Curtin University – Department of Computing

Assignment Cover Sheet / Declaration of Originality

Complete this form if/as directed by your unit coordinator, lecturer or the assignment specification.

Last name:	Mahen	Student ID:	21029112
Other name(s):	Justyn		
Unit name:	Database Systems	Unit ID:	ISYS2014
Lecturer / unit coordinator:	Nimalika Fernando Thudugala Mudalige	Tutor:	Friday 8am-10am
Date of submission:	24/10/2023	Which assignment?	(Leave blank if the unit has only one assignment.)

I declare that:

- The above information is complete and accurate.
- The work I am submitting is *entirely my own*, except where clearly indicated otherwise and correctly referenced.
- I have taken (and will continue to take) all reasonable steps to ensure my work is *not accessible* to any other students who may gain unfair advantage from it.
- I have *not previously submitted* this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I understand that:

- Plagiarism and collusion are dishonest, and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I plagiarise or collude, I risk failing the unit with a grade of ANN ("Result Annulled due to Academic Misconduct"), which will remain permanently on my academic record. I also risk termination from my course and other penalties.
- Even with correct referencing, my submission will only be marked according to what I have done myself, specifically for this assessment. I cannot re-use the work of others, or my own previously submitted work, in order to fulfil the assessment requirements.
- It is my responsibility to ensure that my submission is complete, correct and not corrupted.

Signature: Justyn Date of signature: 24/10/2023

(By submitting this form, you indicate that you agree with all the above text.)

Introduction

For this final assessment for the 'Database Systems' unit, I have selected to complete the assignment based on scenario B which is regarding Nobel Prize Laureates. This report covers the design, implementation, functionality and usage of my database that is based on scenario B. All reasonings for the choice of all entities, attributes, queries and any other implementations will be discussed in my report with evidence from tables, charts and my database implementations that was created using MySQL on a Linux virtual machine that is hosted via WSL Ubuntu in VsCode.

Design of Database

After reading through scenario B, I have deduced that I will be creating 3 entities for the database; Prize, Winner and Affiliation. The reason for this is because each prize, winner and affiliation can be uniquely identified and has their own independent existence where each object can hold different data that it unique to itself while also relating/having a relationship with another object such as Prize having a relationship with Winner and Winner having a relationship with Affiliation.

The following tables visualize all entities, their attributes, their relationships, cardinalities and participation in relationships. Under each table is an explanation for their design.

Entity Sets	**Keys**	**Attributes**
Prize	PrizeID	Field, Year_Awarded, Medal, Diploma, Cash_Award
Winner	WinnerID	First_Name, Last_Name, Gender, Date of Birth (DOB), Date of Death (DOD), Country
Affiliation	AffiliationID	Name, City, Country, Type

For the prize entity, I have given it attributes field which relates to the study field of the prize won, year_awarded which shows which year the nobel prize was awarded, medal which informs what the name of the prize is, diploma which states the motivation as to why the prize was awarded to someone and a cash_award to display how much money at that time was awarded to the winner. Cash_Award is not an adjusted amount and does not account for inflation hence the value is the exact amount awarded to a person at that current time.

With the winner entity, I have listed attributes that inform the first and last name of the winner, their gender (male, female, other), their date of birth, their date of death (if they have died) and their country of origin.

The affiliation entity relates to the organisation a winner might be working for or with. The attributes tell about the name of the organisation, their city and country location and the type of organisation the affiliation is. Some examples of organisation types are University, Research Institute, Non Profit Organisation, etc.

Relationship Sets	**Between which Entity Sets**	**Attributes of Relationship Sets**
Received_By	Prize & Winner	N/A
Associated_With	Winner & Affiliation	N/A

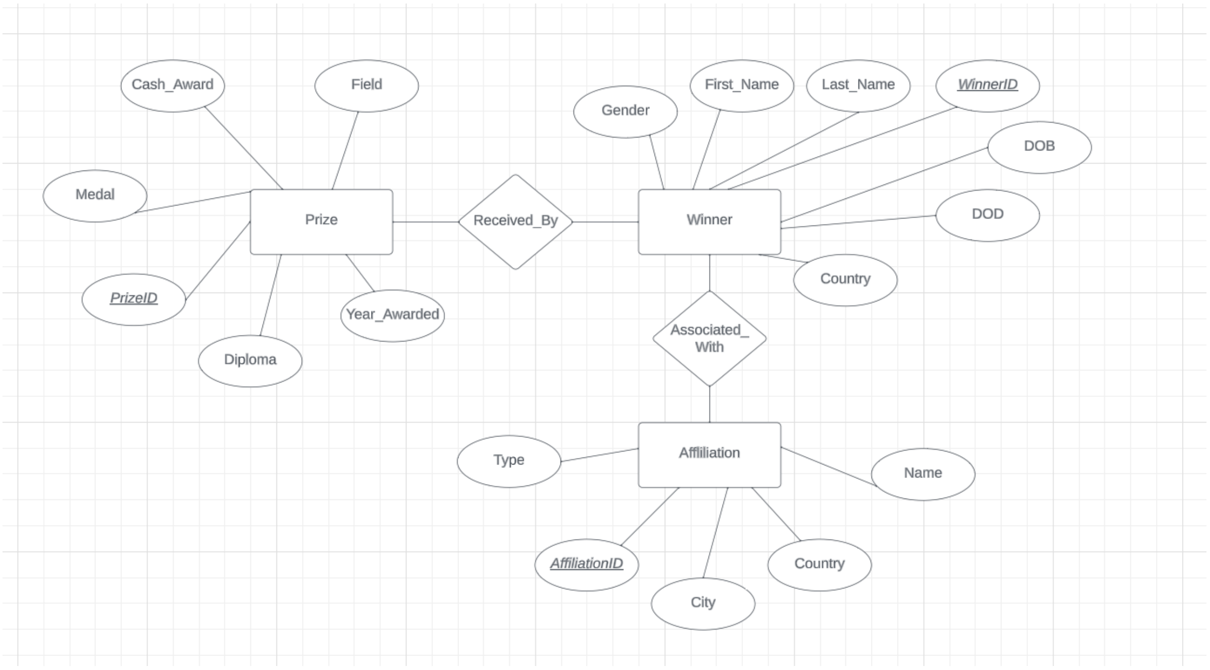
I have created 2 relationship sets for this database. Their names relate to what is happening between 2 entities such as a prize is received by a winner and a winner could be associated with an affiliation.

The reason why I explained that a winner 'could be' associated with an affiliation is because not all winners are associated with an organization.

Relationship Sets	**Cardinality Constraints**	**Participation**
Received_By	M:N	Total (Winner & Prize)
Associated_With	M:N	Partial (Winner), Total (Affiliation)

Both relationship sets have many to many cardinality constraints between their entities. Many prizes are received by many winners and many winners are associated with many affiliations. The reasoning for a partial participation for winner between winner and affiliation is because not all winners are affiliated with an organisation.

The following ER diagram drawn using Chen notation reflects the design of my database that has been previously explored and explained:



Relational Schema

Prize	***Winner***	***Affiliation***	***Received_By***	***Associated_With***
PrizeID (PK)	WinnerID (PK)	AffiliationID (PK)	PrizeID (FK)	WinnerID (FK)
Medal	First_Name	Name	WinnerID (FK)	AffiliationID (FK)
Field	Last_Name	City		
Year_Awarded	Gender	Country		
Diploma	DOB	Type		
Cash_Award	DOD			
	Country			

Because the relationship sets are many to many, the relationship sets have to form their own tables hence received_by and associated_with are tables that link prizes to winners and winners to affiliations through the use of converting the entities' primary keys into foreign keys for the relationship set tables.

