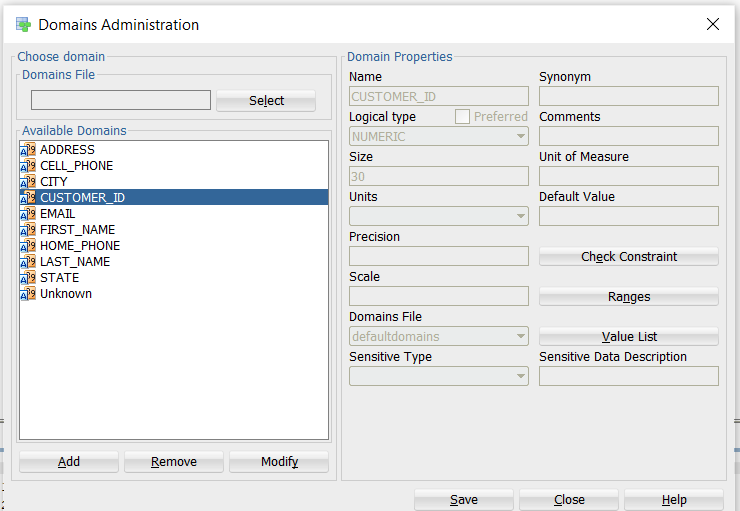
## Adding Domains:

In planning for your data needs, you have determined that several kinds of fields will occur in multiple kinds of records, and many fields can share a definition.

For example, you have decided that CUSTOMER table will have following structure:



1. Click **Tools**, then **Domains Administration**.
2. In the [Domains Administration](https://docs.oracle.com/cd/E39885_01/doc.40/e48205/dialogs_data_modeling.htm#BABBFGAA) dialog box, add domains with the above definitions. Click **Add** to start each definition, and click **Apply** after each definition.



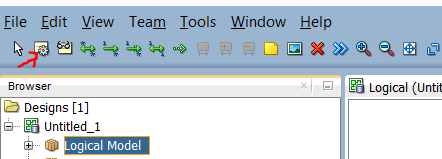
When you have finished defining these domains, click **Save**. This creates a file named defaultdomains.xml in the **datamodeler/domains directory** or **datamodeler\domains** folder under the location where you installed Data Modeler.

1. Click Close to close the dialog box.

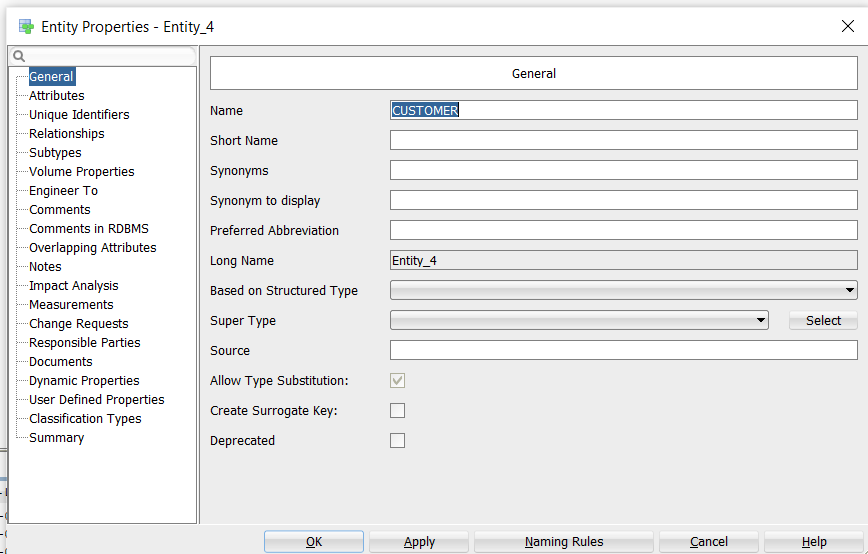
## Creating the CUSTOMER Entity

Create the Books entity as follows:

1. In the main area (right side) of the Data Modeler window, click the **Logical tab.**
2. Click the New Entity icon.

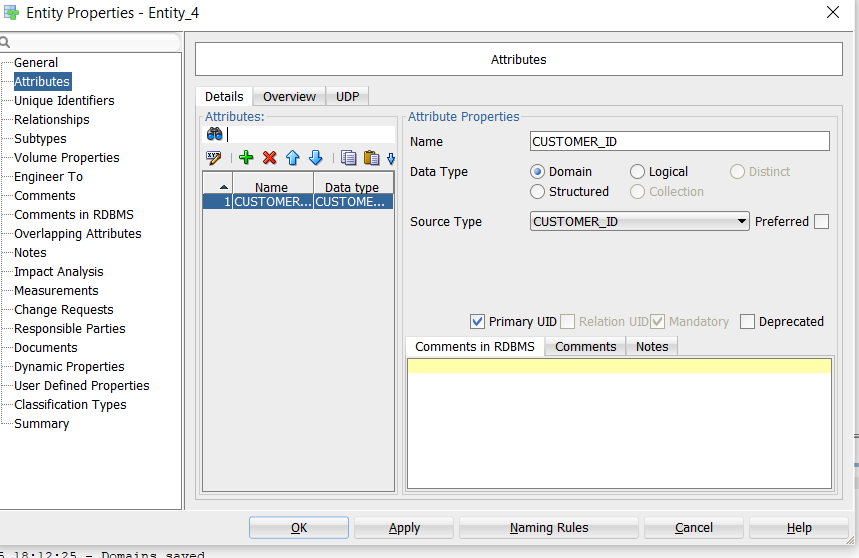


1. Click in the logical model pane in the main area; and in the Logical pane press, diagonally drag, and release the mouse button to draw an entity box. The [Entity Properties](https://docs.oracle.com/cd/E39885_01/doc.40/e48205/dialogs_data_modeling.htm#BABHACHD) dialog box is displayed.
2. Click **General** on the left, and specify as follows:

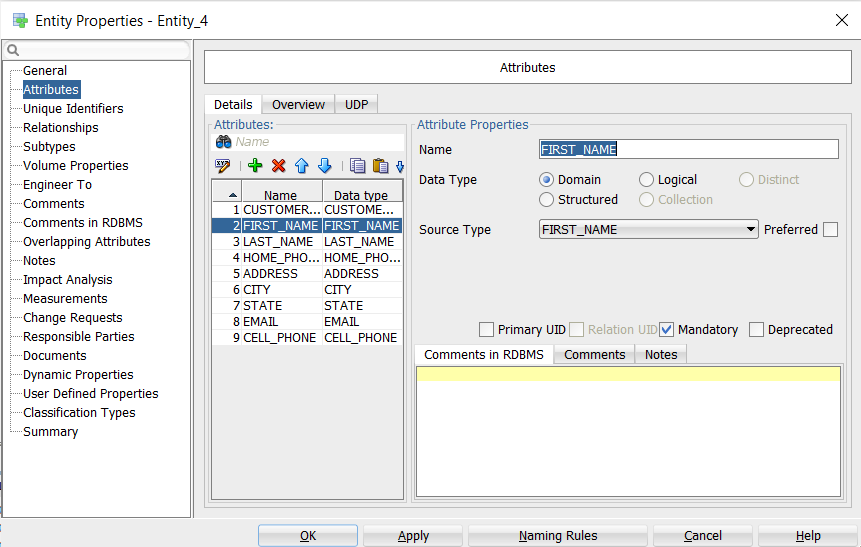


**Name**: CUSTOMER

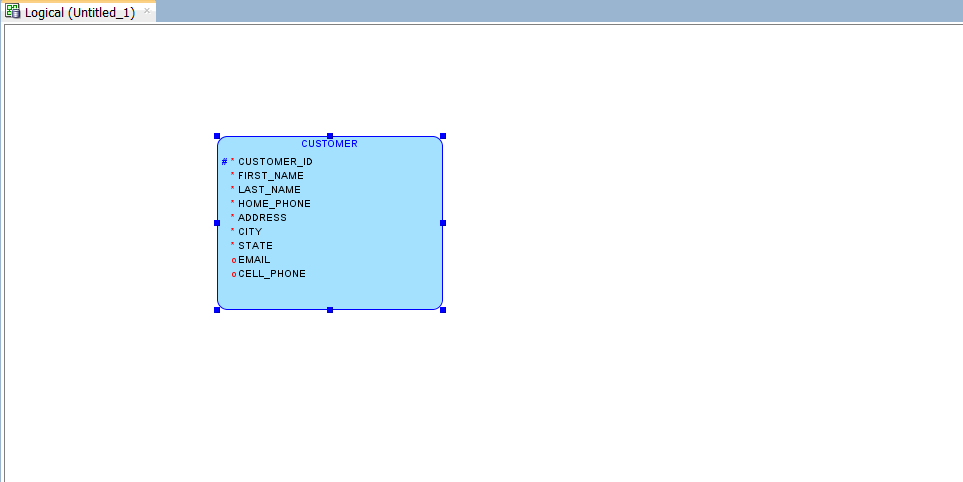
1. Click **Attributes** on the left, and use the **Add (+)** icon to add the following attributes, one at a time. (For datatypes, select from the Domain types) and from Source Type select CUSTOMER\_ID, check Primary UID to make it key



1. Add all attributes and click OK to finish creating CUSTOMER Entity

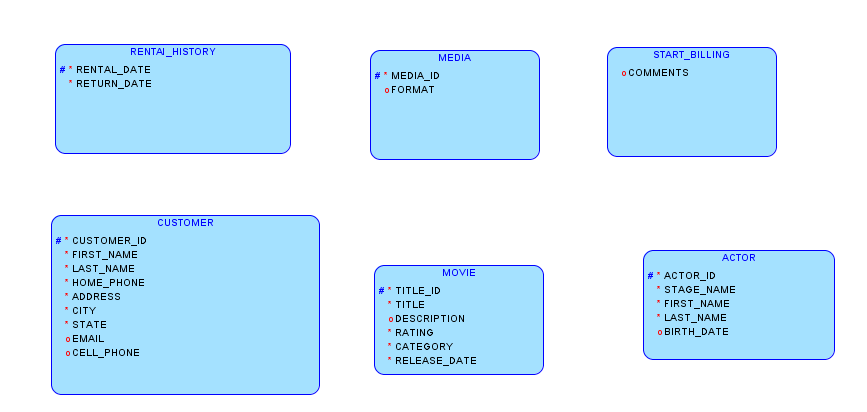


After finishing the customer entity is created:

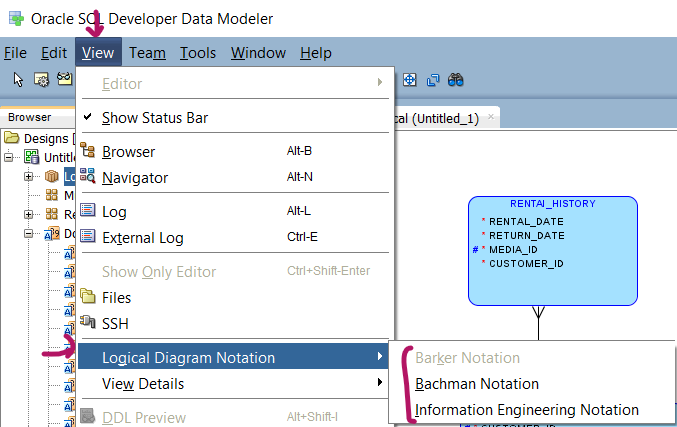


**PRACTICE EXERCISE:** Repeat the above steps for creating entities for the below remaining entities:

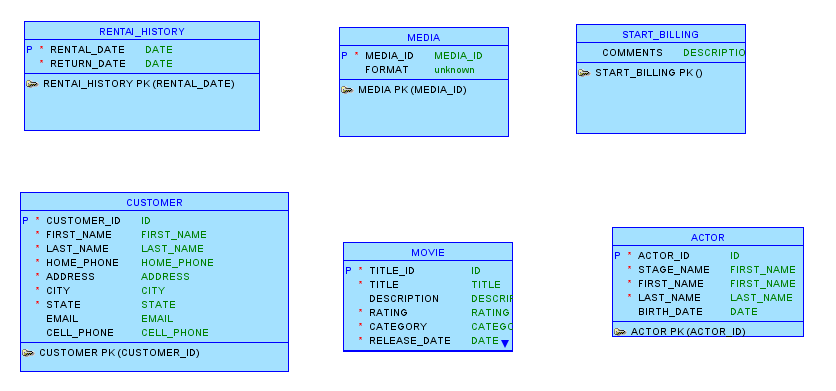
When you are done with all entities, the logical model pane in the main area should look like the following figure.



Note that for this figure, Barker Notation is used (you can change to Bachman notation by clicking View, then Logical Diagram Notation, then Bachman notation).



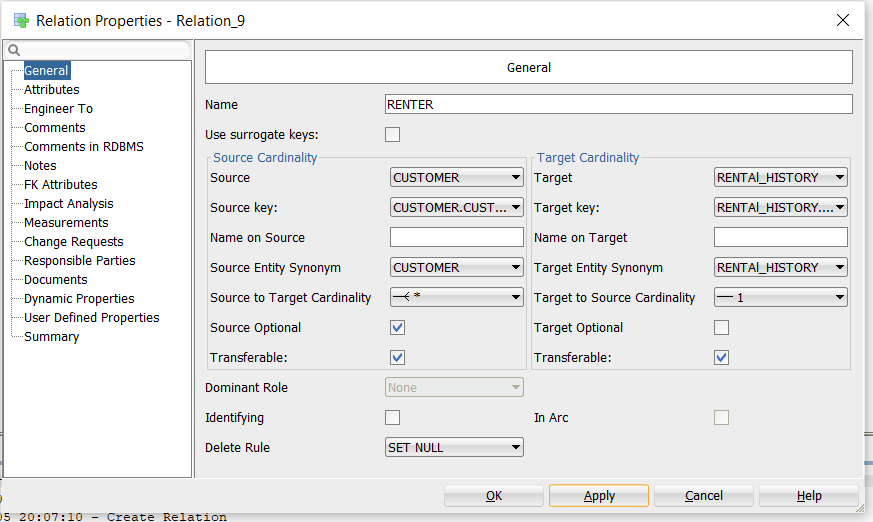
Bachman Notation



# Creating Relations Between Entities

CUSTOMER and RENTAl\_HISTORY has one-to-Many relationships from customer to RENTAL\_HISTORY.

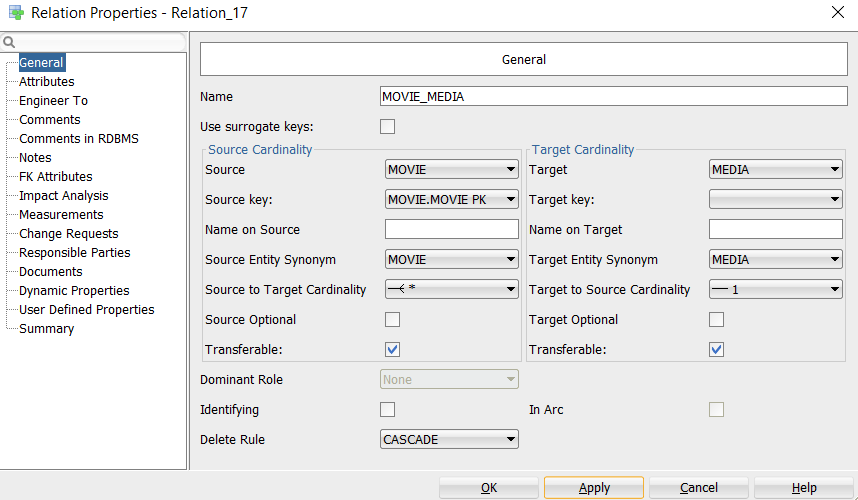
1. Click the 1:N Relation icon. 
2. Click first in the CUSTOMER box, then in the RENTAL\_HISTORY box. A line with an arrowhead is drawn from CUSTOMER to RENTAL\_HISTORY and following property dialogue will appear:



1. Check source optional box, since every customer may not rent a media, however, rental history should be mandatory if some customer rent a media, therefore leave target optional unchecked.

MOVIE and MEDIA has one-to-Many relationships from MOVIE to MEDIA

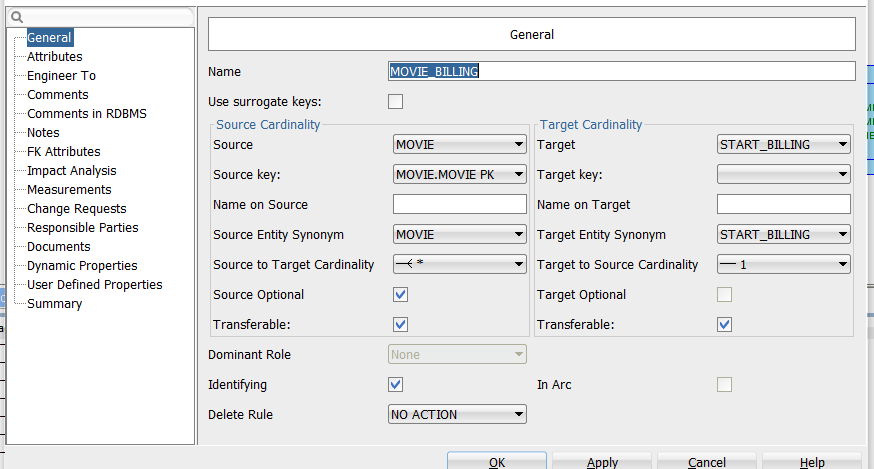
Add one-to-many relationship from MOVIE to MDIA. Note media cannot exist without movie keep source and target unchecked and set delete rule as CASCADE as shown in the below figure.



Optionally, double-click a line (or right-click a line and select Properties) and view the [Relation Properties](https://docs.oracle.com/cd/E39885_01/doc.40/e48205/dialogs_data_modeling.htm#BABDJCBE) information.

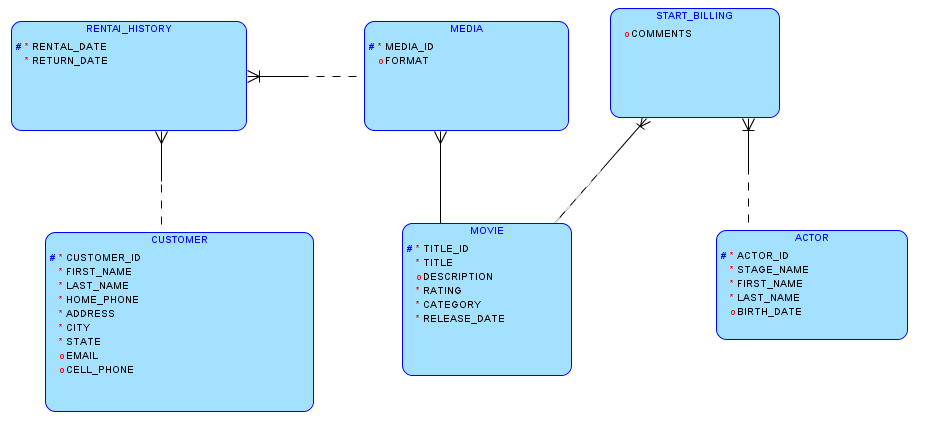
MOVIE and START\_BILLING Relationship:

MOVIE and START\_BILLING has one-to-Many relationships from MOVIE to START\_BILLING as shown in the below figure. Please note that identifying box is checked to make the TITLE\_ID both PK and FK.

