**Instructions:**

**In a word document, answer the following questions.**

**Keep question in the exact order they are listed in.**

**Questions should be in bold font.**

**Answers should be in non-bold font.**

**All questions are worth the same points**

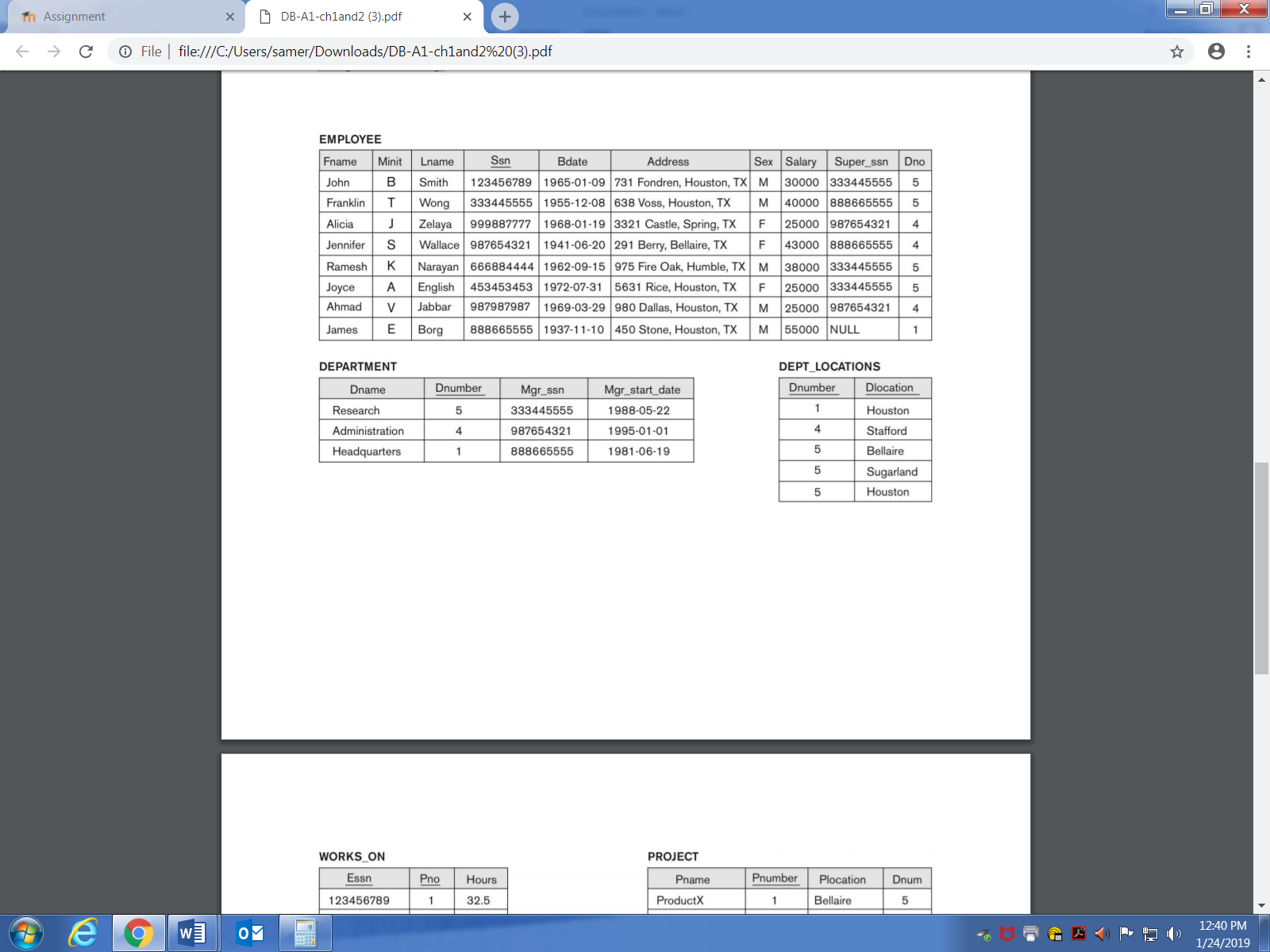
**Question 1 (Chapter 1 )**

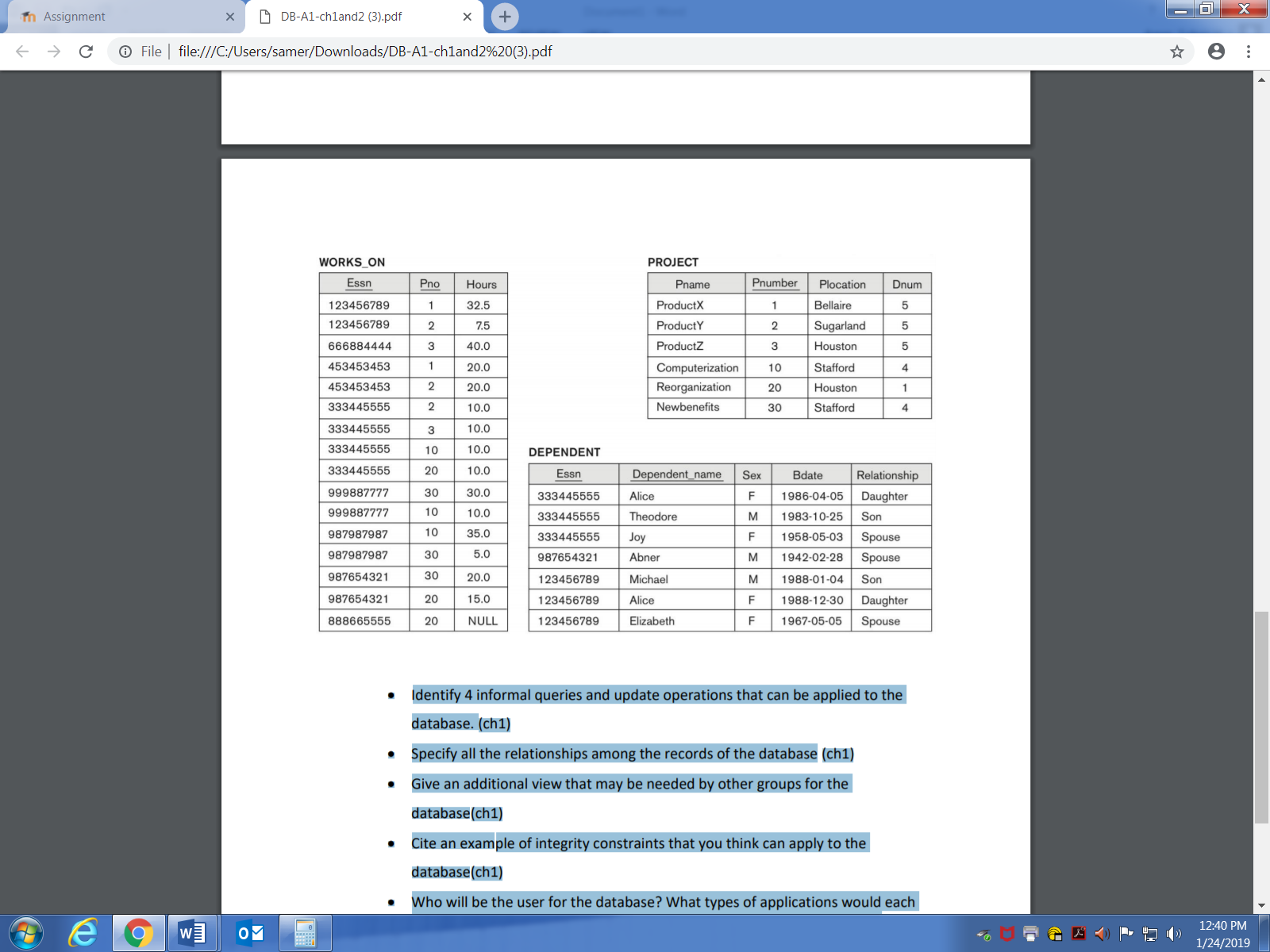
**Provide an example that shows the difference between controlled and uncontrolled redundancy**

**Using the following database**

Controlled redundancy is like what’s in the database. In the EMPLOYEE table, there is information that can be seen in other tables. As an example: EMPLOYEE and WORKS\_ON have SSN, EMPLOYEE and DEPT\_LOCATIONS have Dnumber/Dno. These data entries would ideally check against each other to make sure they’re consistent. This controlled redundancy uses denomalization to decrease search time by not having to search multiple tables.

