# horizontal lineDatabase Systems

Assignment 2

I confirm that this is my own work and that use of material from other sources, including the Internet, has been properly and fully acknowledged and referenced.

|  |  |
| --- | --- |
| Name: | Pang, Jinhao |
| Date: | 2022.09.21 |
| NYU ID: | N19475049 |
| Course Section Number: | csci-ga.2433-001 |



**Total in points** (100 points total): \_\_\_\_\_

**Professor’s Comments:**

|  |
| --- |
|  |

**1.9. What is the difference between controlled and uncontrolled redundancy? Illustrate with examples.**

The main difference is that the controlled redundancy can automatically check the modified record and update the corresponding files to prohibit inconsistencies among them. For example, in Figure 1.6 mentioned in the textbook, student\_name is changed. If redundancy is not controlled, it may cause an error because of the inconsistency with the Student file, while controlled redundancy will change the related files to prohibit inconsistencies.

**1.12. Cite some examples of integrity constraints that you think can apply to the database shown in Figure 1.2.**

1. every course record must have a unique value for Course\_number.

2. every student record must have a unique value for Student\_number.

3. The value of the Year data item within each Section record must be a two-digit integer

4. The value of Course\_name must be a string of no more than 30 alphabetic characters.

**2.14. If you were designing a Web-based system to make airline reservations and sell airline tickets, which DBMS architecture would you choose from Section 2.5? Why? Why would the other architectures not be a good choice?**

Three-Tier or n-Tier Architectures. Compared to Two-Tier and Basic Client/Server Architectures, n-Tier Architectures have more advantages in web applications because of the changing role of clients and servers. To make airline reservations and sell airline tickets, it would be better to add middle tiers between Client and Database server to improve the database security and business logic. As for Centralized DBMSs Architecture, all functionality, application program execution, and user interface processing are based on one machine, which cannot be used for a web-based system. People have to use architecture with client-server separation to access the application deployed on the server.

**2.15. Consider Figure 2.1. In addition to constraints relating the values of columns in one table to columns in another table, there are also constraints that impose restrictions on values in a column or a combination of columns within a table. One such constraint dictates that a column or a group of columns must be unique across all rows in the table. For example, in the STUDENT table, the Student\_number column must be unique (to prevent two different students from having the same Student\_number). Identify the column or the group of columns in the other tables that must be unique across all rows in the table.**

STUDENT: Course\_name

COURSE: Course\_number

PREREQUISITE: Prerequisite\_number

SECTION: Section\_identifier

GRADE\_REPORT: Student\_number, Section\_identifier

**5.16. Consider the following relations for a database that keeps track of student enrollment in courses and the books adopted for each course:**

**STUDENT(Ssn, Name, Major, Bdate)**

**COURSE(Course#, Cname, Dept)**

**ENROLL(Ssn, Course#, Quarter, Grade)**

**BOOK\_ADOPTION(Course#, Quarter, Book\_isbn)**

**TEXT(Book\_isbn, Book\_title, Publisher, Author)**

**Specify the foreign keys for this schema, stating any assumptions you make.**

The attributes Ssn in relation schema ENROLL is a foreign key of ENROLL that references relation STUDENT.

The attributes Course# in relation schema ENROLL is a foreign key of ENROLL that references relation COURSE.

The attributes Course# in relation schema BOOK\_ADOPTION is a foreign key of BOOK\_ADOPTION that references relation COURSE.

The attributes Book\_isbn in relation schema TEXT is a foreign key of TEXT that references relation BOOK\_ADOPTION.

**5.18. Database design often involves decisions about the storage of attributes. For example, a Social Security number can be stored as one attribute or split into three attributes (one for each of the three hyphen-delineated groups of numbers in a Social Security number—XXX-XX-XXXX). However, Social Security numbers are usually represented as just one attribute. The decision is based on how the database will be used. This exercise asks you to think about specific situations where dividing the SSN is useful.**

When the request is to filter the People by different conditions. (We can know that the first set of three digits is called the Area Number. The second set of two digits is called the Group Number. The final set of four digits is the Serial Number. From the Internet). By storing SSN in three attributes, we can filter people who live in the same area or blocks to better understand the user data.

**5.20. Recent changes in privacy laws have disallowed organizations from using Social Security numbers to identify individuals unless certain restrictions are satisfied. As a result, most U.S. universities cannot use SSNs as primary keys (except for financial data). In practice, Student\_id, a unique identifier assigned to every student, is likely to be used as the primary key rather than SSN since Student\_id can be used throughout the system.**

**a. Some database designers are reluctant to use generated keys (also known as surrogate keys) for primary keys (such as Student\_id) because they are artificial. Can you propose any natural choices of keys that can be used to identify the student record in a UNIVERSITY database?**

A combination of full name, last four digits of the phone number, and exact birthday will be a choice for this situation.

**b. Suppose that you are able to guarantee uniqueness of a natural key that includes last name. Are you guaranteed that the last name will not change during the lifetime of the database? If last name can change, what solutions can you propose for creating a primary key that still includes last name but remains unique?**

We cannot guarantee that the last name will never change for some cultural reason, so we can create a unique primary key with full name at the end of the key. We should make sure the segment before the first name is unique, so the whole key is unique. Then when the last name changes, after updating the primary key, we still get a unique primary key.

**c. What are the advantages and disadvantages of using generated (surrogate) keys?**

Advantages:

1. we do not need to consider if the primary key is unique and how to create it.
2. Query single key column faster with better performance

Disadvantages:

1. They are artificial without actual meaning, which may encounter problems in business requirements.
2. May have duplicates causing problems in generating primary key.