**DATABASE MANAGEMENT AND DATABASE ANALYTICS**

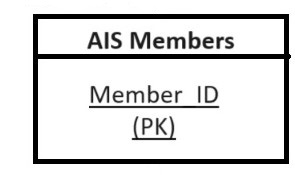
**Take Home Practicum 1**

**Vinay Kacharam**

**Question : 1**

**Supertype Entity: "AIS Members" :**

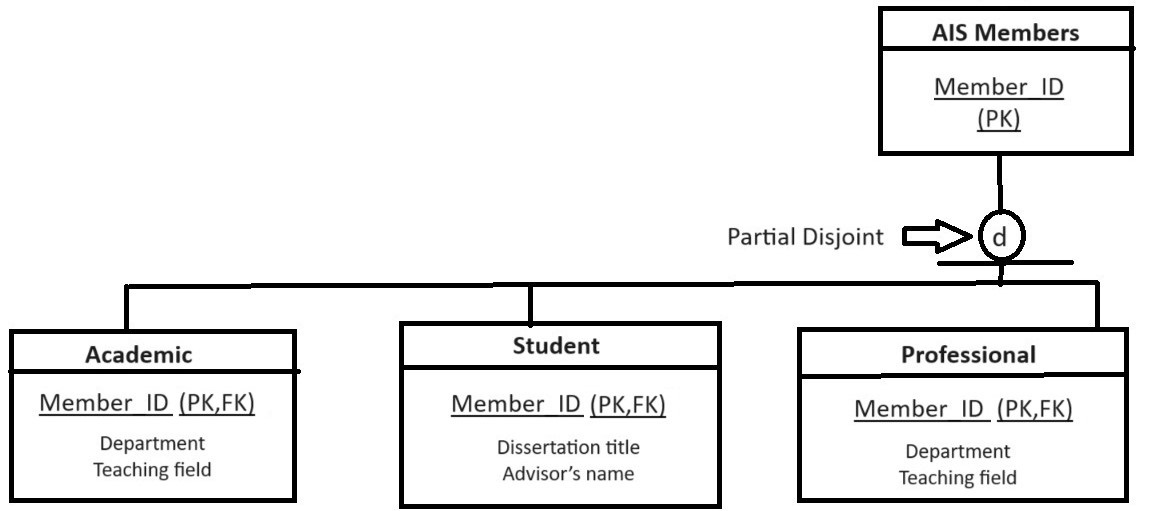
Think of this as the general template for all AIS members. It contains information that applies to all members, such as a unique ID, name, and contact details. This is where you store common details shared by everyone.



As shown in the above diagram , Since there isn't explicit information regarding the primary key, it's reasonable to assume that we can introduce an attribute named **"Member\_ID"** to serve as the **Primary Key (PK)** for this entity. This “Member\_ID” attribute uniquely identifies each member of the group,the other attributes can come below it as well.

**Subtype Entities: "Academic," "Student," and "Professional" :**

Now, imagine that within the AIS Members, there are three specific Members : Academics , Students , and Professionals .



**Disjoint with partial specialization :**

We can observe that the three subtypes are connected to the main entity through a **Disjoint with partial specialization**. In this specialization, it's important to note that an individual cannot simultaneously belong to more than one of these categories. However, it is possible that there are other member types besides of these three categories.

