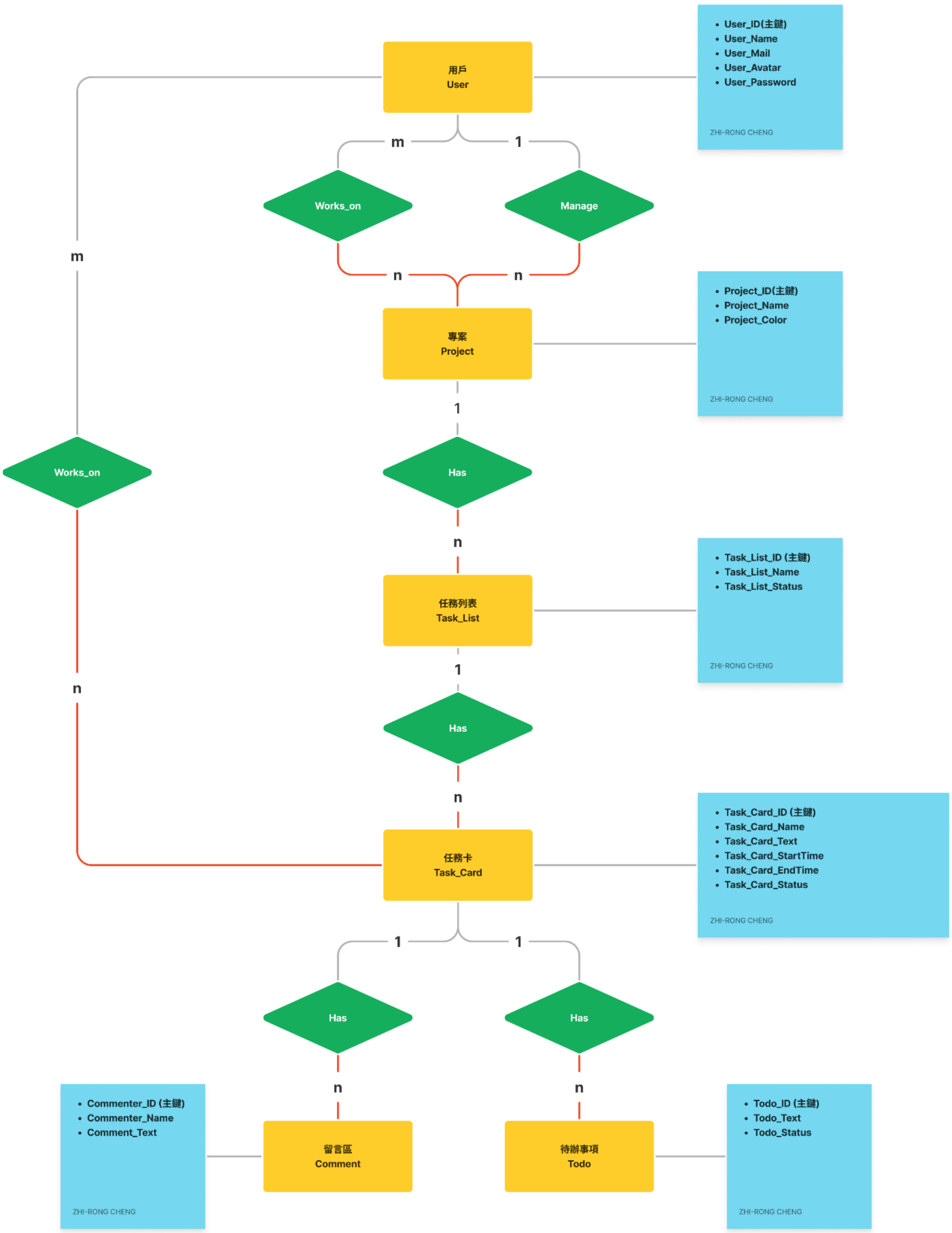
ProjectMgmt - Relational Schema 設計報告

1. 根據以下 ER Diagram 進行 ER-Relational Mapping



- Step 1: for each regular entity type E
    - Create a relation R
    - Include all simple attributes of E
    - Include only the simple component attributes of a
    - composite attribute
    - Choose one of key attributes of E as primary key for R

在此我們做了以下轉換:

User					
User_ID	User_Name	User_Mail	User_Avatar	User_Password	
PK					
Project					
Project_ID	Project_Name	Project_Color			
PK					
Task_List					
Task_List_ID	Task_List_Name	Task_List_Status			
PK					
Task_Card					
Task_Card_ID	Task_Card_Name	Task_Card_Text	Task_Card_StartTime	Task_Card_EndTime	Task_Card_Status
PK					
Todo					
Todo_ID	Todo_Text	Todo_Status			
PK					
Comment					
Commenter_ID	Commenter_Name	Comment_Text			
PK					

