**DSCI 723 Term project**

**Dumpster Fire Golf Course Management Case**

**Sheyma Abdikebir**

Final Submission: April 24, 2024

Contents

[List of figures 2](#_Toc164883185)

[Abstract 3](#_Toc164883186)

[Note 4](#_Toc164883187)

[Course 4](#_Toc164883188)

[Hole 5](#_Toc164883189)

[Fairway 5](#_Toc164883190)

[Greens 5](#_Toc164883191)

[Rough 6](#_Toc164883192)

[Operation 6](#_Toc164883193)

[Problem 6](#_Toc164883194)

[List of functional dependencies 9](#_Toc164883195)

[Data Dictionary 12](#_Toc164883196)

[Potential ethical and privacy issues 32](#_Toc164883197)

[References 33](#_Toc164883198)

List of figures

[Figure 1: ER diagram 7](#_Toc164883249)

[Figure 2: Relational Schema 8](#_Toc164883250)

[Figure 3: Database Diagram 11](#_Toc164883251)

Abstract

Dumpster Fire LLC has many golf courses that need maintenance, but the business currently does not adopt a centralized database to manage these tasks. The absence of this database has impacted on the business’s ability to monitor these crucial activities. Some features of the golf course, such as the green, require regular maintenance to create a good experience for golfers and to keep the resources in the required condition. The company was criticized by golfers for lacking a database.

To mitigate this problem, the owners and the staff of the individual golf courses were interviewed to collect the required information for building the database. Developing the system will introduce various anticipated advantages and drawbacks. The main benefit is the improved customer experience. Since the database will allow the business to record information on the maintenance schedules and other relevant data, it will be able to create consistent experience for the golfers. Additionally, since maintaining golf courses require a noteworthy investment, the database will provide a wealth of information that can allow the business to uncover trends and insights, allowing them to devise an effective maintenance and resource allocation strategies.

As for the risks, there might be an anticipated risk associated with data security since this is a centralized database. If there is a failure at one point in the system, then centralized databases will potentially result in data loss. This data loss will adversely impact the business if there are no backups. Finally, any changes to the database will be made using this single server, making centralized databases expensive to maintain.

Note:

The attributes for each entity were carefully selected after contemplating the role of each entity and the type of attributes that were necessary to describe it. Additionally, the project’s time constraint and the nature of the attributes (i.e. whether temporal or static) were factored in when choosing the attributes. Those that were trivial, temporal, or cannot be objectively measured were excluded from the final ER diagram. Also, compared to other attributes, since aesthetic features are not essential and can’t be measured objectively, they were not included in the diagram.

Course

* The diagram shows CourseID and CourseName as unique keys, but CourseID will be chosen as the primary key.
* For the hazards, these can be grouped into bunkers, water, and vegetation. Since each instance of golf course can have multiple hazards, it is sensible to treat it as a multi-valued attribute instead of creating a separate entity which would increase complexity of the diagram.
* For amenities, it might be essential to track all types of amenities provided in the courses. A golf course owner might want to include/remove an amenity in the future. Hence, to avoid insertion/deletion anomaly, a separate entity was created.
* Some attributes such as the tee boxes, green speed, green, rough etc. were not included as entity attributes since these are better suited to describe the hole or demand a separate entity.
* Since course rating and slope rating give us a picture into the course’s difficulty level, ‘overall challenge’ was not included in the diagram to avoid redundancy.

Hole

* Slope was not included since it is appropriate to include slope as an attribute for green. Additionally, although Pin Placement changes daily, it was included in the diagram as it impacts putting strategy, making it crucial.

For fairways, rough, greens, since there is a separate entity that will track maintenance, there was no need to include maintenance and divot management as attributes. Also, since health varies, it was excluded.

Fairway

The ER diagram noted several attributes such as texture, contours, rolls etc. that are linked to the surface quality and impact playability. Only firmness was selected to be part of the ER diagram. This selection was based on researching how fairway’s surface quality is measured. Most of the resources discussed measuring the firmness in contrast to other attributes. Also, since visibility is temporal, the ER diagram excluded ‘accessibility’.

