**DSC 450: Database Processing for Large-Scale Analytics**

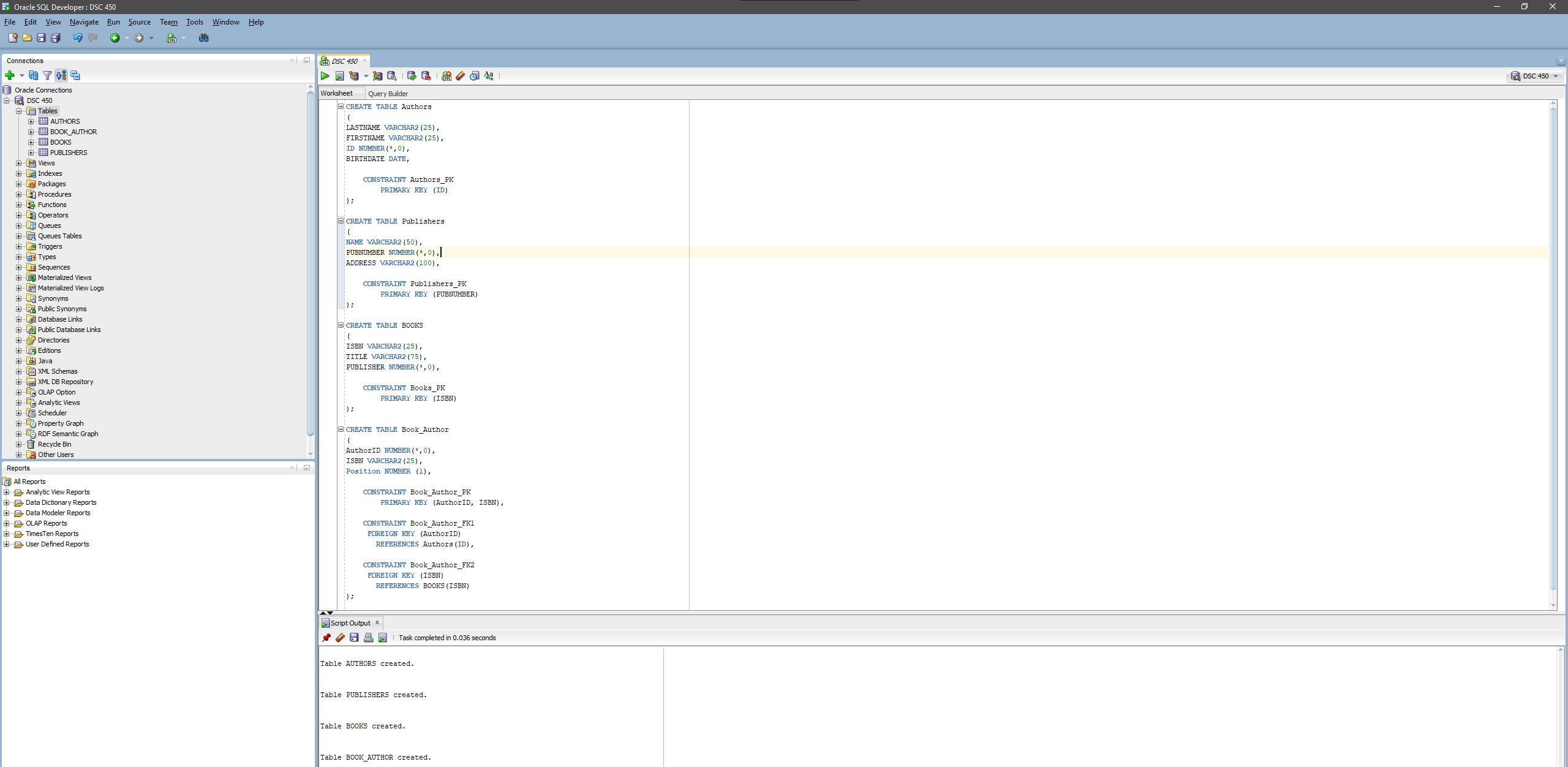
**Assignment Module 2**

**Name – Goutham Selvakumar**

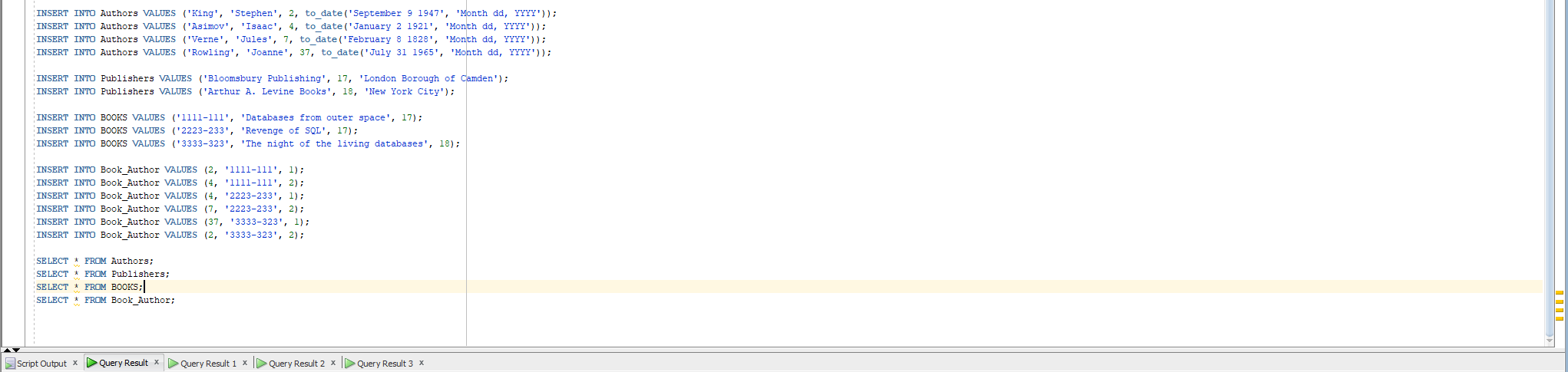
**ID - 2092286**

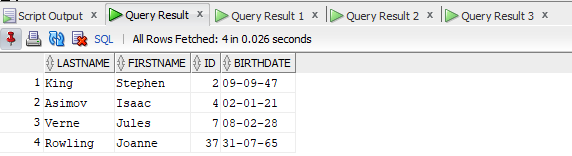
**Part 1**

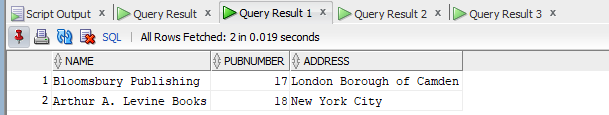
1. **Using your logical schema from previous assignment (Part 2-a), write the necessary SQL DDL script to create the tables. Be sure to specify every primary key and every foreign key. Make reasonable assumptions regarding the attribute domains (for example, setting every column to *VARCHAR2 (100)* is not reasonable).**

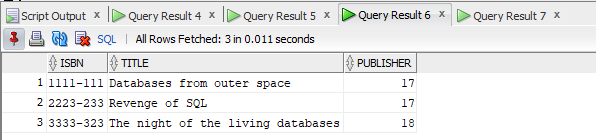
****

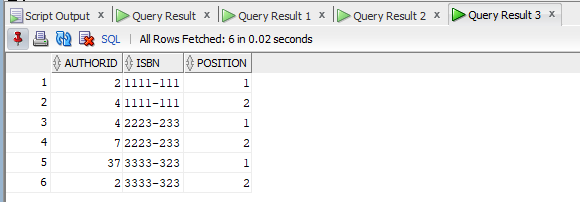
1. **Write SQL INSERT statements to populate your database from Part 2-a with the following data (NOTE: remember that strings would need to use single quotes, e.g., 'Asimov'). Be sure to verify that your statements worked correctly and loaded the data in Oracle (see tutorial on how to connect to Oracle using SQLDeveloper).** 
