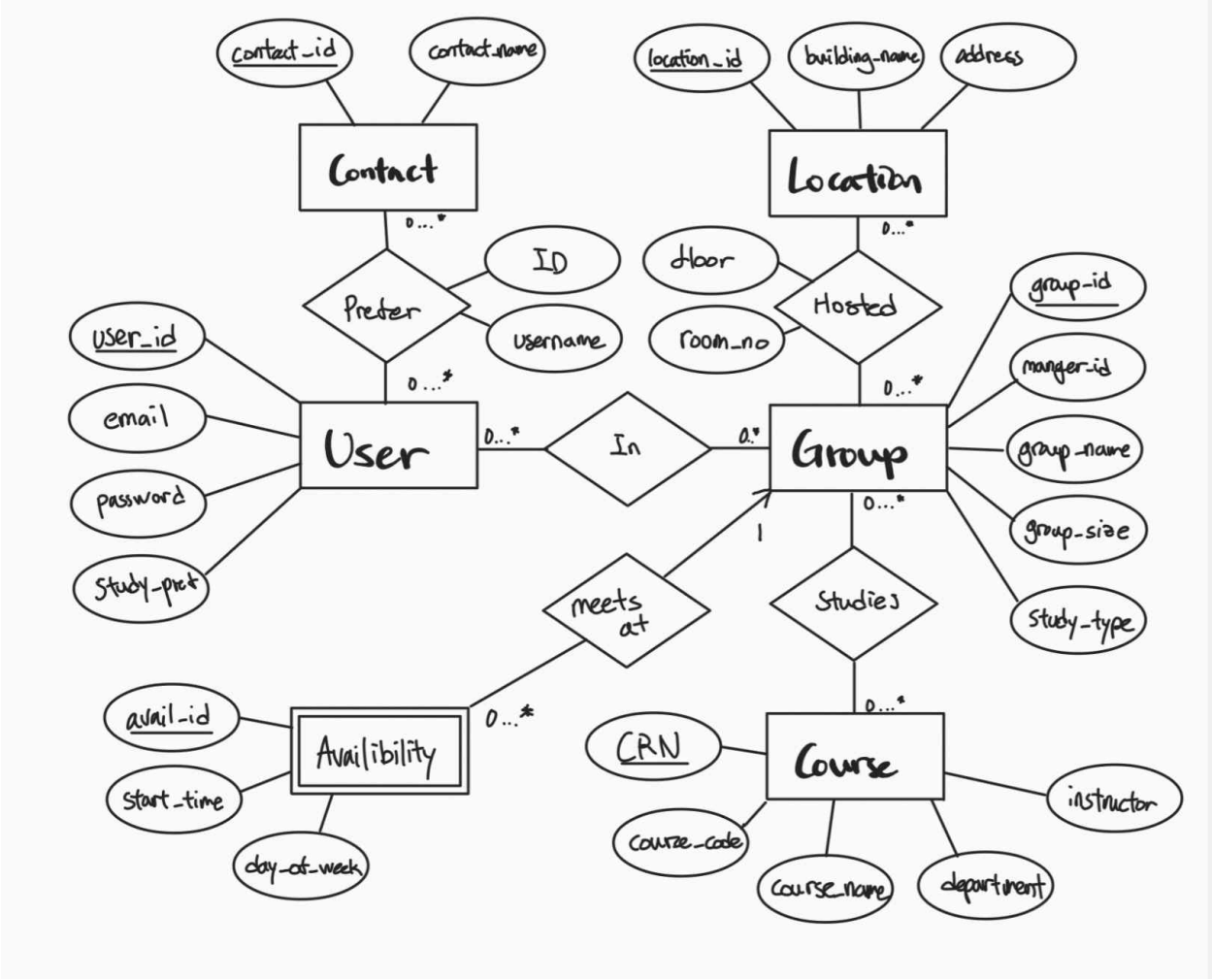
1+3. ER Diagram:



2. Entities and Relationships

Ideology:

All of the entities that we have are independent categories that are not dependent on other entities. Hence we use many-to-many relationships to describe how they relate to each other.