Greens

HARTWIGER (2014) mentioned that analyzing speed, slope, and firmness of the green is an essential factor that can aid in identifying factors that impact playability and those that create bottlenecks. The author, additionally, mentioned some objective measurement tools, so these factors were included in the ER diagram. According to Windows & Bechelet (2015), smoothness and trueness are essential attributes for top tournament professionals since this helps to perfect the surface. Unless Dumpster Fire LLC is hosting tournaments, these attributes are not of prime importance. Hence, they were not included in the diagram to further simplify it.

Rough

Nicoludis (2021) stated that maintaining a consistent grass level on the rough is expensive and is not feasible. Hence length consistency was excluded from the diagram.

Operation

An entity called operation was created as it might be necessary to track all types of operation that is conducted by the golf course. Creating a separate entity will prevent insertion anomaly if there was a need to include new maintenance type.

Problem

An entity called problem was created to track potential problems. Since problems do not occur frequently, adding those as an attribute to the entity task will result in many empty rows in the database. Hence, it is appropriate to create an entity that will track the history of all problems. Also, since problem description and summary provide the same information, summary was not included.

A diagram of a company

Description automatically generated

Figure 1: ER diagram

A diagram of a computer flowchart

Description automatically generated

Figure 2: Relational Schema

List of functional dependencies

Course\_ID 🡪 Course\_name, Country, City, Street\_Address, Zipcode, State, No\_ofHoles, Course\_rating, Slope\_rating, Yardage, Par

Amenity\_no, Course\_ID 🡪 Type

Hole\_no, Course\_ID 🡪 Yardage, Par, Doglegs, Pin\_placement

Fairway\_ID 🡪 Firmness, width, length, Hole\_no, Course\_ID

Rough\_ID 🡪 Height, Length, Recov\_difficulty, thickness, texture, Penalty, Hole\_no, Course\_ID

Green\_ID 🡪 Length, Width, Square\_Footage, Speed, Green\_type, Firmness, Slope, Difficulty, Hole\_no, Course\_ID

Operation\_ID 🡪 Type

Plan\_Sched\_ID🡪 Plan\_StartTime, Plan\_EndDate, Plan\_EndTime, Plan\_StartDate, Green\_ID, Rough\_ID, Fairway\_ID, Op\_ID

Actual\_Sched\_ID 🡪 Status, Act\_StartDate, Act\_EndDate, Act\_StartTime, Act\_EndTime, Plan\_Sched\_ID

Task\_ID 🡪 Tk\_Startdate, Tk\_EndDate, Comments, Tk\_Start\_Time, Tk\_End\_Time, Actual\_Sched\_ID, Employee\_ID

Serial\_no 🡪 Manufacturer, Name, Type, DatePurchased, Warranty\_status

Pb\_ticketID 🡪 Associated\_equip, Pb\_Date, Pb\_Time, Fix\_Date, Fix\_Time, Pb\_descrip, Employee\_ID

Employee\_ID 🡪 F\_Name, L\_Name, Home\_Phone, Email\_Address, City, State, Street\_Address, ZipCode, Hr\_Rate

A screenshot of a computer

Description automatically generated

Figure 3: Database Diagram

Data Dictionary

List of tables

1. Course 2) Amenity 3) Course\_Hazards 4) Hole 5) Fairway 6) Green 7) Rough 8) Hole\_Teebox 9) uses 10) Equipment 11) Plan\_Schedule 12) Task 13) Problem 14) Employee 15) Operation 16) Actual\_Schedule

Table: dbo.Course

Store specific details on each golf course, including location information, slope rating etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data type | N | Description |
| City | nvarchar(50) |  | City where each course is situated |
| Country | nvarchar(50) |  | Country where each course is situated |
| Course\_ID | nchar(8) |  | Primary key for course records |
| Course\_Name | nvarchar(50) |  | Each course’s name |
| Course\_rating | int |  | Rating of each course |
| No\_ofHoles | tinyint |  | The number of holes for each course |
| Par | tinyint |  | The Par value for each course |
| Slope\_rating | tinyint |  | Slope ratings for each course |
| State | nvarchar(50) |  | State where each course resides in |
| Street\_Address | nvarchar(25) |  | Street address comprising of street name and street number |
| Yardage | int |  | Yardage values for each course |
| ZipCode | nvarchar(15) |  | Zipcode/postal code for each course |

Linked from

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Amenity | dbo.Course.CourseID=dbo.Amenity.CourseID | FK\_Amenity\_Course  Foreign key constraint referencing Course.CourseID |
| dbo.Course\_Hazards | dbo.Course.CourseID = dbo.Course\_Hazards.CourseID | FK\_Course\_Hazards\_Course  Foreign key constraint referencing Course.CourseID |
| dbo.Hole | dbo.Course.CourseID = dbo.Hole.CourseID | FK\_Hole\_Course  Foreign key constraint referencing Course.CourseID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Course\_ID | PK\_Course  Primary key (clustered) constraint |
| Course\_Name | U\_Course\_Name  Unique nonclustered index. |

Used by:

|  |
| --- |
| Name |
| dbo.Course |
| * dbo.Amenity |
| * dbo.Course\_Hazard |
| * dbo.Hole |

Table: dbo.Amenity

Stores information on the different types of amenities that are provided by the golf course.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Amenity\_no | int |  | Primary key for each amenity  Identity/ Auto increment column |
| Course\_ID | nchar(8) |  | Primary key. Foreign key to Course.CourseID |
| Type | nvarchar(50) |  | States the type of amenity |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Course | dbo.Amenity.Course\_ID = dbo.Course.Course\_ID | FK\_Amenity\_Course  Foreign key constraint referencing Course.CourseID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Amenity\_no, Course\_ID | PK\_Amenity  Primary key (clustered) constraint |

Uses

|  |
| --- |
| Name |
| dbo.Amenity |
| * dbo.Course |

Table: dbo.Course\_Hazards

Stores information on each course’s hazards such as water bodies, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Hazards | nvarchar(20) |  | Primary Key for each course hazard |
| Course\_ID | nchar (8) |  | Primary key. Foreign key to Course.CourseID |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Course | dbo.Course\_Hazards.Course\_ID = dbo.Course.Course\_ID | FK\_Course\_Hazards\_Course  Foreign key constraint referencing Course.CourseID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Hazards, Course\_ID | PK\_Course\_Hazards  Primary key (clustered) constraint |

Uses

|  |
| --- |
| Name |
| dbo.Course\_Hazards |
| * dbo.Course |

Table: dbo.Hole

Stores information on attributes of each hole such as par, yardage, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Hole\_no | int |  | Primary Key for hole records  Identity/ Auto increment column |
| Course\_ID | nchar (8) |  | Primary key. Foreign key to Course.CourseID |
| Yardage | int |  | Yardage values for each record |
| Par | tinyint |  | Par values for each record |
| Doglegs | nvarchar (15) |  | Doglegs for each record such as left/right doglegs |
| Pin\_placement | nvarchar (15) |  | Location where the flagstick is on the green |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Course | dbo.Hole.Course\_ID = dbo.Course.Course\_ID | FK\_Hole\_Course  Foreign key constraint referencing Course.CourseID |

Linked from

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Fairway | dbo.Fairway.(Hole\_no, CourseID) = dbo.Hole.(Hole\_no, CourseID) | FK\_Fairway\_Hole  Foreign key constraint referencing Hole.(Hole\_no, CourseID) |
| dbo.Green | dbo.Green.(Hole\_no, CourseID) = dbo.Hole.(Hole\_no, CourseID) | FK\_Green\_Hole  Foreign key constraint referencing Hole.(Hole\_no, CourseID) |
| dbo.Rough | dbo.Rough.(Hole\_no, CourseID) = dbo.Hole.(Hole\_no, CourseID) | FK\_Rough\_Hole  Foreign key constraint referencing Hole.(Hole\_no, CourseID) |
| dbo.Hole\_Teebox | dbo.Hole\_Teebox.(Hole\_no, CourseID)= dbo.Hole.(Hole\_no, CourseID) | FK\_Hole\_Teebox\_Hole  Foreign key constraint referencing Hole.(Hole\_no, CourseID) |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Hole\_no, Course\_ID | PK\_Hole  Primary key (clustered) constraint |

Uses

|  |
| --- |
| Name |
| dbo.Hole |
| * dbo.Course |

Used by

|  |
| --- |
| Name |
| dbo.Hole |
| * dbo.Fairway |
| * dbo.Green |
| * dbo.Rough |
| * dbo.Hole\_Teebox |

Table: dbo.fairway

Stores details on fairways such as their width, firmness etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Fairway\_ID | int |  | Primary Key for fairway records  Identity/ Auto increment column |
| Course\_ID | nchar (8) |  | Foreign key to Course.CourseID |
| Firmness | nvarchar (15) |  | Firmness of each record |
| Hole\_no | int |  | Foreign key to Hole.Hole\_no  Identity/ Auto increment column |
| Length | int |  | Length of each fairway |
| Width | int |  | Width of each fairway |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Hole | dbo.Fairway.(Hole\_no, CourseID) = dbo.Hole.(Hole\_no, CourseID) | FK\_Fairway\_Hole  Foreign key constraint referencing Hole.(Hole\_no, CourseID) |

Linked from

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Plan\_Schedule | dbo.Plan\_Schedule.Fairway\_ID = dbo.Fairway.Fairway\_ID | FK\_Plan\_Schedule\_Fairway Foreign key constraint referencing Fairway.Fairway\_ID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Fairway\_ID | PK\_Fairway  Primary key (clustered) constraint |

Uses

|  |
| --- |
| Name |
| dbo.Fairway |
| * dbo.Hole |

Used by

|  |
| --- |
| Name |
| dbo.Fairway |
| * dbo.Plan\_Schedule |

Table: dbo.Green

Stores details on Green such as their difficulty, type etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Green\_ID | int |  | Primary Key for Green records  Identity/ Auto increment column |
| Course\_ID | nchar (8) |  | Foreign key to Course.CourseID |
| Difficulty | nvarchar(15) |  | Difficulty of each Green |
| Hole\_no | int |  | Foreign key to Hole.Hole\_no  Identity/ Auto increment column |
| Firmness | nvarchar(15) |  | Firmness of each Green |
| Grass\_type | nvarchar(40) |  | Grass type of each Green |
| Length | int |  | Length of each Green |
| Slope | int |  | Slope of each Green |
| Speed | int |  | Speed of each Green |
| Square\_Footage | int |  | Square footage of each Green |
| Width | int |  | Width of each Green |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Hole | dbo.Green.(Hole\_no, CourseID) = dbo.Hole.(Hole\_no, CourseID) | FK\_Green\_Hole  Foreign key constraint referencing Hole.(Hole\_no, CourseID) |

Linked from

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Plan\_Schedule | dbo.Plan\_Schedule.Green\_ID = dbo.Green.Green\_ID | FK\_Plan\_Schedule\_Green  Foreign key constraint referencing Green.Green\_ID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Green\_ID | PK\_Green  Primary key (clustered) constraint |

