

URL to front-end pages: <http://flip1.engr.oregonstate.edu:3467/>

Reviews:

Review #1:

Hi Sankalp and Michael,

The overview describes the need for a relationship database by stating the purpose of the application and the Information needed to be stored with it. The purpose is to help OSU students find housing, and the Information needed to do this involves relationships between Users, Tenants, Seekers, Information request, and Properties.

The overview is specific about the number of Users that the application may have (40,000), the location (Corvallis, OR) and university involved (OSU), and attributes that will need to be stored in the database.

The overview refers to several unique entities, including: mentions ratings (Reviews), addresses (Properties), Owners (Tenant), and Seekers.

The outline does a good job describing the purpose of each entity, their relationships, their attributes, each attributes datatypes, and some constraints.

The outline incorrectly formulates some of the relationships (saying a relationship is 1:1 where the other table says 1:M), but the ERD has them correct. I do not see a M:M relationship. The ERD is detailed well.

I would recommend using a consistent case style for attributes. There are currently some attributes with snake case (User.user_id), pascal case (Seekers.UserID), and other non-standard cases (Reviews.user_id, Properties.Sq\ ft). Table names are inconsistently pluralized. The overview also refers to tables and attributes inconsistently.

The draft is a good start. After making names more consistent and including a M:M relationship, you have a really good design.

Review #2:

Hi Sankalp and Michael, it looks like you guys put together a great project design. Your overview clearly states the problem of students being able to find somewhere to live and taking a database of which rooms are vacant/occupied. Your overview also lists a specific number of 40,000 needed rooms and used the population of OSU as your reference, which was a great idea.

There are at least 4 entities in your project design (6), and it looks like they all serve a purpose. You also listed all of the datatypes for the entities and their relationships to each other for each entity.

To me, it appears that all of the 1:M relationships are correctly formulated. For example, One Property can have multiple Tenants, so you identified it as 1:M correctly. I don't see any Many-to-Many relationships in your project draft, though. The ERD absolutely presents the project well and all of its relationships; all of the details present in the entities section can be seen in the ERD.

The names of entities and attributes are consistent all around the document, including your ERD.

The only suggestion I would give you guys is to find somewhere to add a M:M relationship, but it is an overall great project draft! I really like the idea of splitting up the Users into the Owner and Seeker groups.

Review #3:

- Does the overview describe what problem is to be solved by a website with a relational database back end?
 - This seeks to store information of people looking for rooms such that they can be paired together.
- Does the overview list specific facts?
 - 40,000 students
 - Might want to come up with some numbers on the number of properties, number of seekers etc.
 - Are at least four entities described and does each one represent a single kind of object/idea to be stored?
 - user, tenant, seekers, properties, reviews
 - Does the outline of entity details describe the purpose of each entity as well as its...
 - Attributes and their respective datatypes and constraints?
 - Looks good
 - The relationships between entities?
 - Looks good
 - Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a sufficiently detailed view of the database?
 - Looks good
 - Are the names of entities and attributes--including plural/singular forms and capitalization--consistent across the document?
 - Looks like the capitalization in Seekers_Users_user_id is wrong. Should only be capital at the beginning if it fits with the rest of the pattern.

Review #4:

- Does the overview describe what problem is to be solved by a website with a relational database back end? The overview describes the problem and solution clearly. It states the need for this database and the methods of which the team set out to achieve this. The target group is also listed as the 40,000 students attending OSU.
- Does the overview list specific facts? Yes, the overview does list the target audience and the size. It also lists some functionality where its users need to be able to customize information like names, addresses and preferences.
- Are at least four entities described and does each one represent a single kind of object/idea to be stored?

Yes there are 6 distinct entities described in the overview, namely, users, seekers, tenants, reviews, properties and information request.

- Does the outline of entity details describe the purpose of each entity as well as its...
 - Attributes and their respective datatypes and constraints?
 - The relationships between entities?

