Garden Glory Project

**Garden Glory Project Information**

*IT468 Projects in Database*

**Company Background**

Garden Glory is a partnership that provides gardening and yard maintenance services to individuals and organizations. Garden Glory is owned by two partners, who employ two office administrators and a number of full- and part-time gardeners. Garden Glory will provide one-time garden services, but it specializes in ongoing service and maintenance. Many of its customers have multiple buildings, apartments, and rental houses that require gardening and lawn maintenance services.

*Below is a sample list of owners, properties, and services that* *Garden Glory collects about* *for its business operation (you can refer to textbook* ***Figure 2-34, pp. 115*** *to find the sample list as well).*



**D1: DB Analysis & Query**

* Due Oct 12 by 10pm
* Points 60
* Submitting a text entry box or a file upload

**Deliverable 1: DB Analysis & Query (60 points)**

**D1: DB Project Analysis & Query**: Develop a reasonable and realistic plan and outline the steps to accomplish the integrated project by the end of the course. *You can also refer to Garden Glory project questions in §1~3 of the Textbook (at the end of each chapter).*

## **(TJ)1. How could a well-managed database help the company better achieve its goals?**

The first step of using a database to help Garden Glory, is to know what their goals are. Suppose that Garden Glory wants to be known as the most reliable, experienced, and cost effective solution on the market. For them, a well-managed database can help them achieve these goals.

Kroenke and Auer (2015, p. 4[[1]](#footnote-1)) point out that people use databases to keep track of things. In the case of Garden Glory, a database holds the key to scheduling jobs, locations of jobs, the types of jobs needed, and the timing of jobs. Garden Glory can also keep track of other information through searches of the database.

For instance, they could search properties based on a criteria of “Type” or “Zip Code.” They could further break this down to “City.” If they wanted to see how often they visit a certain area they could look at a breakdown of “Service Dates.”

Garden Glory could use the information from their database to help them turn a better profit by scheduling jobs in clustered areas (districts). This improves the efficiency of staff usage. Imagine, no wasted employee time traveling from one side of the city to another. They could schedule jobs in one district area one day, and then move to the other district the next day.

Another advantage of maintaining a good database for Garden Glory is the reduction of costs and materials. By keeping their jobs clustered in districts, in theory fuel consumption should go down and production should go up. By knowing the type of jobs that need tending they can group employees into teams. Those who are strictly doing yard maintenance go to those properties, while employees who must perform duties that require special tools are allocated to the properties that require the specific equipment.

If Garden Glory wanted to increase business, they could look to the database to see what areas have the least amount of customers. They could then use this information to grow their business base by targeting this area for new customers.

For Garden Glory, a well-managed database is a much more efficient, faster way of looking at the requirements of their business, than that of keeping a group of lists. Lists are one dimensional, while databases are multidimensional. Garden Glory can see the big picture with a database. If they used an Excel spreadsheet (list) they would have a harder time gleaning the information needed to see possibilities, resources, and/or potential problems.

Hint: refer to Text Chapter 1 for " Advantages of database approach " and apply that understanding to the GG project case

**2. What are the modification problems that are likely to occur if Garden Glory attempts to maintain the list below in a spreadsheet?**

Hint: refer to Text Chapter 1 for " Modification Problems " and apply that understanding to the GG project case

**3. Using these data presented in the list, state assumptions about functional dependencies among the columns of data. Justify your assumptions on the basis of these sample data and also on the basis of what you know about service businesses. Indicate what assumptions need to be checked with the users (GG) and what suggestions you may have about the determinants/keys (e.g., do you need a surrogate key?)**

Hint: refer to Text Chapter 2 for " Functional Dependency " and apply that technique to the GG project case

**4. Based on the above analysis and assumptions, split the sample list into tables such that each has only one theme. Demonstrate that the modification problems you identified in E have been eliminated.**

Hint: refer to Text Chapter 2 for " Normalization Process " and apply that technique to the GG project case. You can also refer to the 9 possible designs provided in the GG Project Questions section at the end of Chapter 2 in Text book, on pp.114-115, and choose the best one you believe.

**5. Assume that based on your suggestions and their additional requirements, Garden Glory designs a trail database saved as Garden\_Glory.accdb in MS Access 2013. Query the trail database using SQL statements and answer the following questions. For each SQL statement you write, show the results based on your data (you could press the "PrtSc - Print Screen" key to capture the results image) .**

Hint: refer to Text Chapter 3 for " SQL Queries " and apply that knowledge to the GG project case.

**5.1) Write SQL statements to list all columns for all tables.**

**5.2) Write an SQL statement to list the LastName , FirstName and CellPhone for all employees having an experience level of Master .**

**5.3) Write an SQL statement to list the LastName , FirstName and CellPhone for all employees having an experience level of Master and a FirstName that begins with the letter J .**

**5.4) Write an SQL statement to list the Name of employees who have worked on a property in New York .**

**5.5) Write an SQL statement to list the names of employees who have worked on a property owned by a Corporation.**

SELECT EMPLOYEE.LastName, EMPLOYEE.FirstName

FROM EMPLOYEE INNER JOIN ((OWNER INNER JOIN PROPERTY ON OWNER.OwnerID = PROPERTY.OwnerID) INNER JOIN SERVICE ON PROPERTY.PropertyID = SERVICE.PropertyID) ON EMPLOYEE.EmployeeID = SERVICE.EmployeeID

WHERE (([OWNER]![Type]="Corporation"));



**5.6) Write an SQL statement to show the name and sum of HoursWorked for each employee.**

SELECT EMPLOYEE.LastName, EMPLOYEE.FirstName, Sum(SERVICE.HoursWorked) AS SumOfHoursWorked

FROM EMPLOYEE INNER JOIN SERVICE ON EMPLOYEE.EmployeeID = SERVICE.EmployeeID

GROUP BY EMPLOYEE.LastName, EMPLOYEE.FirstName;



**5.7) Write an SQL statement to show the sum of HoursWorked for each ExperienceLevel of EMPLOYEE. Sort the results by ExperienceLevel in descending order.**

SELECT EMPLOYEE.ExperienceLevel, Sum(SERVICE.HoursWorked) AS SumOfHoursWorked

FROM EMPLOYEE INNER JOIN SERVICE ON EMPLOYEE.EmployeeID = SERVICE.EmployeeID

GROUP BY EMPLOYEE.ExperienceLevel

ORDER BY EMPLOYEE.ExperienceLevel DESC;



**5.8) Write an SQL statement to show the sum of HoursWorked for each Type of OWNER but exclude services of employees who have an ExperienceLevel of Junior and exclude any Type with less than three members.**

SELECT OWNER.Type, Sum(SERVICE.HoursWorked) AS SumOfHoursWorked

FROM EMPLOYEE INNER JOIN ((OWNER INNER JOIN PROPERTY ON OWNER.OwnerID = PROPERTY.OwnerID) INNER JOIN SERVICE ON PROPERTY.PropertyID = SERVICE.PropertyID) ON EMPLOYEE.EmployeeID = SERVICE.EmployeeID

GROUP BY OWNER.Type, EMPLOYEE.ExperienceLevel, OWNER.Type

HAVING (([EMPLOYEE]![ExperienceLevel]<>"Junior") AND ([OWNER]![Type]<>"Individual"));



1. **Kroenke, D. M., & Auer, D. J. (2015). *Database Concepts.* (7th, Ed.) Hoboken, NJ: Pearson.** [↑](#footnote-ref-1)