Database Management Assignment 2

Due date: Before the next week's class (week 5)

The assignment has two independent parts (Part I and II). Each part includes SQL coding questions **to be answered in sequence**. You will submit **a single .sql file.** Clearly label each question with its part (Part I or II) and question number. Use comments (either -- or /* *comment* */) as needed for clarity.

PART I: Please write SQL statements for the following questions.

1. Create a new table containing these four columns: ***Emp#, Lastname, Firstname,*** *and* ***Job_class***. The table name should be **EMPLOYEES**. The Emp# column contains a numeric ID and should allow a five-digit number. Use a maximum length of 12 for the Firstname and Lastname columns. The Job_class column should be able to store character strings up to a maximum length of four, but the column values shouldn't be padded (shouldn't be filled with blank) if the value has less than four characters.

2. Now that you have created the EMPLOYEES table, add two columns to the table. One column, EmpDate, contains each employee's date of employment, and its default value should be the system date. The second column, EndDate, contains the employees' termination dates. EndDate does not have a default value.

3. Modify the Job_class column of the EMPLOYEES table so that it allows a maximum width of six characters.

4. Delete the EndDate column from the EMPLOYEES table.

5. Create a new table ***JL_EMPS*** by using a subquery from EMPLOYEES; copy three columns, Emp#, Lastname, and Firstname from the EMPLOYEES table.

6. TRUNCATE the JL_EMPS table, and then verify that the table still exists but no longer contains any data.

7. Delete the JL_EMPS table permanently so that it isn't moved to the recycle bin.

8. Delete the EMPLOYEES table so that it can be restored.

9. Restore the EMPLOYEES table.

PART II: Please write SQL statements for each of the following questions

Currently, the contents of the Category column in the BOOKS table are the actual names for each category. The structure presents a problem if one user enters COMPUTER for the Computer category and another enters COMPUTERS. To avoid this issue and other problems that might occur, the database designers have decided to create a CATEGORY table containing a code and description for each category. The structure for the CATEGORY table should be as follows:

Column name	Data type	Width
CatCode	CHAR	3
CatDesc	VARCHAR2	14

There are a couple of requirements for the table:
(1) The CatCode column should be the primary key of the CATEGORY Table.
(2) All categories must be assigned with category descriptions.

The data for the CATEGORY table is as follows:

CATCODE	CATDESC
BUS	BUSINESS
CHN	CHILDREN
COK	COOKING
COM	COMPUTER
FAL	FAMILY LIFE
FIT	FITNESS
SEH	SELF HELP
LIT	LITERATURE

1. Create the CATEGORY table (with all columns and the above constraints) and populate it with the given data.
2. Add a column, CatCode, to the BOOKS table with a data type matching with CatCode in the CATEGORY table. We created the BOOKS table in class (JLDB_Build_w4.sql).
3. Add a constraint requiring all category codes entered into the BOOKS table (if the values are entered) to exist in the CATEGORY table. Set the CatCode values for the existing records in the BOOKS table based on each book's current Category value.
4. Verify that the correct categories have been assigned in the BOOKS table by simply displaying them.
5. Save the changes permanently to the database.
6. Now, delete the Category column from the BOOKS table.
7. Create a CATEGORY1 table from CATEGORY and drop the CATEGORY table – we realized that the table needs to be recreated (no need to recreate for this assignment).

NOTE: you must provide all SQL commands that execute the above requirements.