   1. (King, Stephen, 2, September 9 1947)
   2. (Asimov, Isaac, 4, January 2 1921)
   3. (Verne, Jules, 7, February 8 1828)
   4. (Rowling, Joanne, 37, July 31 1965)
   5. (Bloomsbury Publishing, 17, London Borough of Camden)
   6. (Arthur A Levine Books, 18, New York City)
   7. (1111-111, Databases from Outer Space, 17)
   8. (2223-233, Revenge of SQL, 17)
   9. (3333-323, The Night of the Living Databases, 18)
   10. (2, 1111-111, 1)
   11. (4, 1111-111, 2)
   12. (4, 2223-233, 1)
   13. (7, 2223-233, 2)
   14. (37, 3333-323, 1)
   15. (2, 3333-323, 2)



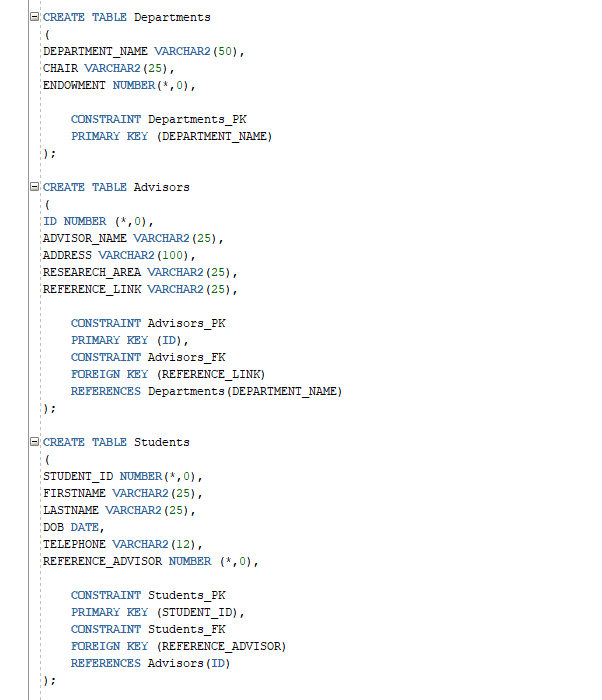








1. **Using logical schema from previous assignment (Part 2-b) write the necessary SQL DDL script to create the tables. Be sure to specify every primary key and every foreign key.**

****

1. **Write a python function to validate SQL insert statements. The function will take a string containing a SQL insert statement and print two kinds of messages. 1) “Invalid insert” or 2) Inserting [list of values] into [the target] table. The values and the table name would be based on each particular statement. For validating the statement, you only have to check that the insert statement starts with INSERT INTO and that statements ends with a semicolon (;).**

Examples:

validateInsert(**”INSERT INTO Students VALUES (1, 'Jane', 'B+');”**)

* Inserting (1, 'Jane', 'A-') into Students table

validateInsert(**“INSERT INTO Students VALUES (1, 'Jane', 'B+')”**)

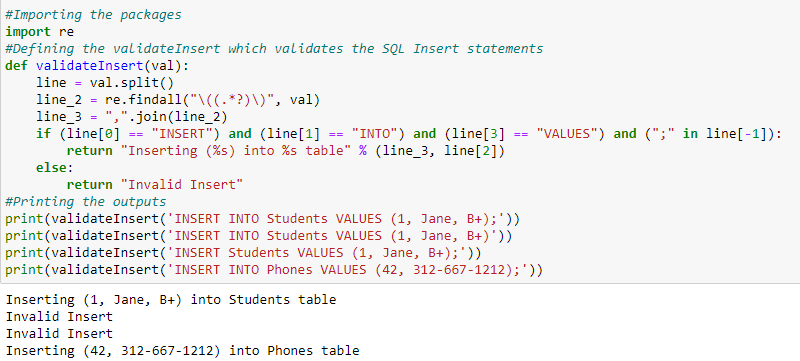
* Invalid insert

validateInsert(**“INSERT Students VALUES (1, 'Jane', B+);”**)

* Invalid insert

validateInsert(**“INSERT INTO Phones VALUES (42, '312-667-1212');”**)

* Inserting (42, '312-667-1212') into Phones table



**Part 2**

**Consider a MEETING table that records information about meetings between clients and representatives in the company. Each record contains the names of the client and the representative’s name as well as the office number, floor and the building. Finally, each record contains the city that the building is in and the date of the meeting. The table is in First Normal Form and the primary key is (Client, Office).**

**(Date, Client, Office, Floor, Building, City, Representative)**

**You are given the following functional dependencies:**

**Building → City**

**Office → Floor, Building, City**

**Client → Representative**

**Client, Office → Date**

**Meeting Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Client | Office | Floor | Building | City | Executive |
| March 19th 2023 | Goutham Selvakumar | Google | 12th | GB11 | California | Miles |
| April 11th 2023 | Rakesh Vijayakumar | Google | 15th | GB11 | California | James |
| June 9th 2023 | Ranjith Rajendran | Google | 2nd | GB11 | California | Kingston |
| September 5th 2023 | Surya Mani | Google | 7th | GB17 | New York | Larry |

1. **For the functional dependency Building → City, explain the redundancy problem and possible consequences through an example (you can make up your own building names as you see fit).**

We can assume that between the City and the Building there is a redundancy present among them. Furthermore, we can see that the “GB11” always corresponds to the city “California”, the building “GB17” corresponds to the city “New York”, so the building column actually responds to the city column. Moreover, there is a problem being faced if the redundancy is not removed while adding (or) updating the new rows in the table.

1. **Remove any existing partial dependencies and convert the logical schema to the Second Normal Form. Please remember that when performing schema decomposition you need to denote the primary key for every new table as well as the foreign key that will allow us to reconstruct the original data.**

R (Date,Client,Office,Floor,Building,City,Executive) are the partial dependencies.

Row1 (Client, Executive)

Row2 (Date,Office,Floor,Building,City)

1. **Remove any existing transitive dependencies to create a set of logical schemas in Third Normal Form. Again, remember to denote primary keys and foreign keys (including which primary key those foreign keys point to).**

Building is dependable on the City and Office which in terms tells us that the office depends indirectly upon the city which explains the transitive dependency.

Row3 (Office, Floor, Building, City) Row4 (Client,Executive)

Row5 (Date)

**Part 3**

Consider a table that stores information about students, student name, GPA, honors list and the credits that the student had completed so far.

(First, Last, GPA, Honor, Credits)

You are given the following functional dependencies

First, Last → GPA, Honor, Credits

Last → Credits (yes, this FD does not make any practical sense)

1. **Is this schema in Second Normal Form? If not, please state which FDs violate 2NF and decompose the schema accordingly.**

Since there are no functional dependencies present among, also the first and last are the only dependent. It is considered to be in the second normal form.

1. **Is this schema in Third Normal Form? If not, please state which FDs violate 3NF and decompose the schema accordingly.**

There is a case of Transitive Functional Dependency present which represents that the Last depends on credits which is not good for FD, whereas the First and Last depends on the GPA, honor, and credits.

(Last, Credits)

(First, Last, GPA, Credits)