Yes, the outline does show the relationships of each entity and their attributes. They also show the constraints of each attribute.

- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a sufficiently detailed view of the database?

There looks like there are multiple 1:M relationships but no M:M relationships in the ERD and overview.

- Are the names of entities and attributes--including plural/singular forms and capitalization--consistent across the document?

Capitalization in the overview and the drawn ERD diagram are slightly different. In the overview, we see the first letter of each attribute capitalized but not in the ERD.

Overall i think this is a great idea and database. There needs to be a M:M relationship and a more consistent capitalization over all the documents.

Review #5:

Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

- The schema does follow the database outline and ER logical diagram exactly.

Are the names of entities and attributes—including plural/singular forms and capitalization--consistent across the document?

- All names have consistent plural/singular forms, capitalization, and other formatting across the document.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

- It is easy to read. The only thing I might change is to move Tenant Directory one a little further to the right so it's not so close to Seekers but it's not necessary.

Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

- Yes, the intersection table TenantDirectory is properly formed with two foreign keys to facilitate a many-to-many relationship between tenants and properties.

Does the sample data suggest any non-normalized issues (e.g., partial dependencies or transitive dependencies)?

- No, the sample data does not suggest any non-normalized issues.

Is the SQL file syntactically correct? This can be easily verified by using phpMyAdmin and your CS 340 database (do not forget to make a backup of your own database before you do this!).

- Yes everything is syntactically correct.

In the SQL file, are the data types appropriate considering the description of the attributes in the database outline?

- The database outline doesn't completely match up with the SQL file when it comes to datatypes. For example, rent and sq_ft are defined as varchar in the database and int in the SQL. Also, you have age as a varchar in both SQL and the outline but it might be better as a numeric type.

In the SQL file, are the primary and foreign keys correctly defined relative to the Schema? Are the appropriate CASCADE operations declared?

- The primary and foreign keys are correctly defined in the SQL file relative to the schema and no CASCADE operations are declared in the SQL file.

In the SQL file, are relationship tables consistent with the ERD/Schema?

- Yes, the relationship tables in the SQL file are consistent with the ERD/Schema.

In the SQL file, is all example data shown in the PDF INSERTED into the database?

- Yes, all example data shown in the PDF is inserted into the database.

Review #6:

Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

Yes, the schema presents a physical model that follows what is outlined in the database and the ERD from what I can see.

Are the names of entities and attributes--including plural/singular forms and capitalization--consistent across the document?

Yes, everything was consistent across the document. All the table names were capitalized and the values in the table were all lowercase with _ used for spaces. This formatting stayed consistent.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

Yes, everything was clear and easy to read.

Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

Yes, the table Tenant Directory has 2 foreign keys (user_id and address) and it has a many-to-many relationship.

Does the sample data suggest any non-normalized issues (e.g., partial dependencies or transitive dependencies)?

As far as I can see the sample data does not suggest any non-normalized issues.

Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to make a backup of your own database before you do this!).

Yes, the SQL file is syntactically correct and everything ran fine.

In the SQL file, are the data types appropriate considering the description of the attributes in the database outline?

The SQL file mostly matches the outlined base. The only difference I noticed was user_id not being declared as UNIQUE.

In the SQL file, are the primary and foreign keys correctly defined relative to the Schema? Are the appropriate CASCADE operations declared?

Primary and foreign keys seem to be defined correctly. CASCADE is not declared inappropriately since it is not used and I can not think of a use for it in this database.

In the SQL file, are relationship tables consistent with the ERD/Schema?

- Yes, the relationship looks consistent with what has been outlined.

In the SQL file, is all example data shown in the PDF INSERTED into the database?

- Yes, the examples have been INSERTED into the table and the two files are consistent.

Review #7:

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

Yes, the schema follows the database outline and the ERD exactly.

- Are the names of entities and attributes--including plural/singular forms and capitalization--consistent across the document?

Yes, the naming convention used here was the "_" and this was consistent throughout the entire document.

- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

Yes, the schema is easy to read since the diagram was clear and the relationship lines were not crossed/overlapped.

- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

Yes, in this case the Tenant Directory has two FKs and facilitates a M:N relationship between Properties and Tenants properly.

- Does the sample data suggest any non-normalized issues (e.g., partial dependencies or transitive dependencies)?

No, I don't think so, the sample data suggests no redundancies and I am not sure if normalization could be further applied here. Although, it could be argued that a certain combination of rooms, bathrooms and sqft would always give out a fixed rent and therefore, could be normalized in this sense. However, I don't think that is really necessary here and would just make the schema more complicated in a sense.

- Is the SQL file syntactically correct? This can be easily verified by using PhpMyAdmin and your CS 340 database (do not forget to make a backup of your own database before you do this!).

Yes, the sql file ran on my end without a problem or syntax error.

- In the SQL file, are the data types appropriate considering the description of the attributes in the database outline?

Yes, the data types for attributes in the sql file follows exactly as the database outline and are appropriate.

- In the SQL file, are the primary and foreign keys correctly defined relative to the Schema? Are the appropriate CASCADE operations declared?

Yes, the primary and foreign keys are correctly defined. However, there was no use of CASCADE operations in the file.

- In the SQL file, are relationship tables consistent with the ERD/Schema?

Yes, I believe it is consistent.

- In the SQL file, is all example data shown in the PDF INSERTED into the database?

Yes, all the sample data provided in the PDF were used in the sql file to insert into the appropriate entities.

Review #8:

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

Yes, from everything I can see, the schema represents the database outline and ERD exactly. I can't see any differences between the two.

- Are the names of entities and attributes--including plural/singular forms and capitalization--consistent across the document?

Yes. Across the document, all of the table names are capitalized and all of the attributes are lowercase. I don't see any situations that go against this model in the document.

- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

Yes, the schema/ERD is easy to read. None of the lines are crossed and the ideas are clearly represented throughout the schema/ERD.

- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

Yes, all of the intersection tables appear to be properly formed; although, I do not see a M:N relationship in the database outline or schema that would need two FKs.

- Does the sample data suggest any non-normalized issues (e.g., partial dependencies or transitive dependencies)?

From what I see, there shouldn't be any non-normalized issues with the sample data. All of the users have their own information and there is no redundancy between the samples of data. All of the attributes that are not keys are independently determined.

- Is the SQL file syntactically correct? This can be easily verified by using phpMyAdmin and your CS 340 database (do not forget to make a backup of your own database before you do this!).

Yes, all of your SQL is syntactically correct. There is also good indentation in the file to make reading the different commands clear.

- In the SQL file, are the data types appropriate considering the description of the attributes in the database outline?

In the Seekers table on the database outline, the user_id is written to be AUTO_INCREMENT but that is not expressed in the SQL definition for the Seekers table. The user_id in InformationRequests is also written to be UNIQUE but it is not declared as unique in your SQL from what I can see. Everything else looks healthy and appropriate between the database outline and SQL to me.

- In the SQL file, are the primary and foreign keys correctly defined relative to the Schema? Are the appropriate CASCADE operations declared?

In the SQL, the primary and foreign keys are correctly defined in relation to the Schema for the project. Everything lines up the way that it should. CASCADE is not used in the SQL file, but according to my understanding of the CASCADE operation, I don't see anywhere that it should have been used in the SQL.

- In the SQL file, are relationship tables consistent with the ERD/Schema?

Yes, the relationships between the tables in the SQL file are the same that they have been defined as in the ERD, schema, and database outline. The tables and attributes all appear to have relationships consistent with how they have been described.

- In the SQL file, is all example data shown in the PDF INSERTED into the database?

Yes, all of the example data shown in the PDF has been inserted into the database. All of the information for Users, Tenants, Seekers, Properties, Information Requests, Tenant Directory, and Reviews is consistent between the PDF document and SQL file.

Overall, I think you guys did a great job! The only recommendation I would give is to construct a Many to Many relationship and add some of the attribute constraints that I mentioned above to your SQL.

Review #9:

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

Yes, the schema does present database outline and a ERD.

- Are the names of entities and attributes--including plural/singular forms and capitalization--consistent across the document?

All the names of entities and attributes are consistent accross the document.

- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

The schema is easy to read. There is no crossed lines in the diagram.

- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

Yes, the intersection tables are properly formed, and M:N relationship is clearly present.

- Does the sample data suggest any non-normalized issues (e.g., partial dependencies or transitive dependencies)?

The sample data has no non-normalized issues. Besides, I would suggest you to be careful about the format of date insertion, it should be xxxx-xx-xx.

- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to make a backup of your own database before you do this!).

The SQL file is syntactically correct to me when I run in my end.

- In the SQL file, are the data types appropriate considering the description of the attributes in the database outline?

Yes, the data types look appropriate to me.

- In the SQL file, are the primary and foreign keys correctly defined relative to the Schema? Are the appropriate CASCADE operations declared?

The primary and foreign keys are correctly defined in relative to the schema.

- In the SQL file, are relationship tables consistent with the ERD/Schema?

The relationship tables are consitent with the ERD.

- In the SQL file, is all example data shown in the PDF INSERTED into the database?

All example data are shown in the PDF.

Review #10:

- *Does the UI utilize a SELECT for every table in the schema?* Data from each table in the schema should be displayed on the UI (Note: it is rarely acceptable for a single query to join all tables and display them).

Yes, there are SELECT statements for all tables in the schema.

- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*

Yes, there is a SELECT statement that filters out users by their first name.

- *Does the UI implement an INSERT for every table in the schema?* There should be UI input fields that correspond to each table and attribute in that table.

Yes, there are INSERT statements for all tables in the schema.

- *Does each INSERT also add the corresponding FK attributes, including at least one M:N relationship?* For example, if there is an M:N relationship between *Orders* and *Products*, INSERTing a new order, should also INSERT row(s) into the intersection table *OrderDetails*. Otherwise, the new Order won't be associated with any products, and an order containing no products likely doesn't make much sense.

Yes, all INSERT statements add corresponding FKs, and at least one M:N relationship that does this between Tenants and Properties.

- *Is there at least one DELETE and does at least one DELETE remove things from an M:N relationship?* For example, if an order is deleted from the *Orders* table, it should also delete the corresponding rows from the *OrderDetails* table, BUT it should not delete any *Products* or *Customers*.

Yes, there is at least one DELETE statement and it is for the M:N relationship between Tenants and Properties.

- *Is there at least one UPDATE for any one entity?* For example, in the case of *Products*, can the attributes of an existing row/record be updated?

Yes, there is at least one UPDATE statement and it is for the M:N relationship between Tenants and Properties.

- *Is at least one 1:M relationship NULLable?* There should be at least one 1:M relationship with partial participation. For example, perhaps users can have a row in *Accounts* without actually ordering anything (i.e., having no relationship with any record in *Orders*). Thus it should be feasible to edit an order and remove its relationship with an account (i.e., change the order's foreign key to NULL).

Yes, there is at least one 1:M relationship that is NULLable.

- *Do you have any other suggestions for the team to help with their HTML UI?* For example, maybe they should use AS aliases to replace obscure column names such as *fname* with "First Name".

Everything looks great so far, the only suggestion I have is to add a way to distinguish between the different rows of the tables like a line or box.

Review #11:

Hello Group 6. I see a select from each table dedicated towards showing each entity. I also see a selection meant to be used for each update. I also do see an insert for each entity. It also seems like you take into consideration regarding the foreign keys as you seem to add to Tenants when you add to TenantDirectory. That being said, I'm not super confident regarding how this would work, but wouldn't you technically have to also add it to Directory? Also wouldn't adding a Tenant involve inserting a new spot for TenantDirectory? I guess I would just make sure with someone else that you've added in consideration of the foreign keys correctly.