- Step 2: for each weak entity type W
    - Create relation R
    - Include all simple attributes of W as attributes of R
    - Include primary key of relations that correspond to owner entity type as foreign key of R
    - Primary key of R = (primary key of owner, partial key of W)

  我們沒有 weak entity type, 故跳過這個 Step

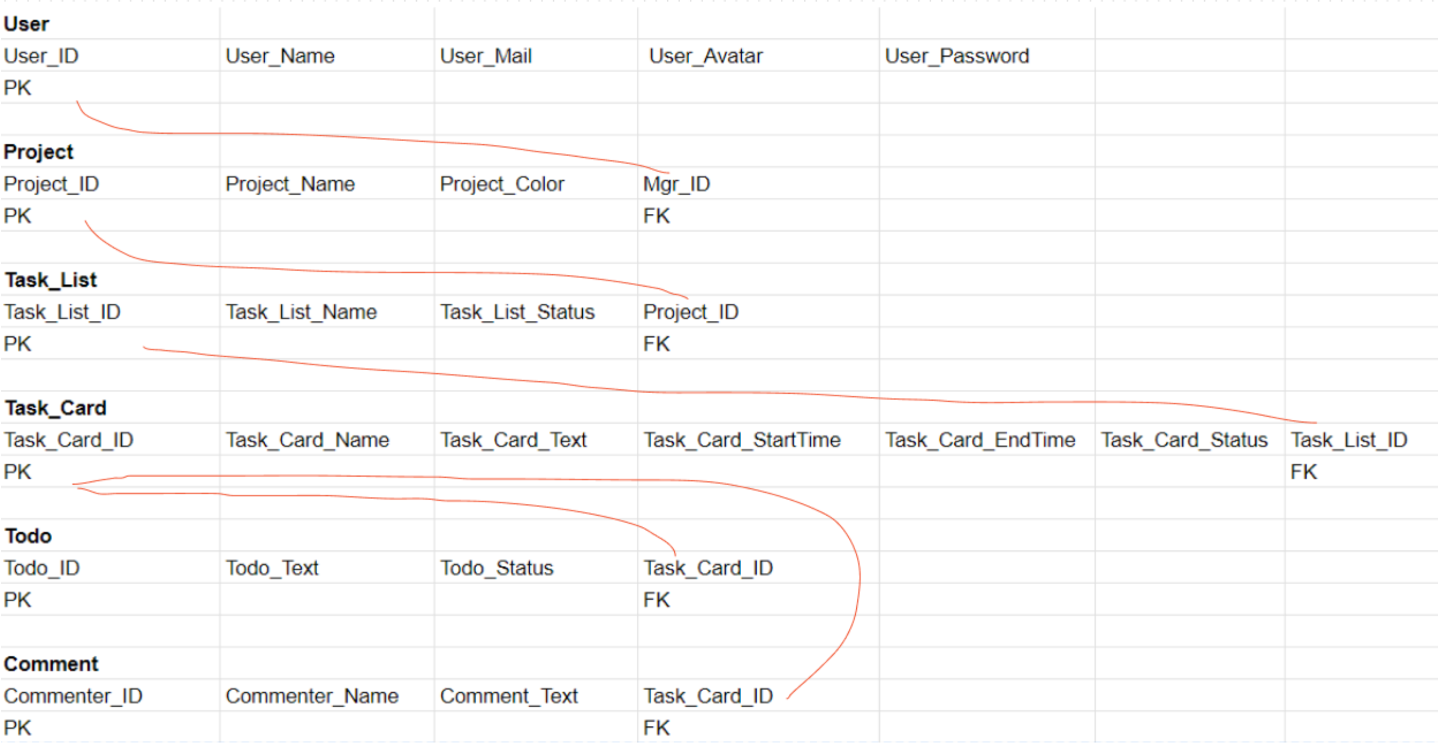
- Step 3: for each binary 1:1 relationship type R

- Identity the relations S & T participating in R
- Choose one relation S (better to choose entity type with total participation) & include the primary key of T as foreign key in S
- Include all simple attributes of R as attributes of S

我們沒有 binary 1:1 relationship type，故跳過這個 Step

- Step 4: for each binary 1:N relationship type R
  - Identify relation S at the N-side of relationship
  - Include primary key of relation T (the other side) as foreign key in S
  - Include any simple attributes of R as attributes of S