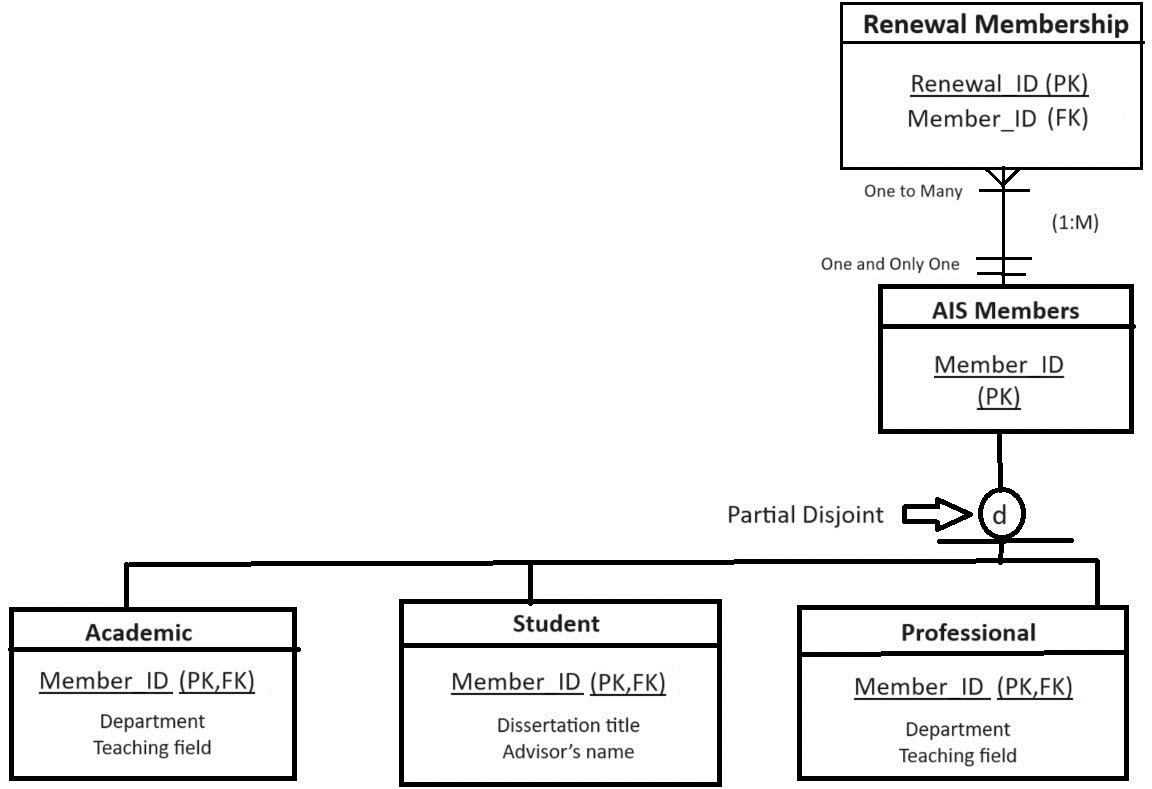
**Foreign Key :**

Expanding upon the previous discussion, it's evident that the subtypes—Academics, Students, and Professionals—inherit certain attributes from the supertype entity, "AIS members," as previously mentioned. Now, let's introduce an additional concept known as a "**Foreign key**," illustrated in the diagram above.

In the above diagram, we can see that the primary key "Member\_ID" from the supertype entity, "AIS Members," is employed as a **Foreign key (FK)** in the three subtype entities. This is because these subtype entities are derived from the supertype and need to establish a relationship with it, making use of "Member\_ID" as the linking key.

**Crow’s Foot Model:**

This comes under one of the important parts of the ER diagrams . Crow's foot notation is used in Barker's Notation, Structured Systems Analysis and Design Method (SSADM) and information engineering. Crow's foot diagrams represent entities as boxes, and relationships as lines between the boxes. The relation comes into the figure to create proper classification and simplification of the data given in the problem with cardinalities as shown in the diagram:



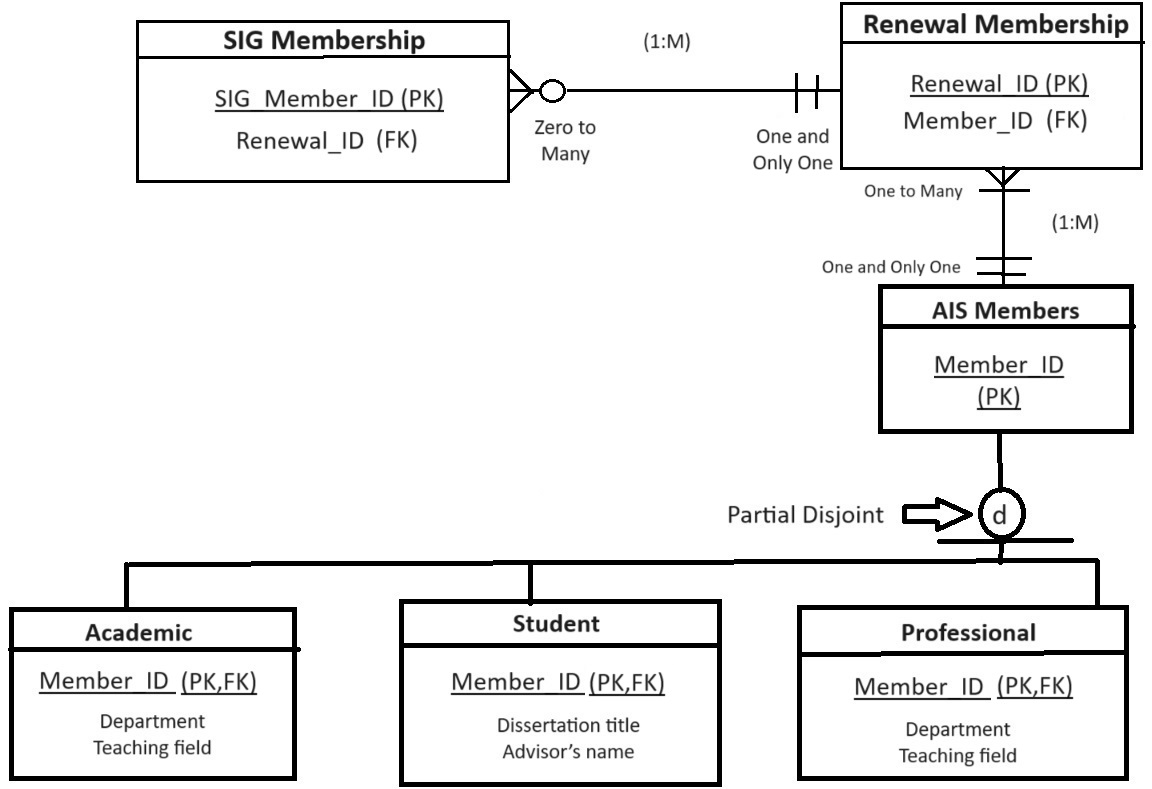
**AIS Members** : One (Minimum Value) to One (Maximum Value)

**Renewal Membership :** One (Minimum Value) to Many (Maximum Value)

We can see in the above diagram .A newly introduced entity, "Renewal Membership," necessitates the creation of a primary key. In this case, **"Renewal\_ID" (PK)** is chosen as the primary key for this entity. This decision is informed by the statement indicating that a member can renew their membership at least once, or multiple times, and each renewal is linked to only one member. This description distinctly defines a 1:M (one-to-many) relationship between the two entities.

To facilitate this relationship and provide insight into the members associated with the subscriptions, **"Member\_ID"(FK)** is employed as a foreign key within the " Renewal Membership " entity. This allows us to establish a clear connection between renewals and the members they pertain to in the database.

In the below diagram, we have introduce another entity **“SIG Membership”**, prompts the need for another primary key, and in this case, **"SIG\_member\_ID" (PK)** is designated as the primary key for this entity. In this entity we have an take a **“Renewal\_ID” (FK)** as the foreign key in this ER diagram.

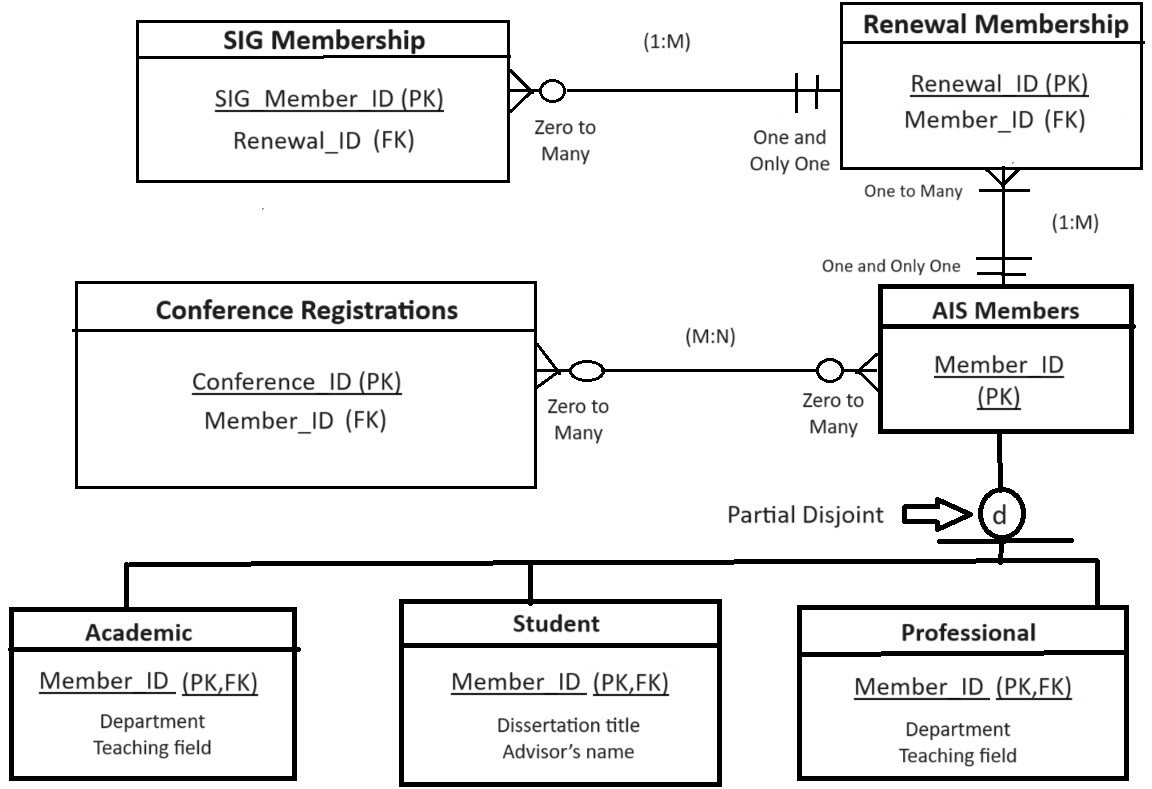


**Renewal Membership** : One (Minimum Value) to One (Maximum Value)

**SIG Membership** : Zero (Minimum Value) to Many (Maximum Value)

The membership renewal transaction may include many SIG memberships but may not include any at all, since joining a SIG is optional. A particular SIG membership needs to be tied to one and exactly one membership renewal. In this context, we observe the presence of a **1:M (one-to-many)** relationship, further emphasizing the one-to-many connections between these entities.

In the below diagram, we have introduce another entity “**Conference Registrations**”, prompts the need for another primary key, and in this case, **"Conference\_ID" (PK)** is designated as the primary key for this entity. In this entity we have an take a **“Member\_ID” (FK)** as the foreign key for the entity**.** This includes the “**Crows Foot Model**” again as shown below.



