# ANSWERS TO REVIEW QUESTIONS 1

**1. Discuss each of the following terms:**

**a. Data:** Raw facts from which the required information is derived. Data have little meaning unless they are grouped in a logical manner.

vertical

**b. Field:** A character or a group of characters (numeric or alphanumeric) that describes a specific characteristic. A field may define a telephone number, a date, or other specific characteristics that the end user wants to keep track of.

horizontal

**c. Record:** A logically connected set of one or more fields that describes a person, place, event, or thing. For example, a CUSTOMER record may be composed of the fields CUST_NUMBER, CUST_LNAME, CUST_FNAME, CUST_INITIAL, CUST_ADDRESS, CUST_CITY, CUST_STATE, CUST_ZIPCODE, CUST_AREACODE, and CUST_PHONE.

**d. File:** Historically, a collection of file folders, properly kept in a filing cabinet. Although such manual files still exist, we more commonly think of a (computer) file as a *collection of related records* that contain information of interest to the end user. For example, a sales organization is likely to keep a file containing customer data. Keep in mind that the phrase "*related records*" reflects a relationship based on function. For example, customer data are kept in a file named CUSTOMER. The records in this customer file are related by the fact that they all pertain to customers. Similarly, a file named PRODUCT would contain records that describe products – the records in this file are all related by the fact that they all pertain to products. You would not expect to find customer data in a product file, or vice versa.

**2. What is data redundancy, and which characteristics of the file system can lead to it?**

Data redundancy exists when unnecessarily duplicated data are found in the database. For example, a customer's telephone number may be found in the customer file, in the sales agent file, and in the invoice file. Data redundancy is typical in a (computer) file system, because it is unable to represent and manage data relationships. Data redundancy may also be the result of poorly-designed databases that allow the same data to be kept in different locations. (Here's another opportunity to emphasize the need for good database design!)

# ANSWERS TO PROBLEMS 1

**Given the file structure shown in figure below, answer problems 1 through 4.**

| | PROJECT_CODE | PROJECT_MANAGER | MANAGER_PHONE | MANAGER_ADDRESS | PROJECT_BID_PRICE |
|---|---|---|---|---|---|
| ▶ | 21-5Z | Holly B. Parker | 904-338-3416 | 3334 Lee Rd., Gainesville, FL 37123 | $16,833,460.00 |
| | 25-2D | Jane D. Grant | 615-898-9909 | 218 Clark Blvd., Nashville, TN 36362 | $12,500,000.00 |
| | 25-5A | George F. Dorts | 615-227-1245 | 124 River Dr., Franklin, TN 29185 | $32,512,420.00 |
| | 25-9T | Holly B. Parker | 904-338-3416 | 3334 Lee Rd., Gainesville, FL 37123 | $21,563,234.00 |
| | 27-4Q | George F. Dorts | 615-227-1245 | 124 River Dr., Franklin, TN 29185 | $10,314,545.00 |
| | 29-2D | Holly B. Parker | 904-338-3416 | 3334 Lee Rd., Gainesville, FL 37123 | $25,559,999.00 |
| | 31-7P | William K. Moor | 904-445-2719 | 216 Morton Rd., Stetson, FL 30155 | $56,850,000.00 |

**1. How many records does the file contain, and how many fields are there per record?**

The file contains seven records (21-5Z through 31-7P) and each of the records is composed of five fields (PROJECT through BID_PRICE.)

**2. What problem would you encounter if you wanted to produce a listing by city? How would you solve this problem by changing the file structure?**

The city names are contained within the MANAGER_ADDRESS field and decomposing this character (string) field is very hard, when needed. If the ability to produce city listings is important, it is best to store the city name as a separate attribute.

**3. If you wanted to produce a listing of the file contents by last name, area code, city, state, or zip code, how would you change the file structure?**

The more we divide the address into its component parts, the greater its information capabilities. For example, by dividing MANAGER_ADDRESS into its component parts (MGR_STREET, MGR_CITY, MGR_STATE, and MGR_ZIP), we gain the ability to easily select records on the basis of zip codes, city names, and states. Similarly, by subdividing the MANAGER name into its components MGR_LASTNAME, MGR_FIRSTNAME, and MGR_INITIAL, we gain the ability to produce more efficient searches and listings. For example, creating a phone directory is easy when you can sort by last name, first name, and initial. Finally, separating the area code and the phone number will yield the ability to efficiently group data by area codes. Thus MGR_PHONE might be decomposed into MGR_AREA_CODE and MGR_PHONE. The more you decompose the data into their component parts, the greater the search flexibility.

