

Group 1
Nathan Jewell
Molly Johnson

Actions based on the feedback:

We are only fixing the two discrepancies found by the second reviewer between our outline and ERD (see green text above). See red text above for explanations regarding feedback we did use to update our draft with.

Upgrades to the Draft Version:

As described in the green text above, we updated our database outline regarding the discrepancies between our outline and the ERD for people having 0 homes, 0 neighbors, and/or 0 jobs.

(See specific changes for each step at end of document)

WEBSITE LINK: <http://web.engr.oregonstate.edu/~johnsmol/>

Project Outline and Database Outline - Updated Version:

Neighborhood Database Outline

Project Outline

Our database represents a neighborhood. This is a fictional neighborhood with people from many occupations and different family structures living together as neighbors. A neighborhood can contain many different components and they can relate to each other in interesting and useful ways. One can see who a person's neighbors are, who's living in each home, if people with similar jobs or who work for the same companies live near each other or not, and other useful information. Most people may be familiar with the general concept of a neighborhood, but being able to more closely study the important and interconnected relationships between the parts of a complex neighborhood makes this topic a smart choice for this database project.

Database Outline, in Words

Entities in the database:

- **Address** - The address is the most crucial entity as it is the primary key, providing unique identification, via a unique id number. Each address will relate to exactly one house. The address attributes are as follows:
 - **id number**: This is an auto-incrementing integer ID of up to 10 digits used to identify each address with a unique primary key. Cannot be blank, default is the appropriate auto-incremented number.
 - **house number**: Numerical portion of the address. An integer with a max length of 5. Cannot be blank, and there is no default.

- **cardinal direction:** Cardinal direction portion of the address. Can be North, S, E, W, N, SW, NE, SE, NW. It is a varchar with a maximum of 2 characters. It can be blank since not every address has a cardinal direction, and there is no default.
- **street name:** Street name portion of the address. It is a varchar with a max of 63 characters. Cannot be blank, and there is no default.
- **suffix:** The end portion of the address after the street name. There are many valid suffix options (Street, Avenue, Bend, Boulevard, etc.) so this isn't restricted to certain choices. Is a varchar with a max of 15 characters. Can be blank, and there is no default.
- **apt/unit:** The apartment/unit of the address, indicating which unit is indicated at an address where multiple units share the same rest of the address. A TINYINT. Can be blank, and there is no default.
- **city:** Is the city where this address is located. Consists of a varchar with a max of 127 characters. Cannot be blank since every address for this neighborhood must be in a city, and the default city is Corvallis since this neighborhood only exists in one city.
- **state:** Is the state where this address is located. Consists of a varchar with a max of 127 characters. Cannot be blank as every address must be in a state, and the default state is Oregon since this neighborhood only exists in one state.
- **5-digit postal code:** This is the postal code describing the location of the address according to the United States Postal Service. Consists of a varchar of 6 characters (since some zip codes can start with a 0 which would be dropped off if ints were used). Cannot be blank since every address for this neighborhood must have a zip code according to the USPS, and there is no default.
- **House** - The house entity represents the physical structure of each house and its unique characteristics. The House entity is related to the address entity in that each house has exactly one address. It is also related to the person entity, as one person may own many houses, or many people may own one or more homes together.
 - **id number:** This is an auto-incrementing integer ID of up to 10 digits used to identify each house with a unique primary key. Cannot be blank, default is the appropriate auto-incremented number.
 - **architectural style:** Describes the general style/architecture of the house (i.e. ranch, Tudor, Victorian, mobile home, etc.). Consists of a varchar with a max of 127 characters. Cannot be blank as all houses can be described with some structural style description, and there is no default.
 - **year constructed:** The year the house was built. Consists of a year type attribute. Cannot be blank since all houses were built at some time, and there is no default.
 - **Year last purchased:** The calendar year when this home was last sold (To its now-current owner). Consists of a year type attribute.
 - **#bedrooms:** How many bedrooms are in the home.
 - **#bathrooms:** How many bathrooms are in the home. (Half baths are acceptable)
 - **Square footage:** Livable square feet in the construction.
 - **levels:** An integer which represents how many levels the house has (one level if entire house is on the same floor, two if there's a main floor and an upper floor or a basement and a main floor, etc.). Cannot be blank since all houses have at least one level, and there is no default.