I see a delete option for each entity. Similar thought regarding insert and relationship of foreign keys. There also seems to be an update for each entity. I would consider whether you can make address and user_id NULL. If the user writes a review, will it be necessary for them to give their address? I guess I can see user_id being important though.

In terms of other suggestions, I found it a bit confusing trying to figure out which was your foreign keys within the Database Outline. In my opinion I would use some kind of symbol to illustrate that. I would also change the name "Tenant Directory" to something like "Tenants_Properties" so it's more clear that it's a intersection table between those two entities. That being said, I think you're DML for the most part looks good. Just took a little bit to understand how everything was labelled.

Review #12:

Does the UI utilize a SELECT for every table in the schema?

Yes, the provided DML code includes SELECT queries for all tables in the schema.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

Yes, there is a SELECT query that filters results based on the user's first name as an input.

Does the UI implement an INSERT for every table in the schema?

Yes, the provided DML code includes INSERT queries for all tables in the schema.

Does each INSERT also add the corresponding FK attributes, including at least one M:N relationship?

Yes, the INSERT queries add the corresponding foreign key attributes, the schema provided shows a many-to-many relationship between Tenants and Properties, using foreign keys in Tenant Directory.

Is there at least one DELETE and does at least one DELETE remove things from an M:N relationship?

The DML code includes DELETE queries for various tables.

Is there at least one UPDATE for any one entity?

Yes, the DML code includes UPDATE queries for multiple entities.

Is at least one 1:M relationship NULLable?

The provided documentation does not explicitly include an example of a nullable 1:M relationship. However, some foreign key columns could allow NULL values, making the relationship nullable.

Do you have any other suggestions for the team to help with their HTML UI?

HTML UI looks great, provides all the needed elements in a logical and easily navigable fashion.

Review #13:

- *Does the UI utilize a SELECT for every table in the schema?* Data from each table in the schema should be displayed on the UI (Note: it is rarely acceptable for a single query to join all tables and display them).
 - Yes, the UI utilizes a SELECT query for all 7 entities in the schema.
- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*
 - Yes. For example, there is a query which will select a specific user given from the user/website.
- *Does the UI implement an INSERT for every table in the schema?* There should be UI input fields that correspond to each table and attribute in that table.
 - Yes, the UI implements an INSERT for every table in the schema with the form values corresponding to the attributes in respective tables.
- *Does each INSERT also add the corresponding FK attributes, including at least one M:N relationship?* For example, if there is an M:N relationship between *Orders* and *Products*, INSERTing a new order, should also INSERT row(s) into the intersection table *OrderDetails*. Otherwise, the new Order won't be associated with any products, and an order containing no products likely doesn't make much sense.
 - Yes, each INSERT also adds the corresponding FK attributes.
- *Is there at least one DELETE and does at least one DELETE remove things from an M:N relationship?* For example, if an order is deleted from the *Orders* table, it should also delete the corresponding rows from the *OrderDetails* table, BUT it should not delete any *Products* or *Customers*.
 - Yes, there are DELETE queries for the required entities and only deletes the corresponding rows, nothing extra. e.g. the delete query for reviews does not delete the user nor the property itself.
- *Is there at least one UPDATE for any one entity?* For example, in the case of *Products*, can the attributes of an existing row/record be updated?
 - Yes, there are UPDATE queries for the required entities.
- *Is at least one 1:M relationship NULLable?* There should be at least one 1:M relationship with partial participation. For example, perhaps users can have a row in *Accounts* without actually ordering anything (i.e., having no relationship with any record in *Orders*). Thus it should be

feasible to edit an order and remove its relationship with an account (i.e., change the order's foreign key to NULL).