**4. What data redundancies do you detect, and how could these redundancies lead to anomalies?**

Note that the manager named Holly B. Parker occurs three times, indicating that she manages three projects coded 21-5Z, 25-9T, and 29-2D, respectively. (The occurrences indicate that there is a 1:M relationship between PROJECT and MANAGER: each project is managed by only one manager but, apparently, a manager may manage more than one project.) Ms. Parker's phone number and address also occur three times. If Ms. Parker moves and/or changes her phone number, these changes must be made more than once *and they must all be made correctly... without missing a single occurrence*. If any occurrence is missed during the change, the data are "different" for the same person. After some time, it may become difficult to determine what the correct data are. In addition, multiple occurrences invite misspellings and digit transpositions, thus producing the same anomalies. The same problems exist for the multiple occurrences of George F. Dorts.

**Given the file structure shown in figure below, answer problems 5 through 6**.

| PROJ_NUM | PROJ_NAME | EMP_NUM | EMP_NAME | JOB_CODE | JOB_CHG_HOUR | PROJ_HOURS | EMP_PHONE |
|---|---|---|---|---|---|---|---|
| 1 | Hurricane | 101 | John D. Newson | EE | $85.00 | 13.3 | 653-234-3245 |
| 1 | Hurricane | 105 | David F. Schwann | CT | $60.00 | 16.2 | 653-234-1123 |
| 1 | Hurricane | 110 | Anne R. Ramoras | CT | $60.00 | 14.3 | 615-233-5568 |
| 2 | Coast | 101 | John D. Newson | EE | $85.00 | 19.8 | 653-234-3254 |
| 2 | Coast | 108 | June H. Sattlemeir | EE | $85.00 | 17.5 | 905-554-7812 |
| 3 | Satellite | 110 | Anne R. Ramoras | CT | $62.00 | 11.6 | 615-233-5568 |
| 3 | Satellite | 105 | David F. Schwann | CT | $26.00 | 23.4 | 653-234-1123 |
| 3 | Satelite | 123 | Mary D. Chen | EE | $85.00 | 19.1 | 615-233-5432 |
| 3 | Satellite | 112 | Allecia R. Smith | BE | $85.00 | 20.7 | 615-678-6879 |

**5. Identify and discuss the serious data redundancy problems exhibited by the file structure?**

**Note: It is not too early to begin discussing proper structure. For example, ideally, each row should represent a single entity. Therefore, each row's fields should define the characteristics of one entity, rather than include characteristics of several entities. The file structure shown here includes characteristics of multiple entities. For example, the JOB_CODE is likely to be a characteristic of a JOB entity. PROJ_NUM and PROJ_NAME are clearly characteristics of a PROJECT entity. Also, since (apparently) each project has more than one employee assigned to it, the file structure shown here shows multiple occurrences for each of the projects. (Hurricane occurs three times, Coast occurs twice, and Satellite occurs four times.)**

Given the file's poor structure, the stage is set for multiple anomalies. For example, if the charge for JOB_CODE = EE changes from $85.00 to $70.00, that change must be made four times. Also, if employee Allecia R. Smith is deleted from the file, you also lose information about the existence of the JOB_CODE = BE, its hourly charge of $85.00, and the PROJ_HOURS = 20.7.

Incidentally, note that the file contains different JOB_CHG_HOUR values for the same CT job code, thus illustrating the effect of changes in the hourly charge rate over time. The file structure appears to represent transactions that charge project hours to each project. However, the structure of this file makes it difficult to avoid update anomalies and it is not possible to determine whether a charge change is *accurately* reflected in each record. Ideally, a change in the hourly charge rate would be made in only one place and this change would then be passed on to the transaction based on the hourly charge. Such a structural change – easy to accomplish in a relational database environment, but difficult to manage in a file system -- would ensure the historical accuracy of the transactions.

**6. How many different data sources are likely to be used by the file?**

The data sources are probably the PROJECT, EMPLOYEE, and JOB. The JOB source would contain the billing charge per hour for each of the job types – a database designer, an applications developer, and an accountant would generate different billing charges per hour.