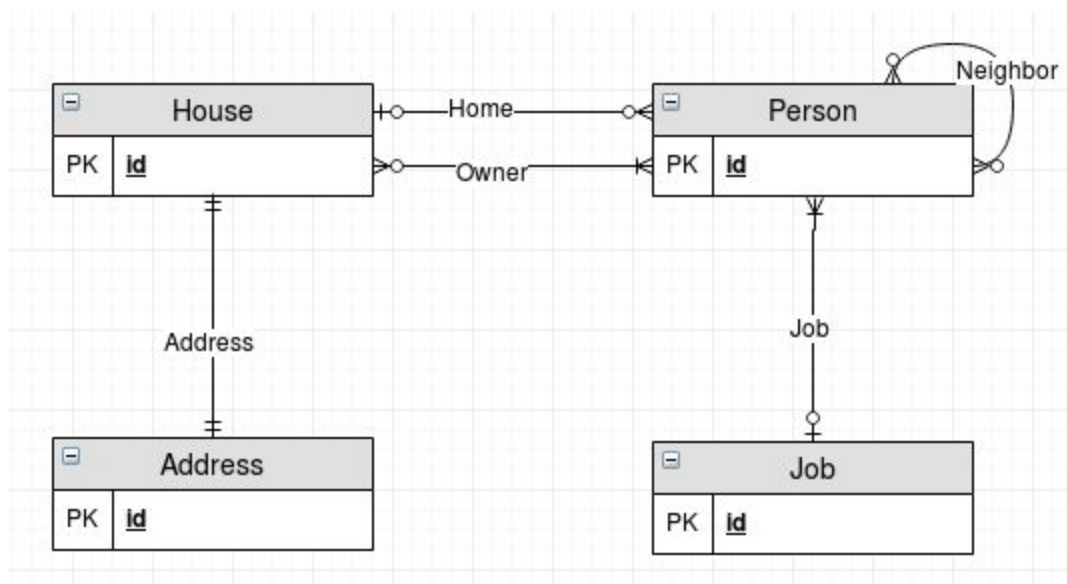
- **home value:** The current worth (in US dollars) of the house. Consists of an integer. Cannot be blank as all houses are worth some amount of money, and there is no default.
- **home address:** The home address, which is connected to the Address entity. Consists of an integer that's at most 5 digits, used to connect it to its related Address entity. Cannot be blank as every home has an address, the default is randomly assigned.
- **Person** - These are the people who reside in the neighborhood. Each person is related to 0 to many people since every home in this neighborhood has 0 to many neighbors. Each person resides in 0 to 1 houses, as they must reside in a home in the neighborhood or be homeless in the neighborhood to be counted as being a member of the neighborhood. Each person has 0 to one job in this neighborhood. People can own 0 to many houses.
 - **id number:** This is an auto-incrementing integer ID of up to 10 digits used to identify each person with a unique primary key. Cannot be blank, default is the appropriate auto-incremented number.
 - **Firstname:** The person's first name. A varchar.
 - **Lastname:** The person's last name. A varchar.
 - **job:** The person's occupation, which is connected to the job entity. Consists of an integer that's at most 5 digits, used to connect it to its related job entity. Can be blank since not everyone is employed. Unemployed is the default.
 - **home:** The person's primary residence, which is connected to the house entity. Consists of an integer of at most 5 digits, used to connect it to its related Address entity. Cannot be blank as everyone is either a person has an address or is homeless, and homeless is the default.
 - *[This will be abstracted as a separate table in the schema]* **property:** The person's house(s) they own, which is connected to the House entity. Consists of an integer of at most 5 digits, used to connect it to its related House entity. Can be blank as not everyone owns the house they live in. Default is homeless.
- **Job** - An occupation, relates to people in the neighborhood. Each person has exactly one job (or is unemployed), and each job can have multiple people who work in that field.
 - **id number:** This is an auto-incrementing integer ID of up to 10 digits used to identify each address with a unique primary key. Cannot be blank, default is the appropriate auto-incremented number.
 - **title:** The title of the job (plumber, accountant, lawyer, cashier, waiter, nurse, etc.). Is a varchar of at most 63 characters. Cannot be blank, and there is no default.
 - **income:** The income of people in this job. An integer of at most 10 digits. Cannot be blank since every job has an income, and there is no default.
 - **company:** The name of the company associated with the job. A varchar of at most 127 characters. Cannot be blank, and there is no default.
 - **education/training required:** Min. education/training level required for the job. A varchar of at most 63 characters. Cannot be blank, and there is no default.