- No, I don't think so (although I might be misunderstanding this question). Although there is at least one 1:M relationship, I don't think it is NULLable. For example, between Information Requests and Seekers. If I were to try to remove the user_id (i.e. make it Null) it would not let me since user_id is specified as "NOT NULL" in the database outline.
- *Do you have any other suggestions for the team to help with their HTML UI?* For example, maybe they should use AS aliases to replace obscure column names such as fname with "First Name".
 - Yes, I think it would be a good idea to replace the column names with ones that gives more context and are more user friendly.
 - Also maybe adding more columns to give more context. For example, in the Tenants tab in the website, you could add the first name and last name relating to the user_id to provide more context of who the owner or tenant is.

Revision Plan:

1. Rename the Tenant Directory table to Tenants Properties
 - a. Since it is just an intersection table, it should follow naming conventions
 - b. Updated it in all the SQL files as well
2. Add table styling to the website
 - a. Makes the tables easier to read
3. Add foreign key symbol to outline: ^

Upgrades to the Draft version

1. Added address dropdown to tenant page.
2. Added submit buttons to creation pages
3. Removed traces of Photo attributes from DML, DDL, and HTML

Revised Project Design:

Title: NestQuest

Team Members: Sankalp Patil and Michael Molineaux

Overview:

Our project is to develop a web application that allows students at Oregon State University to find roommates and places to live. It also allows landlords to post information about their properties to attract tenants. Our database will be a relational database tasked with storing and listing information about vacant rooms in Corvallis, Oregon. This application is intended for students at Oregon State University. There are approximately 40,000 students, most of which live in apartments or townhomes off campus. There are over 400 apartments, townhomes, houses, and other properties where students reside off campus. We estimate that the average size of apartments will be about 1200, with the majority of apartments being good quality. As such, our database needs to have the storage capacity and flexibility to accommodate a large variety of living situations. Users need to be able to insert, delete, and modify certain information from this database given their current living situation and goals. This database will contain information such as names, addresses, ratings, and preferences - such as non-smoking, pets, and noise levels. Additionally, people will be classified into two categories. Owners are people who currently reside in a Property and are looking for roommates. Seekers are people who are looking for a place to live.

Database Outline:

Primary keys have a star*

Foreign keys have a carat^

❖ **Users:** Holds the data of anyone who uses the application

- user_id*^: int, auto_increment
- fname: varchar
- lname: varchar
- email: varchar
- phone: varchar
- smoking: varchar
- pets: varchar
- gender: varchar
- age: varchar
- Relationships:

- A 1: 0 or M relationship between Users and Reviews (A User can write many Reviews).

- A 0 or 1 relationship between Users and Tenants (If a User exists, they can either be a Tenant or not a Tenant.)
- A 0 or 1 relationship between Users and Seeker(If a User exists, they can either be a Seeker or not a Seeker.)

❖ **Tenants:** Holds the data of people residing in a Property who are looking for roommates

- user_id*: int
- role: varchar
- Relationships:
 - A 1: 1 or M relationship between Tenants and Tenant Directory (Many Tenants can be living in many Properties).
 - A 1:1 relationship between Tenants and Users (If a Tenant exists, they must be a User)

❖ **Seekers:** Holds data of people currently looking for roommates and a place to live.

- user_id*: int, auto_increment
- price_upper: int
- price_lower: int
- Relationships:
 - A 1: 0 or M relationship between Seekers and Information request (A Seeker can send out multiple Information requests).
 - A 1:1 relationship between Seekers and Users (If a Seeker exists, they must be a User)

❖ **Information Requests:** Hold data about requests for Information about Properties sent by Seekers

- user_id*: int, not NULL, unique
- date_contacted: DATE, not NULL
- address*: varchar, not NULL
- Relationships:
 - A 1:1 relationship to Seeker(One Information request must have one Seeker)
 - A 1:1 relationship to Property(One Information request can be about only one Property)

❖ **Properties:** The Properties that Tenants are living at where openings are available

- address*: varchar, not NULL, unique
- rooms: int, not NULL
- bathrooms: int, not NULL
- sq_ft: varchar, not NULL
- rent: varchar, not NULL

