HW 1

ENTITIES:

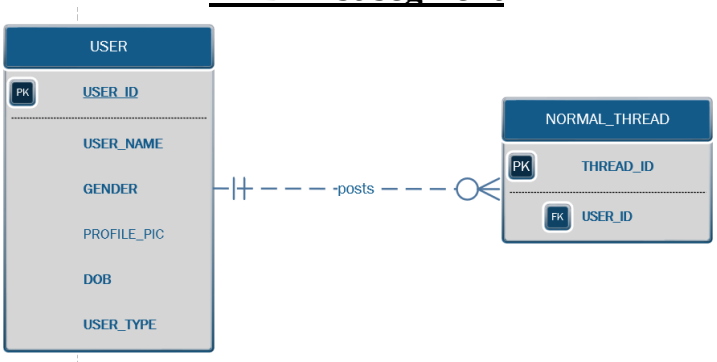
ER diagram consists of following entities-

1. USER (USER_ID,USER_NAME, GENDER, PROFILE PIC, DOB, USER_TYPE)
2. ADMIN (USER_ID,PROMO_DATE)
3. THREAD (THREAD_ID, TEXT_CONTENT, USER_TYPE)
4. ANNOUNCEMENT (THREAD_ID, USER_ID, V_ADDRESS)
5. NORMAL_THREAD(THREAD_ID, USER_ID)
6. T_PICTURE (PIC_ID,THREAD_ID, P_ADDRESS)
7. T_LIKING (USER_ID, THREAD_ID, LIKE_TYPE)
8. TAG (TAG_ID, TAG_NAME)
9. ADD_TAG (TAG_ID, THREAD_ID)
10. REPLY (REPLY_ID, THREAD_ID, USER_ID, TEXT_CONTENT)
11. R_PICTURE (PIC_ID, THREAD_ID,P_ADDRESS)
12. R_LIKING (USER_ID, REPLY_ID, LIKE_TYPE)

RELATIONSHIPS:

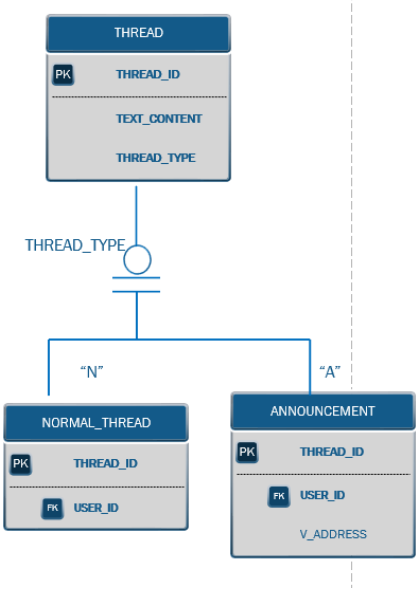
ENTITY	RELATIONSHIP	CONNECTIVITY	ENTITY
USER	posts	1:M	NORMAL_THREAD
USER	sends	1:M	REPLY
USER	likes	M:N	THREAD
USER	likes	M:N	REPLY
THREAD	contains	M:N	TAG
THREAD	includes	1:M	T_PICTURE
THREAD	has	1:M	REPLY
REPLY	includes	1:1	R_PICTURE
ADMIN	posts	1:M	ANNOUNCEMENT

