UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Foundations of Databases A.Y. 2022-2023
Homework 2 – Conceptual and Logical Design

Master Degree in Computer Engineering
Master Degree in Cybersecurity
Master Degree in ICT for Internet and Multimedia

Deadline: November 26, 2022

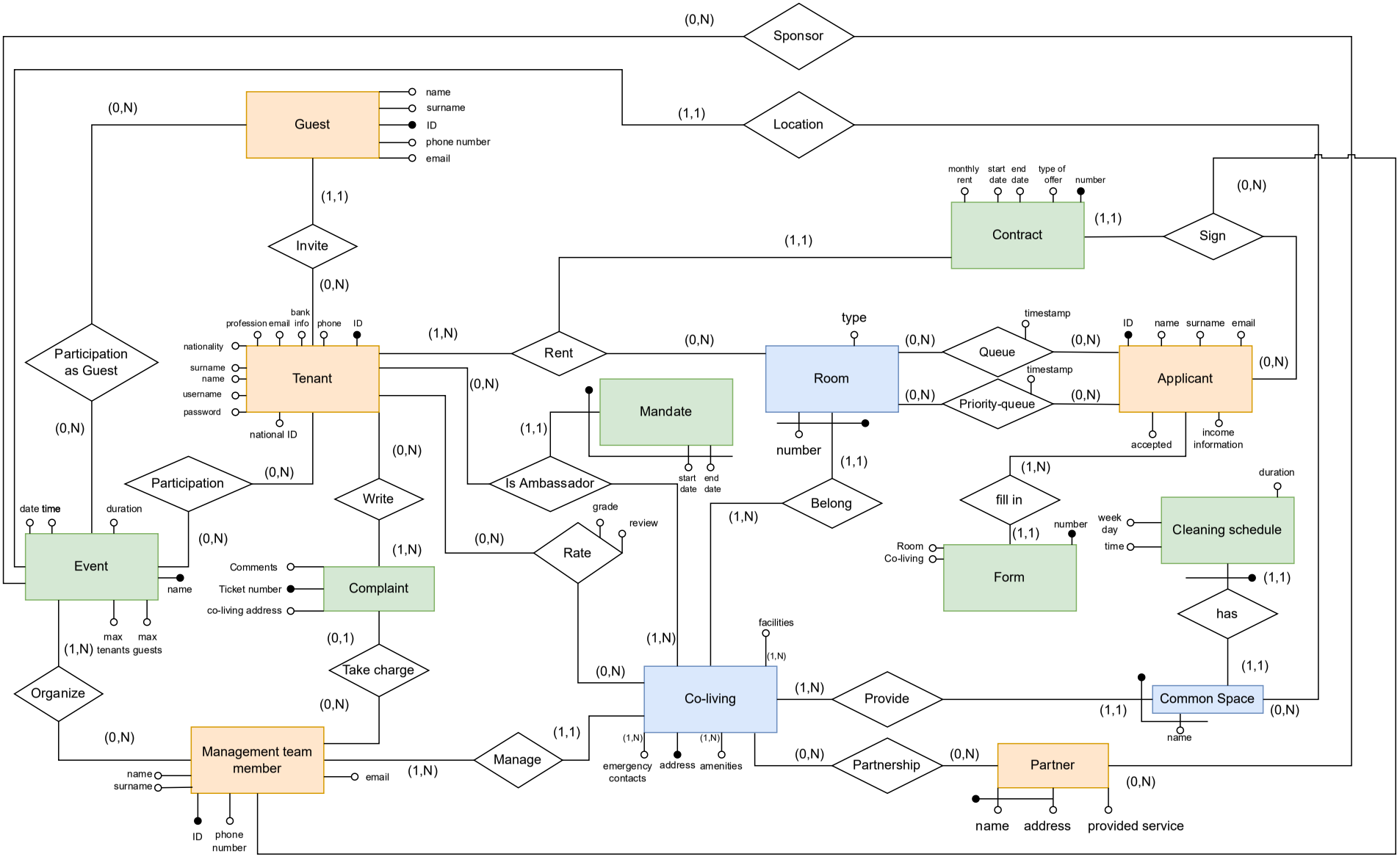
Team acronym	CLDB	
Last Name	**First Name**	**Student Number**
Zattarin	Nicole	2057138
Battut	Anaïs	2065759
Fragne	Anaïs	2073077
Zanin	Daria	2089062
Broetto	Riccardo	2088250
Gasparini	Marco	2087927
Gobbin	Alberto	1232223
D'Este	Luca	2082123
Sulku	Euxhenio	2080611
Ortiz de Zárate	Nicolás	2044778

Conceptual Design

Variations to the Requirement Analysis

- The tenant can no longer purchase on-demand services; companies that make a partnership with a co-living will offer their services to all its tenants.
- The management team doesn't take care anymore of emergency requests (the concept was not clearly defined and probably not relevant to the database).
- The amount of data that the applicant has to provide has been reduced.
- All the information about the tenant's stay are now part of the contract (more consistent and logical).
- The choice of the ambassador is done by a system external to the database, only the information about who is chosen and for how long he/she will be ambassador is stored.
- The number of rooms is no longer an attribute of the co-living (derived).
- The number of tenants per room has been set to one.
- Where necessary, the fiscal code has been replaced by the national ID, given the fact that it's not a standard in every country.
- The guarantors information are no longer required.
- The management team will no longer check rent payments and keep track of them, they will only be able to see the monthly amount for every tenant.