- utilities: varchar, not NULL
- description: varchar
- Relationships:
 - A 1: 0 or M relationship to Information Request (One Property can have multiple Information requests)
 - A 1: 0 or M relationship to Tenant Directory (One Property can house multiple Tenants)
 - A 1: 0 or M relationship to Reviews (One Property can have none or multiple Reviews)

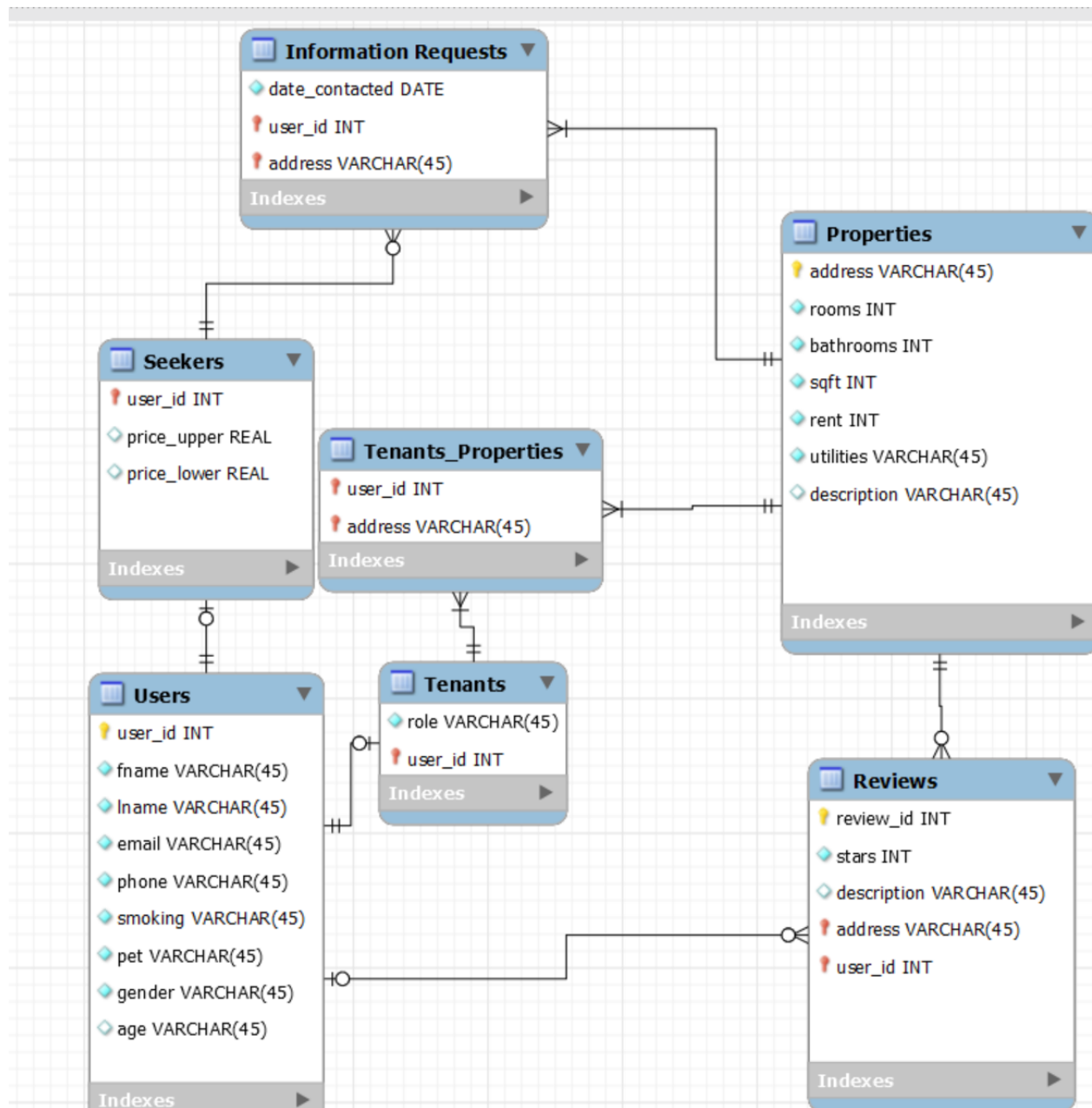
❖ **Reviews:** These are Reviews written by Users of the app about Properties