Uncontrolled redundancy allows for multiple entries of a record to be entered, allowing for inconsistency and duplication of effort. It would be as if one had to enter all information in every table. This would lead to having to enter their name, SSN, department, etc. multiple times and inevitably the transcriber would cause an error.





* **Identify 4 informal queries and update operations that can be applied to the database.**

List all the project names, list a dependent’s information, update an employee’s address, update an employee’s salary.

* **Specify all the relationships among the records of the database**

EMPLOYEE’s Ssn is in WORKS\_ON and DEPENDENT, Super\_ssn is in DEPARTMENT, Dno is in DEPT\_LOCATIONS and DEPARTMENT and PROJECT; DEPT\_LOCATION’s Dnumber is in WORKS\_ON and PROJECT.

* **Give an additional view that may be needed by other groups for the database**

Some people may not need access to the database at all. It would be good to have employee’s able to check their hours in the WORKS\_ON table. If they constantly swap projects, have employee’s check Pno too. Say someone in admin could check all the information on an EMPLOYEE to be able to do paperwork on them.

* **Cite an example of integrity constraints that you think can apply to the database**

I think it would be good if there was checking for the matching of certain rows in a table against another. Also putting variable types restraints on certain cells such as SSNs being a long and birth dates being a DATETIME.

**Question 2 (Chapter 2)**

* **Discuss the main categories of data models. What are the basic difference among the relational model, the object model and XML model.**

A relational model is going to have tables (this is what I’m most familiar with). It keeps the tables separated in different files and can pull information from it using something like SQL. The object data model uses objects to store and separate information. It will use classes to group like objects, and the objects will contain relevant information (cells) that the object (table) would need. The XML model is a newer standard that uses a hierarchical tree structure. The database is elements with tags which allows for data to be nested inside them.

* **Describe the three-schema architecture. Why do we need mappings among schema levels? How do different schema definition languages support this architecture?**

Three-schema architecture separates user applications from the physical database. This puts it into three levels/schemas. The internal level defines the details of data storage and access paths using a physical model. The conceptual level defines the database structure for the users. It hides the information from the internal level and focuses on data types, relationships, and user operations. The external view defines what is shown to each user based on privileges and hides the rest of the information.