Entity-Relationship Schema


Sponsor
(0,N)
Guest
name
surname
ID
phone number
email
(0,N)
(1,1)
Location
(1,1)
Invite
(0,N)
(1,1)
monthly rent
start date
end date
type of offer
number
Contract
(1,1)
Sign
(0,N)
(1,1)
Participation as Guest
(0,N)
profession
email
bank info
phone
ID
nationality
surname
name
username
password
national ID
Tenant
(1,N)
Rent
(0,N)
type
Room
number
(0,N)
Queue
timestamp
(0,N)
ID
name
surname
email
Applicant
accepted
income information
(0,N)
(0,N)
Priority-queue
timestamp
(0,N)
(0,N)
Participation
(0,N)
(0,N)
Write
(1,1)
Mandate
start date
end date
Is Ambassador
(1,1)
Belong
(1,N)
(1,N)
fill in
(1,1)
number
Room
Co-living
Form
duration
week day
time
Cleaning schedule
(1,1)
has
(1,1)
date time
duration
Event
name
max tenants
max guests
(1,N)
Organize
(0,N)
Comments
Ticket number
co-living address
Complaint
(1,N)
(0,1)
Take charge
(0,N)
(0,N)
Rate
grade
review
(0,N)
facilities
(1,N)
Co-living
(1,N)
Provide
(1,1)
Common Space
(0,N)
name
(0,N)
Management team member
name
surname
ID
phone number
email
(1,N)
Manage
(1,1)
(1,N)
emergency contacts
address
(1,N)
amenities
(0,N)
Partnership
(0,N)
Partner
(0,N)
name
address
provided service

Data Dictionary

Entities Table

Entity	Description	Attributes	ID
Co-living	The living space in which tenants live.	• Address: The street address of the co-living. (text) • Amenities: The list of amenities available in the co-living (text) • Emergency contacts: List of contact information, including name and phone number, of the assigned management team members to contact in case of emergency, or for everyday issues with the co-living. (text) • Facilities: List of facilities available in the co-living (e.g. gym, bar...). (text)	Address

Tenant	The individual who, following a written contract, rents a room in the co-living.	• ID: serial number that uniquely identifies the tenant (serial) • Name: The first name of the tenant (text) • Surname: The last name of the tenant (text) • Username: The username of the tenant (text) • Password: The password created by the tenant (text) • Email: The email used to contact the tenant (text) • Phone: The phone number of the tenant (text) • Profession: The profession of the tenant (text) • Nationality: The nationality of the tenant (text) • Bank information: The bank information of the tenant (text) • National ID: The identification document of the tenant issued by their governing state (text)	ID
Room	The room in the co-living that a tenant rents, according to a signed contract	• Number: The number assigned to the room (integer) • Type: The type of room (i.e. if it is standard or premium) (text)	Number, co-living

Management Team Member	A member of the team of owners that own and manage the co-living.	• ID: the national document that uniquely identifies a management team member (text) • Name: The name of the management team member (text) • Surname: The surname of the management team member (text) • Phone number: The phone number of the management team member (text) • Email: The email of the management team member (text)	ID
Applicant	A prospective tenant of a co-living who has completed an application for a room.	• ID: The national ID that uniquely identifies an applicant (text) • Name: The first name of the applicant (text) • Surname: The last name of the applicant (text) • Accepted: The status of whether or not an applicant has been accepted (boolean) • Income Information: The income information of an applicant (text) • Email: The email used to contact the Applicant (text)	ID

Form	The form that applicants fill to apply for a room in a co-living.	• Number: The form number that identifies the form (serial) • Co-living: The co-living being applied to in the form (text) • Room: The room being applied to in the room (integer)	Number
Contract	The written contract, signed by both the tenant and the management team, containing the terms and conditions for renting a room in the co-living.	• Number: The number that uniquely identifies that contract (serial) • Type of offer: The type of contract that is offered to the tenant (text) • Start date: The start date of the contract (date) • End date: The end date of the contract (date) • Monthly rent: The monthly sum of money due by the tenant to the management team (float)	Number
Mandate	The tasks and responsibilities for an ambassador assigned to a co-living.	• Start date: The start date of a tenant being an ambassador (date) • End date: The end date of a tenant being an ambassador (date)	Start date, end date, co-living

Event	An event hosted at a co-living that tenants and their guests can attend.	• Name: The name that identifies the event (text) • Max guests: The maximum amount of allowed guests (integer) • Max tenants: The maximum amount of allowed tenants (integer) • Date: The date of the event (date) • Time: The time of the event (time) • Duration: duration of the event in minutes (integer)	Name
Guest	People invited by tenants of a co-living to events.	• ID: The national document that uniquely identifies a guest (text) • Name: The first name of the guest (text) • Surname: The last name of the guest (text) • Phone Number: The phone number of the guest (text) • Email: The email of the guest (text)	ID
Common space	The common areas of a co-living that are used by all tenants.	• Name: The name of the common space (text)	Name, co-living

Cleaning Schedule	The schedule for each co-living that includes when cleaning sessions occur and at what time.	• Week day: The days of the week in the cleaning schedule (text) • Time: The times in the cleaning schedule (time) • Duration: The duration of time it takes for a cleaning session in minutes (integer)	Common space
Partner	Local establishments that may enter partnerships with a co-living, such as sponsoring events or providing food catering.	• Name: The name of the partner (text) • Address: The address of the business where the partner is located (text) • Provided service: The service that the partner provides for the event (text)	Name, address
Complaint	A written message sent to the management team containing any complaints in regards to the co-living.	• Ticket number: The unique number that identifies a ticket (serial) • Comments: The written comments containing the complaint (text) • Co-living address: The street address of the co-living (text)	Ticket number

Relationships Table

Relationship	Description	Component Entities	Attributes
Sponsor	Associates a partner to a partnered event	• Partner (0,N) → a partner can sponsor multiple events • Event (0,N)→ an event can be sponsored by multiple partners	-
Location	Associates an event to the common space it takes place in	• Event (1,1) → each event has one and only one location • Common space (0,N) → a common space could host multiple events at different times	-
Invite	Associates a tenant with a guest he invites to an event	• Tenant (0,N) → a tenant can invite multiple guests • Guest (1,1) → a guest can be invited by one and only one tenant for each event	-
Sign	Describes the action of signing a contract, performed by an applicant and a management team member	• Applicant (0,N) → an applicant could sign multiple contracts in different, non overlapping, time frames • Contract (1,1) → each contract has to be signed by one and only one applicant • Management team member (0,N) → a member of the team can sign multiple contracts, but since there are multiple members for each team, it is not mandatory that all members sign all contracts	-

Participation as guest	Associates a guest to an event he/she takes part in	• Guest (0,N) → a guest can participate to multiple events • Event (0,N) → an event can have multiple participating guests, up to a max number of guests	-
Participation	Relates an event to tenants taking part in it	• Event (0,N) → an event can have multiple participating tenants, up to a maximum of max tenants • Tenant (0,N) → a tenant can participate to multiple events	-
Write	Relates a complaint to tenants who have written it	• Tenant (0,N) → a tenant can write multiple complaints • Complaint (1,N) → a complaint must be submitted by at least a tenant, but it could be written by multiple tenants	-
Rent	Relates each tenant to the room he/she lives in and to the contract he/she has signed	• Tenant (1,N) → a tenant in order to be a tenant must rent at least a room, but he could rent multiple rooms in non overlapping periods of time • Room (0,N) → a room is rent by multiple tenants as time goes on • Contract (1,1) → each contract must refer to one and one only pair room and tenant	-