Data Dictionary

Prize					
Description	Keeps data relating to the prize awarded				
Attribute	Type	Size	Primary Key	Description	Other Constraints
PrizeID	INT	-	Yes	Unique Prize ID provided by Nobel Foundation	Automatic Incrementation
Field	VARCHAR	30	No	Study field relating to the prize	NOT NULL
Year_Awarded	YEAR	-	No	The year the prize was awarded to winner	NOT NULL
Medal	VARCHAR	100	No	Title of Medal	NOT NULL
Diploma	TEXT	-	No	Motivation for diploma	NOT NULL
Cash_Award	INT	-	No	Prize money awarded	NOT NULL

Winner					
Description	Keeps data relating to winners				
Attribute	Type	Size	Primary Key	Description	Other Constraints
WinnerID	INT	-	Yes	Unique Winner ID provided by Nobel Foundation	Automatic Increment
First_Name	VARCHAR	100	No	First name of winner	NOT NULL
Last_Name	VARCHAR	30	No	Last name of winner	
DOB	DATE	-	No	Winner Date of Birth	
DOD	DATE	-	No	Winner Date of Death	
Gender	VARCHAR	6	No	Gender of Winner	
Country	VARCHAR	50	No	Nationality of Winner	NOT NULL

Affiliation					
Description	Keeps data relating to Winner’s Affiliation				
Attribute	Type	Size	Primary Key	Description	Other Constraints

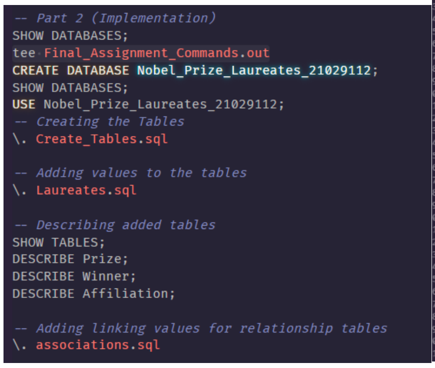
AffiliationID	INT	-	Yes	Affiliaiton ID provided by Nobel Foundation	Automatic Increment
Name	VARCHAR	100	No	Name of Organisation	NOT NULL
City	VARCHAR	50	No	City of Basis of Organisation	NOT NULL
Country	VARCHAR	50	No	County of Origin of Organisation	NOT NULL
Type	VARCHAR	50	No	Type of Organisation	NOT NULL

For all entity tables (non-relationship tables), I have decided to automatically increment the primary keys. This is because a full database with all Nobel Prize winners would be very big and the number would be hard to keep track off hence when adding winners, prizes and affiliations, it would be much easier to have their ID number automatically increase. Majority of attributes cannot be null however if an attribute is allowed to be null, this is due to the fact that that attribute might not relate to an entity at that current time such as a winner having no death date due to still being alive or a when the winner is an organisation, it does not have a last name, a gender, a DOB and a DOD. Gender is limited to 6 because I have only allowed the genders to be listed as male, female or other for simplicity and the longest string that should be there would be female (6 characters). The diploma attribute is a text because some motivations are very long and hence it is easier to store as a text and allows an unlimited length.

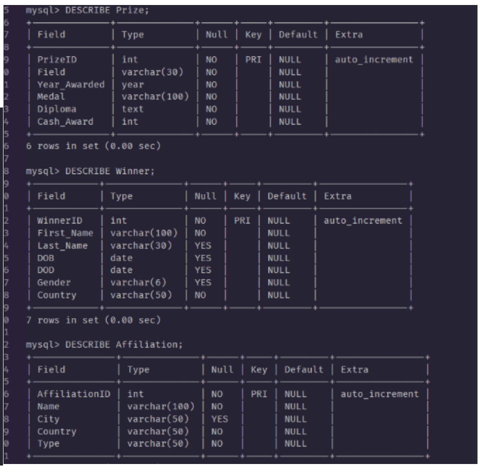
Implementation of the Database

This section of the report will describe and provide evidence of database implementation. As mentioned before, I have worked in a Linux environment through the use of a Linux virtual machine created using WSL Ubuntu on VsCode.

All commands that I have entered have been logged into a file called Final_Assignment_Commands.out by the use of the command tee Final_Assignment_Commands.out which is also in the Final_Assignment_Commands.sql file that creates and populates the entire database. As per what has been said in my user guide, my database can be fully implemented by running the SQL file Final_Assignment_Commands.sql. The following images are the code in the SQL file that creates and populates the database and the description output according to the .out file.



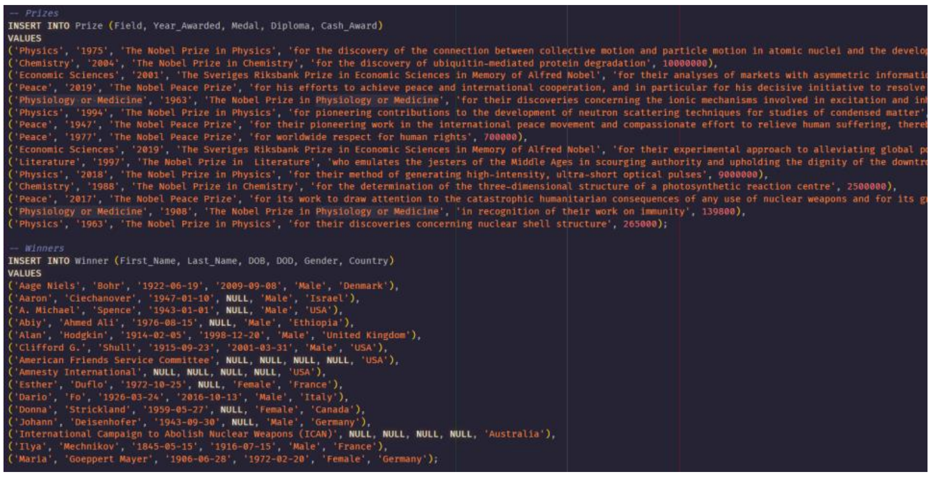
```
-- Part 2 (Implementation)
SHOW DATABASES;
tee Final_Assignment_Commands.out
CREATE DATABASE Nobel_Prize_Laureates_21029112;
SHOW DATABASES;
USE Nobel_Prize_Laureates_21029112;
-- Creating the Tables
\. Create_Tables.sql

-- Adding values to the tables
\. Laureates.sql

-- Describing added tables
SHOW TABLES;
DESCRIBE Prize;
DESCRIBE Winner;
DESCRIBE Affiliation;

-- Adding linking values for relationship tables
\. associations.sql
```



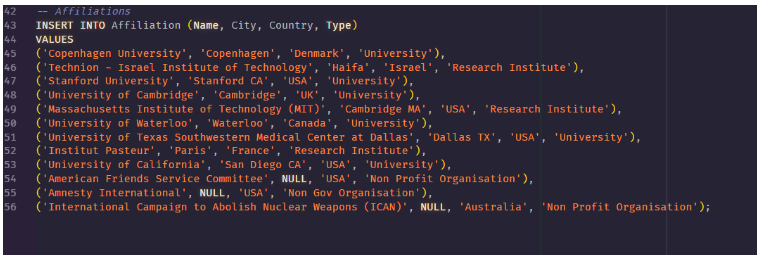
```
mysql> DESCRIBE Prize;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| PrizeID     | int          | NO   | PRI | NULL    | auto_increment |
| Field       | varchar(30)  | NO   |     | NULL    |                |
| Year_Awarded| year         | NO   |     | NULL    |                |
| Medal       | varchar(100) | NO   |     | NULL    |                |
| Diploma     | text         | NO   |     | NULL    |                |
| Cash_Award  | int          | NO   |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)

mysql> DESCRIBE Winner;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| WinnerID   | int          | NO   | PRI | NULL    | auto_increment |
| First_Name | varchar(100) | NO   |     | NULL    |                |
| Last_Name  | varchar(30)  | YES  |     | NULL    |                |
| DOB        | date         | YES  |     | NULL    |                |
| DOD        | date         | YES  |     | NULL    |                |
| Gender     | varchar(6)   | YES  |     | NULL    |                |
| Country    | varchar(50)  | NO   |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)

mysql> DESCRIBE Affiliation;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| AffiliationID | int          | NO   | PRI | NULL    | auto_increment |
| Name          | varchar(100) | NO   |     | NULL    |                |
| City          | varchar(50)  | YES  |     | NULL    |                |
| Country       | varchar(50)  | NO   |     | NULL    |                |
| Type          | varchar(50)  | NO   |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
```

The code runs 3 other separate SQL files that create and populates the tables. Create_Tables.sql creates all the tables, Laureates.sql populates the prize, winner and affiliation tables while associations.sql populates the relationship tables received_by and associated_with.