Relationships in the database:

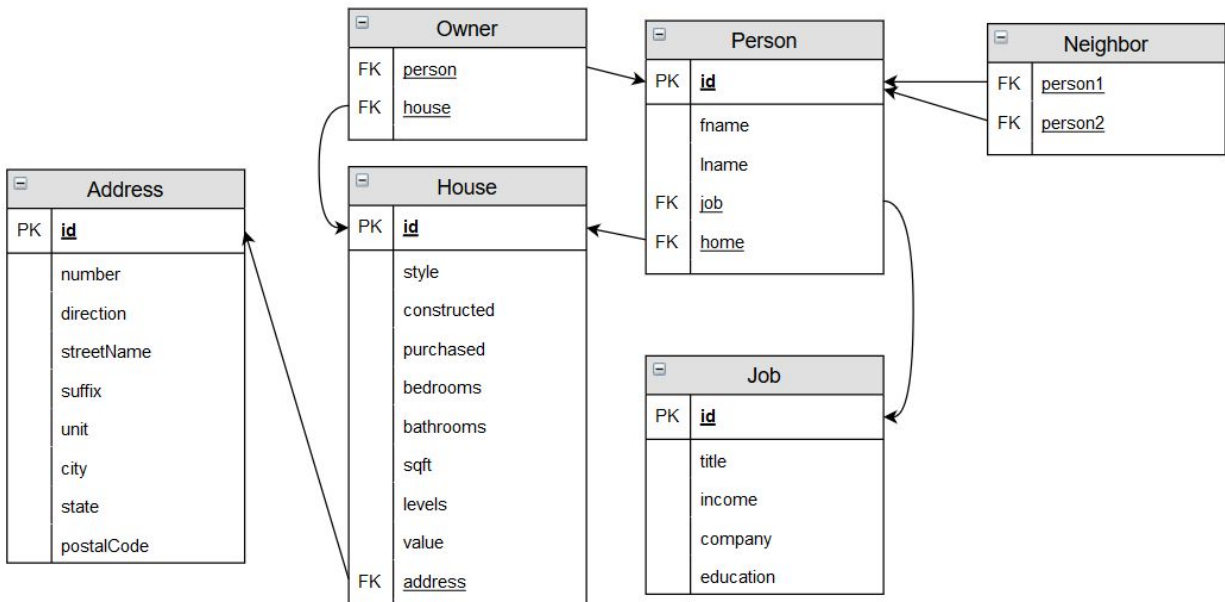
- **Neighbor** - (person -> person, many to many): Multiple people can be neighbors with multiple other people, hence the relationship is many to many.

- **Owner** - (person -> house, many to many): One or more people can own one or more houses, and one or more houses can be owned by one or more people, therefore the relationship is many to many.
- **Location** - (house -> address, one to one): Exactly one house can be connected to exactly one address, and vice-versa. Thus the relationship is one to one.
- **Home(Primary Residency)** - (person -> home, many to one): Each house can have many people reside in it, but each person resides at only one house (though they may own multiple homes at different addresses). Hence the relationship is many to one.
- **Job** - (person -> job, many to one): Each person has exactly one job and but each job can have many people who work in that field. Thus the relationship is many to one.

Entity-Relationship Diagram:



Schema:



Changes from review:

Changed for step 4:

Changed dml to be ready for the functionality we will support.

Made some minor manual fixes in the data as holdovers so it works (neighbor pk conflicts)

Wrote the entire freaking api.

Fixed in step 3

Removed house reference from address (this is why it was missing in sql) - our own design change decision.

Fixed varchar length in doc to match actual choice in sql (using orders of 2).

We are not limiting the house numbers in the same way as specified but instead are using specific mariadb types so that is why there is no mention like you say in the sql.

Fixed in step 2

Review 1:

Are the attributes for each entity in the ERD same as that described in the database outline?

*no. the ERD seems to only have IDs. No other attribute. (did I miss this somewhere in your ERD?) **Piazza post said this was OK as mentioned by other reviewers as well so we made no change.**

Is the participation of entities in the relationships same as that described in the outline?

*yes, this seems to be the case.

Is the cardinality of entities in the relationships same as that described in the outline?

*yes

Based on the Database outline, could any of the relationships be better off described as an Entity instead ?

*no.

Is there something that could be changed/improved in the E R Diagram and/or the overall database design?

*the table don't seem to have any entities aside from IDs **Addressed above**

*the team seems to have create an additional table for many-to-one relationship (home -> person) **This is confusing, we did not create an additional table for this in our ERD.**

SCHEMA:

Are the relationship tables present where required and correctly defined, when compared with the database outline?

*yes

Are foreign keys present where required and correctly defined, when compared with the database outline?

*yes. The representation is a bit unusual.

Do the entity attributes match those described in the outline?

*yes

Is there something that could be changed/improved in the Schema and/or the overall database design?

*it almost feels like the neighbor table should not exist, but should rather be something that's figured out by logic at the application level. The reason I say this is that I don't know if there's a way for the DBA (or user) to know who their neighbors are when they are adding an entry in the database. Maybe it's because I really don't know how the application is expected to work, but just thought I'd make note of this. **This is a valid consideration that got us thinking about how the neighbors table is going to be calculated. It will be something determined by a script of some sort but we do not think it it necessary to do this on every client. Instead we will just update the neighbor table periodically with new neighbor information.**

Review 2:

ERD

Are the attributes for each entity in the ERD same as that described in the database outline?

*No only primary keys, but instructor Piazza post said this was okay.

Is the participation of entities in the relationships same as that described in the outline?

*Yes, lines are provided between the entities to show the participation.

Is the cardinality of entities in the relationships same as that described in the outline?

*Yes, cardinality is provided and matches the description provided.

Based on the Database outline, could any of the relationships be better off described as an Entity instead?

*It might be out of the scope for what is needed, but "Companies" could be a separate entity instead of a Job attribute. But I understand the present design decision.**As you predict, we consider this out of scope for the purposes of our database.**

Is there something that could be changed/improved in the E R Diagram and/or the overall database design?

*The neighbors could be based on the house location. That way, if someone moves houses, their neighbors would be reassigned based on the new house location. **They will be based on**

house location, this is just a logical relationship though and not a data relationship so it is not really represented in the table.

Schema

Are the relationship tables present where required and correctly defined, when compared with the database outline?

*Yes.

Are foreign keys present where required and correctly defined, when compared with the database outline?

*Yes

Do the entity attributes match those described in the outline?

*Yes

Is there something that could be changed/improved in the Schema and/or the overall database design?

*I don't believe so, looks good!

Overall, I think this is a great project concept. Reminds me a lot of Zillow or a County Appraisal District database. Lots of real-world application! Good luck all.

Review 3:

Are the attributes for each entity in the ERD same as that described in the database outline?

*Yes (only the primary keys are shown which was in line with the instructions/clarifications posted on piazza, and they match up).

Is the participation of entities in the relationships same as that described in the outline?

*Yes.

Is the cardinality of entities in the relationships same as that described in the outline?

*In the ERD, it looks like each person does not necessarily have to be related to another person who is a neighbor. I agree with this since someone could live in a new development or in the penthouse unit of an apartment and have no neighbors. However, the outline description says each person must have at least one neighbor, and that is not reflected in the ERD. This makes sense and we have made changes to our outline.

*In the ERD, it looks like each person does not necessarily have to have a home.

However, the outline states that each person must be related to a home to be considered part of the neighborhood. This makes sense and we have made changes to our outline.

Based on the Database outline, could any of the relationships be better off described as an Entity instead ?

*I think the Address entity could be merged into the House entity because they are both one-to-one relationships with one of each required. I do not see a reason to have them as separate entities. They could be, but since we are doing a design exercise we made the choice to keep them separate on purpose even though it is not necessarily the most simplistic for this setup.

Is there something that could be changed/improved in the E R Diagram and/or the overall database design?

*There are a few minor changes I would recommend/questions I have for the authors:

*For the "Person" table, I think it should not be necessary for each person to be related to another person. For example, what if a person is the first person to move into a new housing development and does not have any neighbors yet? What happens if someone owns the penthouse unit of an apartment building and is the only person on that floor?

The premise here is mistaken, our ERD shows that there is a zero/many to zero/many relation between people. It is possible for someone to have no neighbors.

*For the houses and owners, what if there is an apartment building owned by a company? Who would be considered the "owner" of an apartment? The person who owns the company? What if that person does not live in the neighborhood? This was a valid point but as part of the scope of our project we are choosing to ignore it since it will not compromise the integrity. All buildings will be owned by a person and we do not constrain home address to only the owner so there could be renters represented by that.

*For the relationship between jobs and people, what if a person has two jobs (ex: people who work two part-time jobs instead of one full-time job)? Again, while valid, our model is simplified and we are not taking this possibility into account.

*For the jobs, not all people with the same job title have the same income. Is the income intended to be an estimate that is typical for someone with that job title at that company?

Yes, your assumption is correct. This is also a purposeful simplification.

Schema

Are the relationship tables present where required and correctly defined, when compared with the database outline?

*Yes

Are foreign keys present where required and correctly defined, when compared with the database outline?

*Yes

Do the entity attributes match those described in the outline?

*Yes

Is there something that could be changed/improved in the Schema and/or the overall database design?

*There are a few minor changes I would recommend/questions I have for the authors:

*For the "House" table, you may not want the "Year Constructed" field to be required--in the case of very old houses, it may not be possible to determine when the house was built. (I am not sure how likely or not this is, but I know where I live it can be hard to find original records for houses). For this project we will be generating our own data and will make sure all houses have an actual year they were constructed.

*For the "House" table, it seems like the home value might be difficult to keep up-to-date if it is going to be the current value since housing prices fluctuate frequently. Perhaps "year purchased" and "purchase price" attributes might be helpful instead?

Again, since we're not gathering real-world data, in a project of this scope where we're going to generate our own data, we think it makes sense to keep it simplified to one value for each house.

Review 4:

ERD

Are the attributes for each entity in the ERD same as that described in the database outline?

*No. Only shows primary keys. **This is OK as per piazza post.**

Is the participation of entities in the relationships same as that described in the outline?

*Yes.

Is the cardinality of entities in the relationships same as that described in the outline?

*Yes.

Based on the Database outline, could any of the relationships be better off described as an Entity instead ?

*I feel that the neighbor/house relationship could be made into a single entity or incorporate neighbors into the house table to be tracked there since houses will always be neighbors geographically.

Is there something that could be changed/improved in the E R Diagram and/or the overall database design?

*Neighbors are assigned to persons, but what happens if a neighbor moves to a new house?

When this happens some sort of stored procedure will likely just update the neighbor table.

SCHEMA

Are the relationship tables present where required and correctly defined, when compared with the database outline?

*Yes.

Are foreign keys present where required and correctly defined, when compared with the database outline?

*Yes.

Do the entity attributes match those described in the outline?

*Yes.

Is there something that could be changed/improved in the Schema and/or the overall database design?

*The neighbor relationship is a bit difficult to track. **Addressed throughout the above.**

Fixes Based on Feedback from Step 1 + Design

Decisions:

Based on feedback we replaced all mentions of "string" with "varchar". We also changed the "year constructed" attribute of the "house" entity to be the year type instead of a 4-digit integer.

Based on our own design review we modified a couple of relationships and naming strategies. First we added a unique id to every entity which acts as the primary key. Then we changed profession to job since jobs are unique to employers, which we have as an attribute for that entity. We also corrected the ordering of the cardinality for the "home" (Primary Residence) relationship and changed it to be from person->house instead of person->address. We changed the neighbor relation from "one to many" to "many to many" because everyone can have many shared neighbors. Finally we changed the name of the FK reference for the "owner" relationship

in the “person” entity from “house” to “property” and clarified that it will be abstracted as a new table in the schema since it is a many to many relationship.