* **What is the difference between procedural and nonprocedural DMLs?**

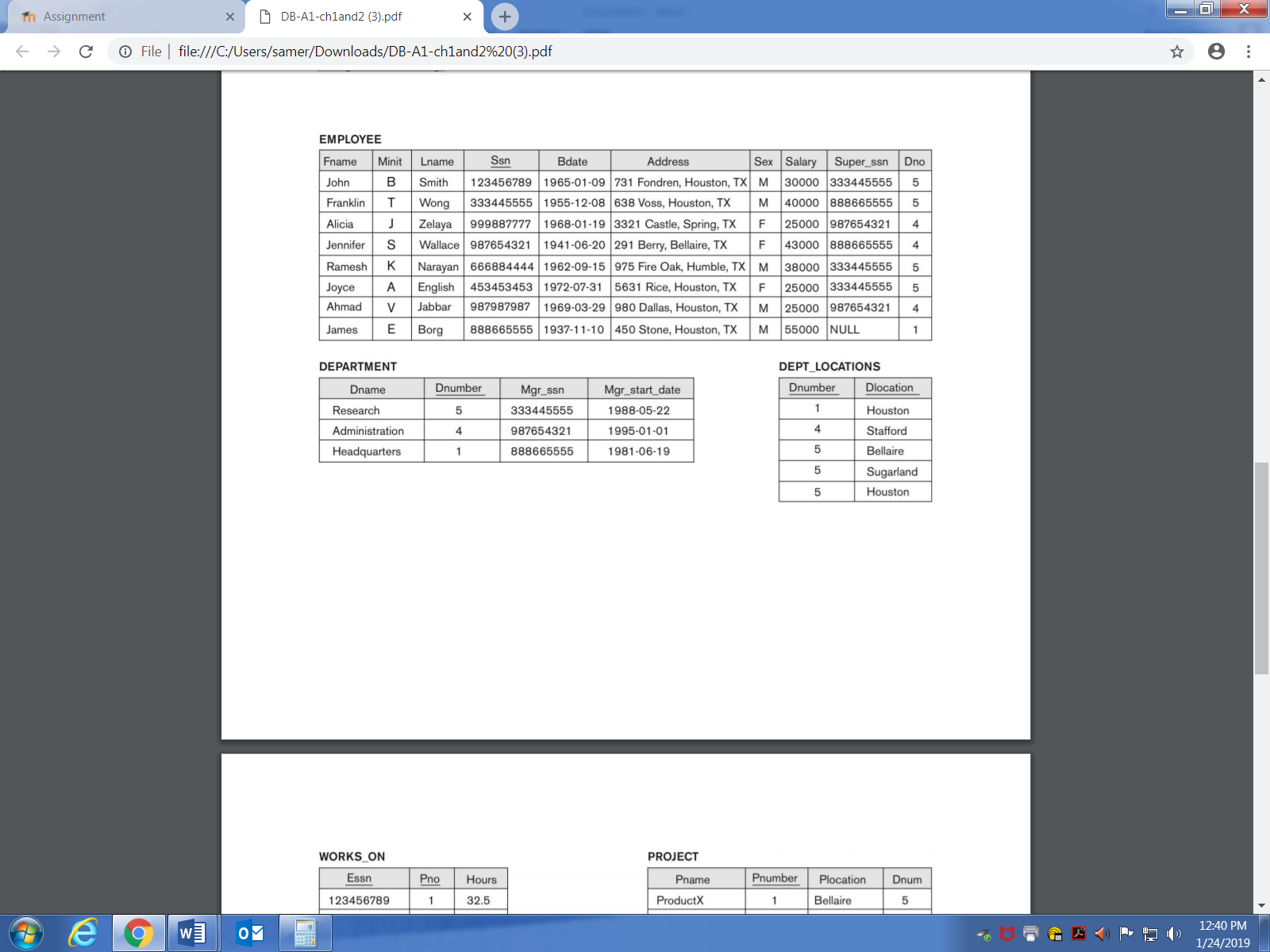
Data Manipulation Languages are split into two categories. Procedural DMLs are lower level and are used to get objects and data from a database. It must be embedded in a programming language. Nonprocedural DMLs are higher level and are used to specify database operations concisely.

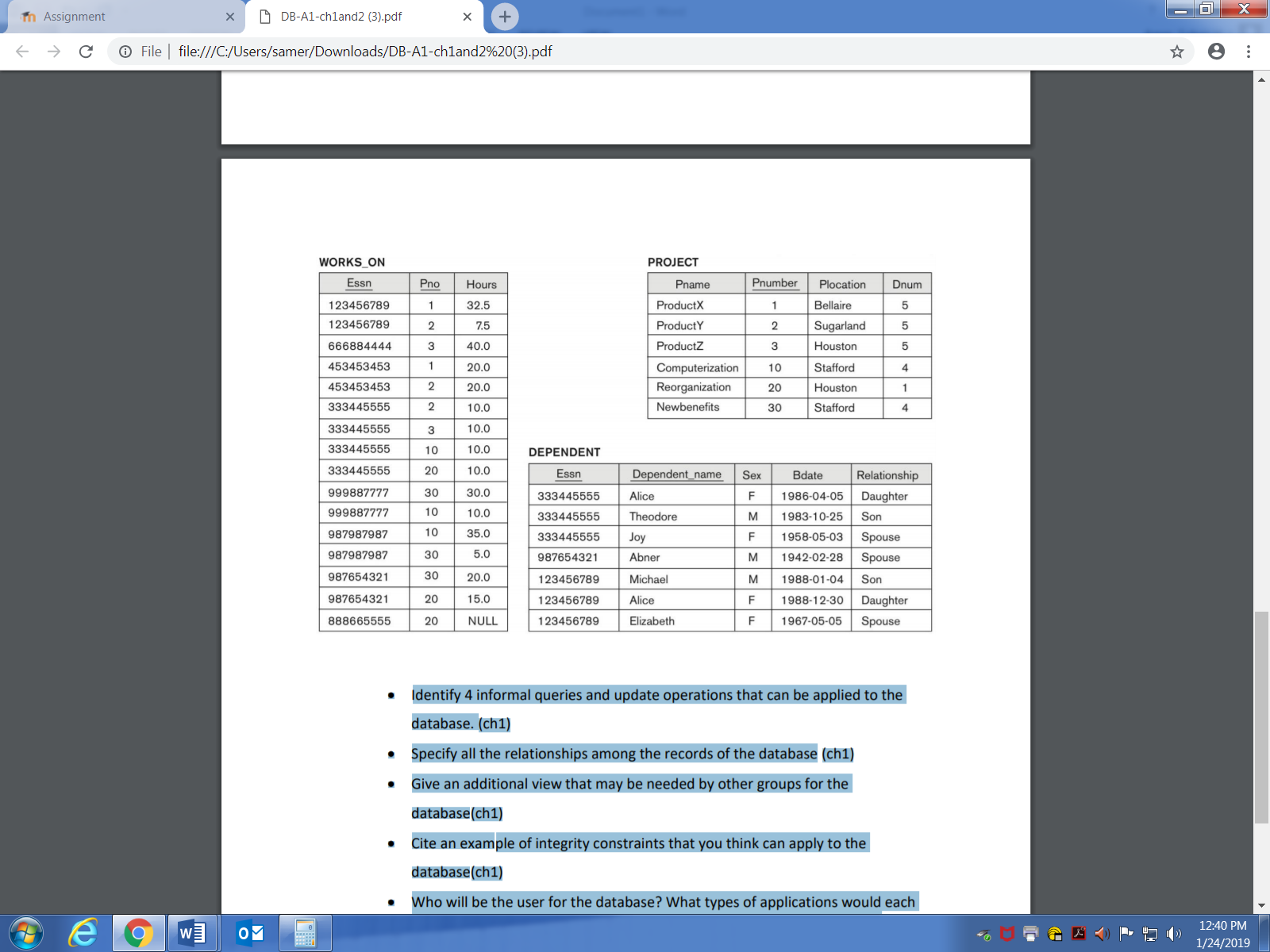
* **What is the difference between the two-tier and three-tier client/server architectures?**

