THE UNIVERSITY OF SYDNEY

# Assignment Coversheet – GROUP ASSIGNMENT

Please fill in your details below. Use one form for each group assignment.

## Assignment Details:

| | | | | | |
|---|---|---|---|---|---|
| Assignment Title | | **Group Assignment** | | | |
| Assignment number | | **1** | | | |
| Unit of Study Tutor | | | | | |
| Group or Tutorial ID | | | | | |
| Due Date | **30/04/2021** | Submission Date | **30/04/2021** | Word Count | **906** |

## Membership

| Family Name | Given Name (s) | | SID | Signature |
|---|---|---|---|---|
| **Srivastava** | **Shubham** | | 500299300 | Shuubham Srivastava |
| **Bhandary** | **Rishik** | | 460298672 | Rishik |
| **Mukherjee** | **Ujjaini** | | 500219755 | Ujjaini Mukherjee |
| | | | | |
| | | | | |
| | | | | |

## Declaration:

1. I understand that all forms of plagiarism and unauthorised collusion are regarded as academic dishonesty by the university, resulting in penalties including failure of the unit of study and possible disciplinary action.
2. I have completed the **Academic Honesty Education Module** on Canvas.
3. I understand that failure to comply with the Academic Dishonesty and Plagiarism in Coursework Policy can lead to the University commencing proceedings against me for potential student misconduct under Chapter 8 of the ***University of Sydney By-Law 1999*** (as amended).
4. This work is substantially my own, and to the extent that any part of this work is not my own I have indicated that it is not my own by acknowledging the source of that part or those parts of the work.
5. The assessment has not been submitted previously for assessment in this or any other unit, or another institution.
6. I acknowledge that the assessor of this assignment may, for the purpose of assessing this assignment may:
   a. Reproduce this assignment and provide a copy to another member of the school; and/or
   b. Use similarity detection software (which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking).
7. I have retained a duplicate copy of the assignment.

| | |
|---|---|
| Please type in your group number | |

| here to acknowledge this declaration: | |
|---|---|

The assignment consists of two tasks: ER Diagram & relational database designing and modeling Schema. In the first task, we have created the ER diagram which shows the conceptual design of the Sydney Entertainment & Theatre's database. Furthermore, in task 2 we have converted ER-diagram into the relational Model which depicts the relationship among all the entities and on the basis of the ER diagram we have created a database for SET. While creating the database we have applied various constraints in order to fulfil the demand. We have explained every point in details in the following report.

# ER-Diagram & Relational Modelling Explanation

**List of Strong Entities:** Production, theatre, Booking, Payment, Customer

**List of Weak Entity**: Performance, Seating Sections, Seat

First Entity is the **production (Strong entity)**

- It consists of 4 attributes and we have selected pname as a primary key
- Production has a set of schedule performances which is a record. And, the record can be an entity therefore, we have created **Performance** as a weak entity and performance_id as its discriminator. Moreover, it also consists of the Start and ends data/Time attribute.
- Performance is a weak entity that has a discriminator Performance_Id and is connected to both the strong entities Production, Theatre, and Booking. This results in a composite Primary Key (Performance_Id, Pname, T_name, B_Id) where T_name should be NOT NULL and has the Foreign keys (Pname, T_name and B_Id) because we have shown the relationship among three different entities that's why we have used it as a foreign key.
- The relationship between production and performance is that every production should schedule at least one performance whereas one performance must be scheduled by at most one production.
- The relationship between performance and the theatre is that**" a performance plays exactly one theatre whereas a theatre can play at least one performance".**

**Constraints and types of production:**
Pname VARCHAR(50), description VARCHAR (100) NOTNULL,PDate DATE NOT NULL,PSeatCost FLOAT NOT NULL, PRIMARY KEY (Pname).

**Constraints and types of performance:**
performance_id INTEGER, Start Time TIME NOT NULL, End Time TIME NOT NULL, date DATE NOT NULL,

Each performance takes place in a certain **theatre;** therefore, we have created **theatre as a strong** entity that is made up of 5 attributes:

- Tname is a primary key because each theatre has a unique name and others are Tdescription, capacity, address, and postal code.
- According to the case study, the theatre is made up of Seating section which also has some attributes and relation which is dependent on the theatre, therefore, we have created "Seating Section" as a weak entity.
- The relation between theatre and Seating section **is "Every theatre contains at least one Seating section. And vice versa. It is many-to-many relation**.

**Constraints and types of theatre:**
Tname VARCHAR(50), Tdescription VARCHAR(100) NOT NULL, capacity INTEGER NOT NULL,address VARCHAR(100) NOT NULL, Postcode INTEGER NOT NULL

Next, the **Seating section is a weak entity** that is comprised of 4 attributes and 1 weak entity relationship.

- Seating Sections is a weak entity has a discriminator Section_Id and is connected to Seat using seat_Id and Theatre using T_name. So, the composite primary key will be (Section_Id, Tname, seat_Id) Where seat_Id and T_name are Foreign keys.
- The other attributes are views and comfort level are composite keys and section seat cost is a single attribute.
- Moreover, each section has a lot of seats. Therefore, we have created a seat as a weak entity and seat_id as a discriminator.
- The relationship between the seat section and the seat is **"Every seat section contains at least one seat".**

**Constraints and types of Section:**
section_id INTEGER, section_seat_cost FLOAT

**Seat (Weak entity)** that only contains the discriminator which is seat_id. It is dependent on the seat section, which is also a weak entity and they share **the "many to one "**weak relationship named as composed of. Seat is a weak entity connected to Production, Performance and Seat Sections with a discriminator Seat_Id and Primary Key (Seat_Id, Pname, Performance_Id, Section_Id). The foreign keys are Pname, Performance_Id, Section_Id.

**Constraints and types of Seat:**
seat_id INTEGER

We have created **Customer (strong entity)** because the customers are the one who is going to integrate with the system.

- It has 6 attributes: Cust_Id (primary key), fname, lname, number,email, dob.
- We have considered Cust_Id as a primary key because it will be uniquely assigned to each customer.
- We have forged two relations with two different entities i.e. Payment and Booking.
- The relationship between Customer and Payment is "Every customer place at least one payment".
- The relationship between Customer and Booking is "Every customer place at least one payment".

**Constraints and types of Customer:**
cust_id INTEGER, fname VARCHAR(50) NOT NULL, lname VARCHAR(50) NOT NULL, number CHAR(10) NOT NULL, dob DATE NOT NULL, email VARCHAR(50) NOT NULL

**A booking entity** is an entity that **shares two ternary relationships**. It has three attributes i.e. B_id, Bdate and the TotalCost which is a derived attribute whose value is variable and is being got calculated by using the production seat cost and section seat cost The first one is a customer will use to book the seat in specific performance and all of them are interconnected to each other by ternary relationship and **the degree of cardinality between booking and performance is that" Each booking should select at least 1 performance and the cardinality between booking and the seat is " many-to-many relationship.** Second, Shows the association among customer, payment, and booking. the degree of **cardinality between booking and payment is many to many relationships**. The customer will select the performance and select and will get the unique IDs in terms of performance_id and seat_id and the whole booking process will also get the unique booking id (bid) for each time the customer books a performance. Booking is a strong entity connected to performance, seat and customer with the primary key B_Id. The **foreign key** is Cust_Id from Customer.

**Constraints and types of booking:**
B_id INTEGER, BDate DATE NOT NULL /*CHECK*/, TotalCost FLOAT NOT NULL

**The payment entity** is the most important that will help the customer to pay the total booking amount. The payment provides different methods/subtypes such as Gift cards, Vouchers & Credit cards. A customer is allowed to split its payment into multiple methods that are the reason we have used the bold line which means that the customer is allowed to use all the subclasses for the payment. However, there is only one constraint where the customer is only allowed to use the voucher only once. In SQL, we have applied a unique constraint on the voucher and due to that customers are not allowed to use the used voucher again.

**Constraints and types of payment:**

We have further segregated our payment methods into entity i.e. Gift(entity): attributes(gcard and pin), Card(entity): attributes(Cardno, name, cexpiry, CVV) & Voucher(entity): attributes(code & vexpiry). Card and Gift share the many to one relationship whereas the voucher is only allowed once due to the SQL constraint.

**ER Diagram**