Uses

|  |
| --- |
| Name |
| dbo.Green |
| * dbo.Hole |

Used by

|  |
| --- |
| Name |
| dbo.Green |
| * dbo.Plan\_Schedule |

Table: dbo.Rough

Stores details on Rough such as their recovery difficulty, penalty etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Rough\_ID | int |  | Primary Key for Rough records  Identity/ Auto increment column |
| Course\_ID | nchar (8) |  | Foreign key to Course.CourseID |
| Height | int |  | Height of each Rough |
| Hole\_no | int |  | Foreign key to Hole.Hole\_no  Identity/ Auto increment column |
| Length | int |  | Length of each Rough |
| Penalty | nvarchar (30) |  | Penalty for each Rough |
| Recov\_difficulty | nvarchar (30) |  | Recovery difficulty for each Rough |
| Texture | nvarchar (15) |  | Texture for each Rough |
| Thickness | nvarchar (12) |  | Thickness for each Rough |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Hole | dbo.Rough.(Hole\_no, CourseID) = dbo.Hole.(Hole\_no, CourseID) | FK\_Rough\_Hole  Foreign key constraint referencing Hole.(Hole\_no, CourseID) |

Linked from

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Plan\_Schedule | dbo.Plan\_Schedule.Rough\_ID = dbo.Rough.Rough\_ID | FK\_Plan\_Schedule\_Rough  Foreign key constraint referencing Rough.Rough\_ID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Rough\_ID | PK\_Rough  Primary key (clustered) constraint |

Uses

|  |
| --- |
| Name |
| dbo.Rough |
| * dbo.Hole |

Used by

|  |
| --- |
| Name |
| dbo.Rough |
| * dbo.Plan\_Schedule |

Table: dbo.Hole\_Teebox

Stores details on each Teebox. These are color coded, indicating different difficulty levels.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Teebox | nvarchar(12) |  | Primary Key for Teebox records |
| Course\_ID | nchar (8) |  | Primary key. Foreign key to Course.CourseID |
| Hole\_no | int |  | Primary Key. Foreign key to Hole.Hole\_no  Identity/ Auto increment column |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Hole | dbo.Hole\_Teebox.(Hole\_no, CourseID) = dbo.Hole.(Hole\_no, CourseID) | FK\_Hole\_Teebox\_Hole  Foreign key constraint referencing Hole.(Hole\_no, CourseID) |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Teebox, Hole\_no, Course\_ID | PK\_Hole\_Teebox  Primary key (clustered) constraint |

Uses

|  |
| --- |
| Name |
| dbo.Hole\_Teebox |
| * dbo.Hole |

Table: dbo.Uses

This is a linking table. Since each task can use multiple equipment and each equipment can be used in multiple tasks, the Uses table stores all the instances in which each equipment was used.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Serial\_no | nchar (8) |  | Primary Key for Uses records. Foreign key to Equipment.Serial\_no |
| Task\_ID | int |  | Primary Key for Uses records. Foreign key to Task.Task \_ID |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Equipment | dbo.Uses.Serial\_no = dbo.Equipment.Serial\_no | FK\_Uses\_Equipment  Foreign key constraint referencing Equipment.Serial\_no |
| dbo.Task | dbo.Uses.Task\_ID = dbo.Task.Task\_ID | FK\_Uses\_Task  Foreign key constraint referencing Task.Task\_ID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Serial\_no, Task\_ID | PK\_Uses  Primary key (clustered) constraint |

Uses

|  |
| --- |
| Name |
| dbo.Uses |
| * dbo.Equipment |
| * dbo.Task |

Table: dbo.Equipment

Stores details on equipment such as their serial number, manufacturer, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| DatePurchased | date |  | Date when equipment was purchased |
| Manufacturer | nvarchar(50) |  | Manufacturer of each equipment |
| Name | nvarchar(35) |  | Name of each Equipment |
| Serial\_no | nchar(8) |  | Primary Key for each equipment |
| Type | nvarchar(30) |  | Type of each equipment |
| Warranty\_status | nvarchar(10) |  | Warranty status for each equipment |

Linked from

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Uses | dbo. Equipment.Serial\_no = dbo. Uses.Serial\_no | FK\_Uses\_Equipment  Foreign key constraint referencing Equipment.EquipmentID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Serial\_no | PK\_Equipment  Primary key (clustered) constraint |