The two-tiered architecture allows for the server to run the queries and transactional executions and for the client to run user interface and application programs. The three-tiered structure separates the client and adds the application server as a middle tier. This allows for increased security and possibly performance.

**Question 3 (chapter 2)**

**Using the following database**





* **Who will be the user for the database? What types of applications would each user need? To which user category would each belong, and what type of interface would each need?**

Possible users would be the following:

IT – people running the database. They have access to all privileges and tables. They would want the programmer interface to allow for more flexibility and operations. They would also want mobile to allow for fixing things on the go.

Admin – higher ups in the company who need access to EMPLOYEEs information for real world paperwork or applications. Access to DEPENDENT, DEPT\_LOCATIONS, DEPARTMENT, and EMPLOYEE. They would use a user-friendly interface and possibly a mobile one.

Managers – these people would need to be able to update and move around EMPLOYEEs to different projects. They would have update and view for WORKS\_ON and PROJECT. These people would want the user-friendly interface.

Employees – these people are the lowest privilege. They would only have view privileges for their specific SSN on WORKS\_ON. This way they can view their project and hours. User-friendly interface.

* **What types of additional information and constraints would you like to represent in the schema?**

Names, locations and relationships would all be varchar to allow for only letters. Middle initial and sex is going to be a char. Dates are going to be datetime. Address is going to be a String. Department and project number is going to be an int. Hours will be a double. SSNs will be a long and unique.

* **One type of constraints dictates that a column or a group of columns must be unique across all rows in the table. Identify the column or the group of columns in the tables that must be unique across all rows in the table.**

SSN in EMPLOYEE should be unique because people shouldn’t share an SSN. There are other places where SSN is okay to be duplicated because it’s referring to the same person in the same column.

**Question 4 (chapter 2)**

* **Design a schema for the database of question 3**

**EMPLOYEE**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super\_ssn | Dno |

**DEPARTMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| Dname | Dnumber | Mgr\_ssn | Mgr\_start\_date |

**DEPT\_LOCATIONS**

|  |  |
| --- | --- |
| Dnumber | Dlocation |

**WORKS\_ON**

|  |  |  |
| --- | --- | --- |
| Essn | Pno | Hours |

**PROJECT**

|  |  |  |  |
| --- | --- | --- | --- |
| Pname | Pnumber | Plocation | Dnum |

**DEPENDENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Essn | Dependent\_name | Sex | Bdate | Relationship |