在此我們做了以下轉換:

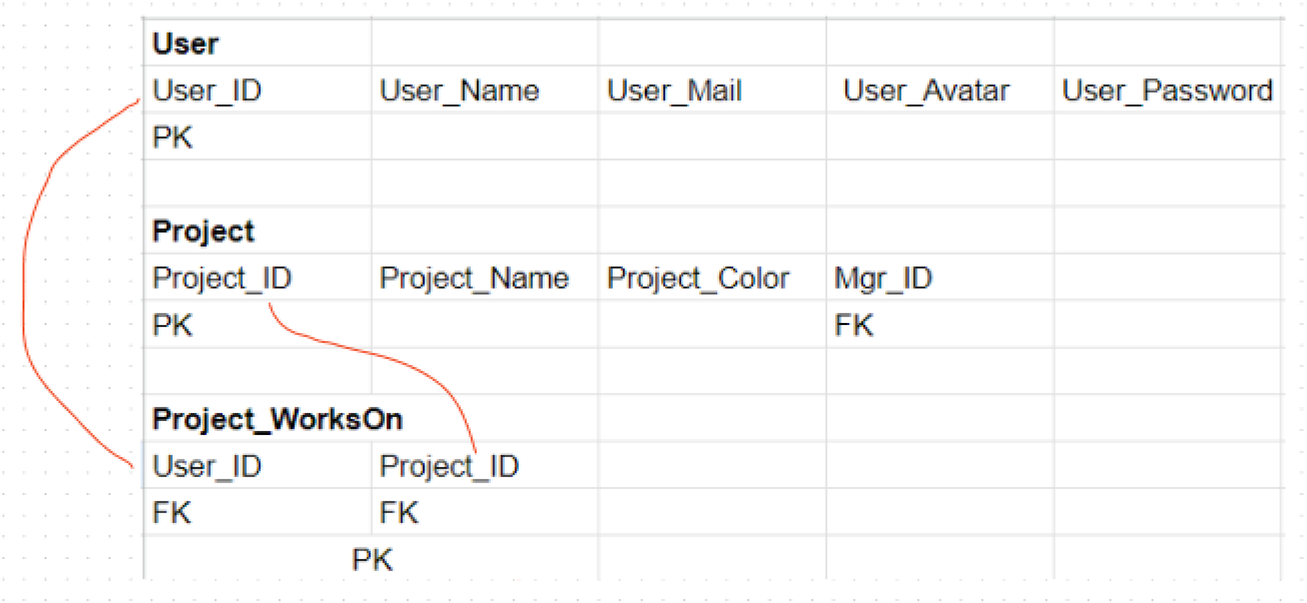


User						
User_ID	User_Name	User_Mail	User_Avatar	User_Password		
PK						
Project						
Project_ID	Project_Name	Project_Color	Mgr_ID			
PK			FK			
Task_List						
Task_List_ID	Task_List_Name	Task_List_Status	Project_ID			
PK			FK			
Task_Card						
Task_Card_ID	Task_Card_Name	Task_Card_Text	Task_Card_StartTime	Task_Card_EndTime	Task_Card_Status	Task_List_ID
PK						FK
Todo						
Todo_ID	Todo_Text	Todo_Status	Task_Card_ID			
PK			FK			
Comment						
Commenter_ID	Commenter_Name	Comment_Text	Task_Card_ID			
PK			FK			

- Step 5: for each binary M:N relationship type R
  - Create a new relation S to represent R
  - Include primary keys of participating relations as foreign key of S
  - Their combination form primary key of S
  - Include any simple attributes of R as attributes of S

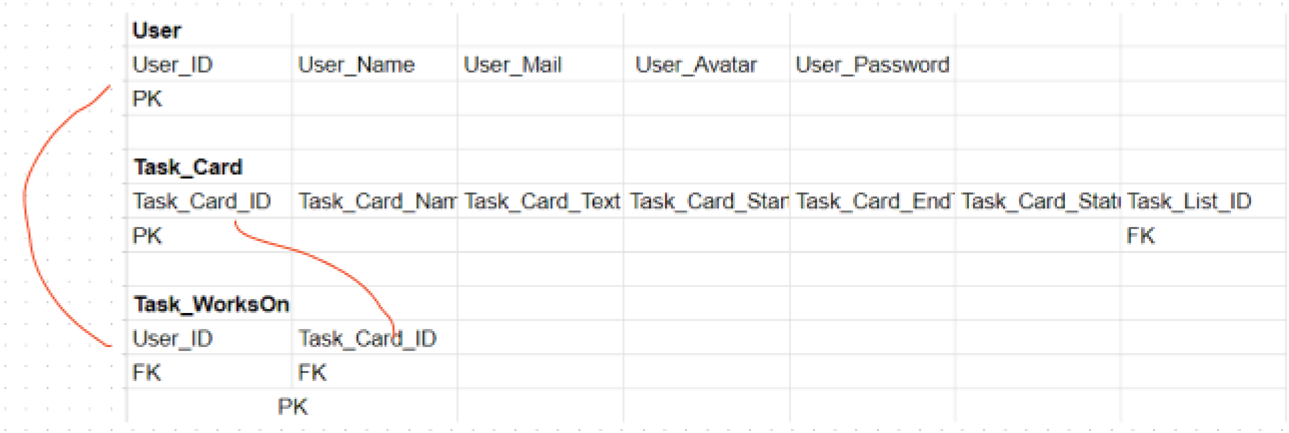
在此我們做了以下轉換:

a. Project_WorksOn



User				
User_ID	User_Name	User_Mail	User_Avatar	User_Password
PK				
Project				
Project_ID	Project_Name	Project_Color	Mgr_ID	
PK			FK	
Project_WorksOn				
User_ID	Project_ID			
FK	FK			
PK				

b. Task_WorksOn



User						
User_ID	User_Name	User_Mail	User_Avatar	User_Password		
PK						
Task_Card						
Task_Card_ID	Task_Card_Nam	Task_Card_Text	Task_Card_Star	Task_Card_End	Task_Card_Stat	Task_List_ID
PK						FK
Task_WorksOn						
User_ID	Task_Card_ID					
FK	FK					
PK						

- Step 6: for each multivalued attribute A
  - Create a new relation R
  - Include an attribute corresponding to A
  - Include primary key K (as a foreign key of R) of relation S as an attribute
  - Primary key of R = (A, K) – If A is composite, include its simple components

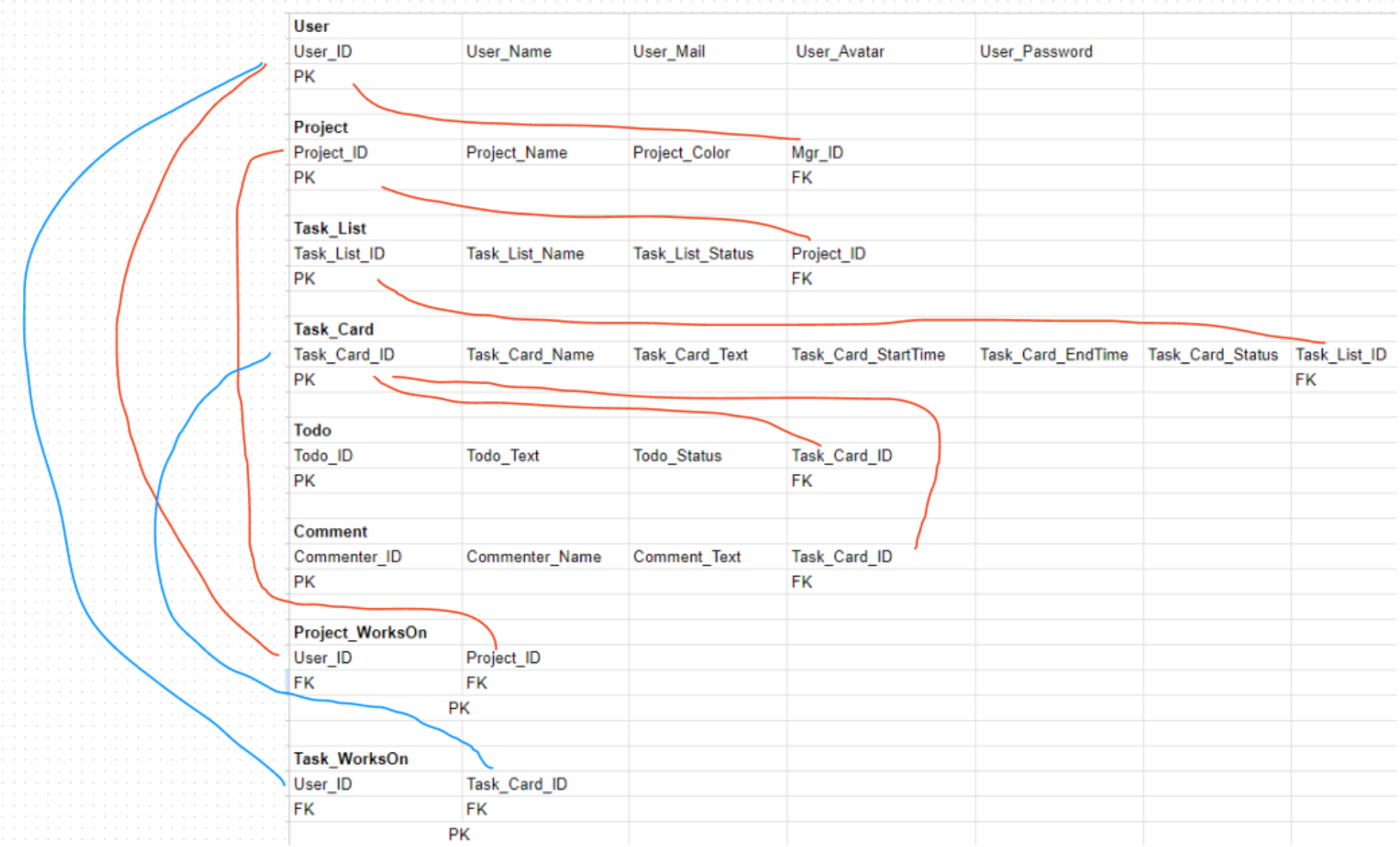
我們沒有 multivalued attribute，故跳過這個 Step

- Step 7: for each n-ary relationship type, n>2
  - Create a new relation S to represent R
  - Include primary key of participating relations as foreign key
  - Include any simple attributes of R as attributes of S

- ○ Primary key of S = all foreign keys that references participating R

我們沒有 n-ary relationship type，故跳過這個 Step

2. 最終 Relational Schema 設計



User						
User_ID	User_Name	User_Mail	User_Avatar	User_Password		
PK						
Project						
Project_ID	Project_Name	Project_Color	Mgr_ID			
PK			FK			
Task_List						
Task_List_ID	Task_List_Name	Task_List_Status	Project_ID			
PK			FK			
Task_Card						
Task_Card_ID	Task_Card_Name	Task_Card_Text	Task_Card_StartTime	Task_Card_EndTime	Task_Card_Status	Task_List_ID
PK						FK
Todo						
Todo_ID	Todo_Text	Todo_Status	Task_Card_ID			
PK			FK			
Comment						
Commenter_ID	Commenter_Name	Comment_Text	Task_Card_ID			
PK			FK			