- review_id*: int, auto_increment, not NULL
- user_id^: int, not NULL
- address^: varchar, not NULL
- stars: int, not NULL
- description: varchar
- Relationships:
 - A 1:1 relationship with Users (Any given review can only be written by one User)
 - A 1:1 relationship to Properties (A review can only be related to one Property)

❖ **Tenants_Properties:** This is an intersection table between Tenants and Properties

- user_id*^ int, not NULL (Note, the pk is the combination of the two)
- address*^ varchar
- Relationships:
 - A 1:1 relationship to Tenants
 - A 1:1 relationship to Properties

Entity Relationship Diagram:



Sample Data

| Users | | | | | | | | |
|---------|--------|-----------|----------------------|------------|---------|-----|--------|-----|
| user_id | fname | lname | email | phone | smoking | pet | gender | age |
| 1 | Rob | Michaels | rob@gmail.com | 1234567890 | yes | no | male | 22 |
| 2 | Amy | Stevenson | amy@outlook.com | 987654321 | no | no | female | 20 |
| 3 | Jim | Davis | jim@oregonstate.edu | 5413214567 | no | yes | male | 25 |
| 4 | Jordan | Brooks | jordan@gmail.com | 3219876540 | yes | yes | female | 20 |
| 5 | Bill | Bobert | bill@oregonstste.edu | 4436758766 | no | no | male | 24 |
| 6 | Jace | Smith | jace@yahoo.com | 4336751369 | no | yes | male | 21 |

| Tenants | |
|---------|--------|
| user_id | role |
| 4 | owner |
| 5 | tenant |
| 6 | tenant |

| Seekers | | |
|---------|-------------|-------------|
| user_id | price_upper | price_lower |
| 1 | 750 | 0 |
| 2 | 950 | 300 |
| 3 | 600 | 0 |

| Properties | | | | | | |
|-------------------|-------|-----------|------|------|---|--|
| address | rooms | bathrooms | sqft | rent | utilities | description |
| 12345 Berry st | 3 | 3 | 1550 | 750 | water, sewage, garbage, washer/dryer | Townhome 2.2 miles from campus |
| 23465 west ave | 2 | 2 | 1450 | 850 | water, sewage, garbage, washer/dryer | Apartment room located 5 minutes from campus |
| 876 robin blvd | 4 | 3 | 1650 | 600 | electricity, water, sewage, garbage, washer/dryer | nice place |
| 23819 bald mtn rd | 4 | 4 | 1800 | 650 | electricity, water, sewage, garbage, washer/dryer | House from the 80s but in great condition. |

| Information Request | | |
|---------------------|---------|-------------------|
| date_contacted | user_id | address |
| 12-11-2022 | 1 | 12345 Berry st |
| 09-28-2022 | 2 | 12345 Berry st |
| 02-24-2023 | 3 | 23819 bald mtn rd |

| Tenant_Properties |
|----------------------|
| user_id address |
| 4 12345 Berry st |
| 4 23465 west ave |
| 5 876 robin blvd |
| 6 23819 bald mtn rd |

| Reviews | | | | |
|-----------|---------|-------------------|-------|---|
| review_id | user_id | address | stars | description |
| 1 | 1 | 23819 bald mtn rd | 4 | great place! The location is nice |
| 2 | 2 | 876 robin blvd | 4 | Included electricity is a huge plus. |
| 3 | 2 | 23465 west ave | 3 | The rooms have a weird smell and the carpets are discolored |