My data sources are the 2 websites that are provided by the Unit Coordinator on Blackboard that relates to all Nobel Prize Laureates and each provide an excel file that contain all Nobel Prize Laureates up to 2022. I have used a sample data size of 15 winners and prizes for my database. My sample data has given 12 different affiliations from the 15 winners as some winners have the same affiliations while others do not have any affiliations. My sample data also takes winners from a broad range of times ranging from 1963 to 2019. I have inserted the data into my database by the use of simple INSERT INTO commands that are in Laureates.sql and associates.sql. The following images display the contents in Laureates.sql and associations.sql.

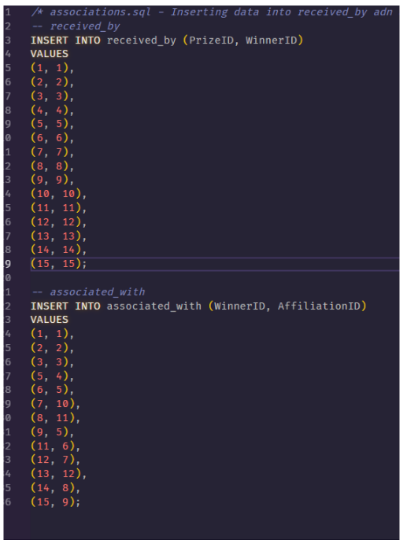


```sql
-- Prizes
INSERT INTO Prize (Field, Year_Awarded, Medal, Diploma, Cash_Award)
VALUES
('Physics', '1975', 'The Nobel Prize in Physics', 'for the discovery of the connection between collective motion and particle motion in atomic nuclei and the develop
('Chemistry', '2004', 'The Nobel Prize in Chemistry', 'for the discovery of ubiquitin-mediated protein degradation', 10000000),
('Economic Sciences', '2001', 'The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel', 'for their analyses of markets with asymmetric informati
('Peace', '2019', 'The Nobel Peace Prize', 'for his efforts to achieve peace and international cooperation, and in particular for his decisive initiative to resolve
('Physiology or Medicine', '1963', 'The Nobel Prize in Physiology or Medicine', 'for their discoveries concerning the ionic mechanisms involved in excitation and in
('Physics', '1994', 'The Nobel Prize in Physics', 'for pioneering contributions to the development of neutron scattering techniques for studies of condensed matter',
('Peace', '1947', 'The Nobel Peace Prize', 'for their pioneering work in the international peace movement and compassionate effort to relieve human suffering, there
('Peace', '1977', 'The Nobel Peace Prize', 'for worldwide respect for human rights', 700000),
('Economic Sciences', '2019', 'The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel', 'for their experimental approach to alleviating global po
('Literature', '1997', 'The Nobel Prize in  Literature', 'who emulates the jesters of the Middle Ages in scourging authority and upholding the dignity of the downtro
('Physics', '2018', 'The Nobel Prize in Physics', 'for their method of generating high-intensity, ultra-short optical pulses', 9000000),
('Chemistry', '1988', 'The Nobel Prize in Chemistry', 'for the determination of the three-dimensional structure of a photosynthetic reaction centre', 2500000),
('Peace', '2017', 'The Nobel Peace Prize', 'for its work to draw attention to the catastrophic humanitarian consequences of any use of nuclear weapons and for its g
('Physiology or Medicine', '1908', 'The Nobel Prize in Physiology or Medicine', 'in recognition of their work on immunity', 139800),
('Physics', '1963', 'The Nobel Prize in Physics', 'for their discoveries concerning nuclear shell structure', 265000);

-- Winners
INSERT INTO Winner (First_Name, Last_Name, DOB, DOD, Gender, Country)
VALUES
('Aage Niels', 'Bohr', '1922-06-19', '2009-09-08', 'Male', 'Denmark'),
('Aaron', 'Ciechanover', '1947-01-10', NULL, 'Male', 'Israel'),
('A. Michael', 'Spence', '1943-01-01', NULL, 'Male', 'USA'),
('Abiy', 'Ahmed Ali', '1976-08-15', NULL, 'Male', 'Ethiopia'),
('Alan', 'Hodgkin', '1914-02-05', '1998-12-20', 'Male', 'United Kingdom'),
('Clifford G.', 'Shull', '1915-09-23', '2001-03-31', 'Male', 'USA'),
('American Friends Service Committee', NULL, NULL, NULL, NULL, 'USA'),
('Amnesty International', NULL, NULL, NULL, NULL, 'USA'),
('Esther', 'Duflo', '1972-10-25', NULL, 'Female', 'France'),
('Dario', 'Fo', '1926-03-24', '2016-10-13', 'Male', 'Italy'),
('Donna', 'Strickland', '1959-05-27', NULL, 'Female', 'Canada'),
('Johann', 'Deisenhofer', '1943-09-30', NULL, 'Male', 'Germany'),
('International Campaign to Abolish Nuclear Weapons (ICAN)', NULL, NULL, NULL, NULL, 'Australia'),
('Ilya', 'Mechnikov', '1845-05-15', '1916-07-15', 'Male', 'France'),
('Maria', 'Goeppert Mayer', '1906-06-28', '1972-02-20', 'Female', 'Germany');
```



```sql
-- Affiliations
INSERT INTO Affiliation (Name, City, Country, Type)
VALUES
('Copenhagen University', 'Copenhagen', 'Denmark', 'University'),
('Technion - Israel Institute of Technology', 'Haifa', 'Israel', 'Research Institute'),
('Stanford University', 'Stanford CA', 'USA', 'University'),
('University of Cambridge', 'Cambridge', 'UK', 'University'),
('Massachusetts Institute of Technology (MIT)', 'Cambridge MA', 'USA', 'Research Institute'),
('University of Waterloo', 'Waterloo', 'Canada', 'University'),
('University of Texas Southwestern Medical Center at Dallas', 'Dallas TX', 'USA', 'University'),
('Institut Pasteur', 'Paris', 'France', 'Research Institute'),
('University of California', 'San Diego CA', 'USA', 'University'),
('American Friends Service Committee', NULL, 'USA', 'Non Profit Organisation'),
('Amnesty International', NULL, 'USA', 'Non Gov Organisation'),
('International Campaign to Abolish Nuclear Weapons (ICAN)', NULL, 'Australia', 'Non Profit Organisation');
```

Laureates.sql Contents



```sql
/* associations.sql - Inserting data into received_by adn
-- received_by
INSERT INTO received_by (PrizeID, WinnerID)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10),
(11, 11),
(12, 12),
(13, 13),
(14, 14),
(15, 15);

-- associated_with
INSERT INTO associated_with (WinnerID, AffiliationID)
VALUES
(1, 1),
(2, 2),
(3, 3),
(5, 4),
(6, 5),
(7, 10),
(8, 11),
(9, 5),
(11, 6),
(12, 7),
(13, 12),
(14, 8),
(15, 9);
```

