**Chapter 5 - Review Questions**

**8. What three data anomalies are likely to be the result of data redundancy? How can such anomalies be eliminated?**

* The most common anomalies considered when data redundancy exists are:

1. Update anomalies.
2. Addition anomalies.
3. Deletion anomalies.

All of these can easily be avoided through data normalization. Data redundancy produces data integrity problems, caused by the fact that data entry failed to conform to the rule that all copies of redundant data must be identical.

**10. What is a surrogate key, and when should you use one?**

* A surrogate key is an artificial PK introduced by the designer with the purpose of simplifying the assignment of primary keys to tables. Surrogate keys are usually numeric, they are often automatically generated by the DBMS, they are free of semantic content (they have no special meaning), and they are usually hidden from the end users.

**13. Suppose that someone tells you that an attribute that is part of a composite primary key is also a candidate key. How would you respond to that statement?**

* I would respond by saying that this argument is incorrect if the composite PK contains no redundant attributes. If the composite primary key is properly defined, all of the attributes that compose it are required to identify the remaining attribute values. By definition, a candidate key is one that can be used to identify all of the remaining attributes, but it was not chosen to be a PK for some reason. In other words, a candidate key can serve as a primary key, but it was not chosen for that task for one reason or another. Clearly, a part of a proper (“minimal”) composite PK cannot be used as a PK by itself.

**Chapter 5 – Problems**

**5. Using the STUDENT table structure shown in Table P5.5, write the relational schema and draw its dependency diagram. Identify all dependencies, including all transitive dependencies.**

* The following is the dependency diagram based on the Table P5.5.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STU\_NUM | STU\_LNAME | STU\_MAJOR | DEPT\_CODE | DEPT\_NAME | DEPT\_PHONE | COLLEGE\_NAME |

Transitive Dependencies

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ADV\_LASTNAME | ADV\_OFFICE | ADV\_BUILDING | ADV\_PHONE | STU\_CLASS | STU\_GPA | STU\_HOURS |

Transitive Dependencies

Transitive Dependencies

The relational schema for the dependency diagram shown in Table P5.5 is:

STUDENT (**STU\_NUM**, STU\_LNAME, STU\_MAJOR, DEPT\_CODE, DEPT\_NAME, DEPT\_PHONE, ADVISOR\_LNAME, ADVISOR\_OFFICE, ADVISOR\_BLDG, ADVISOR\_PHONE, STU\_GPA, STU\_HOURS, STU\_CLASS)

**6. Using the answer to Problem 5, write the relational schema and draw the dependency diagram to meet the 3NF requirements to the greatest practical extent possible. If you believe that practical considerations dictate using a 2NF structure, explain why your decision to retain 2NF is appropriate. If necessary, add or modify attributes to create appropriate determinants and to adhere to the naming conventions.**

* The relational schemas are:

STUDENT (**STU\_NUM**, STU\_LNAME, STU\_MAJOR, DEPT\_CODE, ADVISORY\_NUM, STU\_GPA, STU\_HOURS, STU\_CLASS)

MAJOR (**MAJOR\_CODE**, DEPT\_CODE, MAJOR\_DESCRIPTION)

BUILDING (**BLDG\_CODE**, BLDG\_NAME, BLDG\_MANAGER)

DEPARTMENT (**DEPT\_CODE**, DEPT\_NAME, DEPT\_PHONE, COLLEGE\_CODE)

COLLEGE (**COLL\_CODE**, COLL\_NAME)

The normalized dependency diagrams would look like:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STU\_NUM | STU\_LNAME | STU\_MAJOR | DEPT\_CODE | ADV\_NUM | STU\_CLASS | STU\_GPA | STU\_HRS |

|  |  |  |
| --- | --- | --- |
| MAJOR\_CODE | DEPT\_CODE | MAJOR\_DESCRIPTION |

|  |  |  |
| --- | --- | --- |
| BLDG\_CODE | BLDG\_NAME | BLDG\_MANAGER |

|  |  |  |  |
| --- | --- | --- | --- |
| DEPT\_CODE | DEPT\_NAME | DEPT\_PHONE | COLL\_CODE |

|  |  |
| --- | --- |
| COLL\_CODE | COLL\_NAME |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADV\_NUM | ADV\_LASTNAME | ADV\_OFFICE | ADV\_BUILDING | ADV\_PHONE |

Transitive Dependencies

Transitive Dependencies

Transitive Dependencies