Is ambassador	Associates each co-living to one ambassador and the information about his/her mandate	• Tenant (0,N) → a tenant can be ambassador of multiple co-livings as time goes on, • Co-living (1,N) → the co-living, in non overlapping time frames, is going to have multiple ambassadors, at least one (the first) • Mandate (1,1) → each mandate refers to one and only one ambassador relationship	-
Queue	Relates an applicant to the room he has applied for	• Applicant (0,N) → applicant don't have to participate to the queue if they are in the priority queue, but they still can be added to multiple queues (for different rooms) • Room (0,N) → a room can have multiple applicant in the same queue	Timestamp: time indicator stating when the applicant entered the queue, Date
Priority queue	Relates an applicant who has already been tenant to the room he has applied for	• Applicant (0,N) → applicant don't have to participate to the queue if they are in the queue, but they still can be added to multiple queues (for different rooms) • Room (0,N)→ a room can have multiple applicant in the same priority queue	Timestamp: time indicator stating when the applicant entered the priority queue, Date

Fill in	Describes the action of signing a form by an applicant who wants to rent a room in a co-living	• Applicant (1,N) → an applicant, in order to be recognized as an applicant, must fill at least a form, but they can fill multiple forms • Form (1,1) → a form refers to one and only one applicant	-
Rate	Allows a tenant to review a co-living, with a comment and a grade	• Tenant (0,N) → a tenant can rate multiple co-livings (in which he rent a room in different times) • Co-living (0,N) → a co-living can be rated by multiple tenants	• Grade: mark summarizing the opinion of a tenant about a co-living, int • Review: comment written by a tenant on a co-living, Text
Belong	Associates each room to the co-living it belongs to	• Co-living (1,N) → a co-living has at least one room, more probably it has many rooms • Room (1,1) → a room belongs to one and only one co-living	-
Organize	Associates a management team member to an event he/she has organized	• Management team member (0,N) → a team member can organize multiple events • Event (1,N) → an event must be organized by at least a member of the team, possibly more	-

Take charge	Associates a management team member to a complaint he/she takes charge of	• Management team member (0,N) → a member of the team can take charge of multiple complaints • Complaint (0,1) → a complaint can be taken in charge by at most one member of the team	-
Manage	Associates a management team member to a co-living he/she runs	• Management team member (1,N) → a member of the team must manage at least a co-living, but also more • Co-living (1, 1) → each co-living is assigned to a single manager	-
Provide	Associates a co-living to a common space it provides in its building	• Co-living (1,N) → each co-living provides at least a common space, more commonly more • Common space (1,1) → a common-space belongs to one and only one co-living	-
Partnership	Relates a partner of a co-living to the co-living itself	• Co-living (0,N) → a co-living can have multiple partners, but it can also not provide any partnership • Partner (0,N) → a partner can have multiple partnership with different co-livings. each	-

Has	Relates a common space to its cleaning schedule	• Common space (1,1) → each common space has one single cleaning schedule • Cleaning schedule (1,1) → each cleaning schedule refers to one and only one common space	-

External Constraints

- The tenants can only rate the co-living in which they are staying.
- There cannot be two tenants with the same username.
- The applicants join a queue only after they have been accepted by the management team.
- The tenants can write complaints only regarding the co-living in which they are staying..
- Applicants that have already rented a room in the past have priority on new applicants and are placed in the priority queue relation instead of in the queue relation.
- There cannot be two or more valid contracts for the same room.
- There cannot be two or more valid contracts for a tenant.
- The queue and priority queue have to be updated every time an applicant who is in the queue becomes a tenant.
- In a common space only one event per time can take place.
- The number of guests partecipating to an event must not exceed the number stored in the max guests attribute; in case the number is reached no more guests can enter the event. The same applies to tenants considering the max tenants attribute.
- Two ambassadors for the same co-living cannot have an overlapping mandate, meaning that the start date of the second comes after the end date of the first.

Functional Requirements Satisfaction Check

The database must store:

- Buildings / Co-living data including::
  - Facilities (gym, pool, co-working space)
  - Address of the building
  - Amenities available (tv, kitchen robot. . .)

  - Emergency contact with the name of the management team member and their phone number
  - Common spaces (number of kitchens, rooms, living room, terrace, garden. . .) with tables indicating cleaning schedule
  - List of the number of rooms, each attached to a tenant and a contract that, specify the duration of the stay and the monthly rent.
  - The ambassador name
  - The management team member assigned to it
  - List of events organized by the team member attached to the building
  - Partners

  The first of the information about the building are stored as attributes of the co-living entity, in particular: facilities, address, amenities and emergency contacts are attributes of the co-living entity.
  "Common spaces" is an entity itself, linked to the co-living through the relationship "provide". Each common space has its own cleaning schedule and can be the location of some events thanks to the "location" relationship between commonspace and event entities.
  The co-living has a link "belong" with rooms, indeed each co-living building is characterized by a certain number of rooms.
  Moreover, for each co-living, there is an ambassador, who must be a tenant of that same building. The latter is related to it through the " is ambassador" relationship. The process of assigning the ambassador among different tenants is represented by the mandate entity, which participates to the relationship "is ambassador". The duration of the mandate is defined by the start/end dates attributes, which play the role of identifiers.
  Additionally, a tenant can report problems concerning each co-living. Those problems are represented as a rating given by each tenant, who provides a specific grade to the co-living. Grades and reviews are modeled as attributes of the rate relationship.
  Furthermore, for each co-living building, managers from the management team are assigned (through the "manage" relationship) and will give their contacts for the emergency contact.
  Those team management members can organize events in each building. Finally, buildings could also have partners (companies, bars, clubs, etc..), which may contribute to the organization of events. Those events are modeled as an entity itself, which is characterized by the attributes max tenants, max guests, date and name. Events can also be hosted in a common space, which is modeled by the location relationship between the commonspace and event entities.