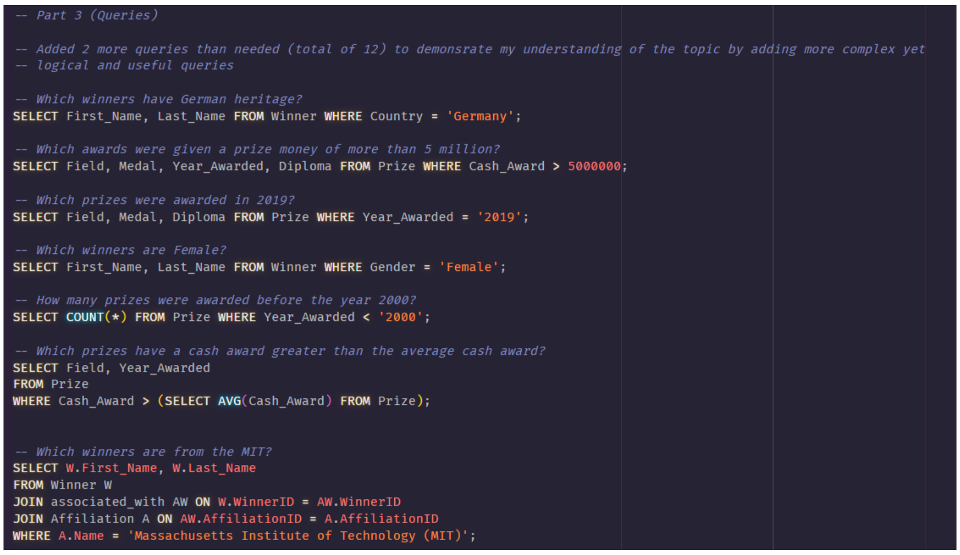
associations.sql Contents

Use of Database

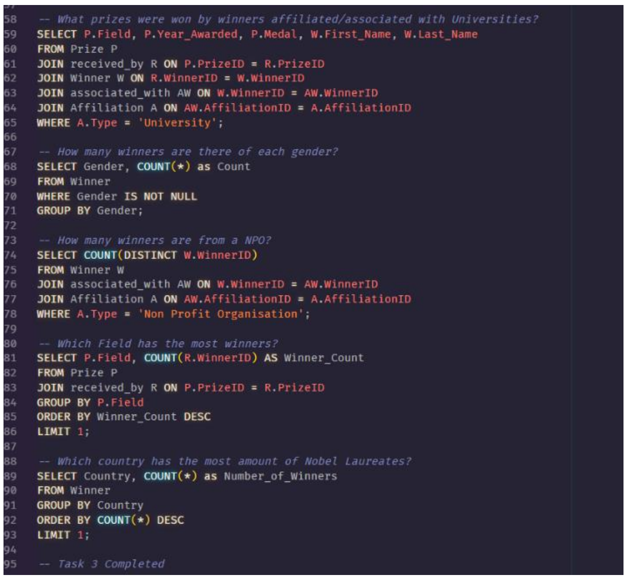
All queries and advanced feature implementations are also put in the Final_Assignment_Commands.sql file. After the implementation and population of the database using the sample data, the file will automatically do all queries and all advanced features (3 Stored Procedures and 2 Triggers). I have put 12 queries which is 2 more than needed as the queries are logical questions and help demonstrate my understanding of the topic. The images soon to be shown below show the implementation of the queries followed by the sample output shown in the .out file. All queries are related to basic things that a user might want to know about the Nobel Prize Winners such as where they are from, which country has the most prize winners, how many winners are female, etc.

I have chosen implement 3 different stored procedures and 2 different triggers for the advanced features. The first stored procedure adds a new prize, the 2nd adds a new winner and checks that the date of birth is logical and the last stored procedure finds all the winners of a certain field. Trigger 1 ensures that a prize is not already added while trigger 2 checks if an affiliation is already added. The following implementations, their usages and sample outcomes are shown below.

Query Implementations:

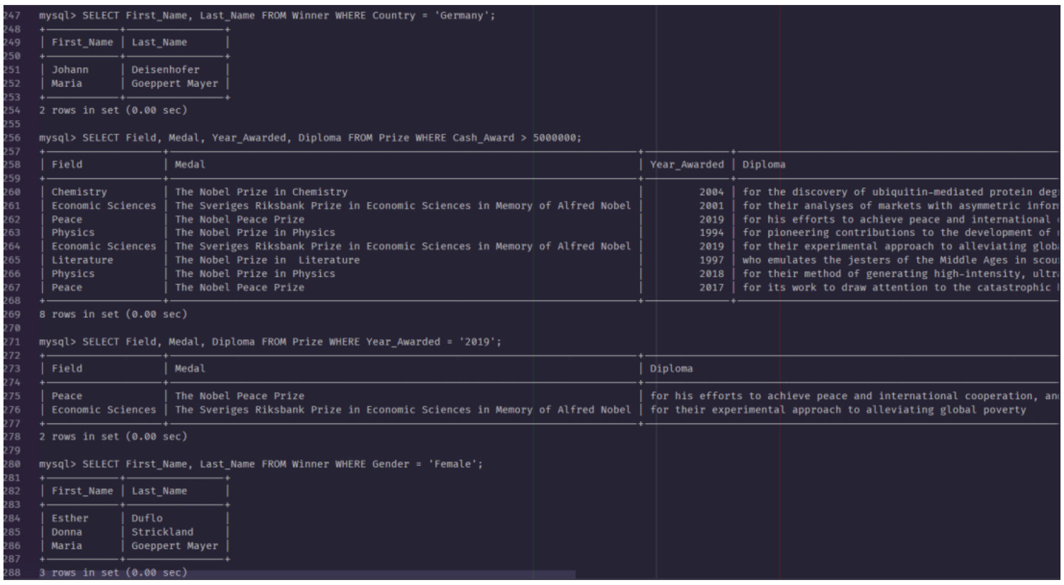


```sql
-- Part 3 (Queries)

-- Added 2 more queries than needed (total of 12) to demonsrate my understanding of the topic by adding more complex yet
-- logical and useful queries

-- Which winners have German heritage?
SELECT First_Name, Last_Name FROM Winner WHERE Country = 'Germany';

-- Which awards were given a prize money of more than 5 million?
SELECT Field, Medal, Year_Awarded, Diploma FROM Prize WHERE Cash_Award > 5000000;

-- Which prizes were awarded in 2019?
SELECT Field, Medal, Diploma FROM Prize WHERE Year_Awarded = '2019';

-- Which winners are Female?
SELECT First_Name, Last_Name FROM Winner WHERE Gender = 'Female';

-- How many prizes were awarded before the year 2000?
SELECT COUNT(*) FROM Prize WHERE Year_Awarded < '2000';

-- Which prizes have a cash award greater than the average cash award?
SELECT Field, Year_Awarded
FROM Prize
WHERE Cash_Award > (SELECT AVG(Cash_Award) FROM Prize);


-- Which winners are from the MIT?
SELECT W.First_Name, W.Last_Name
FROM Winner W
JOIN associated_with AW ON W.WinnerID = AW.WinnerID
JOIN Affiliation A ON AW.AffiliationID = A.AffiliationID
WHERE A.Name = 'Massachusetts Institute of Technology (MIT)';
```