Used by

|  |
| --- |
| Name |
| dbo.Equipment |
| * dbo.Uses |

Table: dbo.Plan\_Schedule

Stores details on the planned schedule, including the start time and date for each maintenance instance.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Plan\_Sched\_ID | int |  | Primary key for each planned schedule records  Identity/ Auto increment column |
| Plan\_StartDate | date |  | The maintenance schedule’s planned start date |
| Plan\_StartTime | time(1) |  | The maintenance schedule’s planned start time |
| Rough\_ID | int |  | Foreign key to Rough.Rough\_ID  Identity / Auto increment column |
| Plan\_EndTime | time(1) |  | The maintenance schedule’s planned end time |
| Plan\_EndDate | date |  | The maintenance schedule’s planned end date |
| Op\_ID | nchar(8) |  | Foreign key to Operation.Op\_ID |
| Green\_ID | int |  | Foreign key to Green.Green\_ID  Identity / Auto increment column |
| Fairway\_ID | int |  | Foreign key to Fairway.Fairway\_ID  Identity / Auto increment column |

Linked from

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Actual\_Schedule | dbo.Plan\_Schedule.Plan\_Sched\_ID =dbo.Actual\_Schedule.Plan\_Sched\_ID | FK\_Actual\_Schedule\_Plan\_Schedule  Foreign key constraint referencing Plan\_Schedule. Plan\_Sched\_ID |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Fairway | dbo.Plan\_Schedule.Fairway\_ID = dbo.Fairway.Fairway\_ID | FK\_Plan\_Schedule\_Fairway  Foreign key constraint referencing  Fairway.Fairway\_ID |
| dbo.Green | dbo.Plan\_Schedule.Green\_ID = dbo.Green.Green\_ID | FK\_Plan\_Schedule\_Green  Foreign key constraint referencing Green.Green\_ID |
| dbo.Rough | dbo.Plan\_Schedule.Rough\_ID = dbo.Rough.Rough\_ID | FK\_Plan\_Schedule\_Rough  Foreign Key constraint referencing  Rough.Rough\_ID |
| dbo.Operation | dbo.Plan\_Schedule.Op\_ID = dbo.Operation.Op\_ID | FK\_Plan\_Schedule\_Operation  Foreign key constraint referencing Operation.Op\_ID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Plan\_Sched\_ID | PK\_Plan\_Schedule  Primary key (clustered) constraint |

Used by

|  |
| --- |
| Name |
| dbo.Plan\_Schedule |
| * dbo.Actual\_Schedule |

Uses

|  |
| --- |
| Name |
| dbo.Plan\_Schedule |
| * dbo.Operation |
| * dbo.Fairway |
| * dbo.Green |
| * dbo.Rough |

Table: dbo.Task

Stores details on each maintenance task, including starting and ending information.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Task\_ID | int |  | Primary key for each maintenance task’s records  Identity/ Auto increment column |
| Actual\_Sched\_ID | int |  | Foreign key to Actual\_Schedule. Actual\_Sched\_ID |
| Comments | nvarchar(255) |  | Comments from employee performing the task |
| Employee\_ID | int |  | Foreign key to Employee.Employee\_ID  Identity / Auto increment column |
| Tk\_End\_Time | time(1) |  | Each task’s end time |
| Tk\_EndDate | date |  | Each task’s end Date |
| Tk\_Start\_Time | time(1) |  | Each task’s start time |
| Tk\_StartDate | date |  | Each task’s start date |

Linked from

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Uses | dbo.Task.Task\_ID =dbo.Uses.Task\_ID | FK\_Uses\_Task  Foreign key constraint referencing Task.Task\_ID |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Actual\_Schedule | dbo.Task.Actual\_Sched\_ID = dbo.Actual\_Schedule.Actual\_Sched\_ID | FK\_Task\_Actual\_Schedule  Foreign key constraint referencing Actual\_Schedule.Actual\_Sched\_ID |
| dbo.Employee | dbo.Task.Employee\_ID = dbo.Employee.Employee\_ID | FK\_Task\_Employee  Foreign key constraint referencing Employee.Employee\_ID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Task\_ID | PK\_Plan\_Task  Primary key (clustered) constraint |