Project_WorksOn						
User_ID	Project_ID					
FK	FK					
	PK					
Task_WorksOn						
User_ID	Task_Card_ID					
FK	FK					
	PK					

- Relational Schema
  - ○ User:
    - ■ User_ID: String NOT NULL
    - ■ User_Name: String
    - ■ User_Mail: String
    - ■ User_Avatar: String
    - ■ User_Password: String

    PK -> User_ID

  - ○ Project:
    - ■ Project_ID: String NOT NULL
    - ■ Project_Name: String
    - ■ Project_Color: String

    PK -> Project_ID
    FK -> Mgr_ID References User(User_ID)
    ON DELETE CASCADE

  - ○ Task_List:
    - ■ Task_List_ID: String NOT NULL

    - Task_List_Name: String
    - Task_List_Status: Boolean

  PK -> Task_List_ID
  FK -> Project_ID References Project(Project_ID)
  ON DELETE CASCADE

- Task_Card
    - Task_Card_ID: String NOT NULL
    - Task_Card_Name: String
    - Task_Card_Text: String
    - Task_Card_StartTime: String
    - Task_Card_EndTime: String
    - Task_Card_Status: Boolean

  PK -> Task_Card_ID
  FK -> Task_List_ID References Task_List(Task_List_ID)
  ON DELETE CASCADE

- Todo
    - Todo_ID: String NOT NULL
    - Todo_Text: String
    - Todo_Status: Boolean

  PK -> Todo_ID
  FK -> Task_Card_ID References Task_Card (Task_Card_ID)

- Comment
    - Commenter_ID: String NOT NULL
    - Commenter_Name: String
    - Comment_Text: String

  PK -> Todo_ID
  FK -> Task_Card_ID References Task_Card (Task_Card_ID)

- Project_WorksOn
    - User_ID: String NOT NULL
    - Project_ID: String NOT NULL

  PK -> (User_ID, Project_ID)
  FK -> User_ID, Project_ID References User(User_ID), Project(Project_ID)

- Task_WorksOn
    - User_ID: String NOT NULL
    - Task_Card_ID: String NOT NULL

  PK -> (User_ID, Task_Card_ID)
  FK -> User_ID, Task_Card_ID References User(User_ID), Task_Card(Task_Card_ID)