- Rooms data including:

  - Number of the room
  - Type of the room (bathroom shared or not) $\rightarrow$ 2 types
  - The building in which it is in
  - The current tenant assigned to the room
  - The current contract with the monthly rent, the start and end date
  - Occupied or vacant (known by the dates of the contract)
  - A waitlist of the applicants who pre-booked the room (the priority list and the normal queue)

Each room is an entity, whose attributes are: its number and its type (if it provides a private bathroom or a shared one).
Each room is linked to its co-living by means of the "belong" relationship. Then, a room is assigned to a current tenant occupying the room by means of the "rent" relationship, and to the contract entity, which is signed by applicants and the management team member of the building. We define the contract entity in order to store all the details regarding the stay of a tenant and to connect the applicant to the corresponding room. Indeed, each contract entity has the following attributes: number, type of offer, start/end dates, monthly rent, applicant ID. Contracts participate to the rent relationship together with the tenant and the room. This allows for the retrieval of all the information regarding the stay when a room is occupied.

Each room is attached to two waitlists relationships (the priority one and the classic one) linked to applicants. Indeed, if an applicant never booked a room in that co-living, once they apply, they are added to the classic queue, with the corresponding position (i.e. attribute "timestamp" of the queue relationship). On the other hand, we want previous tenants to be prioritized in the application process, therefore, if a previous tenant applies once again for a room, they are an applicant added to a priority queue, with the corresponding position (i.e. attribute "timestamp" of the "priority-queue" relationship).

- Tenant data including:
  - Name and surname
  - Username and password
  - E-mail
  - Nationality
  - Phone number
  - Profession: student (degree) or professionals
  - Bank information
  - Id /passport
  - Link to the room that is attached to a contract (which allows to know the deadline of payment for each month, the duration of the stay)
  - If they are an ambassador or not
  - Their rating and reviews on the co-living
  - Their complaints
  - Their participation to some events
  - Guests that they can invite for events or not

  Tenants are entities with the following attributes: name, surname, username, password, e-mail, profession, phone number, ID, nationality, bank information.
  The information about the contract and the stay are all stored in the corresponding contract, which is accessible through the rent relationship. As already pointed out the entity contract has attributes referring

to the start/end dates.
Plus, a tenant may be an ambassador if the relationship "is ambassador" holds between them and the co-living they live in.
Finally, a tenant can write complaints and send them to the management team member attached to the building. They can also participate in events and invite guests whenever they want but have to respect the maximum number of guests of the event.

The system must allow:

- **Future tenants (applicants) to fill out a form with basic personal information and take an appointment → pre-book.**
  This tool is modeled by means of the *form* entity, related to the applicant through the *fill in* relationship. Indeed, the applicant creates an account with its basic information (applicant attributes: ID, name, surname, income information attributes), and in order to be an actual applicant he must fill at least a form with the specification of the *room number* in a specific *co-living* (attributes of the form entity) he is interest into. Later, am applicant can either be accepted or not (accepted attribute of applicant). If accepted, he is added to the correct waitlist, i.e. the queues, with the corresponding timestamp attribute of acceptance.

- **Selected future tenants to book a room by sending the deposit and giving more personal details.**
  An applicant is selected as a future tenant when they are first in the queue/priority queue. To model this, we design a timestamp attribute with the timestamp of when the applicant is added to the queue. When an applicant is first in the queue, the next step is signing a contract. This is represented by means of the "sign" relationship. Once the contract is signed, the instance of the "queue" relationship is deleted, which allows checking which is the most recent timestamp in the queue, in order to select the next applicant to become a tenant.

- **The management team to know which rooms are free or occupied.**
  This information can be retrieved because managers participate in the "sign" relationship, so they have access to all of the information concerning the rent and stay.

- **The management team to choose the next tenant on the waitlist to accept.**
  This information is managed by means of the queue and priority-queue relationships, which store the position of each applicant on the waitlist for a specific co-living/room. Moreover, before being able to join a queue, an applicant must be accepted by the management team ("accepted" attribute).

- **The management team to check the information about each tenant.**
  This information is stored in each tenant entity and in particular it is accessible to the manager of the co-living through the contracts. Indeed, each contract signed by the management team member contains the ID of the applicant/tenant, which allows for the retrieval of all the information about the people living in the co-livings.

- **The management team to check if there is any problem reported in each building.**
  This information can be retrieved on two different levels: implicitly from co-living reviews and directly

from tenants' complaints. Indeed, each tenant entity is related to the corresponding co-living by means of a "rate" relationship, with each rating having "grade" and "review" attributes. Moreover, we design a "complaints" entity, which is related to the tenant by means of the relationship "write" and to the management team member through the "takes charge" relationship. Each complaint provides information in order to help the manager to identify possible issues in the co-living. The attributes are: comments, ticket number, co-living ID.

- **The management team to check when will a room be free.**
  This information can be easily retrieved by checking the start/end dates of each contract, recalling that each contract provides all the information concerning the stay.

- **The management team to handle all the payments and store them.**
  Information regarding the rent price and the dates it is supposed to be paid by each tenant (monthly from the start date) are stored in the contract, which is clearly accessible by members of the management since they participate in the sign relationship.

- **The management team to add or delete co-living spaces.**
  The information is included in the "manage" relationship between the members of the management team and the co-livings.

- **The tenant to log in and check their applications.**
  Each tenant has the "username" and "password" attributes, which allow them to log on the application.

- **The tenant to receive a notification when a room is ready for them or when they are the next one on the waitlist.**
  The information about the position on the waitlist is stored in the "timestamp" attribute in the queue or priority-queue relationships between the applicant and the room. When an applicant is deleted from the queue the corresponding instance is deleted and the position in the queue is fixed by the time of arrival.

- **The tenants to check reviews and grades for each building.**
  This information is stored, as time goes on, by the rating that can be provided by each entity tenant: the "rate" relationship between the co-living and the tenant is characterized by the "grade" and "review" attributes.

- **The tenants to see the next event organized for the building.**
  The tenant entity is linked to the event entity by means of the "participation" relationship. Indeed, to participate in a certain event the tenant can check the attributes of the event, which include date, name and the maximum number of participants and guests. The location is given as another relationship between the event and common spaces.

