

**CS/DSA 4513-001 - Fall 2019 - Dr. Le Gruenwald**  
**INDIVIDUAL PROJECT: A JOB-SHOP ACCOUNTING SYSTEM**  
**Assigned: 10/14/2019; Due: 1:30 PM 11/18/2019 on Canvas**  
**(Late submission will be accepted until 11:59 PM 11/19/2019 with 5% penalty; late submission after this time will NOT be accepted)**

## **I. DESCRIPTION**

A job-shop accounting system is part of an organization that manufactures special-purpose assemblies for customers.

A customer has a unique name, an address, and a category (an integer number from 1-10). A customer can order several assemblies. Each assembly is identified by a unique assembly-id, and has a date-ordered, and assembly-details. To manufacture assemblies, the organization contains a number of processes, each of which is identified by a unique process-id and is supervised by one department. Each department has its own department number and department-data. Each process also has process-data. Processes are classified into three types: paint, fit, and cut. The following information is kept about each type of process:

- Fit: fit-type
- Paint: paint-type, painting-method
- Cut: cutting-type, machine-type

During manufacture an assembly can pass through any sequence of processes in any order; it may pass through the same process more than once.

A job is assigned every time a process begins on an assembly. Information recorded about a job includes a unique job-no, a date the job commenced, and a date the job completed as well as additional information that depends on the type of job. Jobs are classified into three job types: cut-job, paint-job, and fit-job. Information stored about particular job types is:

- Cut-job: type of machine used, amount of time the machine used, material used, and labor time.
- Pain-job: color, volume, and labor time.
- Fit-job: labor time.

An account is maintained by the organization to keep track of expenditure for each process, each assembly, and each department. For each account, the database stores its unique account number and the date the account established. Three types of accounts are maintained:

- Assembly-account to record costs (details-1) for assemblies.
- Department-account to record costs (details-2) for departments.
- Process-account to record costs (details-3) for processes.

As a job proceeds, cost transactions can be recorded against it. Each such transaction is identified by a unique transaction number, and is for a given cost (sup-cost). Each transaction of necessity updates three accounts:

- A process account
- An assembly account
- A department account

The updated process account is for the process used by a job. The updated department account is for the department that manages that process. The updated assembly account is for the assembly that requires the job.

### On-line queries and their frequencies for the job-shop accounting system:

1. Enter a new customer (30/day).
2. Enter a new department (infrequent).
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered (40/day).
4. Enter a new process-id and its department together with its type and information relevant to the type (infrequent).
5. Create a new account and associate it with the process, assembly, or department to which it is applicable (10/day).
6. Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced (50/day).
7. At the completion of a job, enter the date it completed and the information relevant to the type of job (50/day).
8. Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details (50/day).
9. Retrieve the cost incurred on an assembly-id (200/day).
10. Retrieve the total labor time within a department for jobs completed in the department during a given date (20/day).
11. Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process (100/day).
12. Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department (20/day).
13. Retrieve the customers (in name order) whose category is in a given range (100/day).
14. Delete all cut-jobs whose job-no is in a given range (1/month).
15. Change the color of a given paint job (1/week).

## II. TASKS TO BE PERFORMED

**Task 1.** Design an ER diagram and a relational database to represent the Job-Shop Accounting database defined in part I.

**Task 2.** Provide a Data Element Dictionary that lists the names, types, and sizes (in bytes) of all attributes and associated constraints for each table.

**Task 3.**

- 3.1. Discuss choices of appropriate storage structures for each relational table assuming that all types of storage structures discussed in class (Lecture Topic 4) are available. For each table, identify the queries (from the list of the given queries) that access the table, the type of each of those queries (insertion, deletion, random search, or range search), the search keys (if any) involved in each of those queries, the frequency of each of those queries, your choice of the file organization for the table, and your justifications. Use the following format to fill out your answers:

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization	Justifications

- 3.2. Discuss choices of storage structures for each relational table when implementing it in Azure SQL Database (if different from the previous choices specified in 3.1).

**Task 4.** Construct SQL statements to create tables and implement them on Azure SQL Database. All Create statements must include appropriate constraints as defined in Task 2. For each table, you must include SQL statements that create the same storage structure as the one you selected for Azure SQL Database implementation in Task 3.2 (e.g., if you have decided that a table X must

have an index on attribute Y, then you must include an SQL statement to create an index on attribute Y for table X).

**Task 5.** Write SQL statements for all queries (1-15) defined in part I. Write a Java application program that uses JDBC and Azure SQL Database to implement all SQL queries (options 1-15), two additional queries for import and export (options 16-17), and the “Quit” option (option 18) as specified in the menu given below. The program will stop execution only when the user chooses the “Quit” option; otherwise all options must be available for the user to choose at all times. Your program must be commented properly.

## **WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM**

**(1) Description of query 1**

**(2) Description of query 2**

.

.

.