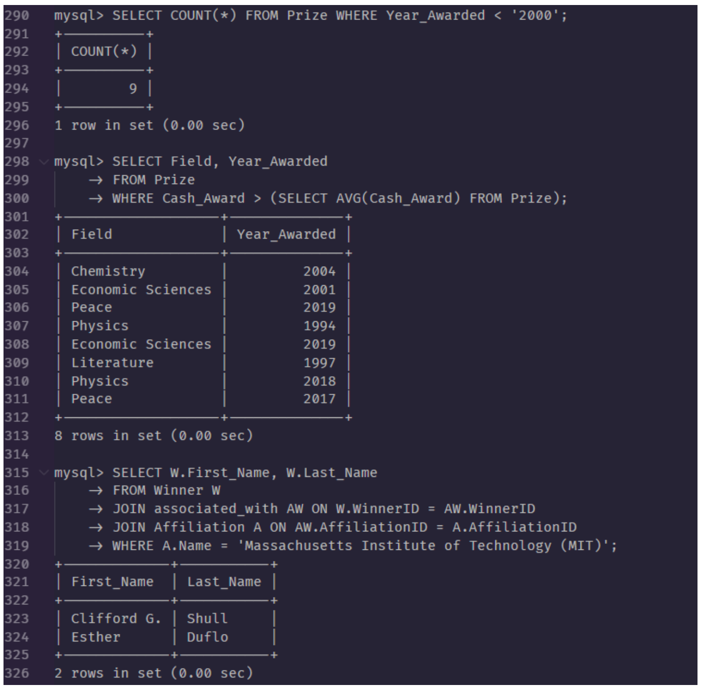


```sql
-- What prizes were won by winners affiliated/associated with Universities?
SELECT P.Field, P.Year_Awarded, P.Medal, W.First_Name, W.Last_Name
FROM Prize P
JOIN received_by R ON P.PrizeID = R.PrizeID
JOIN Winner W ON R.WinnerID = W.WinnerID
JOIN associated_with AW ON W.WinnerID = AW.WinnerID
JOIN Affiliation A ON AW.AffiliationID = A.AffiliationID
WHERE A.Type = 'University';

-- How many winners are there of each gender?
SELECT Gender, COUNT(*) as Count
FROM Winner
WHERE Gender IS NOT NULL
GROUP BY Gender;

-- How many winners are from a NPO?
SELECT COUNT(DISTINCT W.WinnerID)
FROM Winner W
JOIN associated_with AW ON W.WinnerID = AW.WinnerID
JOIN Affiliation A ON AW.AffiliationID = A.AffiliationID
WHERE A.Type = 'Non Profit Organisation';

-- Which Field has the most winners?
SELECT P.Field, COUNT(R.WinnerID) AS Winner_Count
FROM Prize P
JOIN received_by R ON P.PrizeID = R.PrizeID
GROUP BY P.Field
ORDER BY Winner_Count DESC
LIMIT 1;

-- Which country has the most amount of Nobel Laureates?
SELECT Country, COUNT(*) as Number_of_Winners
FROM Winner
GROUP BY Country
ORDER BY COUNT(*) DESC
LIMIT 1;

-- Task 3 Completed
```

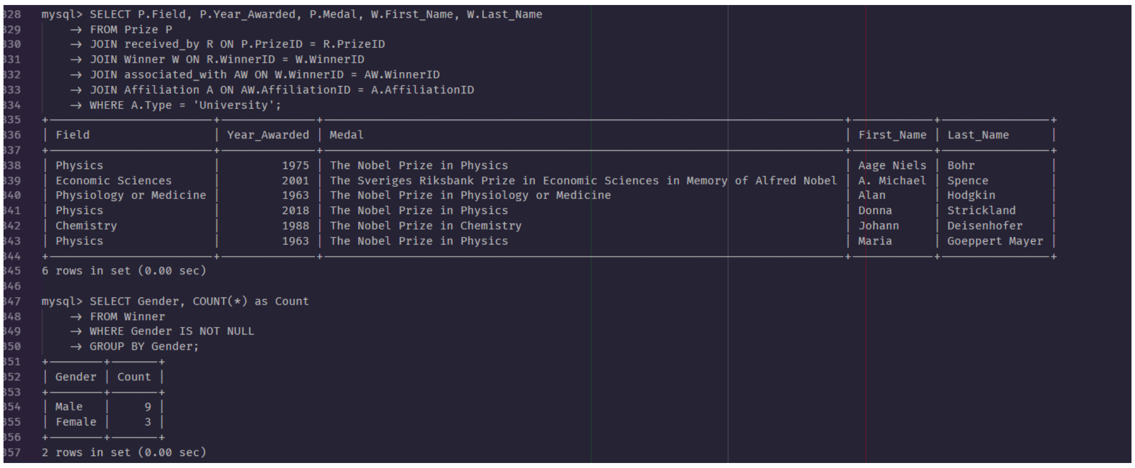
Query Sample Outputs:



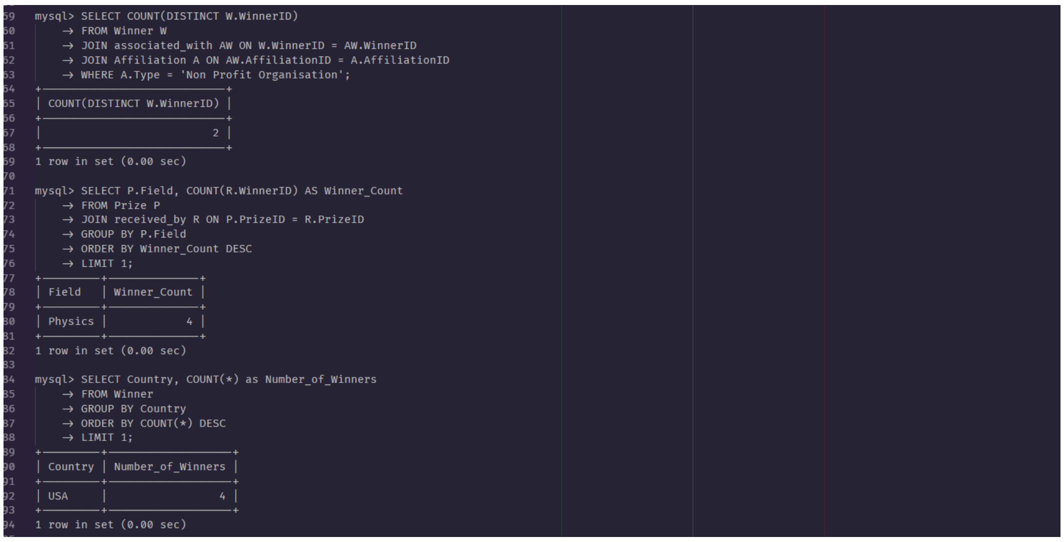
```
mysql> SELECT First_Name, Last_Name FROM Winner WHERE Country = 'Germany';
+------------+----------------+
| First_Name | Last_Name      |
+------------+----------------+
| Johann     | Deisenhofer    |
| Maria      | Goeppert Mayer |
+------------+----------------+
2 rows in set (0.00 sec)

mysql> SELECT Field, Medal, Year_Awarded, Diploma FROM Prize WHERE Cash_Award > 5000000;
+-------------------+------------------------------------------------------------------------+--------------+------------------------------------------------------
| Field             | Medal                                                                  | Year_Awarded | Diploma
+-------------------+------------------------------------------------------------------------+--------------+------------------------------------------------------
| Chemistry         | The Nobel Prize in Chemistry                                           |         2004 | for the discovery of ubiquitin-mediated protein deg
| Economic Sciences | The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel |       2001 | for their analyses of markets with asymmetric infor
| Peace             | The Nobel Peace Prize                                                  |         2019 | for his efforts to achieve peace and international
| Physics           | The Nobel Prize in Physics                                             |         1994 | for pioneering contributions to the development of
| Economic Sciences | The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel |       2019 | for their experimental approach to alleviating glob
| Literature        | The Nobel Prize in  Literature                                         |         1997 | who emulates the jesters of the Middle Ages in scou
| Physics           | The Nobel Prize in Physics                                             |         2018 | for their method of generating high-intensity, ultr
| Peace             | The Nobel Peace Prize                                                  |         2017 | for its work to draw attention to the catastrophic
+-------------------+------------------------------------------------------------------------+--------------+------------------------------------------------------
8 rows in set (0.00 sec)

mysql> SELECT Field, Medal, Diploma FROM Prize WHERE Year_Awarded = '2019';
+-------------------+------------------------------------------------------------------------+------------------------------------------------------------------
| Field             | Medal                                                                  | Diploma
+-------------------+------------------------------------------------------------------------+------------------------------------------------------------------
| Peace             | The Nobel Peace Prize                                                  | for his efforts to achieve peace and international cooperation, an
| Economic Sciences | The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel | for their experimental approach to alleviating global poverty
+-------------------+------------------------------------------------------------------------+------------------------------------------------------------------
2 rows in set (0.00 sec)

mysql> SELECT First_Name, Last_Name FROM Winner WHERE Gender = 'Female';
+------------+----------------+
| First_Name | Last_Name      |
+------------+----------------+
| Esther     | Duflo          |
| Donna      | Strickland     |
| Maria      | Goeppert Mayer |
+------------+----------------+
3 rows in set (0.00 sec)
```



```
mysql> SELECT COUNT(*) FROM Prize WHERE Year_Awarded < '2000';
+----------+
| COUNT(*) |
+----------+
|        9 |
+----------+
1 row in set (0.00 sec)

mysql> SELECT Field, Year_Awarded
    -> FROM Prize
    -> WHERE Cash_Award > (SELECT AVG(Cash_Award) FROM Prize);
+-------------------+--------------+
| Field             | Year_Awarded |
+-------------------+--------------+
| Chemistry         |         2004 |
| Economic Sciences |         2001 |
| Peace             |         2019 |
| Physics           |         1994 |
| Economic Sciences |         2019 |
| Literature        |         1997 |
| Physics           |         2018 |
| Peace             |         2017 |
+-------------------+--------------+
8 rows in set (0.00 sec)

mysql> SELECT W.First_Name, W.Last_Name
    -> FROM Winner W
    -> JOIN associated_with AW ON W.WinnerID = AW.WinnerID
    -> JOIN Affiliation A ON AW.AffiliationID = A.AffiliationID
    -> WHERE A.Name = 'Massachusetts Institute of Technology (MIT)';
+-------------+-----------+
| First_Name  | Last_Name |
+-------------+-----------+
| Clifford G. | Shull     |
| Esther      | Duflo     |
+-------------+-----------+
2 rows in set (0.00 sec)
```