- **The tenants to check when cleaning has been done.**
  This information is stored in the "cleaning-schedule" entity, which is in a relationship "has" with the

"common space" entity. Each common space is provided by each co-living entity. Checking the cleaning schedule attributes, i.e. weekday, duration, and time, tenants can easily verify if the cleaning has been done in that specific common space.

- **The tenants to invite guests.**
  This information is modeled in a straightforward way by the relationship "invite", between the tenant and the guest entities. Each tenant can invite many guests, but if they invite them for an event, they have to respect the constraints fixed by the maximum number of guests attribute of the event entity.

- **The potential applicants to filter the price, type of bathroom, and address of the rooms on the platform when searching for a room.**
  This information is stored in the attributes of the co-living and of each room. Indeed, applicants can have access to this information, which includes: co-living address, room type, and rent.

- **The tenants to check partnerships with the local suppliers and structures.**
  This information can be found on two different levels: as a detail regarding the co-living itself and through events. Indeed, "tenants" are entities, connected to the "event" entity by means of the "participation" relationship, and events may be sponsored by partners of the co-living. Moreover, each co-living is an entity which is in a "partnership" relationship with the "partners" and "tenants" can have access to this information regarding the co-living their room belongs to.

- **The ambassador to represent all the other tenants in case of issues.**
  The ambassador is represented as a relationship between a tenant and the co-living: for each co-living there must be one and one only tenant related to it through the ambassador relationship. A co-living cannot be vacant of an ambassador.

Logical Design

Transformation of the Entity-Relationship Schema

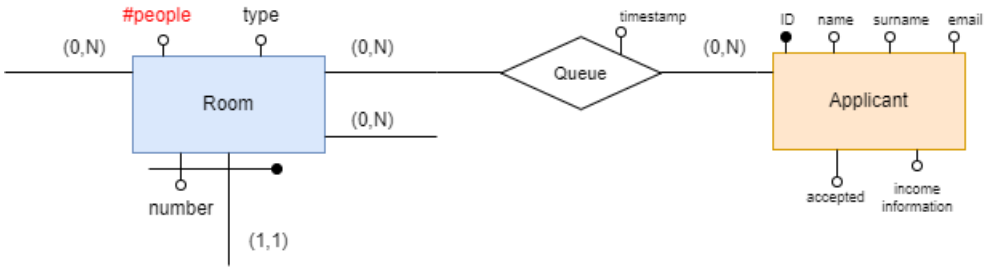
Redundancy Analysis

The schema does not contain neither cycles of entities, nor derived attributes as it can be retrieved from the Analysis of Database Load section. The only redundancy refers to the tenants and applicant entities, but it has been designed in order to maintain the system safer and for a matter of clearness.

Choice of Principal Identifiers

The schema does not contain external identification cycles and the main identifiers comply with the selection criteria. The main exception consists of the use of the ID serial number, nevertheless such choice has been made in order to keep our system safe.

Analysis of Database Load



We compute the load analysis to decide whether to keep or remove the #people in queue derived attribute from the *Room* entity.

Consider the following three operations that involve the redundant attribute #people:

Operation	**Description**	**Frequency**	**Type**
O1: Insert an applicant	Insert an applicant including the queues in which he/she is added	50/week	Online
O2: Print data with the number of people in queue	Print how many people are present in a queue for a room	7/week	Online
O3: Remove an applicant	Remove an applicant also from the queues in which he/she was present	50/week	Online

Table 4: Operations description and frequency

O1, O2, and O3 are online operations because they have to be executed interactively to satisfy a request issued by a user. We consider that an applicant usually applies for 3 rooms in a Co-living. On average, there are 15 applicants waiting in queue for a room.

In table 5, we report the access/volume data related to O1 **with redundancy**. We need to add a new applicant to each queue for the *Room* he/she wants to apply for. We consider that an applicant chooses 3 rooms when applying for a Co-living. Therefore, the *Room* entity has three read accesses to get the current value of #people waiting in queue for each room. We also have three write accesses to update #people waiting in queue for the three rooms.

Operation O1: 50/week				
Concept	**Construct**	**Access**	**Type**	**Average Access**
Applicant	Entity	3	W	3 x 50 x 2 = 300
Queue	Relationship	3	W	3 x 50 x 2 = 300
Room	Entity	3	R	3 x 50 x 1 = 150
Room	Entity	3	W	3 x 50 x 2 = 300
Total Access				1050

Table 5: Access/volume Table for Operation 1 with redundancy

In table 6, we report the access/volume data related to O2 **with redundancy**. Because we have an attribute #people, we only need to read the entity *Room* to print data with the number of people in queue.

Operation O2: 7/week				
Concept	**Construct**	**Access**	**Type**	**Average Access**
Room	Entity	1	R	1 x 7 x 1 = 7
Total Access				7

Table 6: Access/volume Table for Operation 2 with redundancy

In table 7, we report the access/volume data related to O3 **with redundancy**. Just like the access/volume data related to O1, the *Room* entity has 3 read accesses to get #people and 3 write accesses to update this value. The access is equal to 3 because we need to delete the applicant from the 3 rooms where he/she was in queue.

Operation O3: 50/week				
Concept	**Construct**	**Access**	**Type**	**Average Access**
Applicant	Entity	3	W	3 x 50 x 2 = 300
Queue	Relationship	3	W	3 x 50 x 2 = 300
Room	Entity	3	R	3 x 50 x 1 = 150
Room	Entity	3	W	3 x 50 x 2 = 300
Total Access				1050

Table 7: Access/volume Table for Operation 3 with redundancy

In table 8, we report the access/volume data related to O1 **without redundancy**. In this case, we need three write accesses to insert these new instances in *Applicant* and three write accesses to insert the new instances in *Queue* which links the Applicant to the rooms he/she is in queue for.

Operation O1: 50/week				
Concept	**Construct**	**Access**	**Type**	**Average Access**
Applicant	Entity	3	W	3 x 50 x 2 = 300
Queue	Relationship	3	W	3 x 50 x 2 = 300
Total Access				600

Table 8: Access/volume Table for Operation 1 without redundancy

In table 9, we report the access/volume data related to O2 **without redundancy**. To print data with the number of people in queue, there must be a read access to the *Room*. We also need to look at the *Queue* relationship which needs 15 read accesses to get the number of people in queue.