**AIS Members** : Zero (Minimum Value) to Many (Maximum Value)

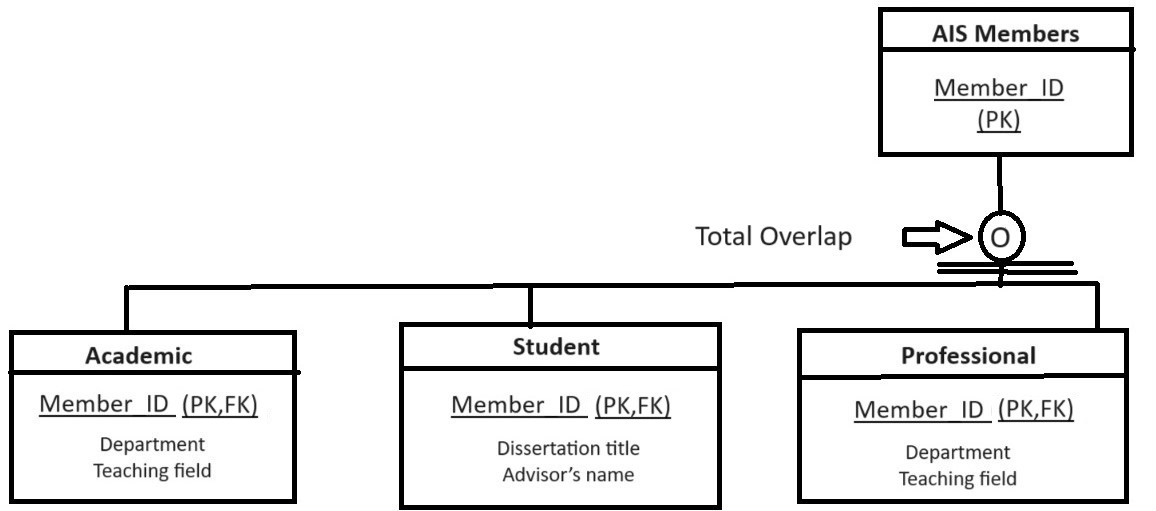
**Conference Registrations** : Zero (Minimum Value) to Many(Maximum Value)

This diagram also shows us the final ER diagram which has been derived from the database information .

An AIS member can register for many conferences over time, A particular conference can have many members registered for it In this context, we observe the presence of a (M:N) Many-to-Many relationship, further emphasizing the Many-to-many connections between these entities.

**Extra Credit Opportunities:**

1)



**Overlapping with total Specialization:**

We can observe that the three subtypes are connected to the main entity through a **Overlapping with total specialization**. In this specialization, it's important to note that an individual member can simultaneously belong to more than one of these categories at a time. However, it is not possible that there are other member types besides of these three categories.

**Question : 2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **SessionID** | **Session\_Name** | **Track** | **Duration** | **Room** | **Start\_Time** | **End\_Time** | **Nbr\_Papers** |
| SAT1 | Social Inclusion | SIGSI | 1:30 | Isla Mujeres 1 | 8:30 | 10:00 | 5 |
| SAT4 | Social Analytics | Social Computing | 1:30 | Cozumel3 | 11:00 | 12:30 | 3 |
| SAT2 | System Analysis & Design | SIGSAND | 1:30 | Isla Mujeres 1 | 8:30 | 10:00 | 5 |
| FRI1 | Leveraging Data in Healthcare | SIGHEALTH | 2:00 | XCaret2 | 16:00 | 18:00 | 4 |
| THU1 | Health Informatics & HIT 2 | SIGHEALTH | 2:00 | Costa Maya 5 | 16:00 | 18:00 | 4 |
| SAT3 | Big Data & Messy Data | Data Science | 1:30 | Tulum 3 | 8:30 | 10:00 | 3 |

1. Use the relation above *(it is in at least 1NF)* to determine **ALL** functional dependencies that the data suggests **MAY** be present. Use A→B notation. To be very clear: identify all minimal (atomic) functional dependencies that have data that suggest this functional dependency may exist, even if you think the functional dependency would fail a logical test.

You should have more blank spaces below than you need. Fill in however many you find.