Used by

|  |
| --- |
| Name |
| dbo.Task |
| * dbo.Uses |

Uses

|  |
| --- |
| Name |
| dbo.Task |
| * dbo.Actual\_Schedule |
| * dbo.Employee |

Table: dbo.Problem

Stores details on problems that each employee faced while performing the maintenance task, including problem description, time, date, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Pb\_ticketID | nchar(8) |  | Primary key for each reported problem |
| Associated\_equip | nvarchar(35) |  | Equipment associated with the reported problem |
| Fix\_Date | date | N | Date when the problem was resolved |
| Employee\_ID | int |  | Foreign key to Employee.Employee\_ID  Identity / Auto increment column |
| Fix\_Time | time(1) | N | Time when the problem was resolved |
| Pb\_Date | date |  | Date when the problem was reported |
| Pb\_descrip | nvarchar(255) |  | Description of the reported problem |
| Pb\_Time | time(1) |  | Time when the problem was reported |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Employee | dbo.Problem.Employee\_ID = dbo.Employee.Employee\_ID | FK\_Problem\_Employee  Foreign key constraint referencing Employee.Employee\_ID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Pb\_ticketID | PK\_Problem  Primary key (clustered) constraint |

Uses

|  |
| --- |
| Name |
| dbo.Problem |
| * dbo.Employee |

Table: dbo.Employee

Stores details on each employee including their first and last names.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Employee\_ID | int |  | Primary key for each employee records  Identity / Auto increment column |
| City | nvarchar(30) |  | Each employee’s city of residence |
| Email\_Address | nvarchar(255) |  | Each employee’s email address |
| F\_Name | nvarchar(60) |  | Stores each employee’s first name |
| Home\_Phone | nvarchar(25) |  | Stores each employee’s home phone number |
| Hr\_Rate | money |  | Stores each employee’s hourly pay rate |
| L\_Name | nvarchar(60) |  | Each employee’s last name |
| State | nvarchar(50) |  | Each employee’s state |
| Street\_Address | nvarchar(25) |  | Each employee’s street address comprising street number and street name |
| ZipCode | nvarchar(15) |  | Each employee’s Zipcode |

Linked from

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Problem | dbo.Employee.Employee\_ID = dbo.Problem.Employee\_ID | FK\_Problem\_Employee  Foreign key constraint referencing Employee.Employee\_ID |
| dbo.Task | dbo.Employee.Employee\_ID = dbo.Task.Employee\_ID | FK\_Task\_Employee  Foreign key constraint referencing Employee.Employee\_ID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Employee\_ID | PK\_Employee  Primary key (clustered) constraint |

Used by

|  |
| --- |
| Name |
| dbo.Employee |
| * dbo.Problem |
| * dbo.Task |

Table: dbo.Operation

Stores details on each operation that was performed, including its type, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Op\_ID | nchar(8) |  | Primary key for each operation records |
| Type | nvarchar(25) |  | Each operation’s type such as mowing, irrigation, etc. |

Linked from

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Plan\_Schedule | dbo.Operation.Op\_ID = dbo.Plan\_Schedule.Op\_ID | FK\_Plan\_Schedule\_Operation  Foreign key constraint referencing Operation.Op\_ID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Op\_ID | PK\_Operation  Primary key (clustered) constraint |

Used by

|  |
| --- |
| Name |
| dbo.Operation |
| * dbo.Plan\_Schedule |

Table: dbo.Actual\_Schedule

Stores details on each actual maintenance schedule, including status, date of occurrence, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Columns | Data type | N | Description |
| Actual\_Sched\_ID | int |  | Primary key for each actual schedule records  Identity/ Auto increment column |
| Act\_EndDate | date |  | Date when the actual schedule ended |
| Act\_EndTime | time(1) |  | time when the actual schedule ended |
| Act\_StartDate | date |  | Date when the actual schedule started |
| Act\_StartTime | time(1) |  | time when the actual schedule started |
| Plan\_Sched\_ID | int |  | Foreign key to Plan\_Schedule. Plan\_Sched\_ID  Identity/ Auto increment column |
| Status | nvarchar(10) |  | Operation status of each actual schedule e.g.: completed, cancelled, etc. |

