Name: Urooba Gohar
Roll No: 22P-9216
Section: BSCS-6A

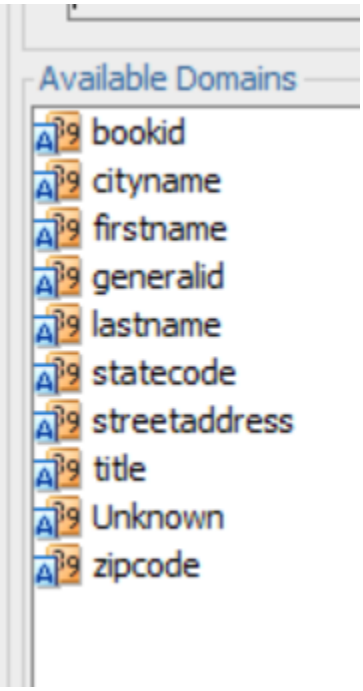
Database Systems Labtask 8

Task 1:

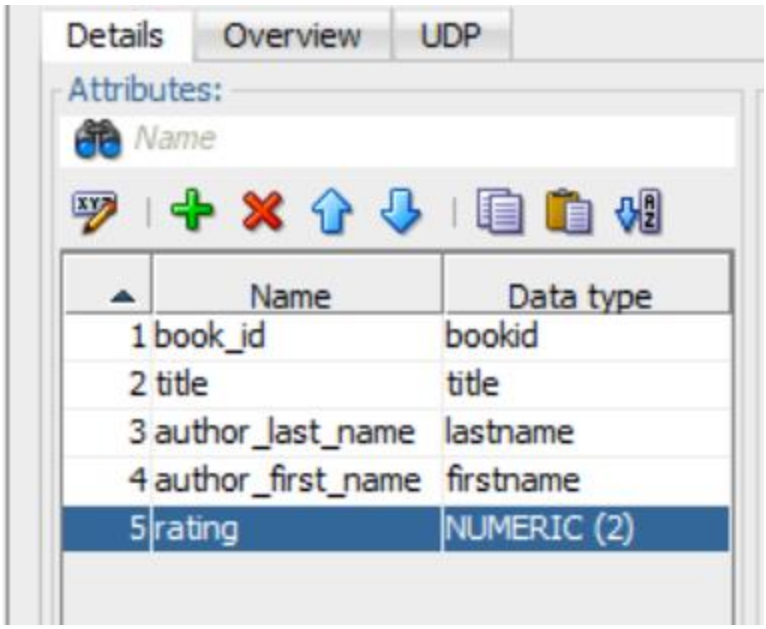
Perform the lab manual example as first task.

Answer:

First we add domains:

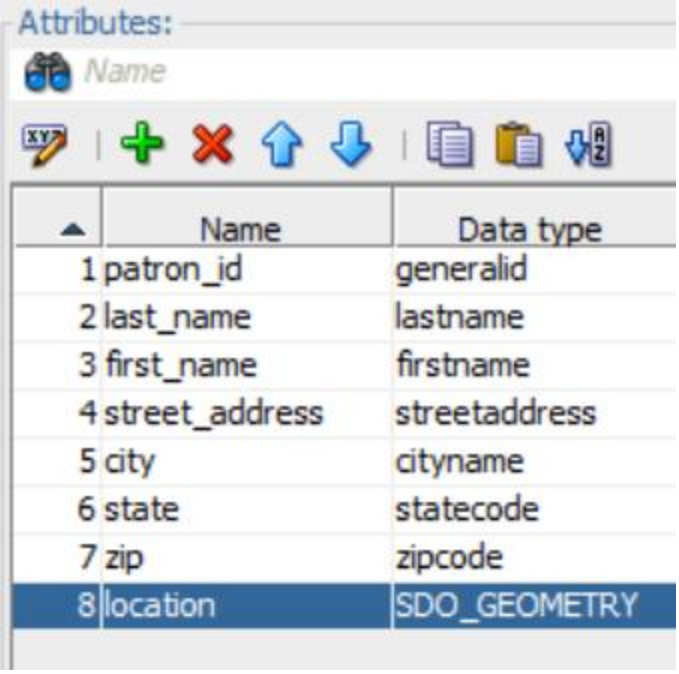


Now we will create the Books entity with the following attributes:



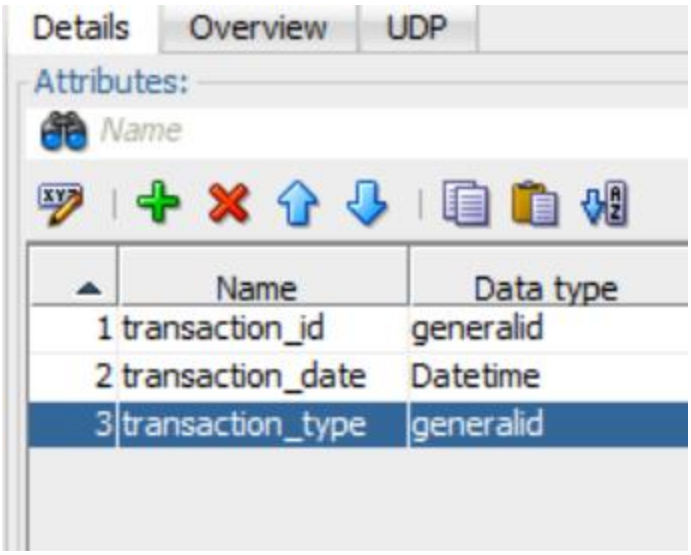
This is how it looks like:

Now we will create the Patrons entity with the following attributes:

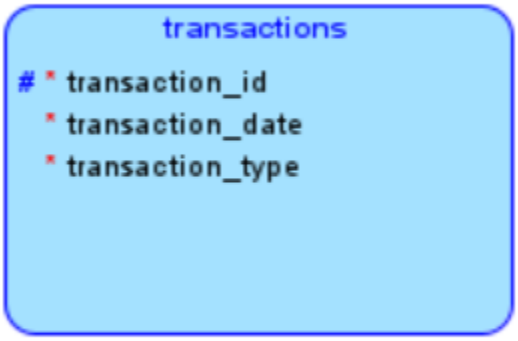


This is how it looks like:

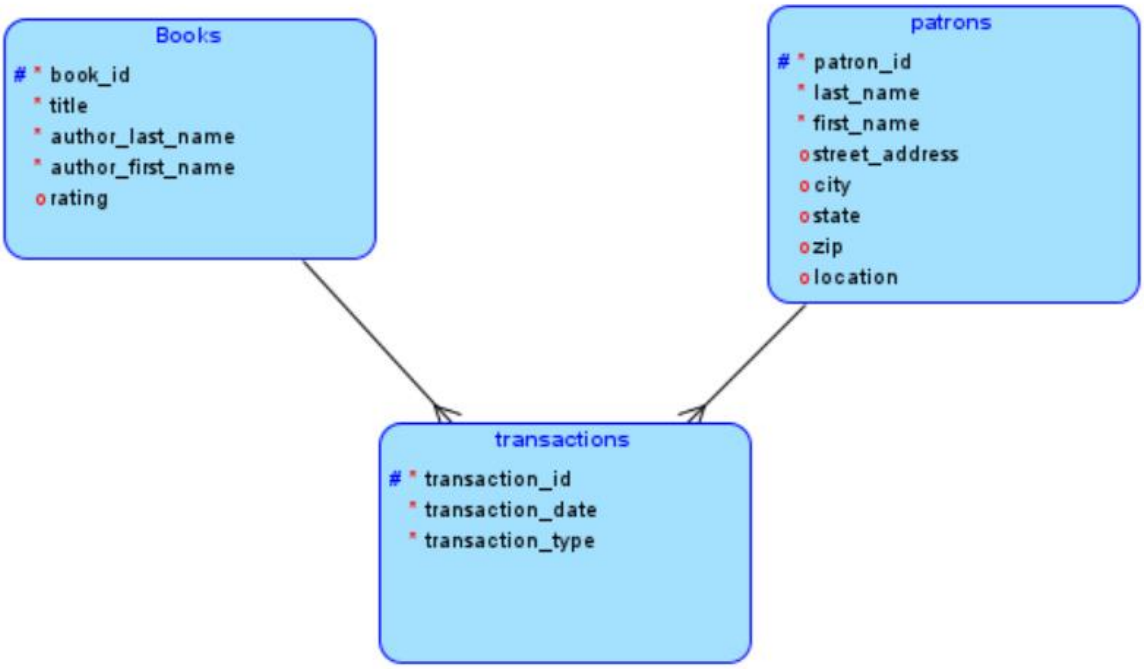
Similarly, we will create the Transactions entity with the following attributes:



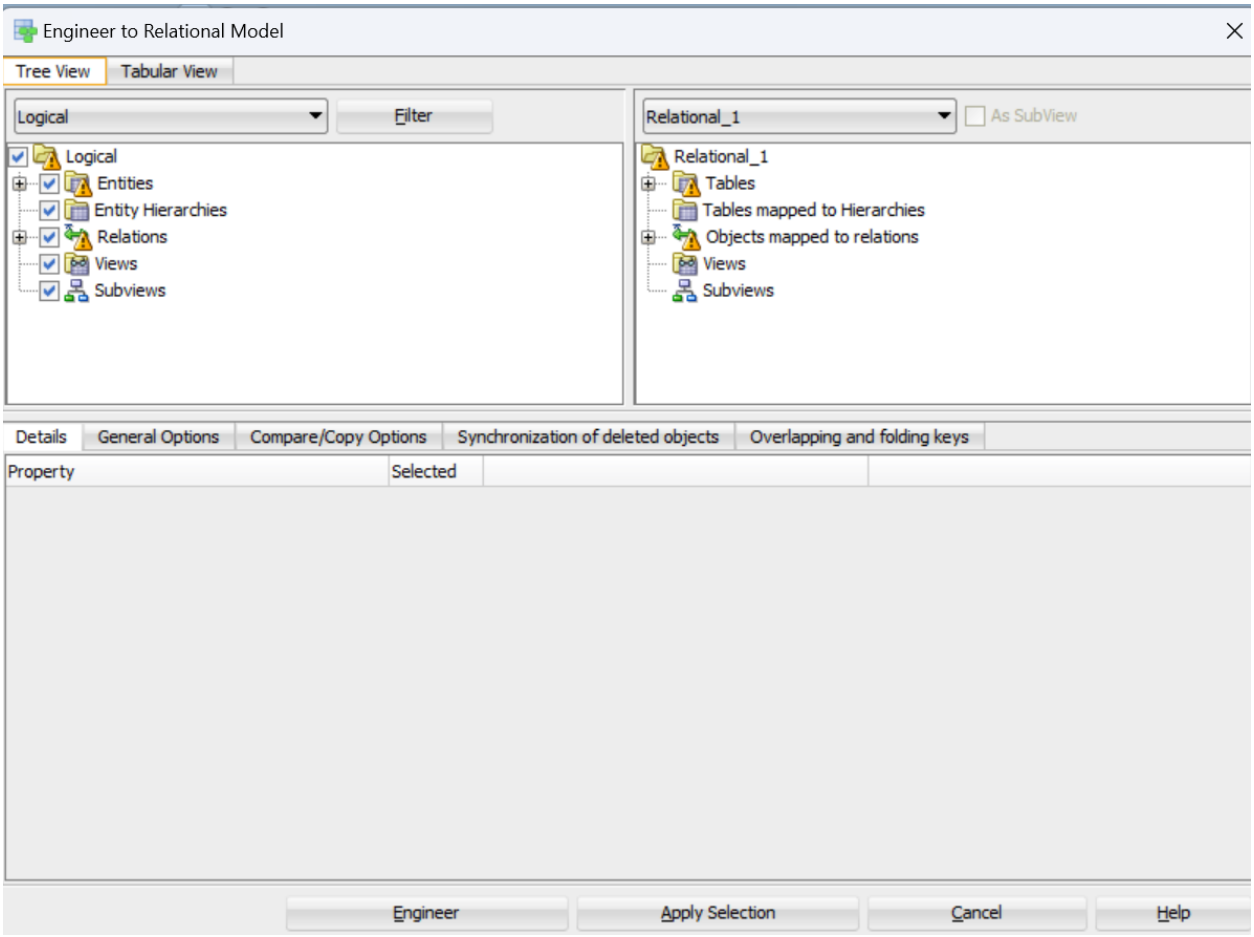
This is how it looks like:



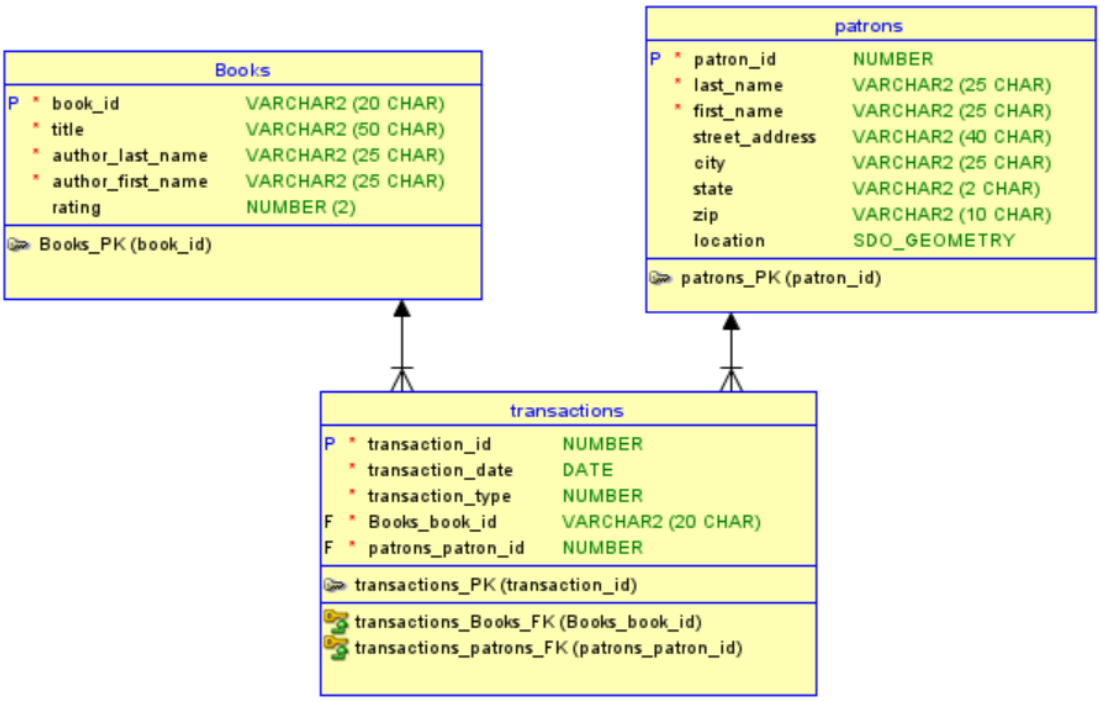
Now we create two one-to-many relationship between Books and Transactions entities and Patrons and Transactions.



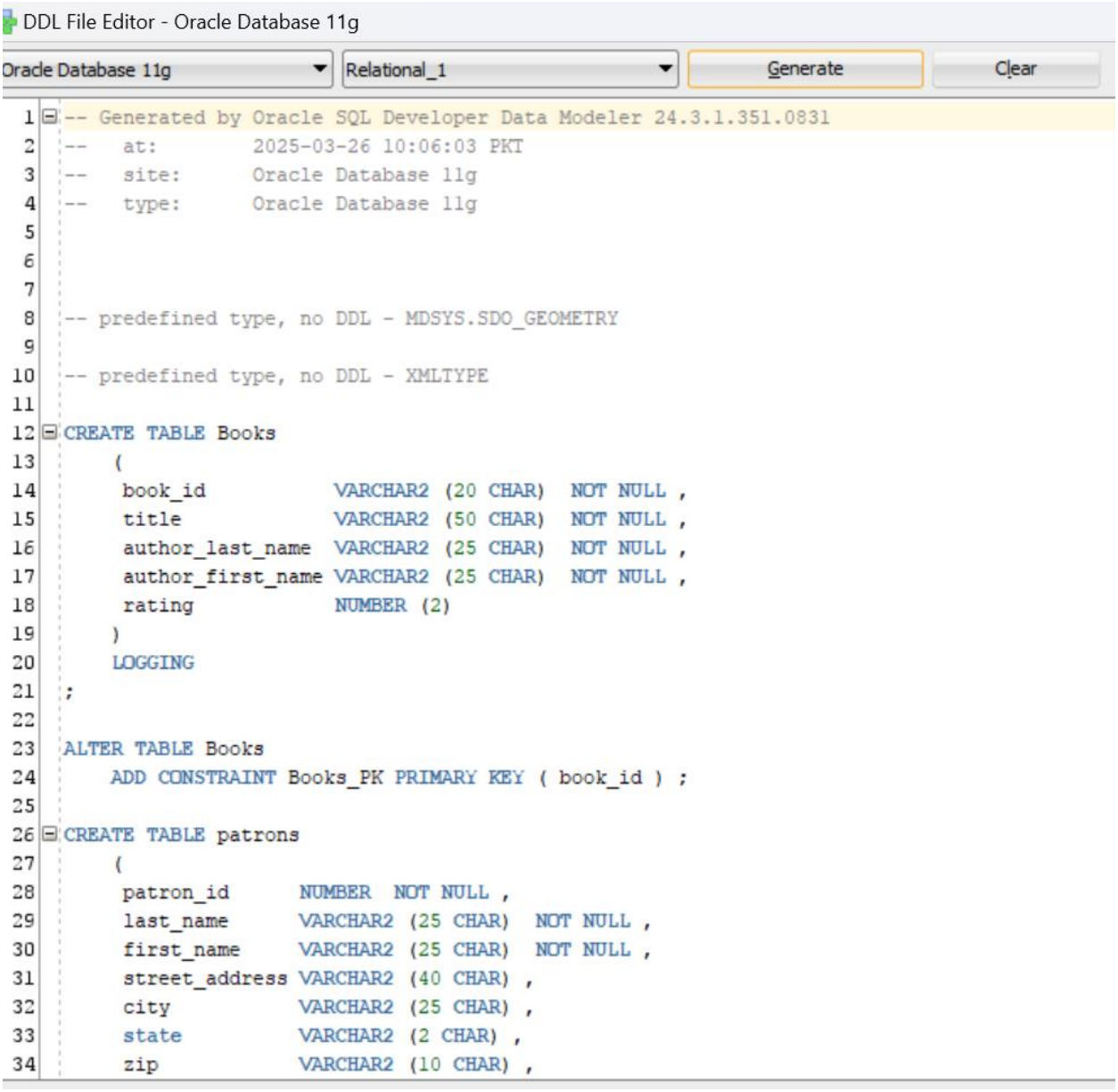
Now we develop the relational model by engineering it.



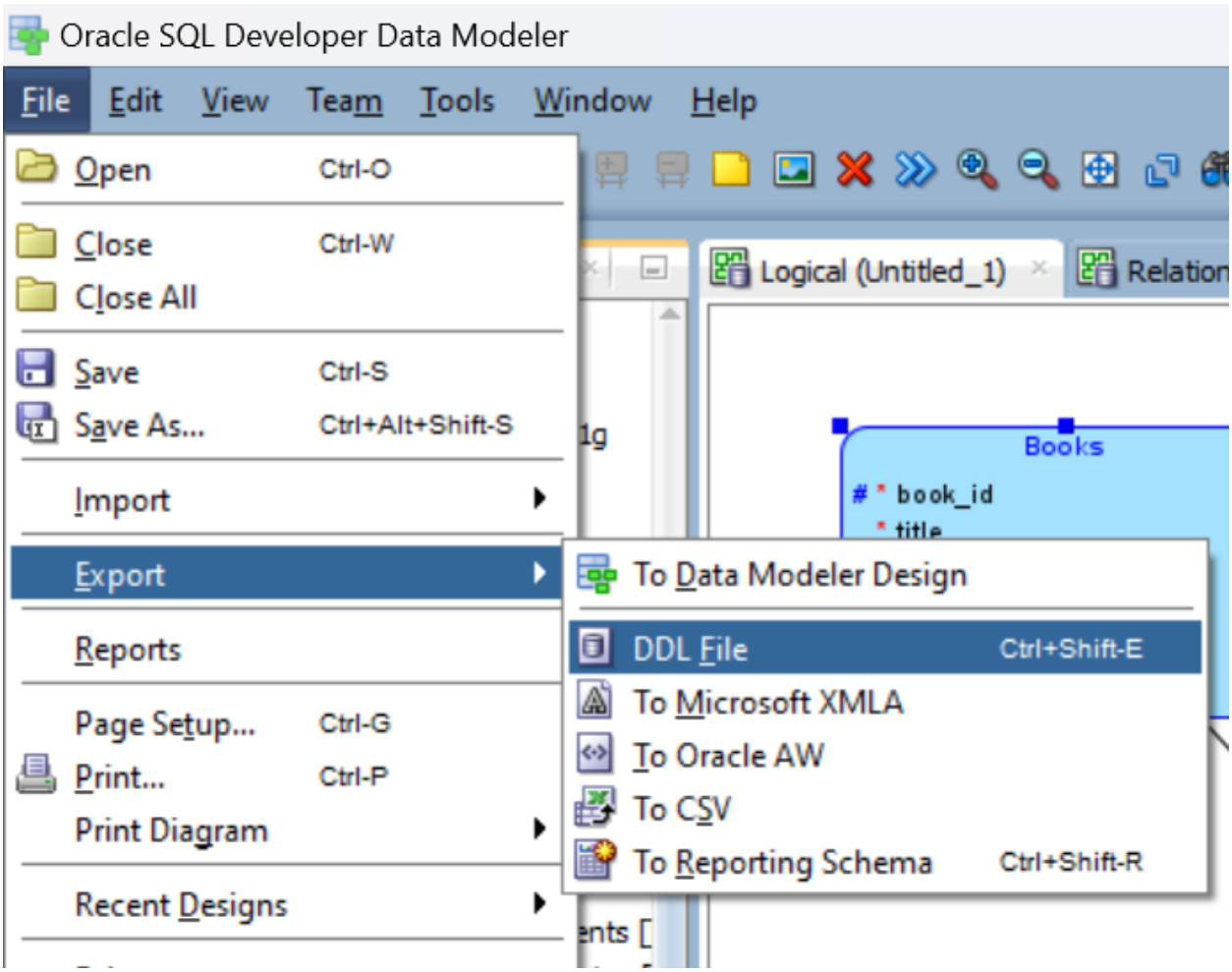
Hence forming the following relationship model:



Now we have to generate the DDL which is:



And we export the DDL and save the whole program:

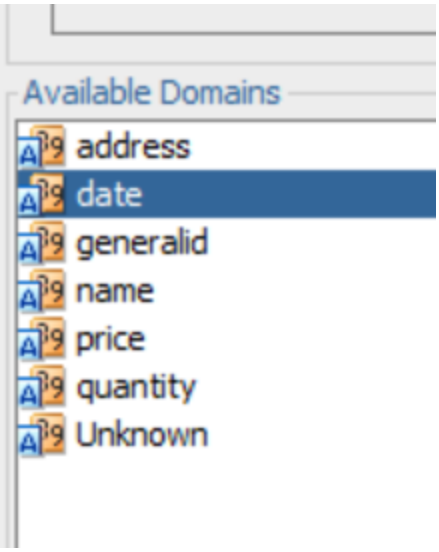


Task 2:

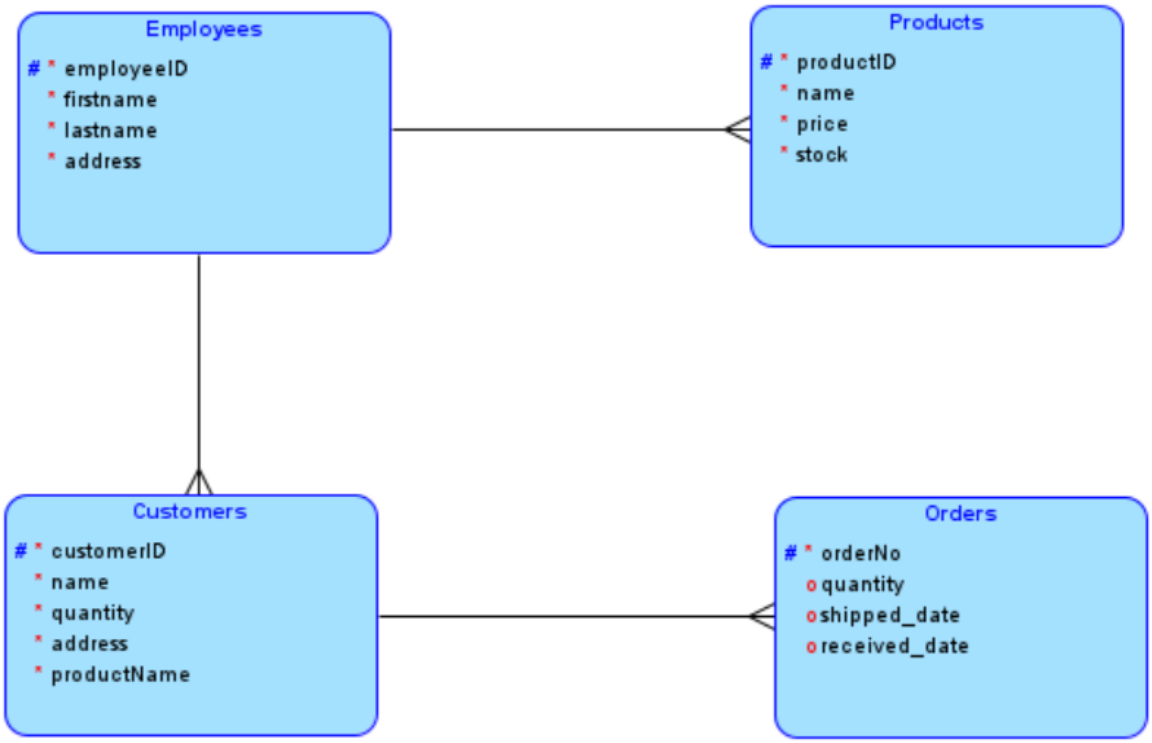
Create a physical design (DDL) from the above logically designed database. Keeping in mind the logical design, create foreign keys in each table where required. Populate each table up to maximum three records.

Answer:

First we add domains:



Now we will make the entities and their relationships:



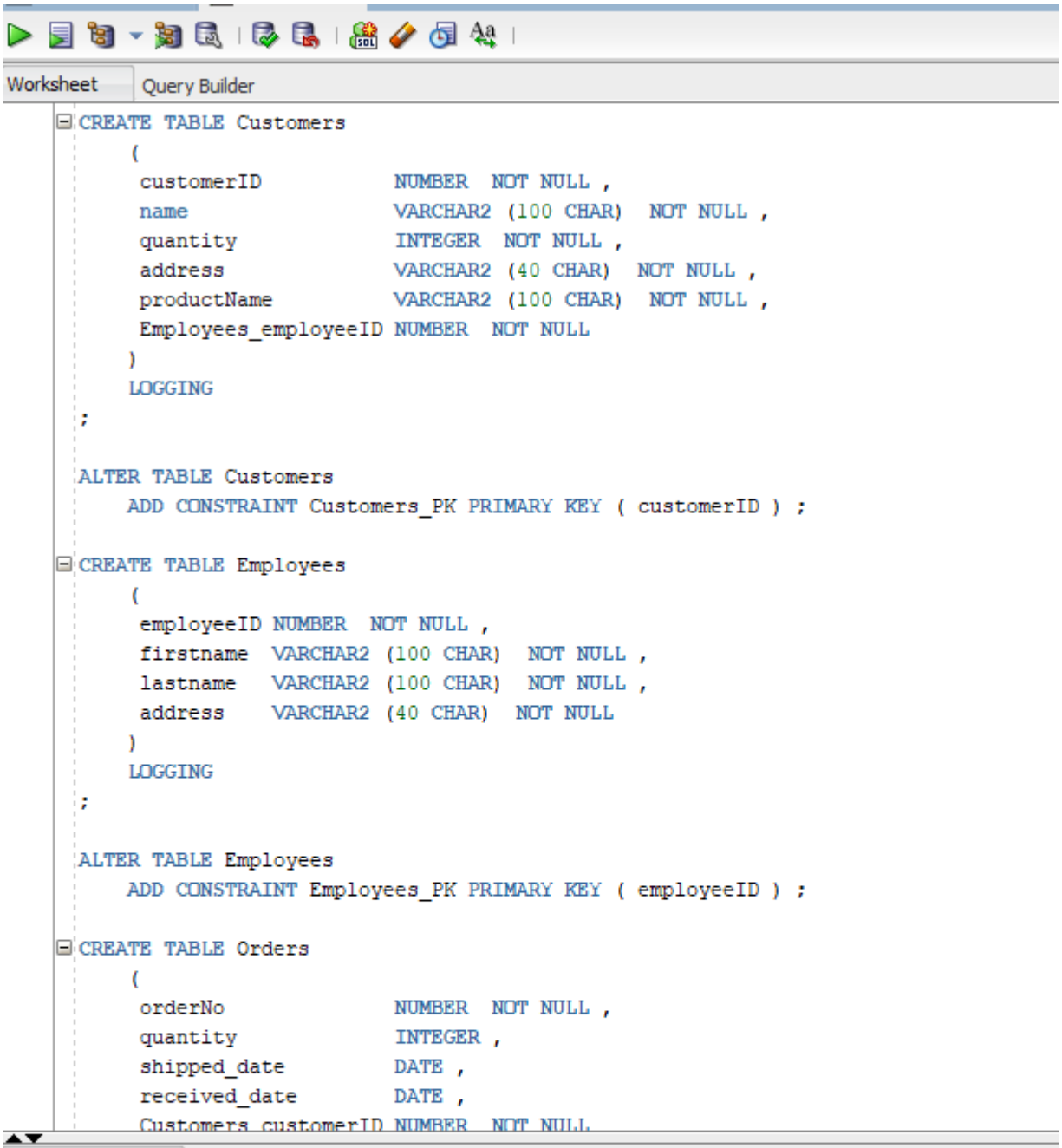
Now generate its DDL:

DDL File Editor - Oracle Database 11g

Oracle Database 11g | Relational_1 | Generate | Clear

```
1  -- Generated by Oracle SQL Developer Data Modeler 24.3.1.351.0831
2  --   at:        2025-03-26 14:31:36 PKT
3  --   site:      Oracle Database 11g
4  --   type:      Oracle Database 11g
5
6
7
8  -- predefined type, no DDL - MDSYS.SDO_GEOMETRY
9
10 -- predefined type, no DDL - XMLTYPE
11
12 CREATE TABLE Customers
13     (
14      customerID           NUMBER  NOT NULL ,
15      name                 VARCHAR2 (100 CHAR)  NOT NULL ,
16      quantity             INTEGER  NOT NULL ,
17      address              VARCHAR2 (40 CHAR)  NOT NULL ,
18      productName          VARCHAR2 (100 CHAR)  NOT NULL ,
19      Employees_employeeID NUMBER  NOT NULL
20     )
21     LOGGING
22 ;
23
24 ALTER TABLE Customers
25     ADD CONSTRAINT Customers_PK PRIMARY KEY ( customerID ) ;
26
27 CREATE TABLE Employees
28     (
29      employeeID NUMBER  NOT NULL ,
30      firstname  VARCHAR2 (100 CHAR)  NOT NULL ,
31      lastname   VARCHAR2 (100 CHAR)  NOT NULL ,
32      address    VARCHAR2 (40 CHAR)  NOT NULL
33     )
34     LOGGING
```

After saving this, we will execute the DDL in oracle:



Worksheet | Query Builder

```
CREATE TABLE Customers
    (
     customerID           NUMBER  NOT NULL ,
     name                 VARCHAR2 (100 CHAR)  NOT NULL ,
     quantity             INTEGER  NOT NULL ,
     address              VARCHAR2 (40 CHAR)  NOT NULL ,
     productName          VARCHAR2 (100 CHAR)  NOT NULL ,
     Employees_employeeID NUMBER  NOT NULL
    )
    LOGGING
;

ALTER TABLE Customers
    ADD CONSTRAINT Customers_PK PRIMARY KEY ( customerID ) ;

CREATE TABLE Employees
    (
     employeeID NUMBER  NOT NULL ,
     firstname  VARCHAR2 (100 CHAR)  NOT NULL ,
     lastname   VARCHAR2 (100 CHAR)  NOT NULL ,
     address    VARCHAR2 (40 CHAR)  NOT NULL
    )
    LOGGING
;

ALTER TABLE Employees
    ADD CONSTRAINT Employees_PK PRIMARY KEY ( employeeID ) ;

CREATE TABLE Orders
    (
     orderNo              NUMBER  NOT NULL ,
     quantity             INTEGER ,
     shipped_date         DATE ,
     received_date        DATE ,
     Customers_customerID NUMBER  NOT NULL
```

Script Output x

Task completed in 0.1 seconds

```
Table CUSTOMERS altered.


Table ORDERS altered.


Table PRODUCTS altered.
```

Now we will populate each table with 3 records. This is for the employees table:

Worksheet | Query Builder

```
INSERT INTO Employees (EMPLOYEEID, FIRSTNAME, LASTNAME, ADDRESS)
VALUES (1, 'Ali', 'Khan', 'Peshawar');

INSERT INTO Employees (EMPLOYEEID, FIRSTNAME, LASTNAME, ADDRESS)
VALUES (2, 'Sara', 'Ahmed', 'Lahore');

INSERT INTO Employees (EMPLOYEEID, FIRSTNAME, LASTNAME, ADDRESS)
VALUES (3, 'Usman', 'Zahid', 'Karachi');
```

```
1 row inserted.

1 row inserted.

1 row inserted.
```

This is for products table:

rksheet | Query Builder

```
INSERT INTO Products (PRODUCTID, NAME, PRICE, STOCK, EMPLOYEES_EMPLOYEEID)
VALUES (1, 'Monitor', 15000.00, 10, 1);

INSERT INTO Products (PRODUCTID, NAME, PRICE, STOCK, EMPLOYEES_EMPLOYEEID)
VALUES (2, 'Keyboard', 3000.50, 25, 2);

INSERT INTO Products (PRODUCTID, NAME, PRICE, STOCK, EMPLOYEES_EMPLOYEEID)
VALUES (3, 'Mouse', 1500.75, 50, 3);
```

```
1 row inserted.


1 row inserted.


1 row inserted.
```

This is for the customers table:

```
sheet  Query Builder
INSERT INTO Customers (CUSTOMERID, NAME, QUANTITY, ADDRESS, PRODUCTNAME, EMPLOYEES_EMPLOYEEID)
VALUES (1, 'Urooba', 2, 'Islamabad', 'Laptop', 1);

INSERT INTO Customers (CUSTOMERID, NAME, QUANTITY, ADDRESS, PRODUCTNAME, EMPLOYEES_EMPLOYEEID)
VALUES (2, 'Aimal', 1, 'Peshawar', 'Phone', 2);

INSERT INTO Customers (CUSTOMERID, NAME, QUANTITY, ADDRESS, PRODUCTNAME, EMPLOYEES_EMPLOYEEID)
VALUES (3, 'Junaid', 5, 'Lahore', 'Keyboard', 3);
```

```
1 row inserted.


1 row inserted.


1 row inserted.
```

This is for the orders table:

```
rksheet  Query Builder
INSERT INTO Orders (ORDERNO, QUANTITY, SHIPPED_DATE, RECEIVED_DATE, CUSTOMERS_CUSTOMERID)
VALUES (101, 2, TO_DATE('2025-03-20', 'YYYY-MM-DD'), TO_DATE('2025-03-25', 'YYYY-MM-DD'), 1);

INSERT INTO Orders (ORDERNO, QUANTITY, SHIPPED_DATE, RECEIVED_DATE, CUSTOMERS_CUSTOMERID)
VALUES (102, 1, TO_DATE('2025-03-21', 'YYYY-MM-DD'), TO_DATE('2025-03-26', 'YYYY-MM-DD'), 2);

INSERT INTO Orders (ORDERNO, QUANTITY, SHIPPED_DATE, RECEIVED_DATE, CUSTOMERS_CUSTOMERID)
VALUES (103, 3, TO_DATE('2025-03-22', 'YYYY-MM-DD'), TO_DATE('2025-03-27', 'YYYY-MM-DD'), 3);
```

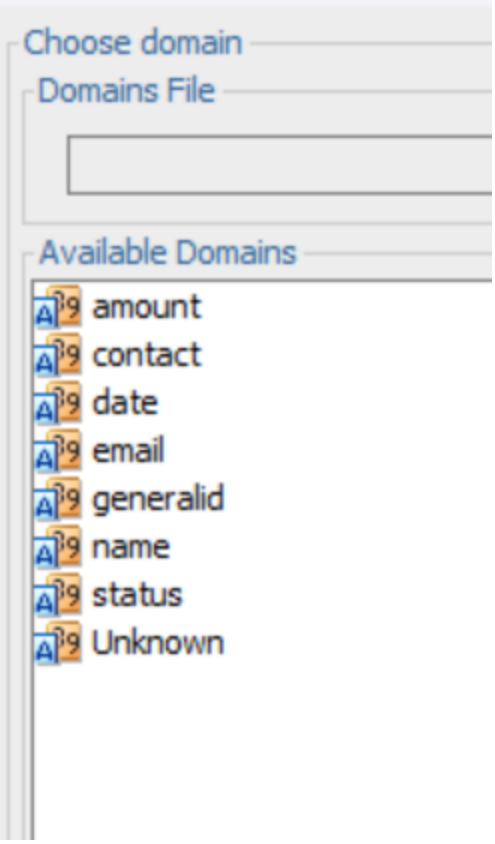
```
1 row inserted.

1 row inserted.

1 row inserted.
```

Task 3:

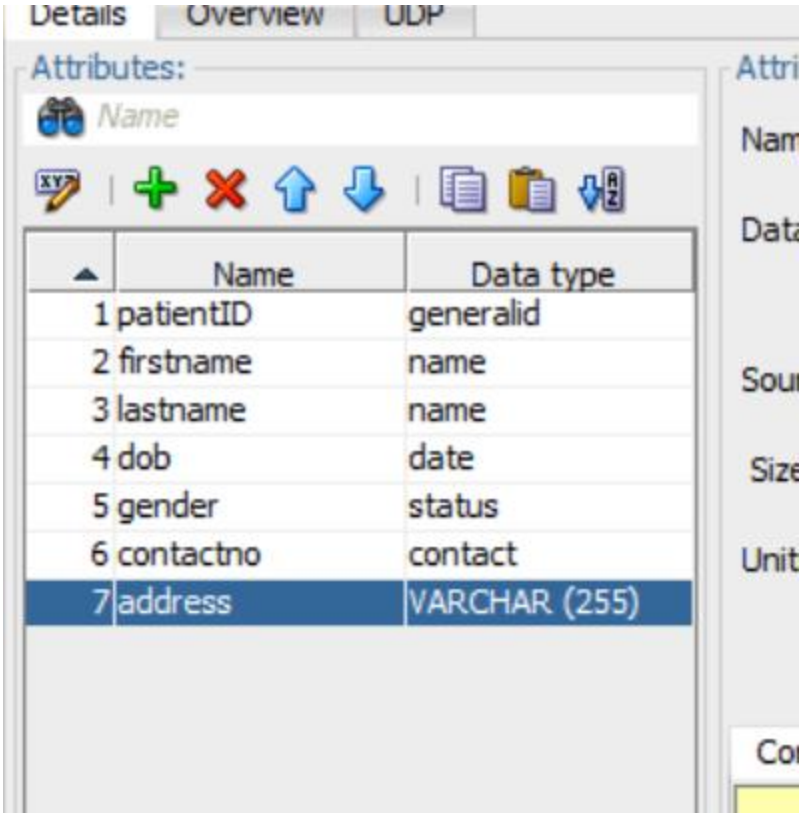
This system is designed to streamline various hospital functions, including patient management, appointment scheduling, and electronic health record (EHR) management. Develop the Logical Model Diagram for the EHR database and build the design using a data modeling tool data modeler. Develop the Relational Model Diagram for the EHR database and build the design using a data modeling tool data modeler (create foreign keys in each table where required.) Generate DDL.

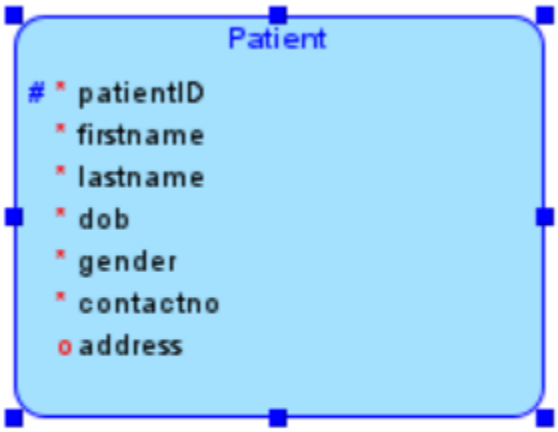
Answer:

First we add domains:

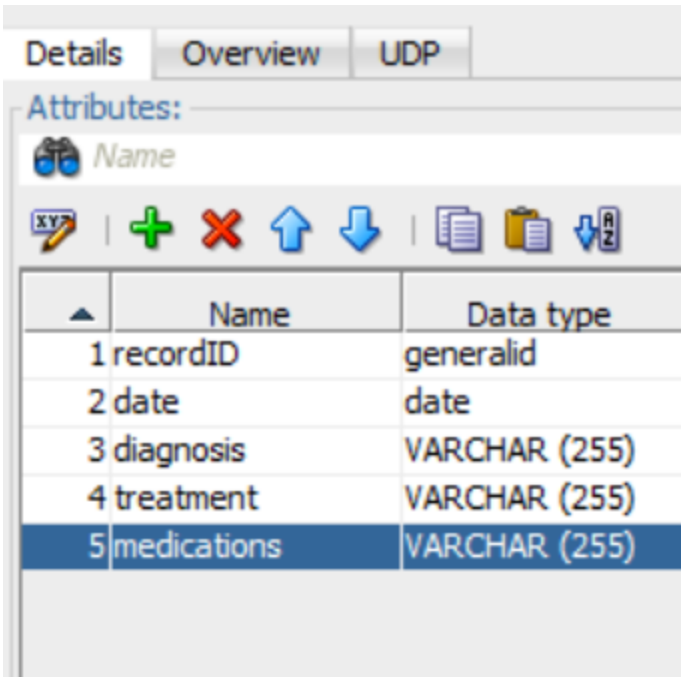


Now we make Patient entity with the following attributes:

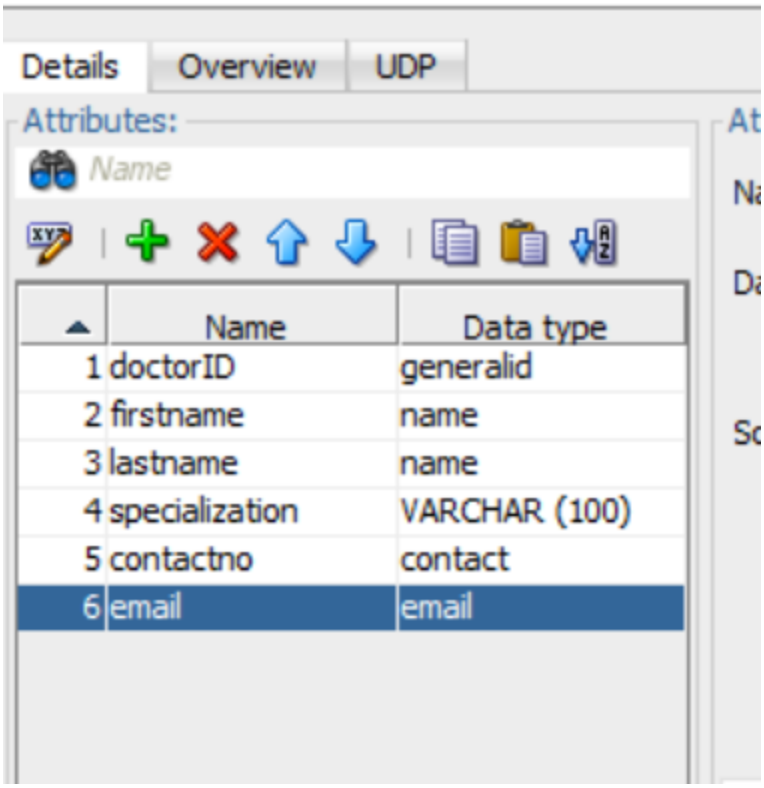


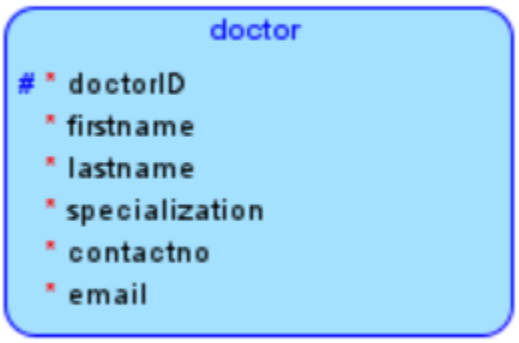


We make Medical Record entity with the following attributes:

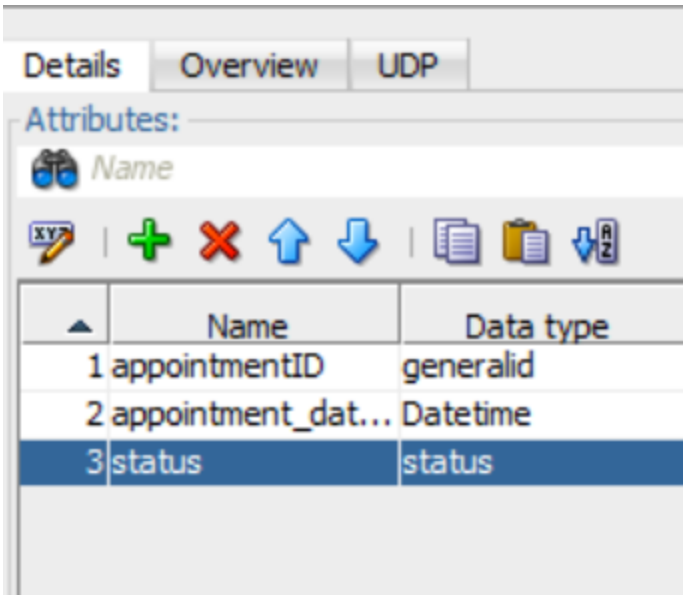


We make Doctor entity with the following attributes:

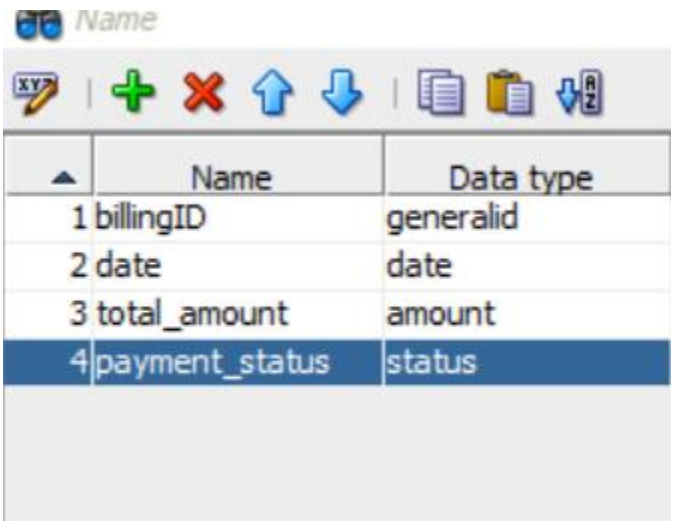


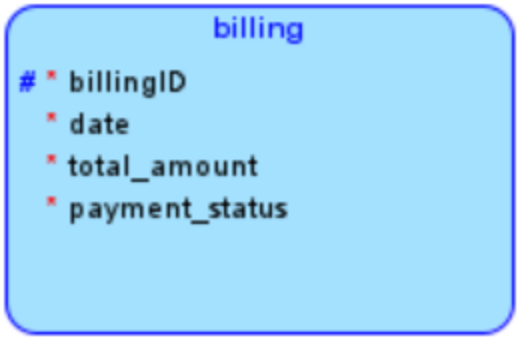


We make Appointment entity with the following attributes:

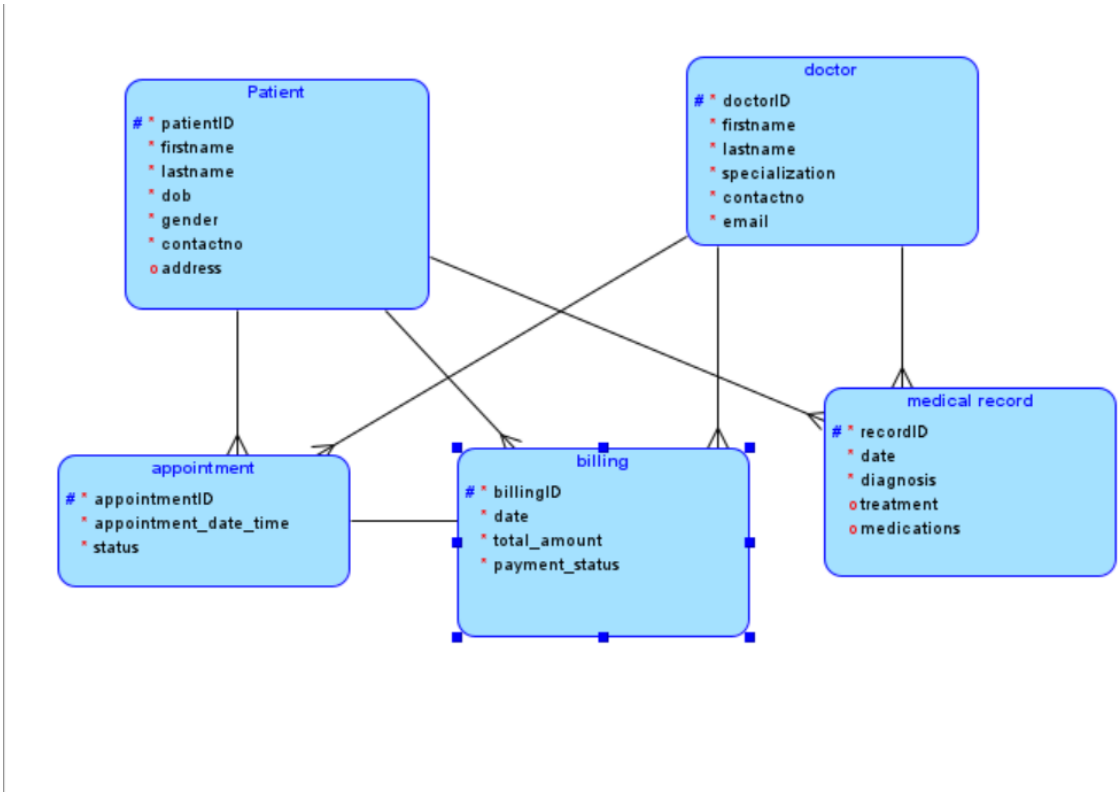


We make Billing entity with the following attributes:

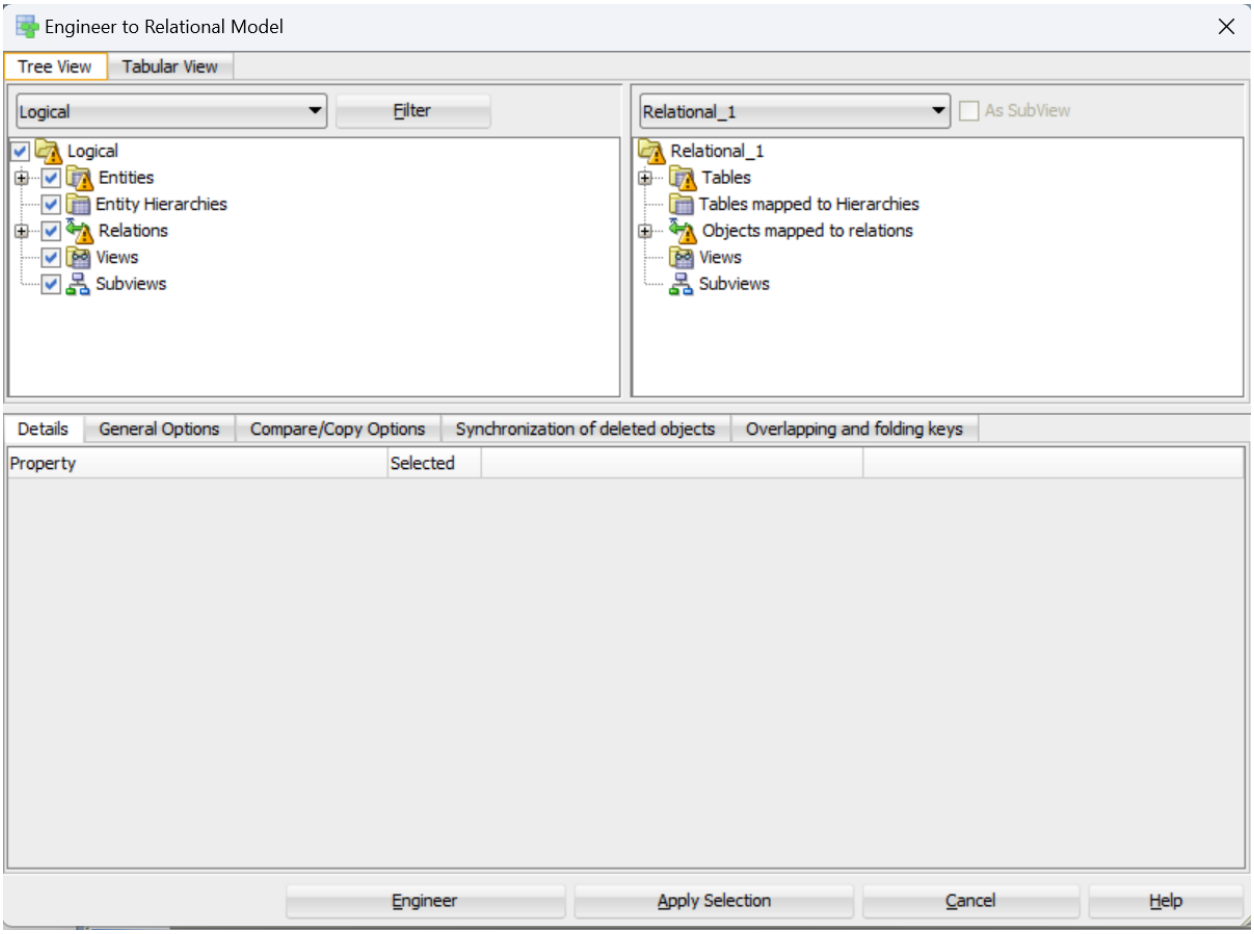




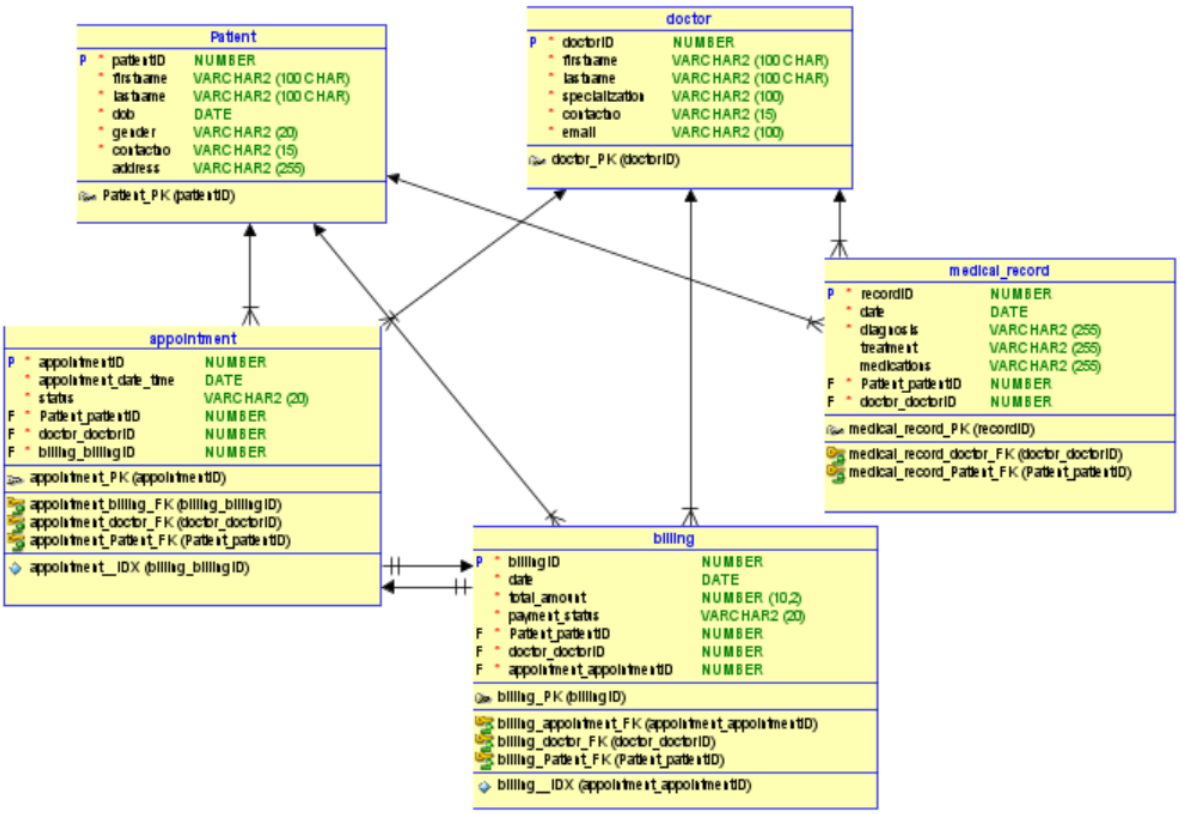
Now we add the relationships for all entities which looks like this:



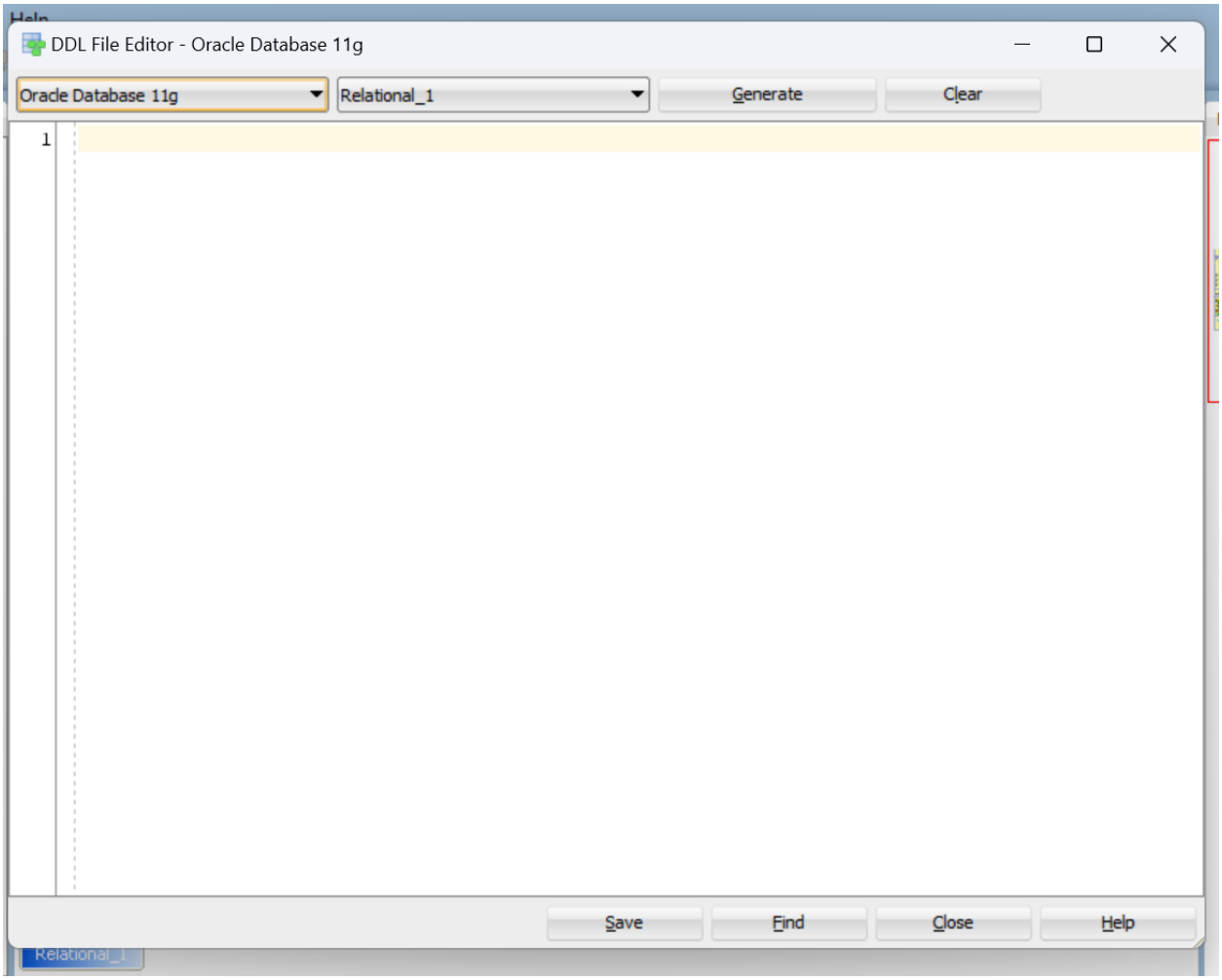
Now we engineer it to convert it into relational model:



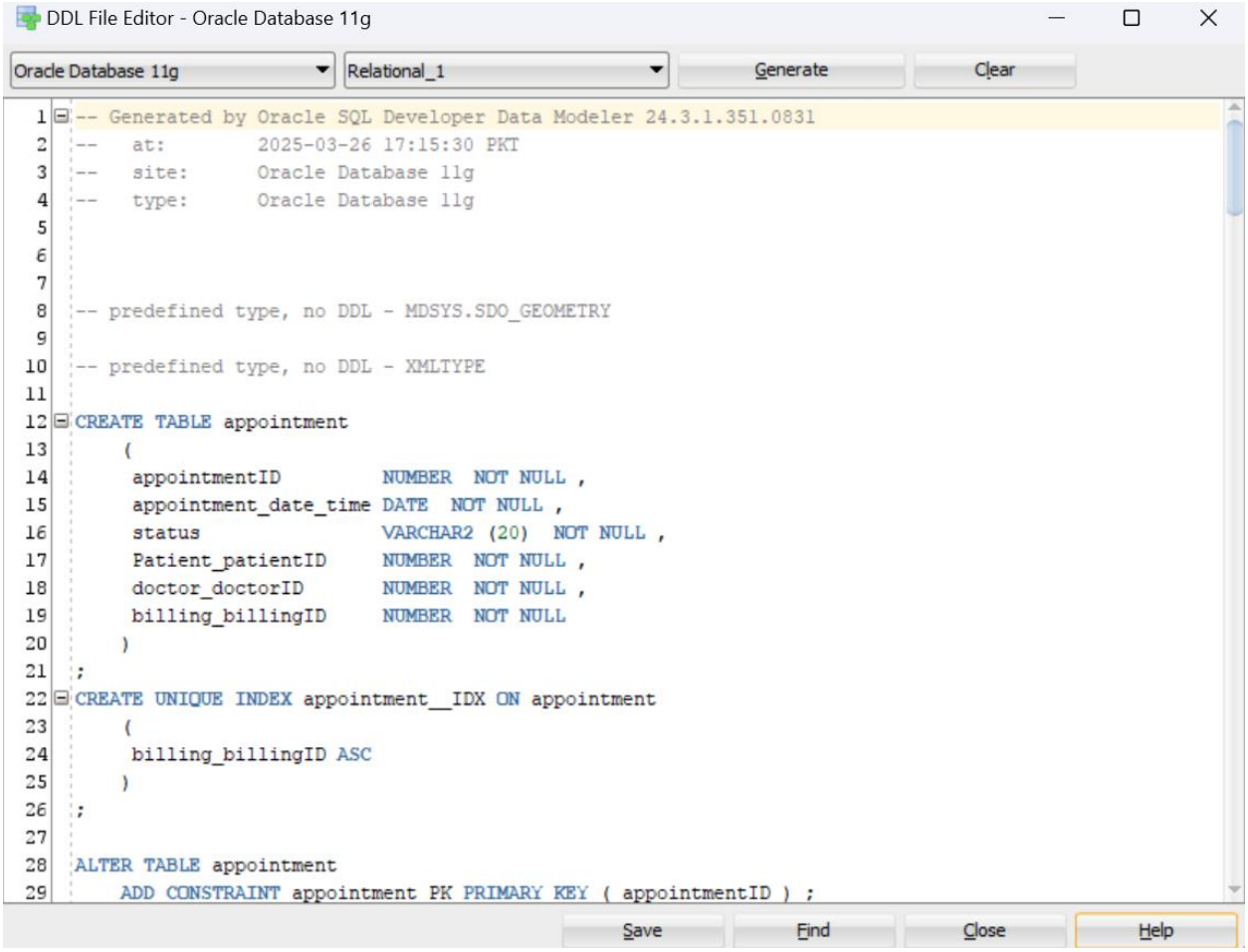
This is the relational model:



Now we generate the DDL:



This is the DDL:

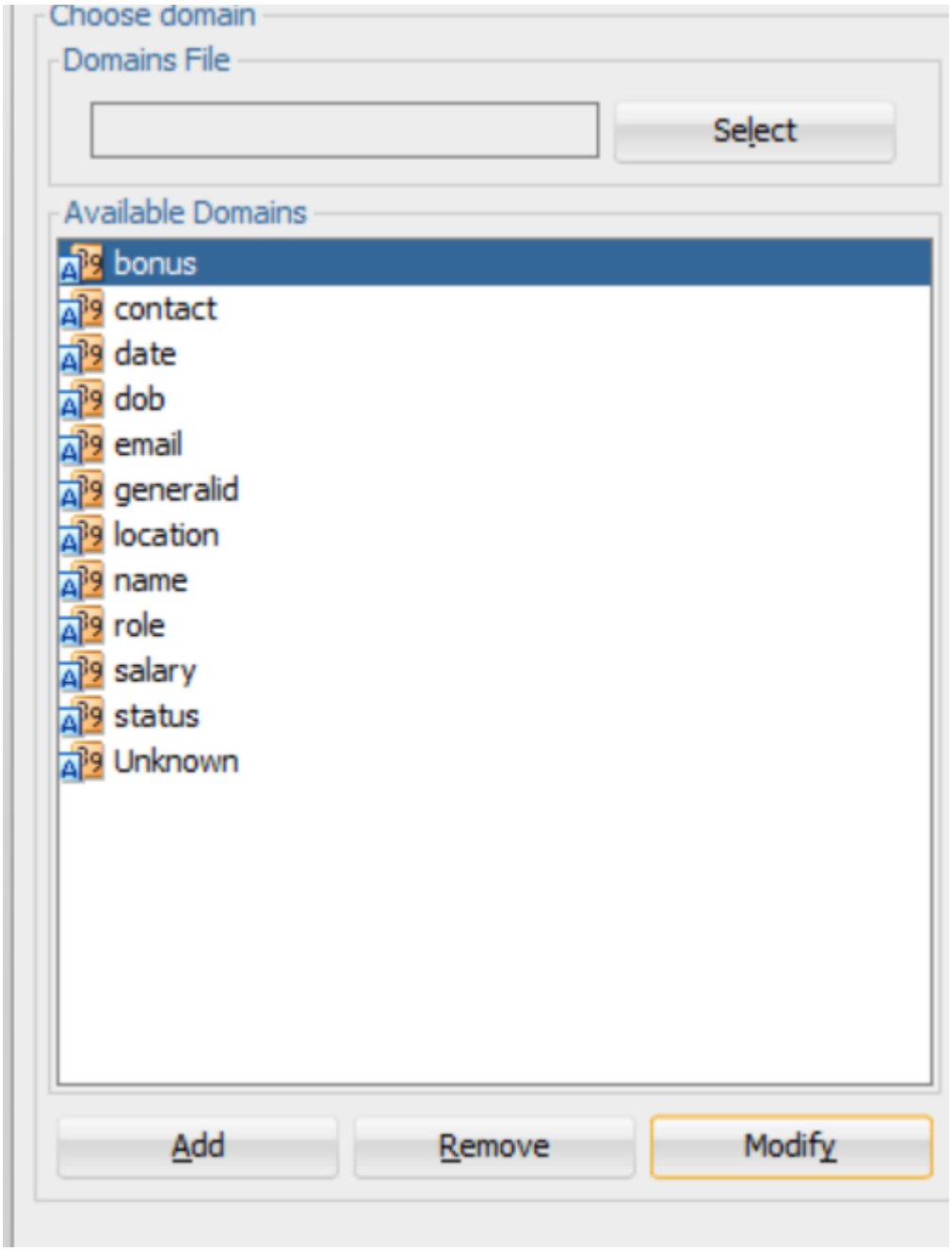


Task 4:

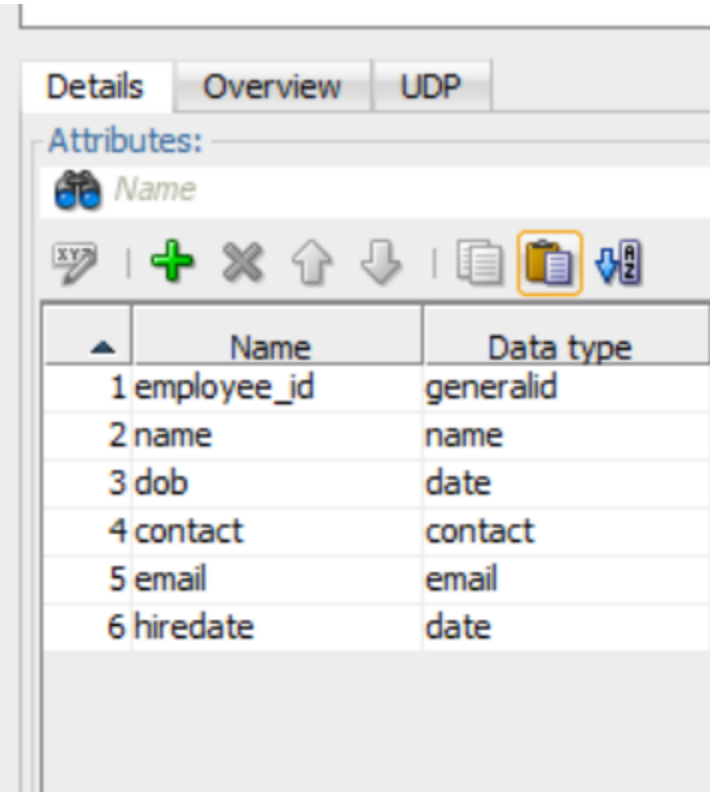
Design a database to manage employees, departments, projects, attendance, and payroll. Logical Model: Create a Logical Model Diagram. Relational Model: Design a Relational Model with foreign keys. DDL Script: Generate SQL DDL to define the schema. Constraints: Ensure Email is Unique & Not Null, enforce data integrity.

Answer:

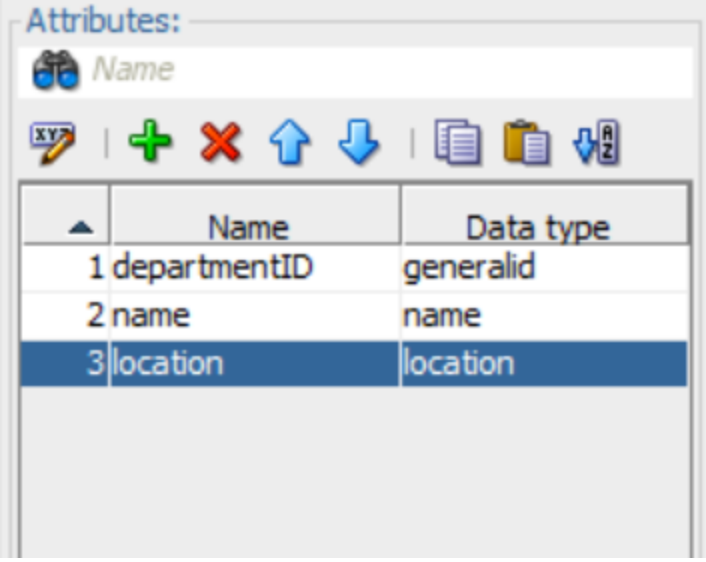
First we add domains:



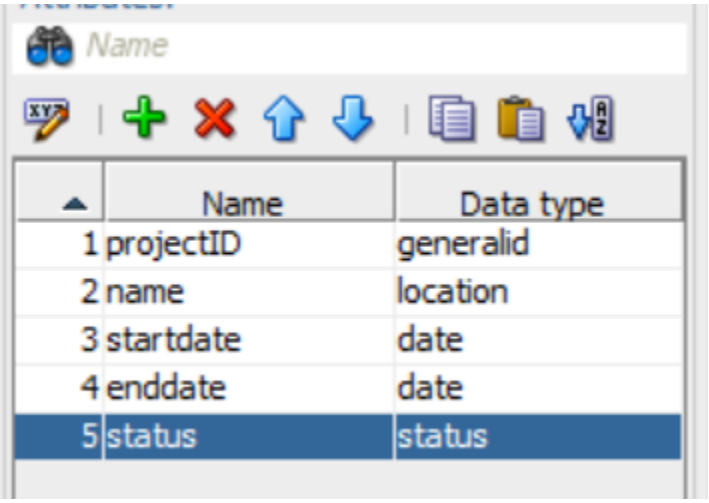
Now we make employees entity with the following attributes:

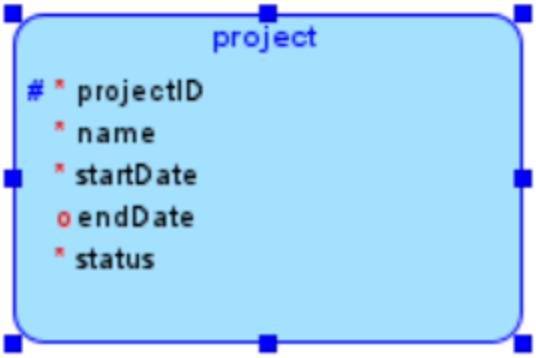


We make department entity with the following attributes:

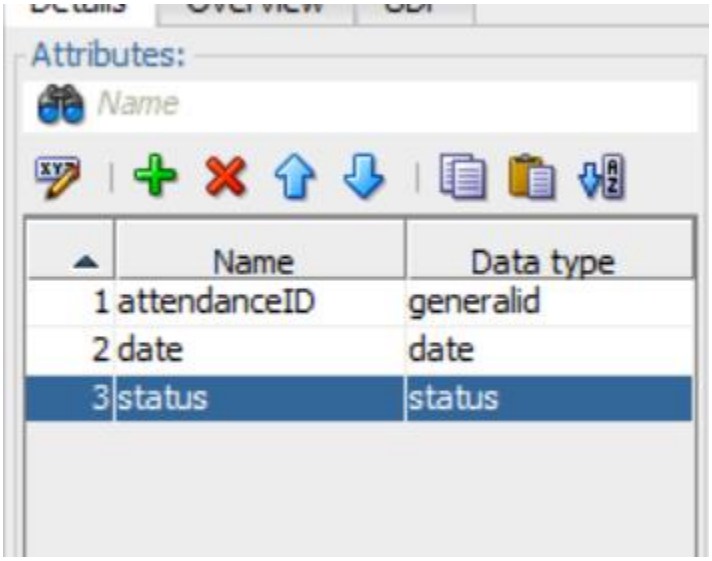


Now make project entity with the following attributes:

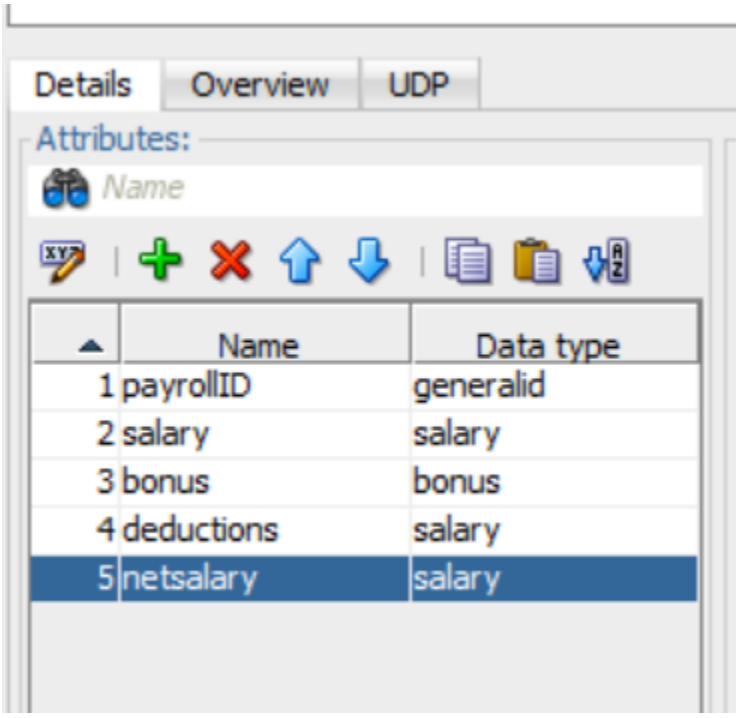




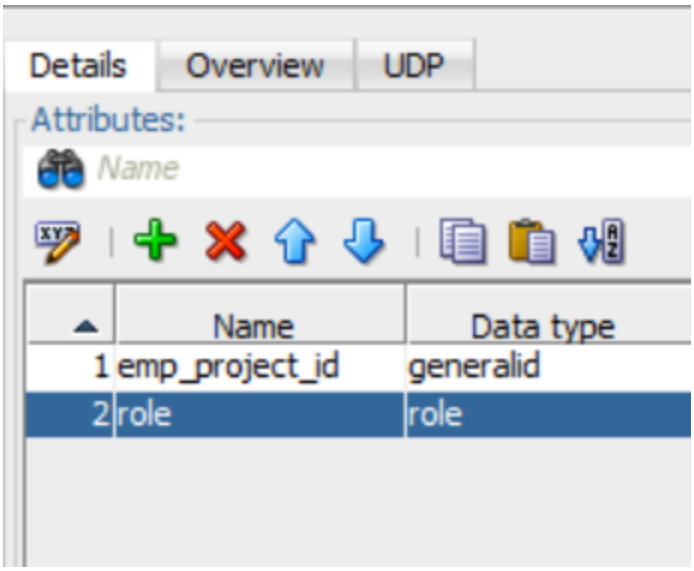
Now we make attendance entity with the following attributes:



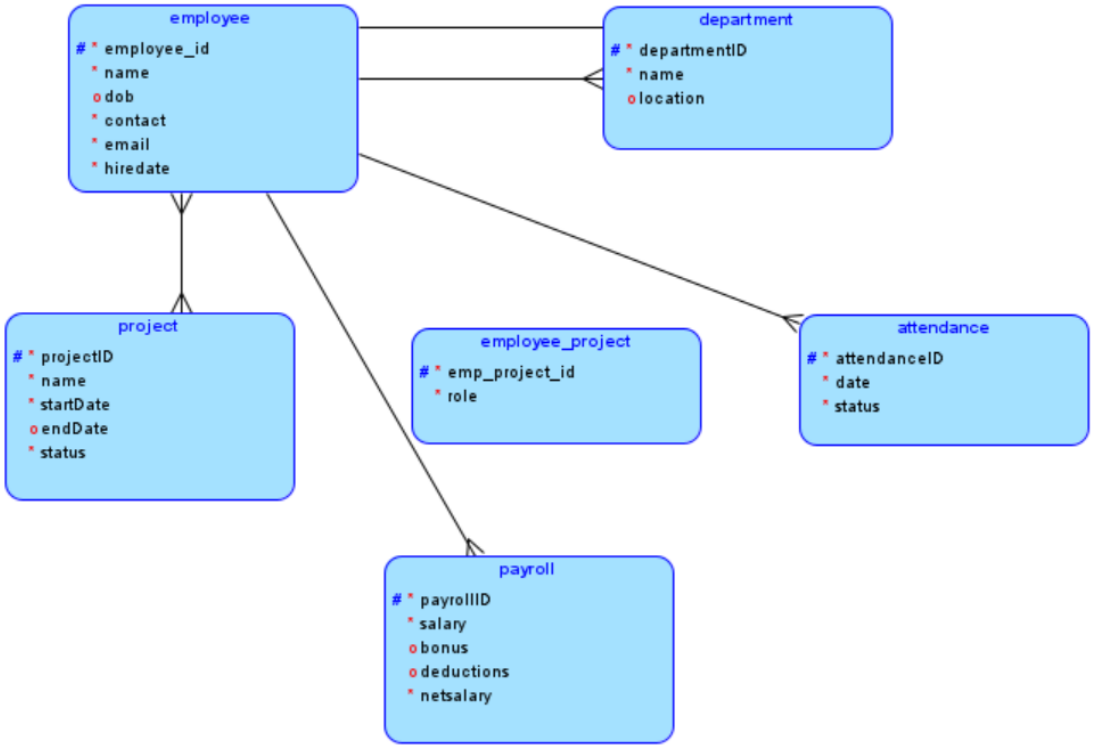
Now we make payroll entity with the following attributes:



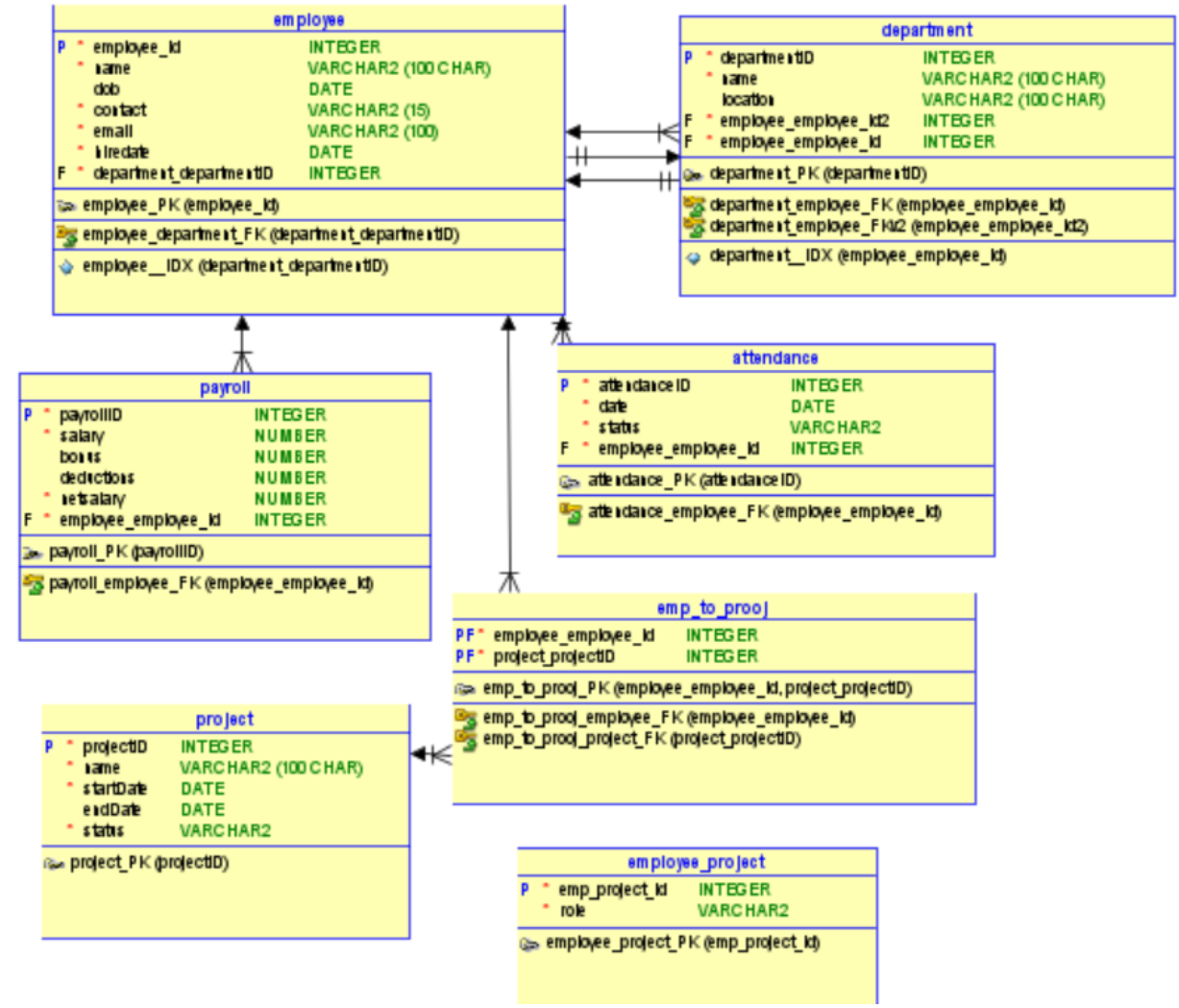
Finally we make employee_project entity with the following attributes:



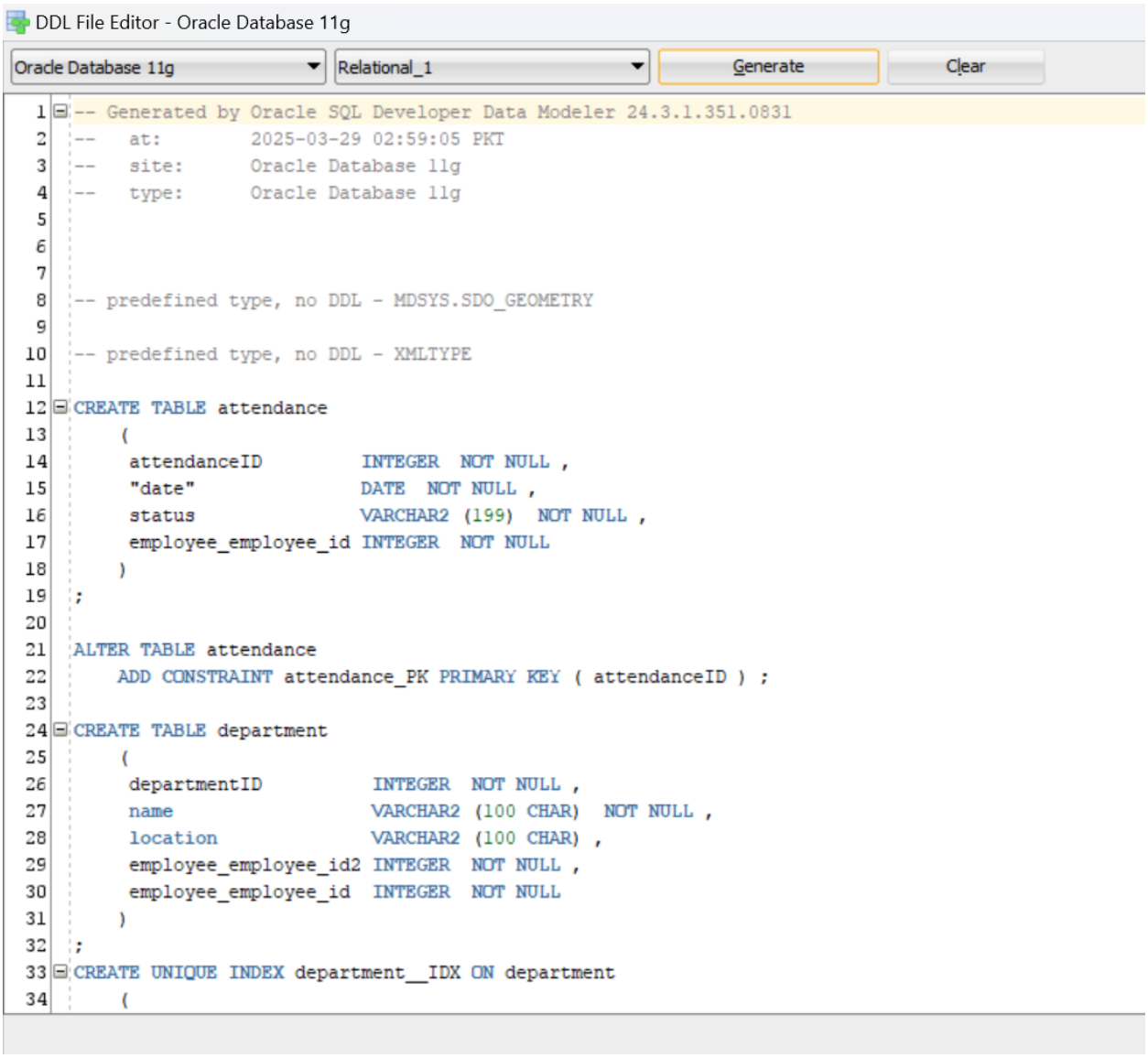
Now we connect the entities according to their relationships:



Now we engineer the model:



Now we generate the DDL script:

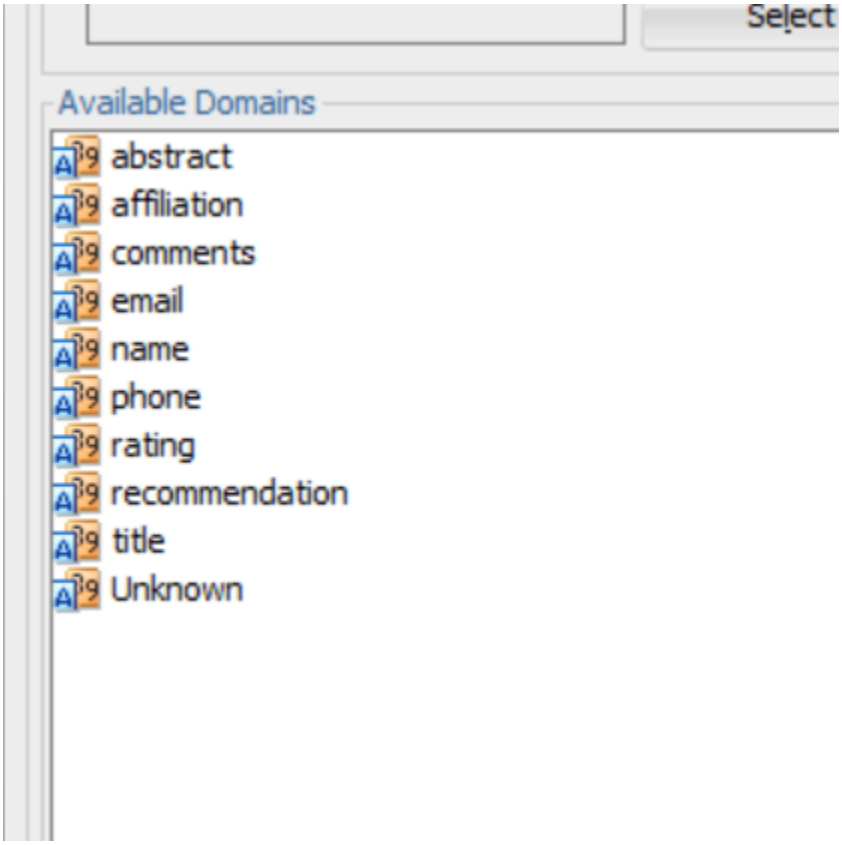


Task 5:

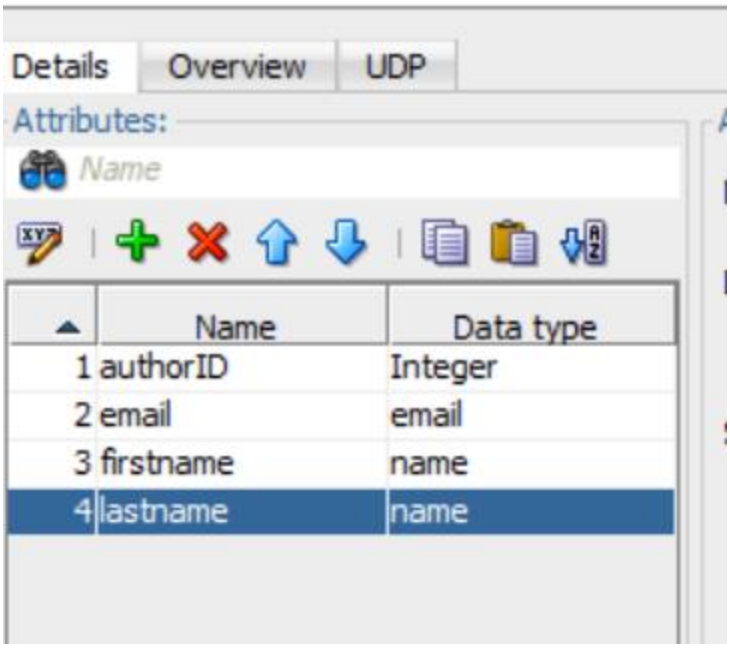
Develop the Logical Model Diagram for the CONFERENCE_REVIEW database and build the design using a data modeling tool data modeler. Develop the Relational Model Diagram for the CONFERENCE_REVIEW database and build the design using a data modeling tool data modeler. Generate DDL.

Answer:

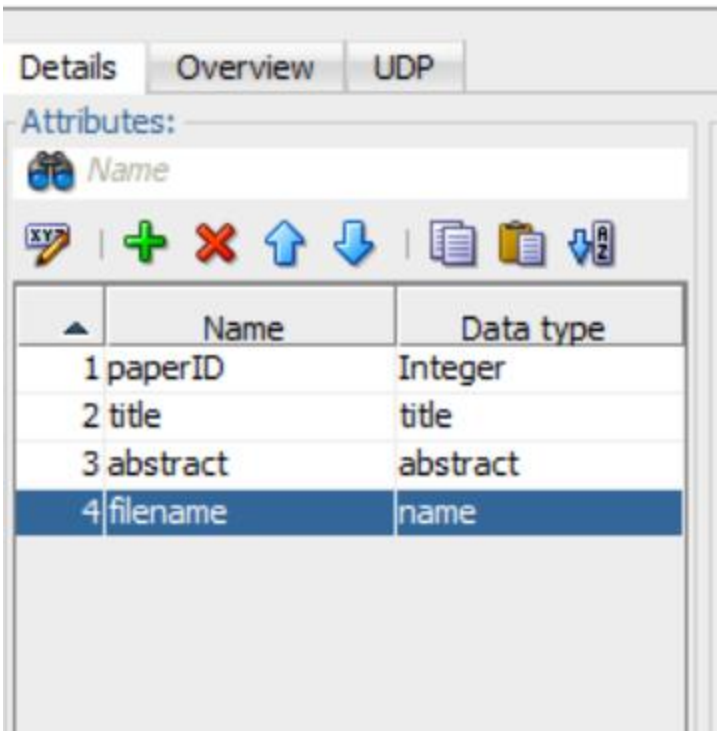
First we add domains:

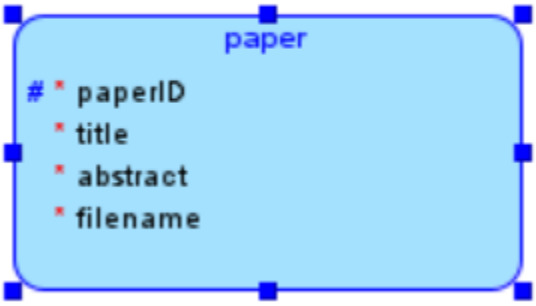


Now we create author entity with the following attributes:

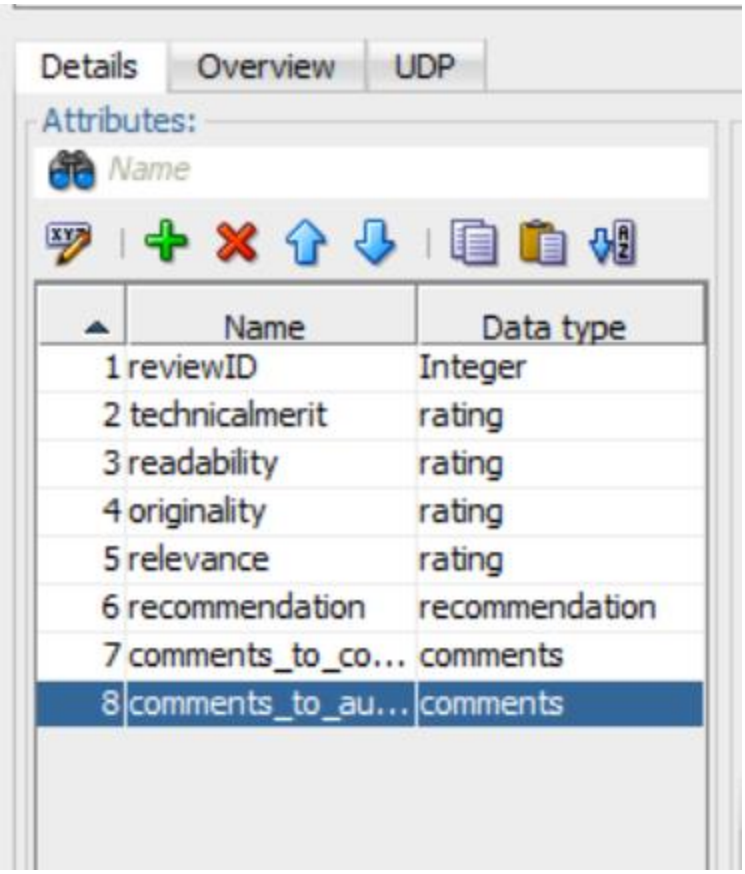


Now we will make paper entity with the following attributes:

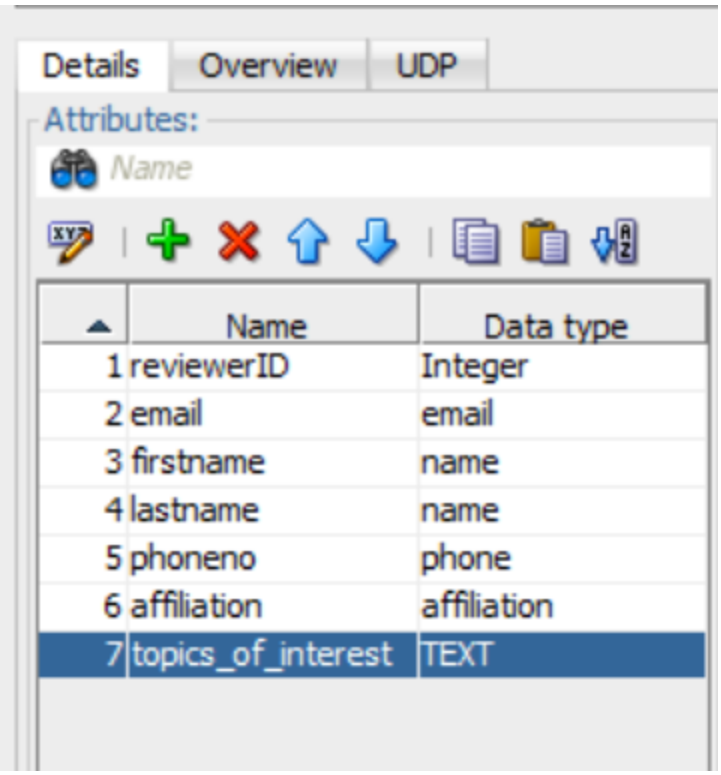




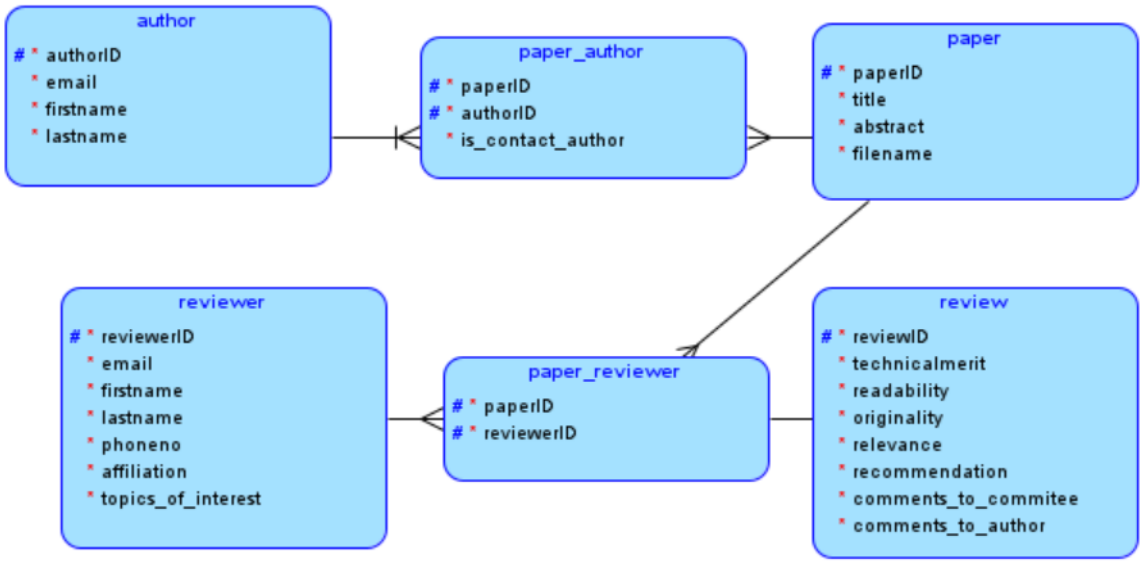
Now we will make review entity with the following attributes:



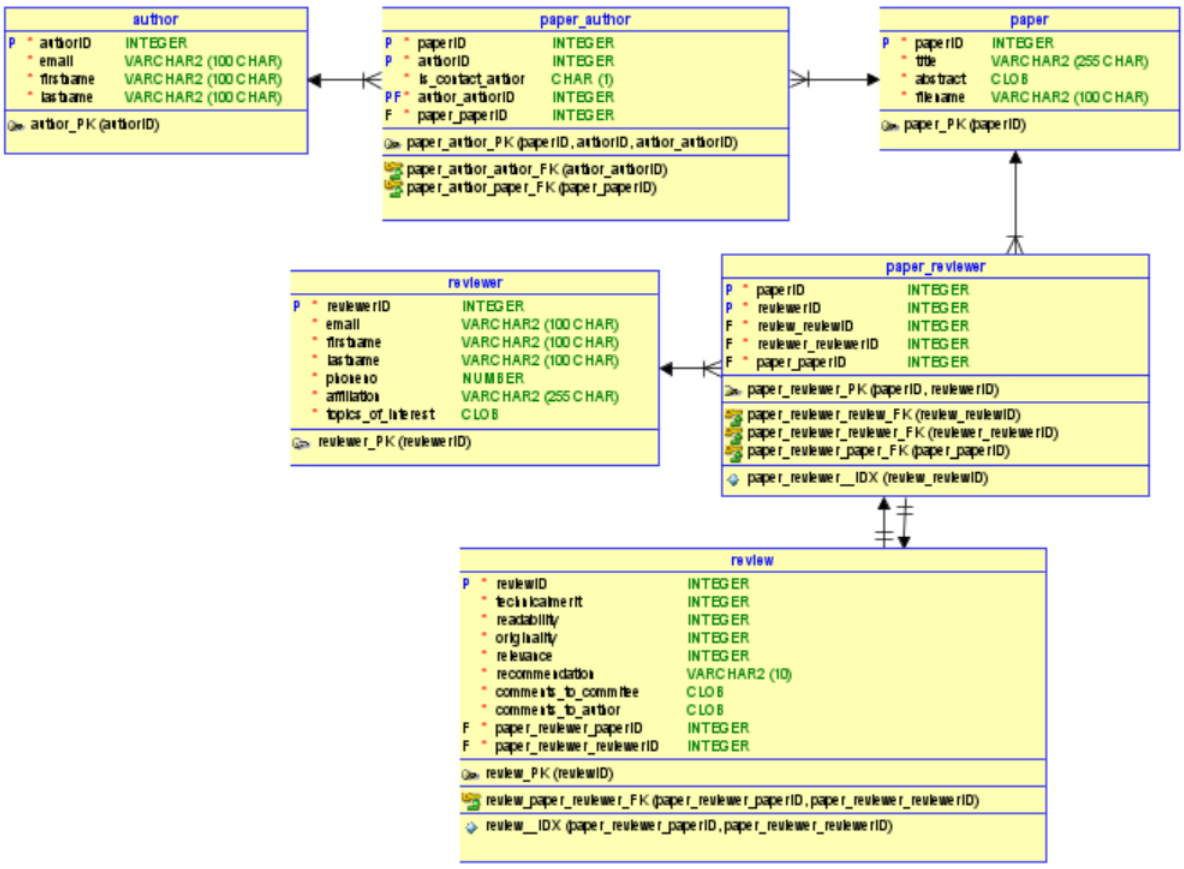
Now make reviewer entity with the following attributes:



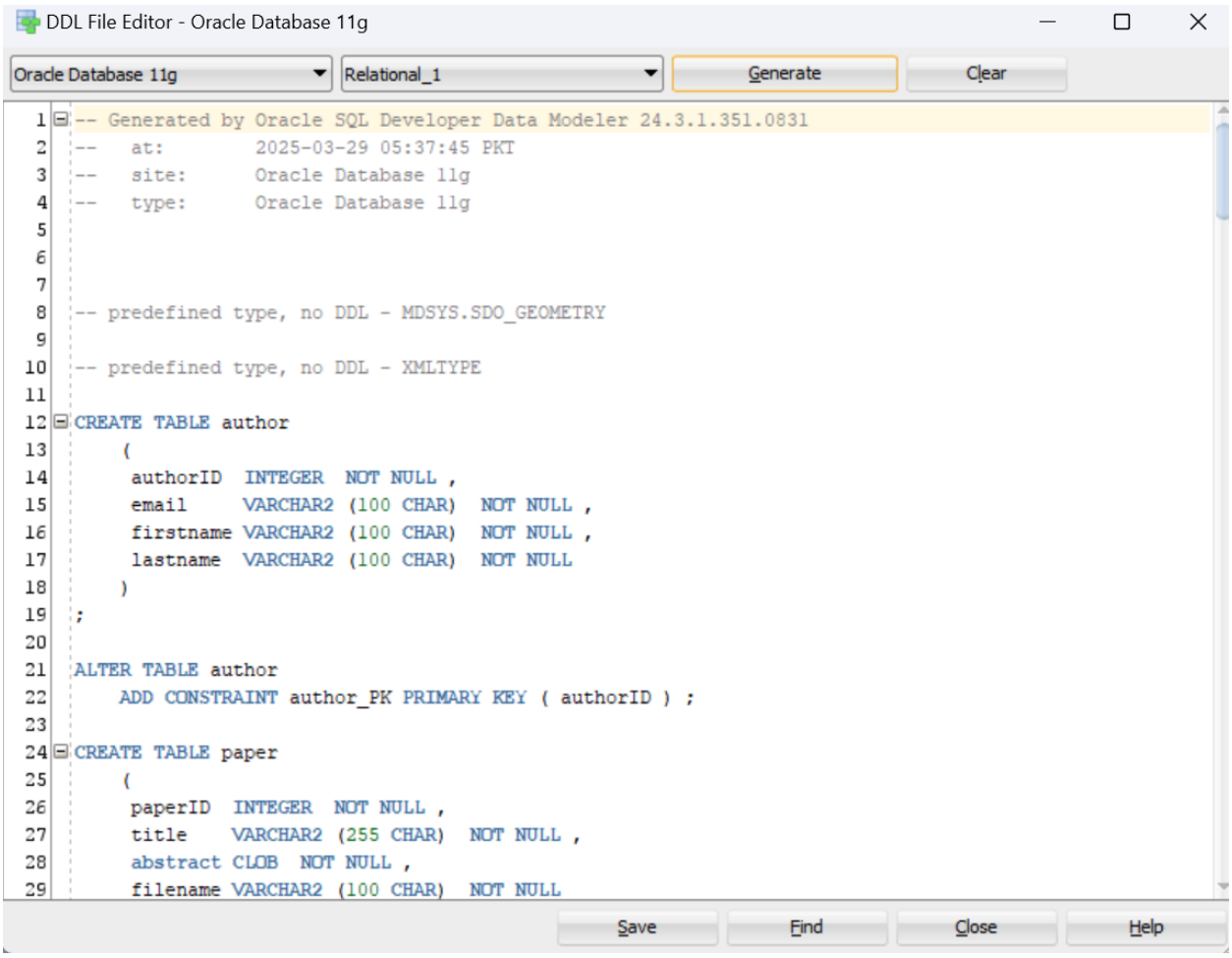
Now we connect relationships between the entities:



Now we engineer it to relational model:



Now we generate the DDL script for it:


DDL File Editor - Oracle Database 11g
Oracle Database 11g
Relational_1
Generate
Clear
-- Generated by Oracle SQL Developer Data Modeler 24.3.1.351.0831
--   at: 2025-03-29 05:37:45 PKT
--   site: Oracle Database 11g
--   type: Oracle Database 11g
-- predefined type, no DDL - MDSYS.SDO_GEOMETRY
-- predefined type, no DDL - XMLTYPE
CREATE TABLE author
(
authorID INTEGER NOT NULL ,
email VARCHAR2 (100 CHAR) NOT NULL ,
firstname VARCHAR2 (100 CHAR) NOT NULL ,
lastname VARCHAR2 (100 CHAR) NOT NULL
)
;
ALTER TABLE author
ADD CONSTRAINT author_PK PRIMARY KEY (authorID) ;
CREATE TABLE paper
(
paperID INTEGER NOT NULL ,
title VARCHAR2 (255 CHAR) NOT NULL ,
abstract CLOB NOT NULL ,
filename VARCHAR2 (100 CHAR) NOT NULL
Save
Find
Close
Help