**ERIC WEBB**

**Assignment#1**

**MSIT 630 Database Systems (Summer, 2019)**

**Total: 60 points**

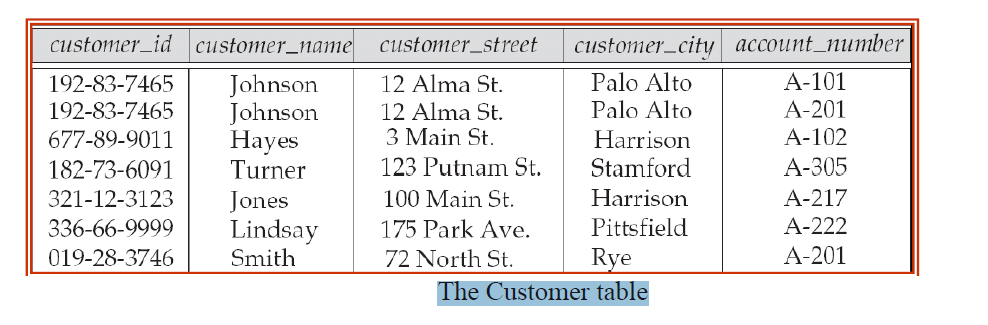
***Due: 5/26/2019 11:59PM***

1. Explain the concept of physical data independence and its importance in database systems (**4 points**)

Physical data independence is important because it allows you to change your actual hardware, for example from ssd to hard disk or vice versa without messing up the actual logic of the table. So when the change is made and you have new physical hardware your business logic is still intact and integrated without hassle. The tables and schemas remain the same but what physical medium they are stored on is different.

**2.** Explain what problems are caused by the design of the following table. (**4 points**)

The Customer table



A Major red flag here is that the Primary Key of “account\_number” is related to two different customers for “A-201” every entry should have its own independent “account\_number”.This would return two entries when querying account\_number “A-201” which would not make your data very consistent.

**3.** List four significant differences between a file-processing system and a DBMS. (**4 points**)

To begin, one difference between a file-processing system and DBMS is that a FPS will have **Redundant Data** where a DBMS will have a central location for the data to be accessed.

For example schools with a FPS will all each have their own individual libraries. Whereas schools with a DBMS will be able to access one central library all together.

Another issue is the topic of **Data Integrity,** where the same records at each locations hold conflicting data**.** In a FPS each school will need its own library and one error that could occur is that each libraries books could have different answers or content for the same book titles. So who can you trust if everyone has a different answer for the same question? That is why it is better to have the data stored in central location.

**Data Security** is another important factor when deciding in a DBMS over a FPS because a DBMS is traditionally more secure. In a FPS each location will have to control its own individual database. So if one locations data is compromised then all the others will be as well. So the more locations then the more exponential possibility of a security breach.

**Scalability** is another deciding factor. With a FPS each location will need physical resources along with man power at each location to support the data. Each location added will be more money, resources, and possibilities for errors.

**4.** Describe the differences in meaning between the terms *relation* and *relation schema*. (**4 points**)

Those relations and their schema.

relation is the predicate for example, Relation schema for a person "relation" can be shown in the following manner:   
  
Person(FirstName, LastName, Age, Gender, Address)

**5.** List two reasons why null values might be introduced into the database. (**4 points**)

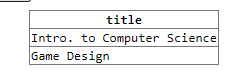
One reason you would use a NULL value is when the information is still unknown or **yet to be determined**. For example ungraded class assignments, you cannot put a value for those assignments before having graded them so the value could be NULL.

Another example would be something that **does not apply** basically leaving empty and carrying on. For instance, if you have contact information about a business stored then the gender category does not apply to that contact information.

6. Write the following queries in SQL, using the university schema, execute your SQL statement on the sample database and show me the query results. (Appendix A, page 1287) (**20 points, 4 points each**)

a. Find the titles of courses in the Comp. Sci department that have 4 credits.

select title from course where dept\_name='Comp. Sci.' and credits = 4;



b. Find the name(s) of the instructor(s) who DON’T earn the lowest salary in Physics department.

create view allButMin as select \* from instructor where dept\_name = 'Physics' except select ID, name, dept\_name, min(salary) from instructor where dept\_name = 'Physics';

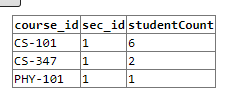
select name from allButMin;



c. Find the enrollment of each section (number of students enrolled) that was offered in Fall 2009.

Create view showSectionEnrollment as select course\_id,sec\_id,count(\* ) as studentCount from takes where year='2009' and semester='Fall' Group By course\_id;

select \* from showSectionEnrollment;



d. Find the minimum enrollment, across all sections offered in Fall 2009.

select course\_id, sec\_id, min(studentCount) from showSectionEnrollment;



e. Find the course ID and section ID of the sections that had the minimum enrollment in Fall 2009.

select course\_id, sec\_id, min(studentCount) from showSectionEnrollment;

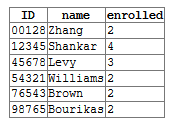


**7.** Write the following queries in SQL, using the university schema, execute your SQL statement on the sample database and show me the query results (Appendix A, page 1287) (**20 points, 4 points each**)

1. Find the names of all students who have taken at least two courses offered by Comp. Sci. department; make sure there are no duplicate names in the result. Note that student in other departments can take courses from Comp. Sci. as well.

create view csStudents as select ID,COUNT(\*)as enrolled from takes WHERE course\_id LIKE 'cs%' GROUP BY ID;

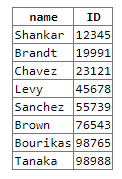
select student.Id,student.name,csStudents.enrolled from student,csStudents where student.ID=csStudents.ID and enrolled >=2;



1. Find the IDs and names of all students who have not taken any course offering in 2009.

create view notEnrolled as select ID,course\_id,sec\_id,semester,year,grade, count(\*) from takes WHERE NOT year='2009' group by ID;

select student.name,student.ID from student,notEnrolled where student.ID=notEnrolled.ID;



1. For each department, find the name and salary of the instructor who earns the minimum salary in that department. You may assume that every department has at least one instructor.

SELECT name,dept\_name,MIN(salary) as lowestSalary FROM instructor group by dept\_name;