Linked from

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Task | dbo.Actual\_Schedule. Actual\_Sched\_ID = dbo.Task. Actual\_Sched\_ID | FK\_Task\_Actual\_Schedule  Foreign key constraint referencing Actual\_Schedule.Actual\_Sched\_ID |

Links to

|  |  |  |
| --- | --- | --- |
| Table | Join | Description |
| dbo.Plan\_Schedule | dbo.Actual\_Schedule. Plan\_Sched\_ID = dbo.Plan\_Schedule. Plan\_Sched\_ID | FK\_Actual\_Schedule\_Plan\_Schedule  Foreign key constraint referencing Plan\_Schedule. Plan\_Sched\_ID |

Unique keys:

|  |  |
| --- | --- |
| Columns | Description |
| Actual\_Sched\_ID | PK\_Actual\_Schedule  Primary key (clustered) constraint |

Uses

|  |
| --- |
| Name |
| dbo.Actual\_Schedule |
| * dbo.Plan\_Schedule |

Used by

|  |
| --- |
| Name |
| dbo.Actual\_Schedule |
| * dbo.Task |

Potential ethical and privacy issues

The newly designed database stores each employees’ personal details such as their address, first and last names. This can have potential ethical and privacy implications if Dumpster Fire LLC does not implement strong data privacy and security measures. It should be noted, however, that the federal trade commission (FTC) acknowledges that even with precautionary measures such as enacting privacy policies in the organization, the security of the database can sometimes be compromised (FTC, 2003). Prior to storing employees’ personal information, Dumpster Fire LLC should receive their consent to avoid ethical issues such as those discussed below.

If security of the database is not maintained, one potential issue is data breach, leading to identity theft. Hence, one way to avoid this issue is by encrypting personal/sensitive information. Another privacy and ethical concern occur if third party entities have access to private information to perform their duties. The issue arises if they fail to uphold high standards of ethical and data privacy practices, potentially exposing the data to cybersecurity attacks. An example of unethical behavior is the selling of these private information to other entities for targeting Dumpster Fire LLC’s employees with marketing messages without their consent, infringing on their privacy rights. Another concern is if Dumpster Fire LLC uses this personal information for unintended purposes such as sending marketing materials to their employees.

Lastly, according to Management Concepts (2023), to ethically gather data, the concept of data retention and de-identification are crucial. The author mentioned that storing information for an extended period can lead to potential identity theft among other security concerns. Additionally, the article mentioned that if data is not de-identified, hackers can gain unauthorized access and re-identify it in order to use it for unauthorized purposes. Hence, Dumpster Fire LLC should not store personal information beyond the required duration to ensure data security.

References

FTC. (2003, November 19). *FTC working to protect consumers and businesses from information security breaches*. Federal Trade Commission. <https://www.ftc.gov/news-events/news/press-releases/2003/11/ftc-working-protect-consumers-businesses-information-security-breaches>

HARTWIGER, C. (2014). The Big Three of Putting Green Setup: Firmness, Slope, and Speed. Green Section Record, 52(4), 1–3.

Management Concepts. (2023, August 29). *The ethics of data collection and analytics*. Broaden your understanding of impactful issues, best practices, and trending topics. <https://resources.managementconcepts.com/download/analytics/the-ethics-of-data-collection-and-analytics/>

Nicoludis, Z. (2021, July 2). *Why the rough isn’t perfect*. USGA. <https://www.usga.org/content/usga/home-page/course-care/green-section-record/59/12/why-the-rough-isn-t-perfect.html>

Windows, R., & Bechelet, H. (2015, February 14). *Perfectly true*. USGA. <https://www.usga.org/articles/2010/02/perfectly-true-55333.html>