1.SessionID → Session\_Name

(Assumption : Each SessionID uniquely determines the Session\_Name)

2.SessionID → Duration

(Assumption : Each SessionID uniquely determines the Duration)

3.SessionID → Room

(Assumption : Each SessionID uniquely determines the Room)

4.SessionID → Start\_Time

(Assumption : Each SessionID uniquely determines the Start\_Time)

5.SessionID → End\_Time

(Assumption : Each SessionID uniquely determines the End\_Time)

6.Track → Duration

(Assumption : Each Track uniquely determines the Duration)

7.Track → Room

(Assumption : Each Track uniquely determines the Room)

8.Track → Start\_Time

(Assumption : Each Track uniquely determines the Start\_Time)

9.Track → End\_Time

(Assumption : Each Track uniquely determines the End\_Time)

10.Track → Nbr\_Paper

(Assumption : Each Track uniquely determines the Duration)

11.Duration→ Start\_Time,End\_Time

(Assumption : Each Duration uniquely determines the Start\_Time,End\_Time)

12.Room→ Start\_Time,End\_Time

(Assumption : Each Room uniquely determines the Start\_Time,End\_Time)

13.Track → SessionID

(Assumption : Each Track\_ID uniquely determines the SessionID)

B. Would any of these functional dependencies cause you to be surprised if it was not removed (i.e., not eliminated as being a legitimate functional dependency) when it is eventually tested logically? Why or why not?  
If you have more than one option for your response, choose the one that would surprise you the most.

**Track → Nbr\_Paper**

**Reasoning:**

This dependency is unexpected since, in a normal conference session database, the number of papers (Nbr\_Papers) is often associated with a session (SessionID) rather than a track (Track). Sessions are typically categorized or thematically organized into tracks, and a track may contain many sessions. However, the quantity of papers may vary depending on the session.

C. Which attribute (if any) could be eliminated without losing any real content, because it seems to be a type of redundancy?

Based on the provided data, it appears that the "Duration" attribute can potentially be eliminated without losing any real content because it seems to be a type of redundancy.

**Reasoning:**

"Duration" can be derived from "Start\_Time" and "End\_Time" attributes. It represents the time span of a session, which can be calculated by subtracting the "Start\_Time" from the "End\_Time." By calculating "Duration" when needed, you ensure that the data remains consistent and accurate, as it will always reflect the actual duration of the session based on the provided start and end times.

So, removing the "Duration" attribute and calculating it as needed would eliminate redundancy without losing any essential information.

D. Comment briefly on the suitability of using **SessionID** as the primary key for the SESSION table in the database. Does it appear to be a good candidate key? Why or why not? Could any problems arise from its use? If so, what are they, and how might you resolve them when designing the database?

It may seem acceptable to use "SessionID" as the primary key for the database's SESSION table, but there are a few things to keep in mind:

**Advantages:**

Uniqueness : is a crucial need for a primary key, and "SessionID" is unique for each session.

Simpleness: It is a clear decision that is simple to comprehend, which is advantageous for database users and developers.

Performance: When using a primary key with a numeric identification like "SessionID" to query and join tables, good query performance is frequently achieved.

**Considerations and Potential Problems:**

Changing Values: Any change to "SessionID" values could trigger cascading modifications in other tables that use this primary key. This can be difficult and prone to mistakes.

Error caused by a human: "SessionID" can be produced manually or automatically. There could be issues with data integrity if these identifiers are assigned or managed by humans in error.

**Possible Solutions:**

Stable Identifiers: If possible, ensure that "SessionID" values are stable and unlikely to change over time. Avoid manual changes whenever possible.

Data Validation: Implement data validation and constraints to prevent duplicate or conflicting "SessionID" values from being inserted.

Audit Trails: Keep an audit trail of changes to "SessionID" values if they must be modified, so you can track and manage any updates.

In summary, "SessionID" can serve as a primary key, but careful attention to data integrity and management is necessary to avoid potential problems, especially if there's a possibility of changing values. Consider using a stable surrogate key or additional data validation measures to enhance database robustness.