Entities:

1) User: Stores user profile data
2) Contact: methods of contact (social media, messaging services) the user uses, which helps other user contact the person to be added to the study group
3) Group: Formed study groups

4) Course: Stores university course data that users can filter by when creating or joining
5) Location: places to meet at for studying

Weak Entities

1) Availability: Stores availability of users to schedule group meetings.

Relationships/Cardinality:

- User and Contact: One user could have several forms of contacts, but one method of contact could have several users using it (Many-Many).
- User and Group: One User could be in multiple (study) Groups, but one Group could have many users as members (Many-Many).
- Group and Location: One group could meet up at multiple locations on campus, but one location could have several groups meeting there (Many-Many).
- Group and Course: One group could be studying for several courses, but one course could have several groups that study for it (Many-Many).
- Group and Availability: One group has a list of available times to meet at (One-Many).

4. Normalization (3NF):

group_id -> {(group attributes), user_id, location_id, course_id}
availability_id -> {(availability attributes), group_id}
user_id -> {(user_attributes), contact_id, group_id}
contact_id -> {(contact attributes), user_id}
course_id -> {(course_attributes), group_id}
location_id -> {(location_attribute), group_id}

Reference: U=User, CN=Contact, G=Group, CR=Course, L=Location, A=Availability
G -> {U, L, CR}
A -> {G}
U -> {CN, G}
CN -> {U}
CR -> {G}
L -> {G}

LEFT	MIDDLE	RIGHT	NONE
A	G,U,CN,CR,L		

1) Find minimal superkey

A+ = A, G, U, L, CR, CN

2) Compute set of minimal bases for FDs

2.1) RHS is Singleton (Yes)

2.2) Remove unnecessary attributes from LHS / Remove redundant FDs (using remaining other FDs)

without	Remains
G -> {U, L, CR}	G+ -> {G}
A -> {G}	A+ -> {A}
U -> {CN, G}	U+ -> {U}
CN -> {U}	CN+ -> {CN}
CR -> {G}	CR+ -> {CR}
L -> {G}	L+ -> {L}

- nothing to remove

3) Create a relation for every minimal basis FD

X(G,U,C,L,R); Y(A,G); Z(U,CN,G); T(CN, U); U(CR,G) V(L,G);

4) Add any of the candidate keys as a relation if we don't have a candidate key as a subset of any of the resulted relations in 3)

- There are no missing relations

So the minimal basis is X(G,U,C,L,R); Y(A,G); Z(U,CN,G); T(CN, U); U(CR,G) V(L,G);
which our ER diagram adheres to.

5. Relational Schema:

```
<User> (
    user_id INT [PK],
    email VARCHAR,
    password CHAR(100),
    study_pref CHAR(100)
)

<Group> (
    group_id INT [PK],
    manager_id INT [FK to User.user_id],
    group_name VARCHAR,
    group_size INT,
    study_type CHAR(100),
)

<Membership> (
    user_id INT [PK] [FK to User.user_id],
    group_id INT [PK] [FK to Group.group_id],
    role VARCHAR
)

<Availability> (
    avail_id INT [PK],
    start_time TIME,
    day_of_week DATE,
    group_id INT [FK to Group.group_id]
)
```

```
<Course> (
    CRN INT [PK],
    course_code CHAR(7),
    course_name VARCHAR,
    instructor VARCHAR,
    department VARCHAR,
)

<Group_Courses> (
    group_id INT [FK to Group.group_id],
    course_id INT [FK to Courses.course_id],
)

<Location> (
    location_id INT [PK],
    building_name VARCHAR,
    address VARCHAR,
)

<Group_Location> (
    group_id INT [FK to Group.group_id],
    location_id INT [FK to location_id],
    floor INT,
    room_no INT,
)

<Contact> (
    contact_id INT [PK],
    contact_name VARCHAR,
)
```

```
<User_Contact> (
    user_id INT [FK to User.user_id],
    contact_id INT [FK to Contact.contact_id],
    ID VARCHAR,
    user_name VARCHAR,
)
```