Operation O2: 7/week				
Concept	**Construct**	**Access**	**Type**	**Average Access**
Room	Entity	1	R	1 x 7 x 1 = 7
Queue	Relationship	15	R	15 x 7 x 1 = 105
Total Access				112

Table 9: Access/volume Table for Operation 2 without redundancy

In table 10, we report the access/volume data related to O3 **without redundancy**. On average, an *Applicant*

is in 3 queues for 3 different rooms. To delete the applicant, there must be a three write accesses to delete the instance from the entity *Applicant* and from the *Queue* relationship.

Operation O3: 50/week				
Concept	**Construct**	**Access**	**Type**	**Average Access**
Applicant	Entity	3	W	3 x 50 x 2 = 300
Queue	Relationship	3	W	3 x 50 x 2 = 300
Total Access				600

Table 10: Access/volume Table for Operation 3 without redundancy

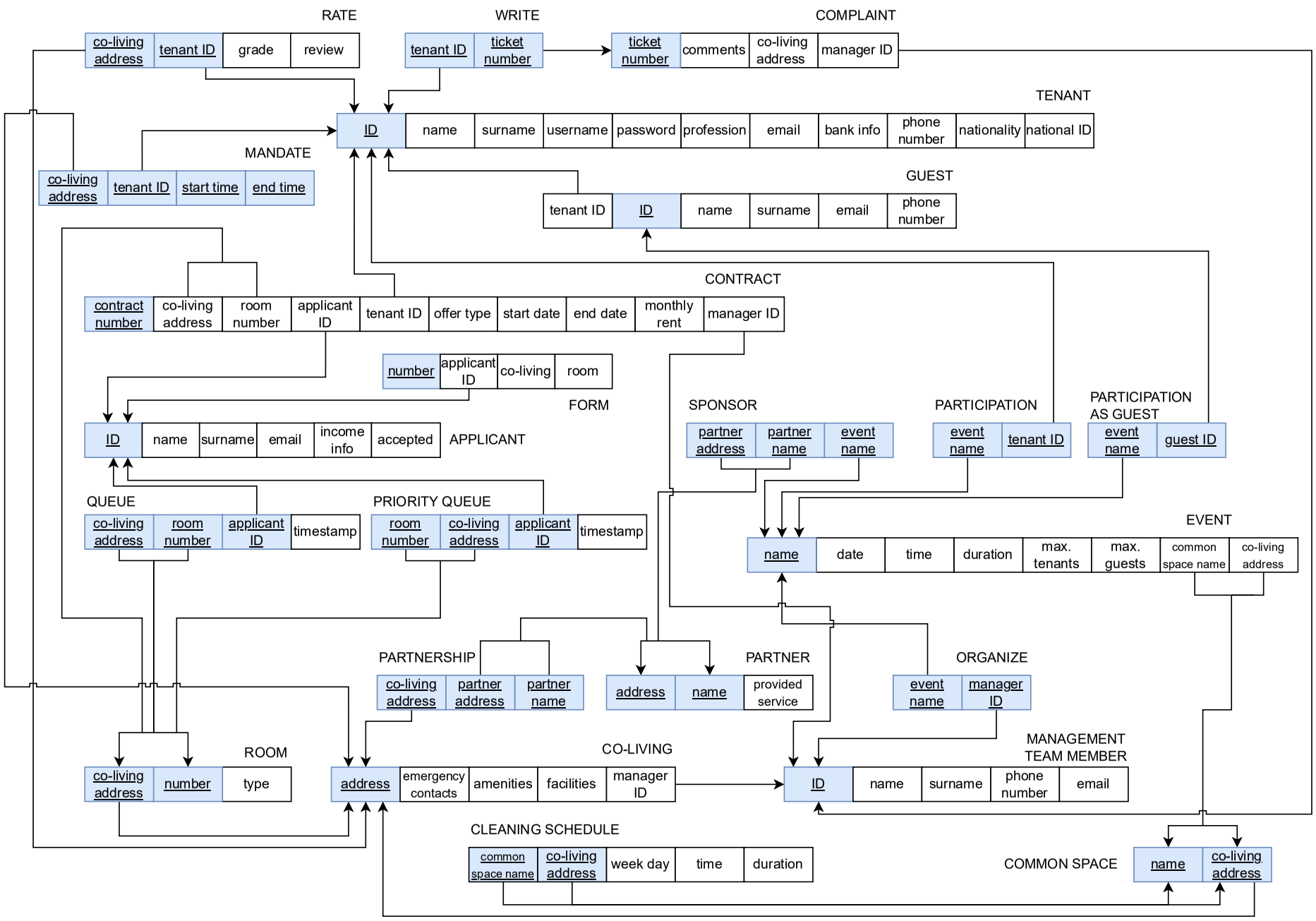
Comparison		
Operation	**With redundancy**	**Without redundancy**
O1	1050	600
O2	7	112
O3	1050	600
Total Accesses/week	**2107**	**1312**

Table 11: Comparison of the number of accesses for each operation

The #people attribute does not improve the operations performance. Indeed, the number of Accesses per week is equal to 1312 without redundancy against 2107 with redundancy. Thanks to the load analysis, we decided not to keep the derivative redundant attribute in our schema.

We can have the same reflection and conclusion for the redundant attribute #people that are present in the priority queue.

Relational Schema


RATE
co-living address
tenant ID
grade
review
WRITE
tenant ID
ticket number
COMPLAINT
ticket number
comments
co-living address
manager ID
TENANT
ID
name
surname
username
password
profession
email
bank info
phone number
nationality
national ID
MANDATE
co-living address
tenant ID
start time
end time
GUEST
tenant ID
ID
name
surname
email
phone number
CONTRACT
contract number
co-living address
room number
applicant ID
tenant ID
offer type
start date
end date
monthly rent
manager ID
number
applicant ID
co-living
room
FORM
ID
name
surname
email
income info
accepted
APPLICANT
SPONSOR
partner address
partner name
event name
PARTICIPATION
event name
tenant ID
PARTICIPATION AS GUEST
event name
guest ID
QUEUE
co-living address
room number
applicant ID
timestamp
PRIORITY QUEUE
room number
co-living address
applicant ID
timestamp
EVENT
name
date
time
duration
max. tenants
max. guests
common space name
co-living address
PARTNERSHIP
co-living address
partner address
partner name
PARTNER
address
name
provided service
ORGANIZE
event name
manager ID
ROOM
co-living address
number
type
CO-LIVING
address
emergency contacts
amenities
facilities
manager ID
MANAGEMENT TEAM MEMBER
ID
name
surname
phone number
email
CLEANING SCHEDULE
common space name
co-living address
week day
time
duration
COMMON SPACE
name
co-living address