**(15) Description of query 15**

**(16) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).**

**(17) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).**

**(18) Quit**

**Task 6.** Run the program created for Tasks 5 to test its correctness. To populate the database, perform 5 queries for each type (1, 2) and 10 queries for each type (3, 4, 5, 6, 7, 8) and show the contents of the affected tables after the 5 queries of each type (1, 2) are completed and after the 10 queries for each type (3, 4, 5, 6, 7, 8) are completed. To show database access is possible, perform 3 queries for each type (9, 10, 11, 12, 13, 14, 15). To show the import and export facilities are available, run each option (16-17) once. To show the Quit option is available, run option (18) at least once. To demonstrate that Azure SQL Database can detect errors, you also need to perform 3 queries of different types that contain some errors.

### **Important Notes for the Java + JDBC + Azure SQL Database Part (Tasks 5 and 6):**

Data manipulation and error checking **must be done** by Azure SQL Database. Your program is only to create the menu, accept choices, form queries, submit them to Azure SQL Database for execution, and display results or error messages.

**Task 7.** Write a Web database application using Azure SQL Database and JSP (or PHP if you are familiar with it) which provides the Web pages for query 1 and query 13. Since both queries take the input data from the user, there should be two Web pages for each query as follows: for query 1, one Web page to allow the user to enter the input data and one to display a message confirming the successful execution of the insertion; and for query 13, there should be one Web page to allow the user to enter the input data and one to display the retrieval results with appropriate headings. To show that your Web application works correctly, run the Web application so that queries 1 and 13 will be executed in this order: first query 13, then query 1, and then query 13 again, making sure that the results of query 1 will change the results of query 13 that follow query 1.

### **III. SUBMISSION**

- 1.** The project report must be typed (diagrams must be computer-generated).
- 2.** The project report must be paginated.
- 3.** The project report must include a cover page, which contains the following information: course name and number, section number, semester and year, instructor's name, author's name, id, and email address, and title of the project.
- 4.** The project report must include a table of contents using the format specified in Section IV.
- 5.** Appropriate section headings must be included to show your solutions for the corresponding tasks/subtasks.
- 6.** In-line comments must be provided for the Java and JSP/PHP programs.
- 7.** Submit the following files (DO NOT SUBMIT ZIP FILES) on Canvas: 1) one single PDF file containing the entire project report with the contents as specified in Section IV (with the file name as IP\_REPORT\_your last name\_your first name); 2) one SQL file (extension .sql) containing the SQL statements to create tables as explained in Task 4 in Section II (with the file name as IP\_Task4\_your last name\_your first name); 3) one Java file (extension .java) containing the Java source program for Task 5 in Section II (with the file name as IP\_Task5\_your last name\_your first name); and 4) Java, JSP (extension .jsp) and HTML files (if any, extension .html), or PHP (extension .php) and HTML files (if any) which comprise your Web application for Task 7 in Section II (just the files with the source code you created/edited, do not try to retain their directory structure used during the running of the application) (each of the files in item (4) must have a file name starting with IP\_Task7\_your last name\_your first name, for example, IP\_Task7\_Smith\_Joe.jsp)
- 8.** The project is due on Canvas at 1:30 PM Monday, November 18, 2019. Late projects will be accepted until 11:59 PM Tuesday, November 19, 2019 with 5% penalty.
- 9.** The overall project is worth 250 points (25% of your course grade); the late submission penalty is 12.5 points

#### IV. REQUIRED FORMAT FOR THE TABLE OF CONTENTS OF THE PROJECT REPORT

For each task, indicate the pages in which it appears. For each subtask, indicate its start and end page numbers. Below is an example of the table of contents.

Tasks Performed	Page Number
Task 1.	1-10
1.1. ER Diagram	1-1
1.2. Relational Database Schema	2-10
Task 2. Data Dictionary	11-11
Task 3.	12-20
3.1. Discussion of storage structures for tables	12-14
3.2. Discussion of storage structures for tables (Azure SQL Database)	15-20
Task 4. SQL statements and screenshots showing the creation of tables in Azure SQL Database	21-50
Task 5. The Java source program and screenshots showing its successful compilation	51-60
Task 6. Java program Execution	61-90
6.1. Screenshots showing the testing of query 1	61-63
6.2. Screenshots showing the testing of query 2	63-65
.	
.	
.	
6.17. Screenshots showing the testing of query 15	80-81
6.18. Screenshots showing the testing of the import and export options	82-84
6.19. Screenshots showing the testing of three types of errors	85-88
6.20. Screenshots showing the testing of the quit option	89-90
Task 7. Web database application and its execution	91-100
7.1. Web database application source program and screenshots showing Its successful compilation	91-98
7.2. Screenshots showing the testing of the Web database application	98-100

#### V. PLAGIARISM

This is an INDIVIDUAL PROJECT. You must do this project by yourself. You must not collaborate with your classmates or anyone else. A violation of this rule will be considered plagiarism and will result in action as specified in the Academic Integrity Code at OU: [http://integrity.ou.edu/files/Academic\\_Misconduct\\_Code.pdf](http://integrity.ou.edu/files/Academic_Misconduct_Code.pdf). Review the Fall 2019 CS/DSA 4513-online section course syllabus for more information concerning plagiarism. If you have questions about this project, come to see your TAs or Dr. Gruenwald.