1. The first database I am interested in is a database centered around food prep. This database contains information about food, nutrition, calories, and even current food inventory. This database would primarily be used in a restaurant/food service environment.
   1. The first user group would be the chef/line cooks. They could easily pull recipes out of the database, giving them basic ingredients and quantities for creating a dish. For instance, if the chef wanted to have lasagna on the menu for dinner, the line cooks (though a simplified interface) could easily pull up the recipe, which indicates exactly what and how much of each ingredient they would need. This database could also contain textual information giving rough steps involved for making a dish. This group would need textual data (ingredient names and process steps), as well as numerical data (quantities of an ingredient, temperatures in an oven). This would be very useful in an environment where consistency of the dishes was important, but there were a lot of rotating dishes and it would be infeasible for cooks to remember exactly what goes into every dish. This user group only reads the data, having no write capability. Below are some example queries:
      1. **Find out how to make a particular dish:**SELECT ingredient.name, ingredient.quantity, recipe.name, recipe.steps  
         FROM ingredient, recipe  
         WHERE recipes.name = ‘lasagna’  
         JOIN …
      2. **Find out how to make every dish on the current menu:**   
         SELECT ingredient.name, ingredient.quantity, recipe.name, recipe.steps  
         WHERE menu.date = CURRENT\_DATE  
         JOIN …
   2. The second user group would be a manager or chef, in a recipe and menu development role. This can develop new recipes—new records that the cooks could subsequently prep. In addition to this, this user group can create a menu which is valid for a range of dates, that joins onto some selection of recipe records. This process can also take advantage of calorie information stored in the ingredients table, although converting between units would require a slightly advanced implementation. This user is looking at textual data (recipe and ingredient names), dates (for menus), and numerical data (ingredient quantities, menu item prices). Below are some example queries:
      1. **Find the dates and cost we had an item on the menu:**  
         SELECT menu.date, menu.cost  
         WHERE recipe.name = ‘lasagna’   
         JOIN <many to many join, require third table to join on?>
      2. **Add a new recipe to the menu:**  
         INSERT INTO recipe(name, ingredient, quantity, unit)  
         VALUES (‘lasagna’, ‘lasagna noodle’, 12, ‘noodle’),  
          (‘lasagna’, ‘ground beef’, 1, ‘lb’),  
          (‘lasagna’, ‘ricotta’, 2, ‘cup’),  
          …
      3. **Update a recipe with a better ingredient quantity:**  
         UPDATE recipe  
         SET recipe.quantity = 2.5  
         WHERE recipe.name = ‘lasagna’  
          AND recipe.ingredient = ‘ricotta’
      4. **Create a menu for an event on a particular day:**  
         INSERT INTO menu(date, recipe, cost)  
         VALUES (‘2019-10-14’, ‘lasagna’, 12.50),  
          (‘2019-10-14’, ‘pot roast’, 14.00),  
          …
   3. The third user group would be a more managerial role, separated from the actual cooking. This user can monitor the restaurants current inventory of ingredients, to make sure that there is always enough ingredients in stock for the current menu lineup. This user can read and edit an inventory table, as well as read the menu and recipe tables. This table also opens up the possibility for more advanced data analytics, such as tracking hot menu items vs time of year, etc. This user is looking at textual data (ingredient names, recipe names), but is primarily interested in numerical data, particularly ingredient inventory. Below are some example queries:
      1. **Find out what ingredients we care about this week:**  
         SELECT ingredient.name, ingredient.inventory  
         WHERE menu.date >= CURRENT\_DATE – 7 DAY  
         JOIN …
      2. **Find out what is out of stock**SELECT ingredient.name, ingredient.inventory  
         WHERE ingredient.inventory = 0
2. The second database to explore is for use at a veterinarian’s office. This database contains information about the patients (pets) treatments, as well as scheduling appointments.