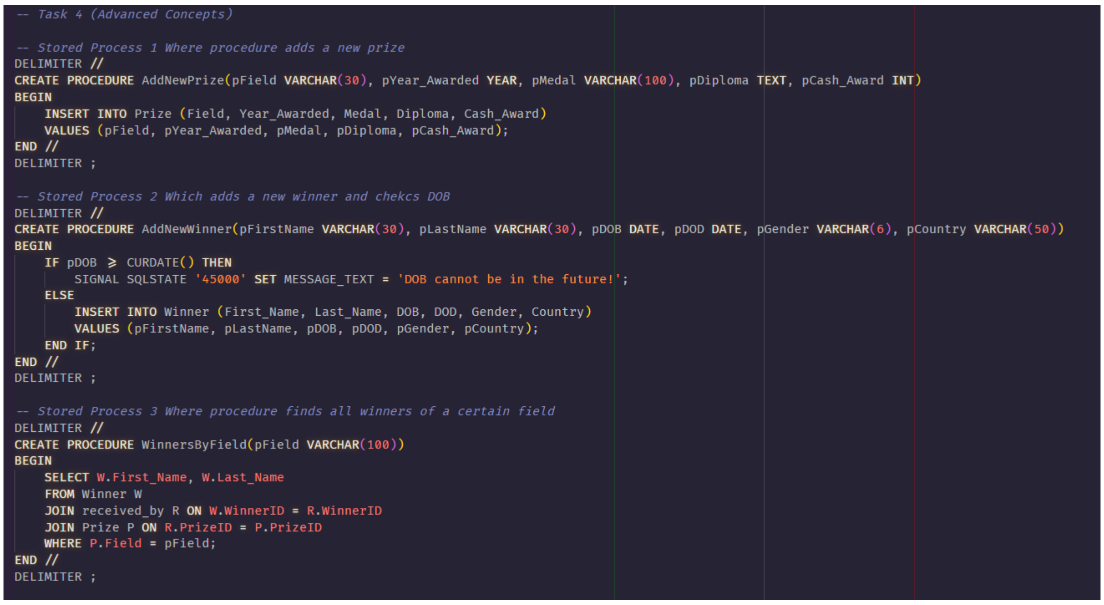


```
mysql> SELECT P.Field, P.Year_Awarded, P.Medal, W.First_Name, W.Last_Name
    -> FROM Prize P
    -> JOIN received_by R ON P.PrizeID = R.PrizeID
    -> JOIN Winner W ON R.WinnerID = W.WinnerID
    -> JOIN associated_with AW ON W.WinnerID = AW.WinnerID
    -> JOIN Affiliation A ON AW.AffiliationID = A.AffiliationID
    -> WHERE A.Type = 'University';
+-------------------------+--------------+----------------------------------------------------------------------------+-------------+----------------+
| Field                   | Year_Awarded | Medal                                                                      | First_Name  | Last_Name      |
+-------------------------+--------------+----------------------------------------------------------------------------+-------------+----------------+
| Physics                 |         1975 | The Nobel Prize in Physics                                                 | Aage Niels  | Bohr           |
| Economic Sciences       |         2001 | The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel | A. Michael  | Spence         |
| Physiology or Medicine  |         1963 | The Nobel Prize in Physiology or Medicine                                  | Alan        | Hodgkin        |
| Physics                 |         2018 | The Nobel Prize in Physics                                                 | Donna       | Strickland     |
| Chemistry               |         1988 | The Nobel Prize in Chemistry                                               | Johann      | Deisenhofer    |
| Physics                 |         1963 | The Nobel Prize in Physics                                                 | Maria       | Goeppert Mayer |
+-------------------------+--------------+----------------------------------------------------------------------------+-------------+----------------+
6 rows in set (0.00 sec)

mysql> SELECT Gender, COUNT(*) as Count
    -> FROM Winner
    -> WHERE Gender IS NOT NULL
    -> GROUP BY Gender;
+--------+-------+
| Gender | Count |
+--------+-------+
| Male   |     9 |
| Female |     3 |
+--------+-------+
2 rows in set (0.00 sec)
```

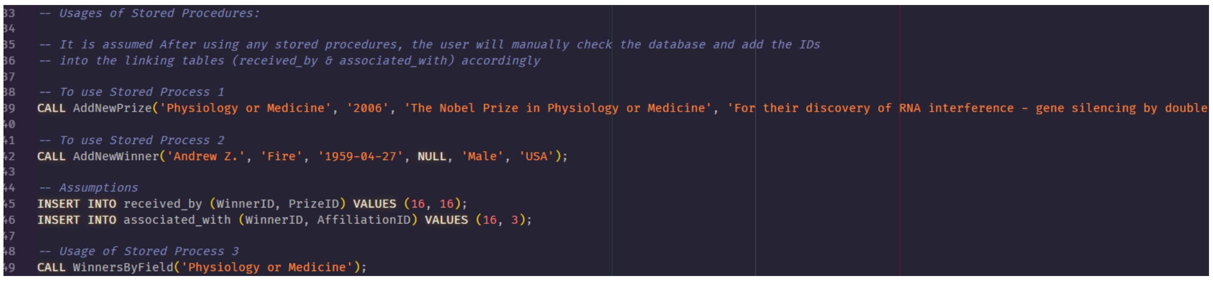


```
mysql> SELECT COUNT(DISTINCT W.WinnerID)
    -> FROM Winner W
    -> JOIN associated_with AW ON W.WinnerID = AW.WinnerID
    -> JOIN Affiliation A ON AW.AffiliationID = A.AffiliationID
    -> WHERE A.Type = 'Non Profit Organisation';
+----------------------------+
| COUNT(DISTINCT W.WinnerID) |
+----------------------------+
|                          2 |
+----------------------------+
1 row in set (0.00 sec)

mysql> SELECT P.Field, COUNT(R.WinnerID) AS Winner_Count
    -> FROM Prize P
    -> JOIN received_by R ON P.PrizeID = R.PrizeID
    -> GROUP BY P.Field
    -> ORDER BY Winner_Count DESC
    -> LIMIT 1;
+---------+--------------+
| Field   | Winner_Count |
+---------+--------------+
| Physics |            4 |
+---------+--------------+
1 row in set (0.00 sec)

mysql> SELECT Country, COUNT(*) as Number_of_Winners
    -> FROM Winner
    -> GROUP BY Country
    -> ORDER BY COUNT(*) DESC
    -> LIMIT 1;
+---------+-------------------+
| Country | Number_of_Winners |
+---------+-------------------+
| USA     |                 4 |
+---------+-------------------+
1 row in set (0.00 sec)
```

Stored Procedures Implementation and Usages:

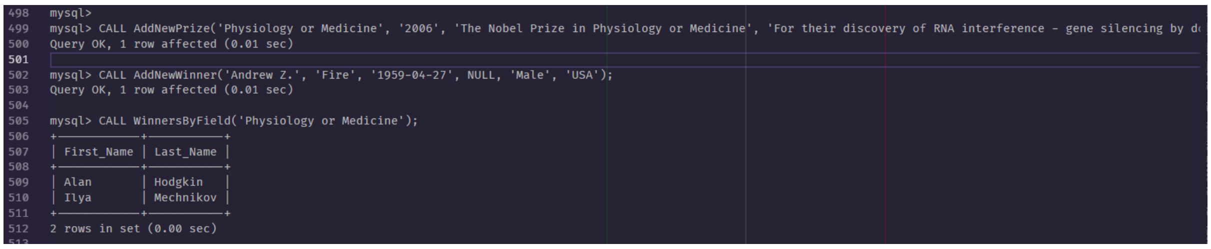


```sql
-- Task 4 (Advanced Concepts)

-- Stored Process 1 Where procedure adds a new prize
DELIMITER //
CREATE PROCEDURE AddNewPrize(pField VARCHAR(30), pYear_Awarded YEAR, pMedal VARCHAR(100), pDiploma TEXT, pCash_Award INT)
BEGIN
    INSERT INTO Prize (Field, Year_Awarded, Medal, Diploma, Cash_Award)
    VALUES (pField, pYear_Awarded, pMedal, pDiploma, pCash_Award);
END //
DELIMITER ;

-- Stored Process 2 Which adds a new winner and chekcs DOB
DELIMITER //
CREATE PROCEDURE AddNewWinner(pFirstName VARCHAR(30), pLastName VARCHAR(30), pDOB DATE, pDOD DATE, pGender VARCHAR(6), pCountry VARCHAR(50))
BEGIN
    IF pDOB >= CURDATE() THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'DOB cannot be in the future!';
    ELSE
        INSERT INTO Winner (First_Name, Last_Name, DOB, DOD, Gender, Country)
        VALUES (pFirstName, pLastName, pDOB, pDOD, pGender, pCountry);
    END IF;
END //
DELIMITER ;

-- Stored Process 3 Where procedure finds all winners of a certain field
DELIMITER //
CREATE PROCEDURE WinnersByField(pField VARCHAR(100))
BEGIN
    SELECT W.First_Name, W.Last_Name
    FROM Winner W
    JOIN received_by R ON W.WinnerID = R.WinnerID
    JOIN Prize P ON R.PrizeID = P.PrizeID
    WHERE P.Field = pField;
END //
DELIMITER ;
```



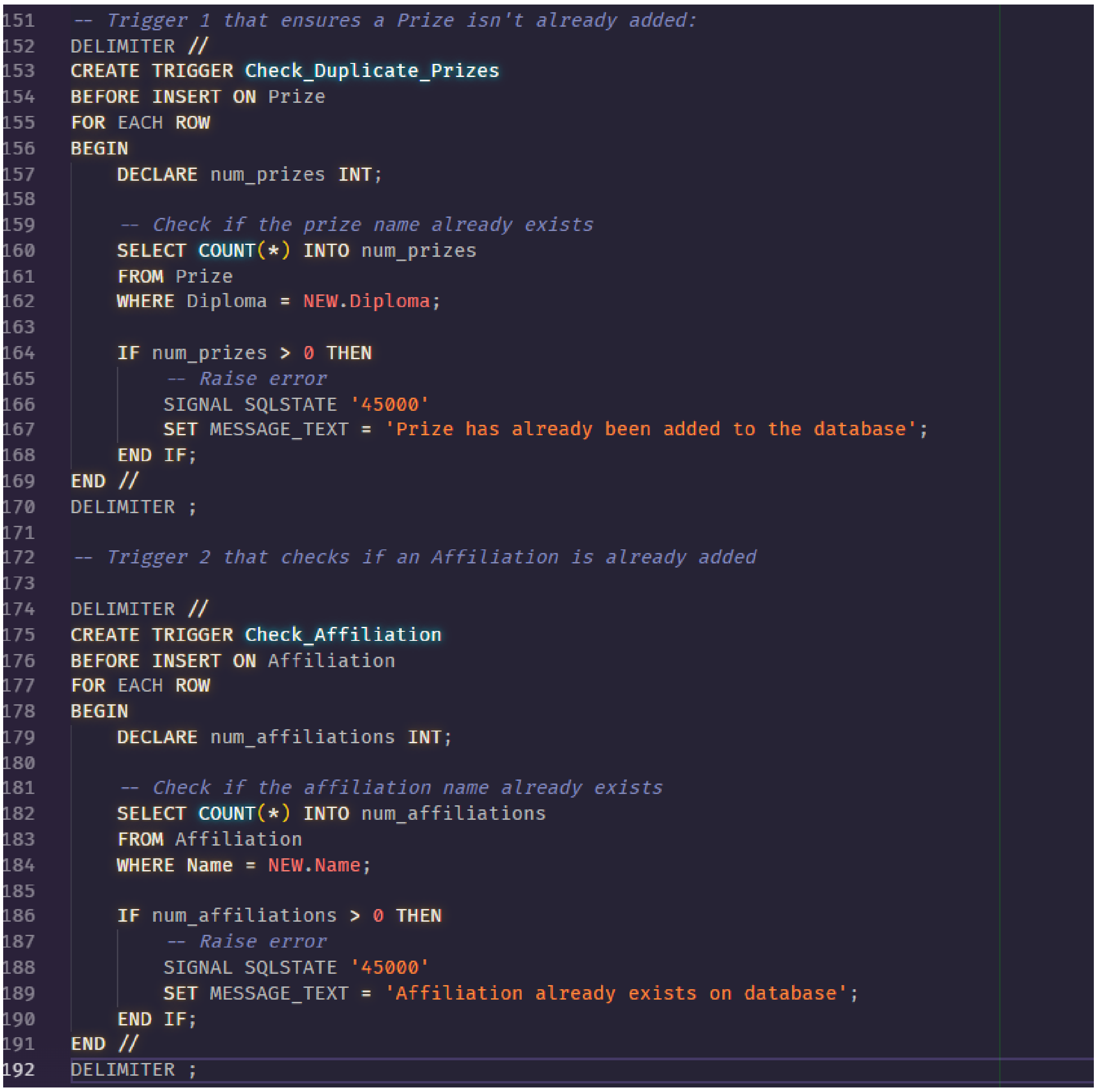
```sql
-- Usages of Stored Procedures:

-- It is assumed After using any stored procedures, the user will manually check the database and add the IDs
-- into the linking tables (received_by & associated_with) accordingly

-- To use Stored Process 1
CALL AddNewPrize('Physiology or Medicine', '2006', 'The Nobel Prize in Physiology or Medicine', 'For their discovery of RNA interference - gene silencing by double

-- To use Stored Process 2
CALL AddNewWinner('Andrew Z.', 'Fire', '1959-04-27', NULL, 'Male', 'USA');

-- Assumptions
INSERT INTO received_by (WinnerID, PrizeID) VALUES (16, 16);
INSERT INTO associated_with (WinnerID, AffiliationID) VALUES (16, 3);

-- Usage of Stored Process 3
CALL WinnersByField('Physiology or Medicine');
```

Stored Procedures Sample Output:



```
mysql>
mysql> CALL AddNewPrize('Physiology or Medicine', '2006', 'The Nobel Prize in Physiology or Medicine', 'For their discovery of RNA interference - gene silencing by d
Query OK, 1 row affected (0.01 sec)

mysql> CALL AddNewWinner('Andrew Z.', 'Fire', '1959-04-27', NULL, 'Male', 'USA');
Query OK, 1 row affected (0.01 sec)

mysql> CALL WinnersByField('Physiology or Medicine');
+------------+-----------+
| First_Name | Last_Name |
+------------+-----------+
| Alan       | Hodgkin   |
| Ilya       | Mechnikov |
+------------+-----------+
2 rows in set (0.00 sec)
```

Trigger Implementations

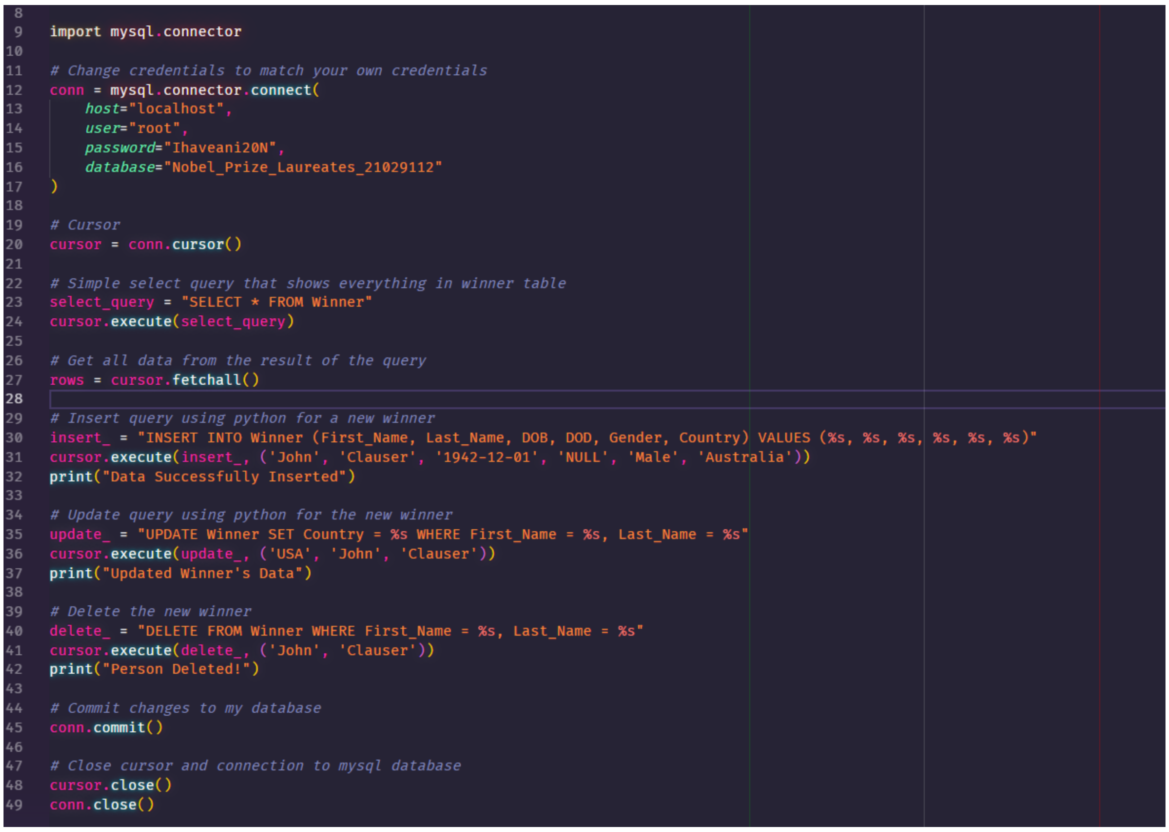


```
-- Trigger 1 that ensures a Prize isn't already added:
DELIMITER //
CREATE TRIGGER Check_Duplicate_Prizes
BEFORE INSERT ON Prize
FOR EACH ROW
BEGIN
    DECLARE num_prizes INT;

    -- Check if the prize name already exists
    SELECT COUNT(*) INTO num_prizes
    FROM Prize
    WHERE Diploma = NEW.Diploma;

    IF num_prizes > 0 THEN
        -- Raise error
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Prize has already been added to the database';
    END IF;
END //
DELIMITER ;

-- Trigger 2 that checks if an Affiliation is already added

DELIMITER //
CREATE TRIGGER Check_Affiliation
BEFORE INSERT ON Affiliation
FOR EACH ROW
BEGIN
    DECLARE num_affiliations INT;

    -- Check if the affiliation name already exists
    SELECT COUNT(*) INTO num_affiliations
    FROM Affiliation
    WHERE Name = NEW.Name;

    IF num_affiliations > 0 THEN
        -- Raise error
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Affiliation already exists on database';
    END IF;
END //
DELIMITER ;
```

Throughout the implementation of the advanced features, I have made 1 assumption that is that I have assumed that after using any stored procedures, the database user will first check their primary key of the new winner and prize that were added using a SELECT statement then manually add the IDs of the new winner, prize and affiliations into the linking tables (received_by & associated_with) accordingly.

Database Connectivity

I have used the python language to attempt to connect to my database however I have encountered issues that appear to be related to my virtual machine's MySQL setup. I have attempted to debug these issues but have not been successfully in doing so, hence this has resulted in my inability to connect to my database using the python language. However, I have coded what I believe would work if my virtual machine's issues were resolved. The code can be found in the file PyConnect.py. I have used the mysql.connector library while attempting to implement database connectivity. The contents of the file start with a simple importing of the library followed by attempting to establish connection with the correct credentials followed by basic select, insert, update and delete queries. The sample data used in these queries are also sourced from the 2 databases that were provided by the Unit Coordinator on blackboard. It starts with me attempting to display all winners then trying to insert a new winner and their country of origin is incorrect. Followed by the update of the correct country of origin and finally the deletion of this new added winner. I have then committed changes in the database and closed the connection. The following snippet demonstrates what I have just discussed.



```python

import mysql.connector

# Change credentials to match your own credentials
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Ihaveani20N",
    database="Nobel_Prize_Laureates_21029112"
)

# Cursor
cursor = conn.cursor()

# Simple select query that shows everything in winner table
select_query = "SELECT * FROM Winner"
cursor.execute(select_query)

# Get all data from the result of the query
rows = cursor.fetchall()

# Insert query using python for a new winner
insert_ = "INSERT INTO Winner (First_Name, Last_Name, DOB, DOD, Gender, Country) VALUES (%s, %s, %s, %s, %s, %s)"
cursor.execute(insert_, ('John', 'Clauser', '1942-12-01', 'NULL', 'Male', 'Australia'))
print("Data Successfully Inserted")

# Update query using python for the new winner
update_ = "UPDATE Winner SET Country = %s WHERE First_Name = %s, Last_Name = %s"
cursor.execute(update_, ('USA', 'John', 'Clauser'))
print("Updated Winner's Data")

# Delete the new winner
delete_ = "DELETE FROM Winner WHERE First_Name = %s, Last_Name = %s"
cursor.execute(delete_, ('John', 'Clauser'))
print("Person Deleted!")

# Commit changes to my database
conn.commit()

# Close cursor and connection to mysql database
cursor.close()
conn.close()
```

Final Discussion

In my opinion, I have taken a good attempt at this final assignment. I have attempted all sections to the best of my knowledge and ability. My sample data size also appears to be decent because it does provide some variety in such that winners are both individuals that have and don't have affiliations and whole organisations themselves. This in turn provides my query sample outcomes with some variety and this can be seen in my queries such as when I have searched for the gender, country and organisation diversity. The only challenge I have faced is during the failed implementation of database connectivity. I have also realised that my database stored procedures are a bit inefficient as after using the currently implemented procedures, the user has to manually check their primary key of the new winner and prize that were added using a SELECT statement then manually add the IDs of the new winner, prize and affiliations into the linking tables (received_by & associated_with) accordingly. To improve on this design, I could have implemented another stored procedure that will automatically add the required data to the relationship tables received_by and associated_with.