ERD first segment



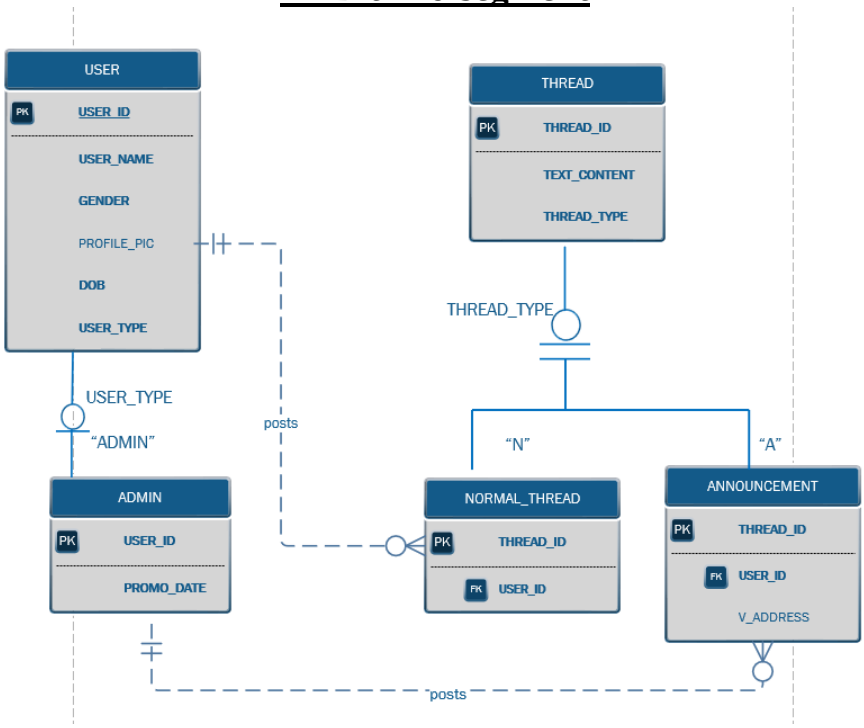
Every user posts threads. While each thread is posted by only one user. Thus, USER posts NORMAL_THREAD is a **1:M relationship**. It is possible that a user will not post any thread. Hence the **participation** of entity THREAD in relationship posts in **optional**. A **weak relationship** exists between USER and NORMAL_THREAD as USER_ID (primary key of USER entity) is only a foreign key in NORMAL_THREAD entity and is not inherited as primary key component in NORMAL_THREAD entity. Additional attribute 'age' is not included in USER table as it can be **derived** from DOB (date of birth) attribute. A user pay choose not to include profile picture. Hence PROFILE_PIC is **optional attribute** and others are set as **required**.

ERD second segment



THREAD is specialized into NORMAL_THREAD and ANNOUNCEMENT subtype with discriminator THREAD_TYPE attribute with domain ("N", "A"). The occurrence of every thread in in subtypes is **mandatory**. Hence it follows **complete constraint**. ANNOUNCEMENT has **unique attribute** video address(V_ADDRESS) which is **optional** as announcement contains at most 1 video. Also NORMAL_THREAD and ANNOUNCEMENT cannot be same. Hence it has **disjoint** constraint.

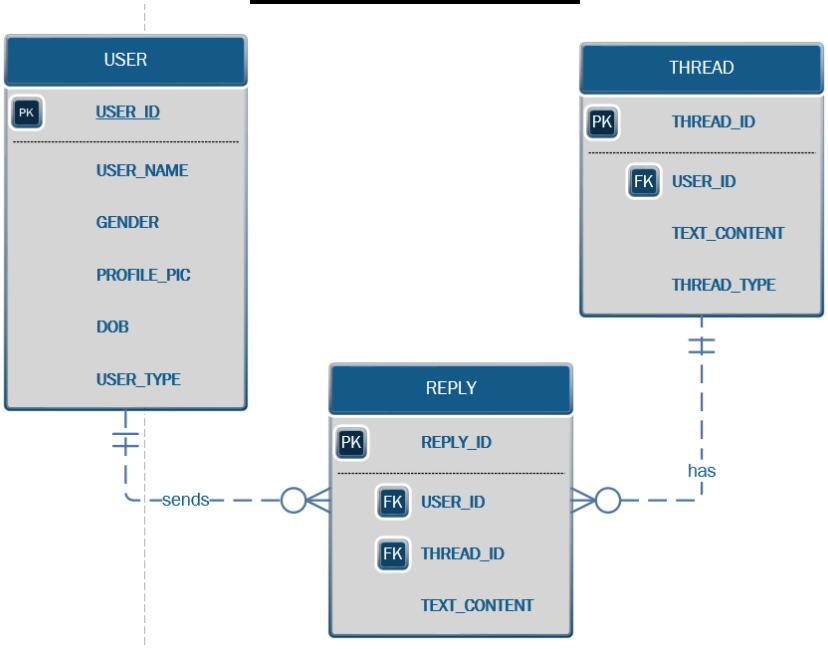
ERD third segment



Administrator is a user with a unique attribute of promotion date that normal user do not possess. If administrator were included in USER entity, then there would be lot of null attributes in promotion date column for normal user. To avoid this, ADMIN is a **subtype entity** of USER super entity type with USER_TYPE as **subtype discriminator** with **domain** ("NORMAL", "ADMIN"). Subtype ADMIN **inherits the all relationships** of super type USER. Thus, ADMIN participates in 1:M posts relationship with NORMAL_THREAD and 1:M sends relationship with REPLY. In addition, subtype ADMIN participates in 1: M posts relationship with ANNOUNCEMENT entity that USER entity can't participate. The occurrence of every user in ADMIN entity subtype is not **mandatory**. Hence it follows **partial constraint**.

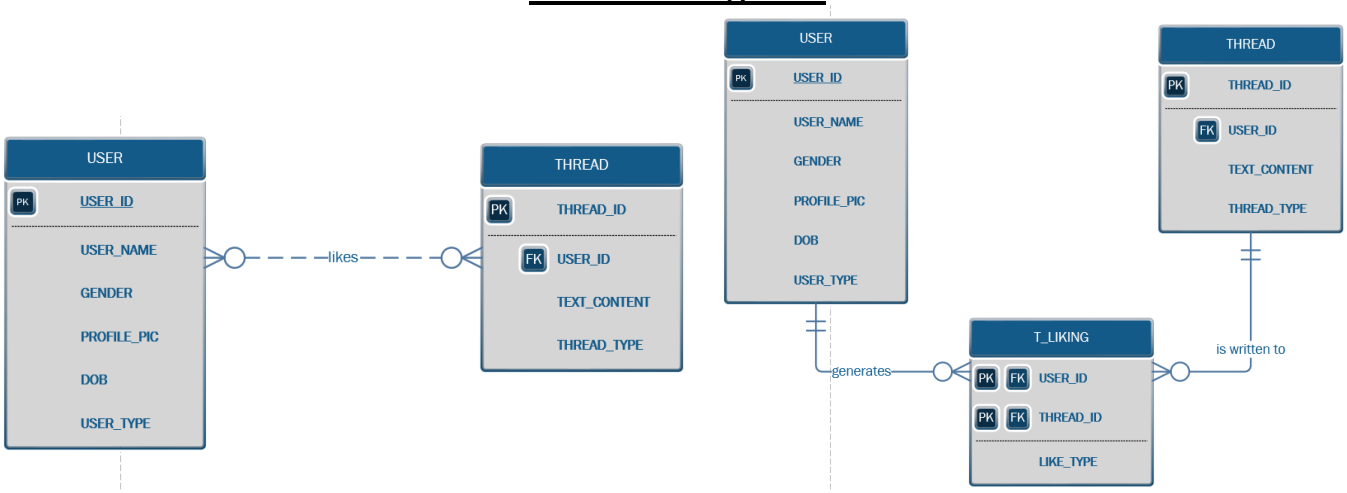
One **design approach** is to connect USER with THREAD entity through posts relationship. But ANNOUNCEMENT will **inherit** this relationship allowing USER to post ANNOUNCEMENT. To avoid this, USER entity is linked to NORMAL_THREAD by posts relationship

ERD fourth segment



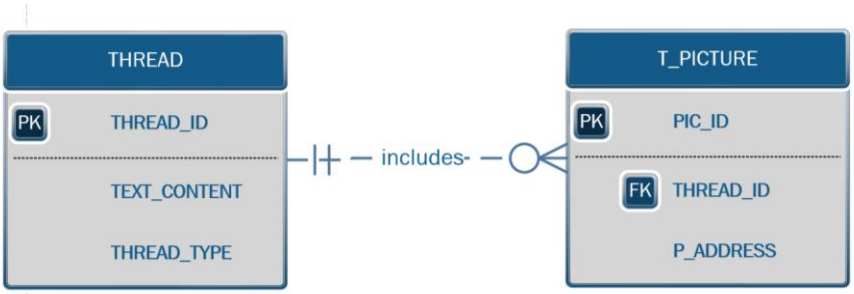
User can produce multiple replies and a reply belongs to only one user. Hence relationship between USER and REPLY is **1:M**. Similarly, one reply corresponds to only one thread and one thread can have many replies. Hence THREAD and REPLY participates in **1:M relationship**. Also REPLY is a **strong entity** as it can exist without THREAD and USER entities.

ERD fifth segment



User likes multiple threads. Also, a thread can be liked by many users. Thus, **M:N relationship** exists between USER and THREAD entity. This M:N relationship between USER and THREAD is decomposed into **two 1:M relationships** by inserting a **bridge table** T_LIKING. The bridge table T_LIKING inherits primary keys of USER(user_id) and THREAD(thread_id) as primary and foreign key. Hence **strong relationship** exists between USER-T_LIKING and THREAD-T_LIKING entities. Additionally, a user can choose to like or superlike a thread. Hence T_LIKING contains a LIKE_TYPE attribute with **domain** ("like","superlike"). Similar relationship exists between USER and REPLY table.

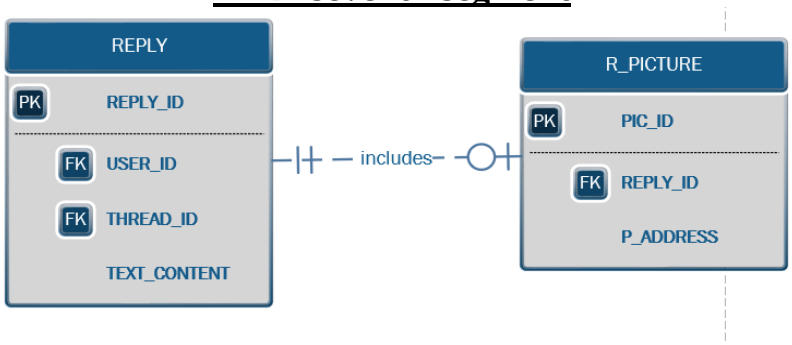
ERD sixth segment



A thread can contain multiple pictures. As the number of pictures is not defined, it is not possible to assign columns to picture attributes in THREAD table. Hence pictures are placed in another table T_PICTURE with foreign key THREAD_ID referencing to THREAD entity. The relationship is **1:M** with **one cardinality** on THREAD side (every picture can belong to only one thread) and **many cardinality** on T_PICTURE side (thread can contain multiple pictures) . Also, a thread can have none pictures. Hence participation of THREAD on T_PICTURE side is **optional**.

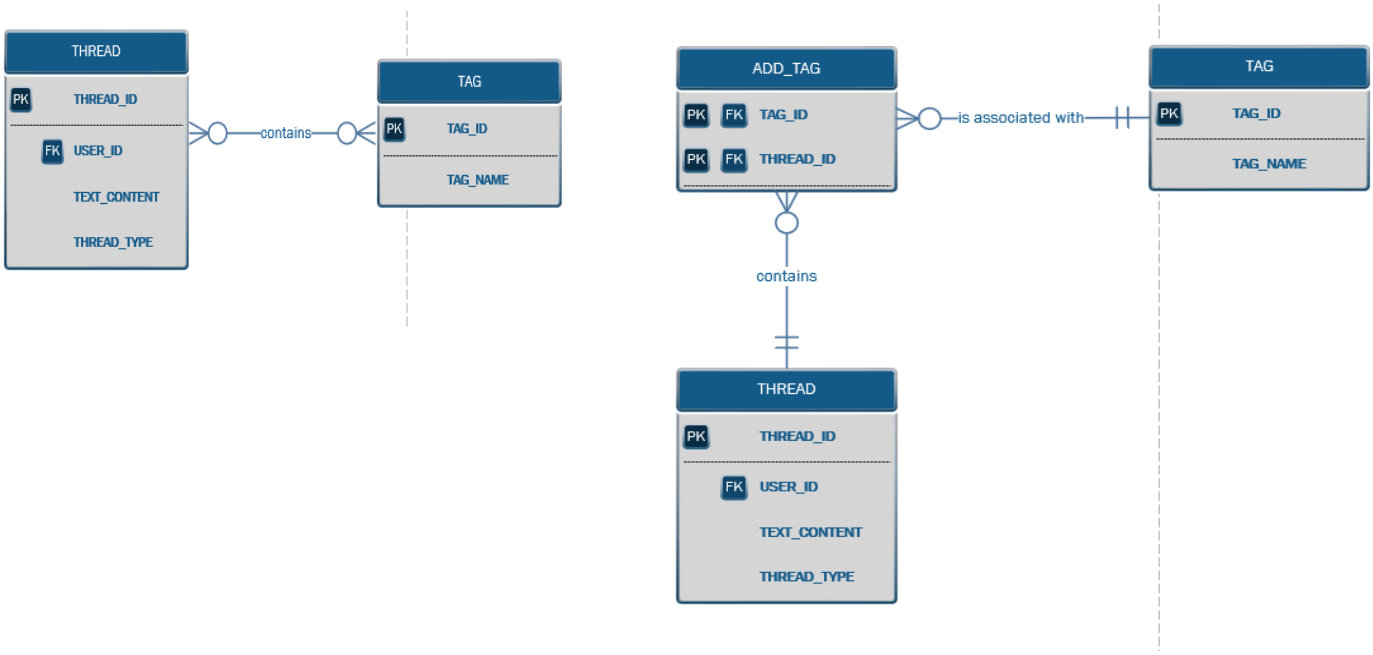
In addition, a picture can be uniquely identified by (THREAD_ID, P_ADDRESS) attributes. Hence the combination will produce huge primary key. Hence P_ID is chosen as **primary key** for T_PICTURE table.

ERD seventh segment



REPLY and R_PICTURE follow similar **1:1 relationship** as THREAD and T_PICTURE. One **design approach** is to combine reply picture R_PICTURE and thread picture T_PICTURE table into single entity. This will result separate column for foreign key THREAD_ID and REPLY_ID. For reply picture, THREAD_ID will be null and REPLY_ID will be null for thread picture. This creates lot of **null values**. Hence thread and reply picture table is kept separate. Also, it is assumed that one picture can belong to only one reply.

ERD eight segment



Thread can contain multiple tags. Also, a tag belongs to many threads. Thus, the relationship between THREAD and TAG entity is **M:N**. This M:N relationship is decomposed into **two 1:M relationships** by inserting **a bridge table** ADD_TAG. A THREAD can contain none or multiple tags. Signifying 1:M relationship with many cardinality on ADD_TAG side. Also, it is possible for a TAG to be attached to none or multiple thread which denotes one on TAG side and many cardinality on ADD_TAG side. The bridge table ADD_TAG **inherits primary keys** of THREAD(thread_id) and TAG (TAG_ID) as **primary and foreign key.** Hence **strong relationship** exists between THREAD-ADD_TAG and ADD_TAG-TAG entities.

COMPLETE ER DIAGRAM