   1. The heart of this database is information about treatments for a given animal. The primary user of this view is vets and vet technicians. These users are querying and inserting information about what treatments were given and when for particular animals. Thus, the database stores a full medical history of the animal. These users are working the dates (when the animal is seen), quantities (vitals measurements), and textual data (what treatments are applied). User might ask when the last time the animal got a vaccine, or how the animals weight changed from one visit to the last. Here are some example queries.
      1. **Find out what the animals weight has been over time:**  
         SELECT animal.weight, appointment.date  
         WHERE animal.name = ‘SPOT’  
          AND customer.lastname = ‘Lesperance’  
         JOIN …
      2. **Find all animals that received a defective vaccine:**SELECT animal.name, customer.lastname, appointment.date  
         WHERE care.medicine = ‘defective vaccine’  
         JOIN …
   2. The second user group would be customers. These users would be logging into a portal on the vet’s website, and could access information about their pets. This would include date information, such as when their pet was last seen, as well as numerical information, such as costs/billing. Here are some example queries:
      1. **Find out when the next appointment is:**SELECT appointment.date  
         WHERE animal.name = ‘SPOT’  
          AND customer.lastname = ‘Lesperance’  
         JOIN …
      2. **Find out the appointment cost:**SELECT appointment.bill  
         WHERE animal.name = ‘SPOT’  
          AND customer.lastname = ‘Lesperance’  
          AND appointment.date = 2019-09-09  
         JOIN …
   3. The third user group is the front desk staff, responsible for creating records for new patients (pets), and scheduling follow up visits. This group is primarily looking at date data. They will mostly just be querying for open slots with no patient booked. Here is an example query:
      1. **Find out when the next open appointment is:**SELECT min(appointment.date), appointment.time  
         WHERE animal.name = ‘’  
          AND customer.lastname = ‘’  
         JOIN …
3. The third database I would like to explore is based off of construction and contracting. This database would be owned and operated by a general contractor.
   1. The first user group would be the general contractors themselves. The contractor would keep track of all materials ordered, and all of the labor performed on site. The contractor would be working with textual data (2x4 lumber, sheathing, etc), as well as numerical data (hours worked, quantity of lumber). This database would work as a manifest to help keep the contractor organized. Here are some example queries:
      1. **How many 2x4s did I order on Saturday:**SELECT material.quantity  
         WHERE material.name = ‘2x4’  
          AND order.date = ‘2019-09-07’  
         JOIN …
      2. **Did my laborer work overtime last week:**SELECT sum(payroll.hours)  
         WHERE payroll.employee = ‘John  
          AND payroll.date >= ‘2019-09-01’  
          AND payroll.date < ‘2019-09-08’  
         JOIN …
   2. The second user group would be the client, who hired the general contractor. They would be interested in how much they are spending. Depending on the contractor and the job being performed, the client might not get to see all of the nitty gritty details. In this implementation, the client only gets to see the final bills. The client will be able to acess date information, as well as numerical information for total costs. The client does not have any write access to the database, just read access (unless we enhanced the database to include billing information)
      1. **How much did the drywall project cost:**SELECT project.cost  
         WHERE project.name like ‘%drywall%’
      2. **How did the plumbing take:**

SELECT project.start\_date, project.end\_date  
WHERE project.name = ‘plumbing’

* 1. The last user group would be subcontractors. Much of their use of the database would be similar to the general contractor, except they would have different authorization. The subcontractors could use this database for managing scheduling, as well as inserting information regarding their work. For instance, the plumbing subcontractor would be able to read higher level information, such as when other portions of the project completed (is the building insulated already, is the electrical in place), but would not have access to things like material costs/labor for projects outside of plumbing. Access is very similar to the second user group, except they have write authorization in very certain areas. Here are some basic example queries:
     1. **Is the framing completed:**SELECT project.end\_date  
        WHERE project.name = ‘framing’
     2. **When is the plumbing scheduled to start:**SELECT project.start\_date  
        WHERE project.name = ‘plumbing: