East West University
Department of Computer Science and Engineering

CSE 302: LAB 07
Course Instructor: Dr. Mohammad Rezwanul Huq

Entity Relationship Modeling using Data Modeler in SQLDeveloper

Lab Objective

Familiarize students with Entity-Relationship Model.

Lab Outcome

After completing this lab successfully, students will be able to:

1. **Understand** E-R Model.
2. **Understand and use** Data Modeler tool for E-R modeling.

Psychomotor Learning Levels

This lab involves activities that encompass the following learning levels in psychomotor domain.

Level	Category	Meaning	Keywords
P1	Imitation	Copy action of another; observe and replicate.	Relate, Repeat, Choose, Copy, Follow, Show, Identify, Isolate.
P2	Manipulation	Reproduce activity from instruction or memory	Copy, response, trace, Show, Start, Perform, Execute, Recreate.

Instructions

- Follow the instructor during the class.
- A simple step-by-step tutorial is uploaded in this link. https://goo.gl/mXvheL

Practice Exercise

Step 1: Add the following domains.

Name	Logical Type	Other Information
Person Name	VARCHAR	Size: 25
Address Line	VARCHAR	Size: 40
City	VARCHAR	Size: 25
State	VARCHAR	Size: 2
Zip	VARCHAR	Size: 10
Book Id	VARCHAR	Size: 20
Numeric Id	NUMERIC	Precision: 7, Scale: 0
Title	VARCHAR	Size: 50

Step 2: Creating the Books Entity Set

Name	Datatype	Other Information and Notes
book_id	Domain: Book Id	Primary UID (unique identifier). (The Dewey code or other book identifier.)
title	Domain: Title	M (mandatory, that is, must not be null).
author_last_name	Domain: Person Name	M (mandatory, that is, must not be null).
author_first_name	Domain: Person Name	25 characters maximum.
rating	Logical type: NUMERIC (Precision=2, Scale= 0)	(Librarian's personal rating of the book, from 1 (poor) to 10 (great).)

Step 3: Creating the Patrons Entity

Attribute Name	Type	Other Information and Notes
patron_id	Domain: Numeric Id	Primary UID (unique identifier). (Unique patron ID number, also called the library card number.)
last_name	Domain: Person Name	M (mandatory, that is, must not be null). 25 characters maximum.
first_name	Domain: Person Name	(Patron's first name.)
street_address	Domain: Address Line	(Patron's street address.)
city	Domain: City	(City or town where the patron lives.)
state	Domain: State	(2-letter code for the state where the patron lives.)
zip	Domain: Zip	(Postal code where the patron lives.)
location	Structured type: SDO_GEOMETRY	Oracle Spatial geometry object representing the patron's geocoded address.

Step 4: Creating the Transactions Entity

Attribute Name	Type	Other Information and Notes
transaction_id	Domain: Numeric Id	Primary UID (unique identifier). (Unique transaction ID number)
patron_id	Domain: Numeric Id	M (mandatory, that is, must not be null). Must match a patron_id value in the Patrons entity.
book_id	Domain: Book Id	M (mandatory, that is, must not be null). Must match a book_id value in the Books entity.
transaction_date	Logical type: Datetime	M (mandatory, that is, must not be null). Date and time of the transaction.
transaction_type	Domain: Numeric Id	M (mandatory, that is, must not be null). (Numeric code indicating the type of transaction, such as 1 for checking out a book.)

Step 5: Creating Relations between Entities

- **Books and Transactions:** one-to-many. Each book can be involved in multiple sequential transactions. Each book can have zero or one active checkout transactions; a book that is checked out cannot be checked out again until after it has been returned.
- **Patrons and Transactions:** one-to-many. Each patron can be involved in multiple sequential and simultaneous transactions. Each patron can check out one or many books in a visit to the library and can have multiple active checkout transactions reflecting several visits; each patron can also return checked out books at any time.

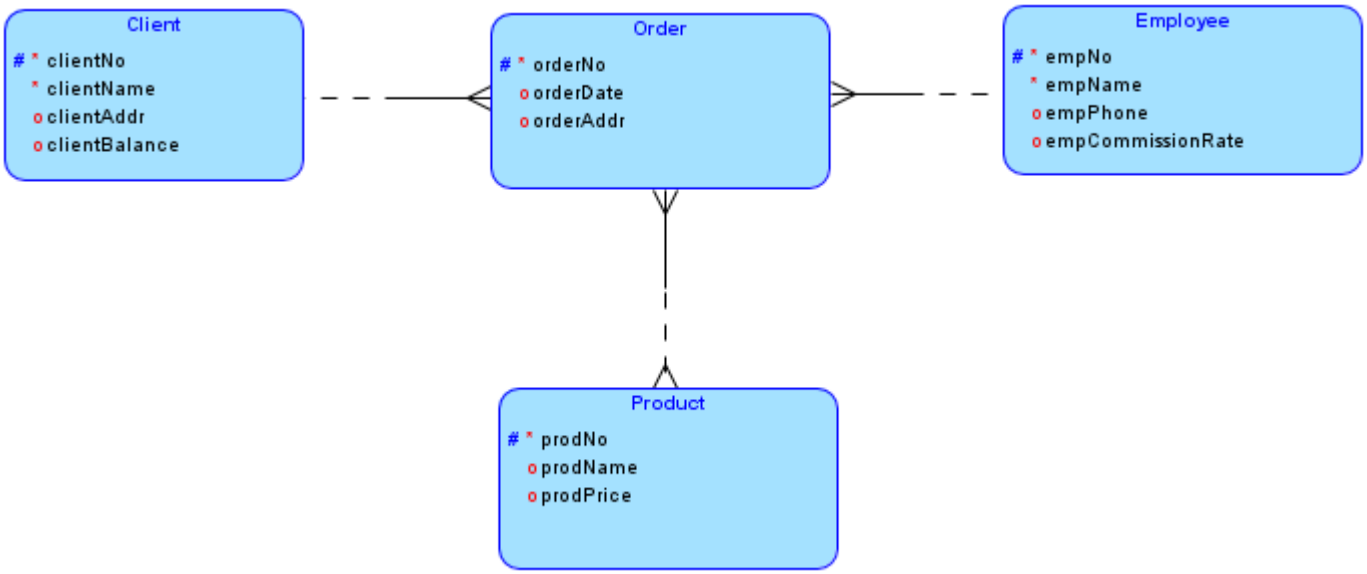
Step 6: Save the Design

Step 7: Develop the Relational Model (Schema Diagram)

Step 8: Generate DDL and Save the script

Exercise:

Draw the following ER Diagram



- Create **four entities** with appropriate attributes first.
- Create the following **three relationships**:
    - **Client to Order: one to many.** A client can give many orders. An order can be placed by one client.
    - **Employee to Order: one to many.** An employee can process many orders. An order is processed by one employee.
    - **Product to Order: many to many.** A product can be included in many orders. An order may have many products.