Data Dictionary

Relation	Attribute	Description	Domain	Constraints
Co-living	Address	The street address of the co-living	text	Primary Key
	Amenities	The list of amenities available in the co-living	text	
	Emergency contacts	List of contact information, including name and phone number, of the assigned management team members to contact in case of emergency, or for everyday issues with the co-living.	text	
	Facilities	List of facilities available in the co-living	text	
	Manager Id	The identification that uniquely identifies a management team member	text	Foreign Key to Management Team Member
Tenant	ID	ID that uniquely identifies the tenant	serial	Primary Key
	Name	The first name of the tenant	text	Not Null
	Surname	The last name of the tenant	text	Not Null
	Username	The username of the tenant	text	Not Null
	Password	The password created by the tenant	text	Not Null
	Email	the email used to contact the tenant	text	Not Null
	Phone number	The phone number of the tenant	integer	Not Null
	Profession	The profession of the tenant	text	
	Nationality	The nationality of the tenant	text	
	Bank info	The bank information of the tenant	text	Not Null
	National ID	The unique identifier issued by the government to track their citizens	text	Not Null
Room	Number	The number assigned to the room	integer	Primary Key with Co-Living Address
	Co-living address	The street address of the co-living which hosts the room	text	Foreign Key to Co-living, Primary Key with Number
	Type	The type of the room	text	Not Null
Management Team Member	ID	The identification that uniquely identifies a management team member	serial	Primary Key
	Name	The name of the management team member	text	Not Null
	Surname	The surname of the management team member	text	Not Null

	Phone number	The phone number of the management team member	integer	Not Null
	Email	The email of the management team member	text	Not Null
Applicant	ID	The national ID that uniquely identifies an applicant	text	Primary Key
	Name	The first name of the applicant	text	Not Null
	Surname	The last name of the applicant	text	Not Null
	Accepted	Status of whether or not an applicant has been accepted	boolean	Not Null
	Email	The email of the applicant	text	Not Null
	Income Information	The income information of an applicant	text	Not Null
Form	Number	The number that identifies the form	integer	Primary Key
	Co-living	The co-living being applied to in the form	text	Not Null
	Applicant ID	The identification that uniquely identifies the applicant who is filling the form t	text	Foreign Key to Applicant, Not Null
	Room	The room being applied to in the form	integer	
Rate	Co-Living address	The street address of the co-living which is being rated by the tenant	text	Foreign Key to Co-Living, Primary Key with tenant ID
	Grade	the score given by the tenants to rate the co-living where they are living	integer	
	Tenant ID	The ID that uniquely identifies the tenant who is rating the co-living	serial	Foreign Key to Tenant, Primary Key with Co-living address
	Review	Opinions left by tenants regarding their experience in the co-living	text	
Write	Tenant ID	The ID that uniquely identifies the tenant who has written the complaint	serial	Foreign Key to Tenant, Primary Key with Ticket Number
	Ticket Number	The number that uniquely identifies the ticket	integer	Primary Key with Tenant ID
Organize	Manager ID	The identification that uniquely identifies the management team member who is responsible for the organization of the event	serial	Foreign Key to Manager, Primary Key with Event Name
	Event name	The name of the event that is being organized	text	Foreign Key to Event, Primary Key with Manager ID

Partnership	Partner Name	The name of the partner who is part of the partnership	text	Foreign Key to Partner, Primary Key with Partner Address and Co-Living address
	Partner Address	The address of the business of the partner who is part of the partnership	text	Foreign Key to Partner, Primary Key with Partner Name and Co-Living address
	Co-living address	The street address of the co-living which is part of the partnership	text	Foreign Key to Co-Living, Primary Key with Partner Name and Partner Address
Sponsor	Partner Name	The name of the partner who is sponsoring the event	text	Foreign Key to Partner, Primary Key with Partner Address and Event Name
	Partner Address	The address of the business of the partner who is sponsoring the event	text	Foreign Key to Partner, Primary Key with Partner Name and Event Name
	Event name	The name of the event which is being sponsored by the partner	text	Foreign Key to Event, Primary Key with Partner Name and Partner Address
Participation	Tenant ID	The identification that uniquely identifies the tenant who has participated in the event	serial	Foreign Key to Tenant, Not Null
	Event name	The name of the event that the tenant has attended	text	Foreign Key to Event
Participation as guest	Guest ID	The identification that uniquely identifies the guest who has participated in the event	serial	Foreign Key to Guest, Primary Key with Event Name
	Event name	The name of the event that the guest has attended	text	Foreign Key to Event, Primary Key with Guest ID
Contract	Contract Number	The number that uniquely identifies that contract	integer	Primary Key
	Co-Living address	The street address of the co-living in which the applicant will agree to stay	text	Foreign Key to Room
	Room Number	The number assigned to the room that the applicant will rent	integer	Foreign Key to Room
	Offer Type	The type of contract that is offered	text	
	Start date	The start date of the contract	date	Not Null
	End date	The end date of the contract	date	Not Null

	Monthly rent	The monthly sum of money due by the tenant to the management team	float	Not Null
	Applicant ID	The identification that uniquely identifies the applicant that is signing the contract	text	Foreign Key to Applicant
	Tenant ID	The ID that uniquely identifies the tenant	serial	Foreign Key to Tenant
	Manager ID	The identification that uniquely identifies the management team member who has signed the contract	serial	Foreign Key to Manager
Mandate	Tenant Id	The ID that uniquely identifies the tenant who is the ambassador	serial	Foreign Key to Tenant, Primary Key with Co-Living address, start time and end time
	Co-Living address	The street address of the co-living in which the ambassador is in charge	text	Foreign Key to Co-Living, Primary Key with tenant ID, start time and end time
	Start time	The start time of a tenant being an ambassador	date	Primary Key with Co-Living address, tenant ID and end time
	End time	The end time of a tenant being an ambassador	date	Primary Key with Co-Living address, tenant ID and start time
Queue	Applicant ID	The ID that uniquely identifies the applicant who is in queue	text	Foreign Key to Applicant, Primary Key with Co-Living address and Room number
	Co-Living address	The street address of the co-living that the applicant is staying in the queue for	text	Foreign Key to Room, Primary Key with Applicant ID and Room Number
	Room number	The number assigned to the room that the applicant is staying in the queue for	integer	Foreign Key to Room, Primary Key with applicant ID and Co-Living address
	Timestamp	Time in which the applicant has joined the queue	date	Not Null
Priority Queue	Applicant Id	The ID that uniquely identifies the applicant who is in priority queue	text	Foreign Key to Applicant, Primary Key with Co-Living address and Room Number

