**DATABASE – S4 PRACTICE-CORRECTION**

# **Exercise 1**

Q1 – Complete the **attributes type**s of Book and Read entities (5 points)

Q2 – Complete the **relation** between the Book and Reader entities (5 points)

|  |  |
| --- | --- |
| **Reader** | |
| ReaderID  Name  Class  Address | numeric string  string  string |

|  |  |
| --- | --- |
| **Book** | |
| BookID  Title  Publishment date  Language | numeric string datetime string |

many

many

|  |  |
| --- | --- |
| **Reader** | |
| ReaderID  Name  Address  …………. | Numeric  String  String |

|  |  |
| --- | --- |
| **Book** | |
| BookID  Title  Language  ……………… | Numeric  String  String |

Q3 – We have created an **additional Associative table** to manage the previous relation between Book and Reader

* Complete the missing parts!

.

one

one

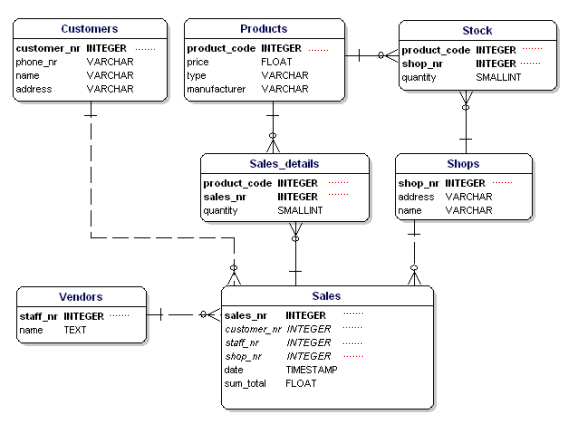
|  |  |
| --- | --- |
| **BookBorrow** | |
| BorrowID  Date borrow  Date return  ReaderID  BookID | Numeric  Datetime  Datetime  Numeric  Numeric |

many

many

# Exercise 2

Q1: complete the missing part of a model diagram below with PK as primary or FK foreign key.

****

PK

PK

PK

FK

FK

PK

FK

PK

FK

FK

PK

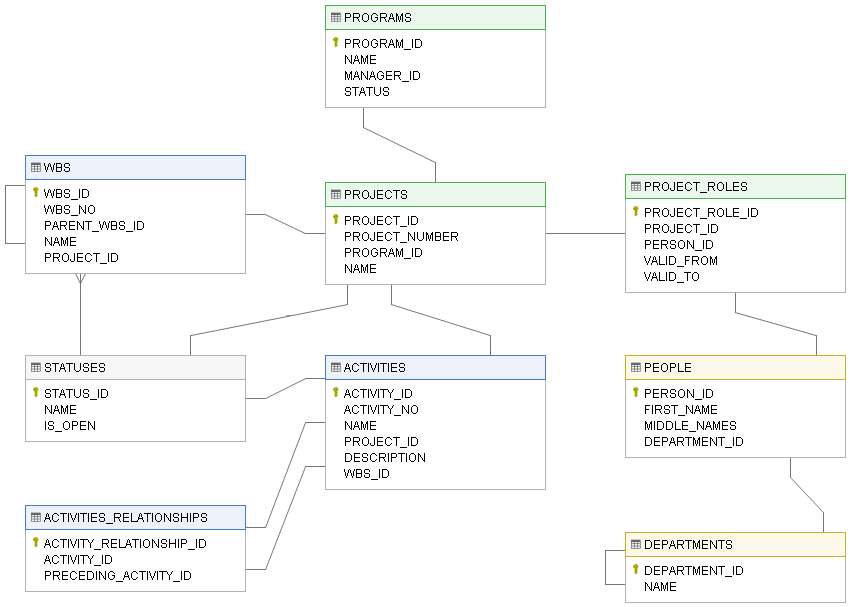
PK

FK

FK

# Exercise 3

Q1: Complete the relation between each entity on the database relation model. Take for example the relation between **PROGRAMS** and **PROJECTS.**

****

one

one

many

many

one

one

one

many

one

one

many

many

one

many

one

one

one

one

one

many

one

one

many

one

# Exercise 4: **Google Classroom database**

Google Classroom is the tool used to manage PNC classes, where the teacher can assign homework to the students of different classes.

**Part 1: Data types**

Q1: Complete the attributes types in the following tables

|  |  |
| --- | --- |
| **User** | |
| user ID  email  password  name  role | **Numeric**  **String**  **String**  **String**  **String** |

|  |  |
| --- | --- |
| **Classroom** | |
| classroom ID  name  section  subject | **Numeric**  **String**  **Numeric**  **String** |

|  |  |
| --- | --- |
| **Assignment** | |
| assignment ID  title  description  deadline | **Numeric**  **String**  **String**  **Datetime** |

|  |  |
| --- | --- |
| **Comment** | |
| comment ID  content  user ID  assignment ID | **Numeric**  **String**  **Numeric**  **Numeric** |

**Part 2: Relation between the entities**

Here are some observations that can help us design the Google Classroom database:

* A user can create many classrooms as teacher
* A user can join many classrooms as student
* A classroom can have many teachers
* A classroom can have many students
* A teacher can post many assignments in a classroom
* A assignment post can have many comments from students or teachers

Note: **ERD** =Entity relation diagram

**Q1:**   **USER** and **CLASSROOM**

1. Type of relation: *one to one, one to many, many to many*?

The relation between USER and ClASSROOM is many to many. Because a User can have many classroom, and a Classroom can have many users.

1. Do you need to create an intersection table? Why?

Yes , I do because , an intersection table can manage relation many to many effectively.

1. Create the ERD representing to represent those 2 entities and their relation

|  |  |
| --- | --- |
| **Classroom** | |
| classroom ID  name  section  subject | **Numeric**  **String**  **Numeric**  **String** |

|  |  |
| --- | --- |
| **User** | |
| user ID  email  password  name  role | **Numeric**  **String**  **String**  **String**  **String** |

one

one

|  |  |
| --- | --- |
| **User\_Classroom** | |
| User\_ClassroomID  user ID  classroom ID | **Numeric**  **Numeric**  many  **Numeric** |

many

**Q2:**   **CLASSROOM** and **ASSIGNEMENT**

1. Type of relation: *one to one, one to many, many to many?*

The relation between ClASSROOM and ASSIGNMENT is many to many. One Classroom can give many assignment. And one assignment can be given to different Classrooms.

1. Do you need to create an intersection table or not? Why?

Yes , I do because , an intersection table can manage relation many to many effectively.

1. Update the previous ERD to represent those 2 entities and their relation

|  |  |
| --- | --- |
| **Classroom** | |
| classroom ID  name  section  subject  one | **Numeric**  **String**  **Numeric**  **String** |

|  |  |
| --- | --- |
| **Assignment** | |
| assignment ID  title  description  deadline | **Numeric**  **String**  **String**  **Datetime** |

one

many

many

|  |  |
| --- | --- |
| **Classroom\_Assignment** | |
| Classroom\_Assignment ID  assignment ID  classroom ID | **Numeric**  **Numeric**  **Numeric** |

**Q3:**   **COMMNENT** and **ASSIGNEMENT**

1. Type of relation: *one to one, one to many, many to many*

The relation between COMMENT and ASSIGNMENT is one to many. Indeed, you can have many Comments on one Assignment, but one comment belongs to one assignment.

1. Do you need to create an intersection table or not? Why?

No I do not because the relation is one to many, so no need to create intersection table.

1. Update the previous ERD to represent those 2 entities and their relation

|  |  |
| --- | --- |
| **Comment** | |
| comment ID  content  user ID  assignment ID | **Numeric**  **String**  **Numeric**  **Numeric** |

|  |  |
| --- | --- |
| **Assignment** | |
| assignment ID  title  description  deadline | **Numeric**  **String**  **String**  **Datetime** |

one

many

**Q4:**   **COMMNENT** and **USER**

1. Type of relation: *one to one, one to many, many to many?*

The relation between COMMENT and User is one to many. Indeed, a User can make many comments, but one comment is made only by one user.

1. Do you need to create an intersection table or not? Why?

No I do not because the relation is one to many, so no need to create intersection table.

1. Update the previous ERD to represent those 2 entities and their relation

|  |  |
| --- | --- |
| **User** | |
| user ID  email  password  name  role | **Numeric**  **String**  **String**  **String**  **String** |

|  |  |
| --- | --- |
| **Comment** | |
| comment ID  content  user ID  assignment ID | **Numeric**  **String**  **Numeric**  **Numeric** |

one

many