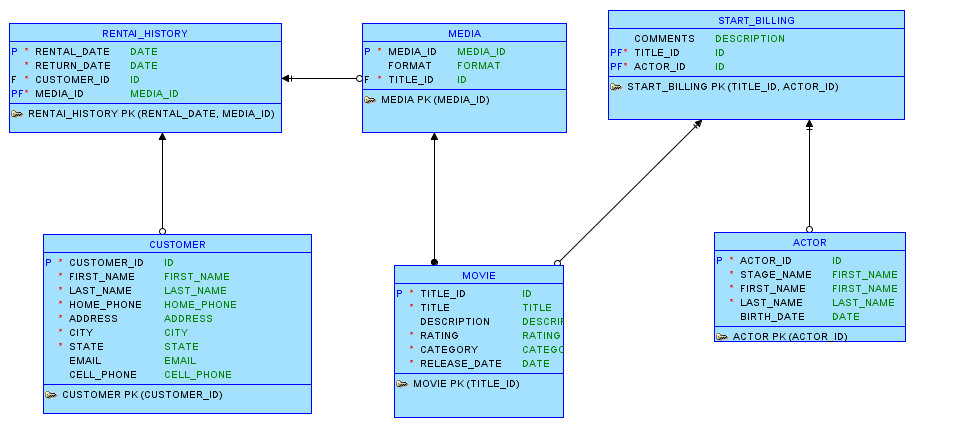


**Practice exercise: Complete all the remaining relationship.**

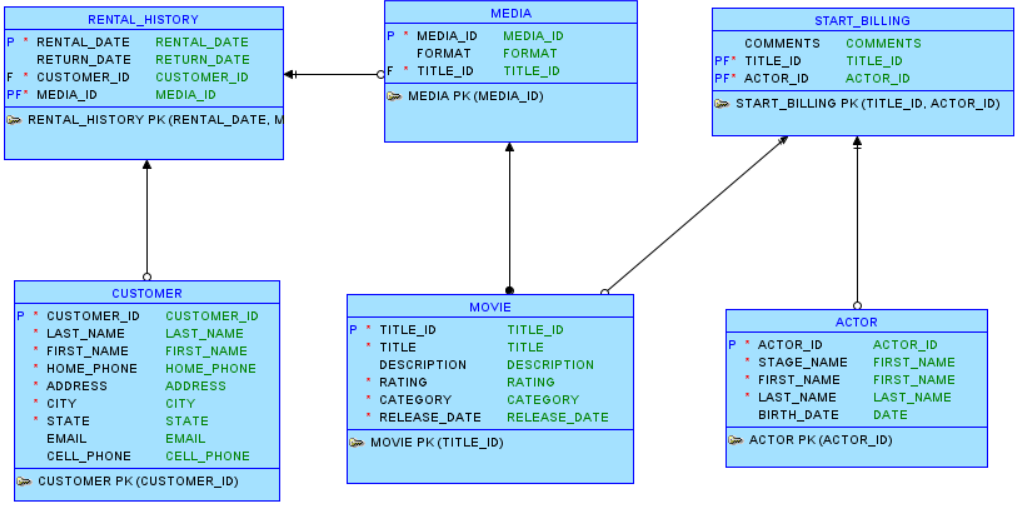
The final ERD should look like the following if using **Barker Notation.**



**Bachman Notation**



**My ERD:**



**Once you have completed the above exercise answer the following questions:**

1. What are the main stages of the database system development lifecycle? Depict the stages using [www.draw.io](http://www.draw.io). For each stage, state if it is mandatory or optional.

A diagram of a design

AI-generated content may be incorrect.

1. Discuss the main approaches to database design. Discuss the contexts where each is appropriate.

There are three main approaches to database design: **top-down, bottom-up, and inside-out**. The **top-down approach** starts from a broad view of the system and gradually defines detailed data elements, making it suitable for new systems with clearly defined requirements. The **bottom-up approach** begins by identifying specific data items and combining them into larger structures, which is useful when working with existing systems or integrating legacy databases. The **inside-out approach** starts from core entities and expands outward, making it ideal when the central data is well understood but the surrounding details may evolve over time.

1. What are the three phases of database design? How are they related to each other?

Physical<Logical<Conceptual

Physical is the physical storage where and how the data is stored. It is stored in which format. It also contains what are the structures and access paths.

Logical is what data is relevant, and which data belongs to which entity. It is mainly the database management system schema.

Conceptual is the highest level. It contains description of data requirements, entity types, relationships, constraints.

1. The following are problems that have been identified during the testing process in the development of a new system. In which part of the life cycle do you think these problem could have originated and been identified by a thorough review following that stage in the development life cycle?
2. The performance of the system is poor – failing to respond quickly enough to meet the stated user requirement of interactive, screen-based use. Physical
3. No backup facilities were included to meet the users' requirement of long-term archival of their data. System Design
4. No user manuals were provided! Implementation

MCQs

**Please attempt your QUIZ under consolidate.**