	Co-Living address	The street address of the co-living that the applicant is staying in the priority queue for	text	Foreign Key to Room, Primary Key with Applicant ID and Room Number
	Room number	The number assigned to the room that the applicant is staying in the priority queue for	integer	Foreign Key to Room, Primary Key with applicant ID and Co-Living address
	Timestamp	Time in which the applicant has joined the priority queue	date	Not Null
Event	Name	The name that identifies the event	text	Primary Key
	Max guests	The maximum amount of allowed guests	integer	
	Max tenants	The maximum amount of allowed tenants	integer	
	Date	The date of the event	date	Not Null
	Duration	The duration time of the event	time	
	Time	The time when the event is hosted	time	
	Common Space name	The name of the common space where the event is hosted	text	Foreign Key to Common Space, Not Null
	Co-Living address	The street address of the co-living which hosts the event	text	Foreign Key to Common Space, Not Null
Guest	ID	The identification that uniquely identifies a guest	serial	Primary Key
	Name	The first name of the guest	text	Not Null
	Surname	The last name of the guest	text	Not Null
	Phone Number	The phone number of the guest	integer	Not Null
	Tenant ID	The ID that uniquely identifies the tenant that has invited the guest	serial	Foreign Key to Tenant, Not Null
	Email	The email of the guest	text	
Common Space	Name	The name of the common space	text	Primary Key with Co-Living address
	Co-Living address	The street address of the co-living which hosts the common space	text	Foreign Key to Co-Living, Primary Key with Name
Partner	Name	The name of the partner(s)	text	Primary Key with Address
	Address	The address of the business where the partner is located	text	Primary Key with Name
	Provided service	The service that the partner provides for the event	text	
Complaint	Ticket number	The unique number that identifies a ticket	integer	Foreign Key to Write, Primary Key
	Comments	The written comments containing the complaint	text	

	Co-living address	The street address of the co-living	text	
	Manager Id	The identification that uniquely identifies a management team member	serial	Foreign Key to Management Team Member
Cleaning Schedule	Week day	The days in the cleaning schedule	text	Not Null
	Time	The times in the cleaning schedule	time	Not Null
	Common Space name	The name of the common space that will be cleaned	text	Foreign Key to Common Space, Primary Key with Co-Living address
	Co-Living address	The street address of the co-living which hosts the common space	text	Foreign Key to Common Space, Primary Key with Common Space name
	Duration	The duration of time it takes for a cleaning session	time	

External Constraints

- The tenants can only rate the co-living in which they are staying: this means that before inserting a new tuple in *Rate* relation, we need to check that in *Contract* relation the tuple related to that tenant has the same Co-living address field.
- There cannot be two tenants with the same username: this means that before inserting a new tuple within *Tenant*, we need to check that the Username field is different from the one of any already existing *Tenant*.
- The applicants join a queue only after they have been accepted by the management team: this means that before a new tuple in *Queue* relation is created, we need to check that the target applicant has the Accepted field set to true.
- The tenants can write complaints only regarding the co-living in which they are staying: this means that for the tenant under consideration the Co-living address field in *Complaint* relation and the Co-living address field in *Contract* relation must be the same.
- Applicants that have already rented a room in the past have priority on new applicants and are placed in the priority queue relation instead of in the queue relation: this means that when an applicant applies for a room, we need to check whether there is an expired *Contract* with the Applicant ID field that matches the ID of the applicant under consideration.
- There cannot be two or more valid contracts for the same room: this means that before creating a new tuple within *Contract*, we need to check that the Room number field is different from the one of any already existing *Contract* that has the End date field still to expire.
- There cannot be two or more valid contracts for a tenant: this means that, after a tenant signs a contract and until its duration is not over, there cannot be a different contract ongoing with the same tenant. Thus, every time we insert a new tuple within *Contract*, we need to check that there does not exist a valid *Contract* with the same Tentant ID.

- The queue and priority queue have to be updated every time an applicant who is in the queue becomes a tenant: this means that after the Start date of a *Contract* has been reached we delete the *(priority)queue* tuple with Applicant ID field matching the Tenant National ID field of *Tenant* relation.
- In a common space only one event per time can take place: this means that before inserting a new tuple within *Event*, we need to check that another *Event* does not exist with the same Common space and Date fields.
- The number of guests partecipating to an event must not exceed the number stored in the max guests attribute; in case the number is reached no more guests can enter the event. The same applies to tenants considering the max tenants attribute. Thus, before inserting a new tuple in *Participation as guest* relation, we need to check that the number of already existing tuples for the target *Event* do not exceed the Max guests field of *Event* relation. The same applies to *Participation* relation.
- Two ambassadors for the same co-living cannot have an overlapping mandate, meaning that the start date of the second comes after the end date of the first: this means that, after a Tenant starts a *Mandate* and until its duration is not over, there cannot be a different *Mandate* ongoing for the same Co-living. Thus, every time we insert a new tuple within *Mandate*, we need to check that there not exist a *Mandate* with overlapping End/Start dates.

Group Members Contribution

All members of the group contributed to the design and update of the ER schema. Each person took care of writing down the main body of a specific section, in particular:

- Entity table: Nicolas Ortiz de Zarate
- Relationship table: Marco Gasparini
- Functional requirements: Nicole Zattarin and Anaïs Fragne
- Analysis of database load: Luca D'Este and Anaïs Battut
- Relational schema: Alberto Gobbin
- Data dictionary: Euxhenio Sulku
- External constraints: Daria Zanin and Riccardo Broetto

Moreover, let us highlight that everybody put effort in reviewing the entire ER schema multiple times, with consequences on all sections. Therefore, each group member contributed in checking and eventually discussing possible solutions, also regarding sections that were not strictly the one they were supposed to take care of.