Logo

Description automatically generatedText

Description automatically generated with low confidence

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| Student ID | : | 2019160 |
| Student UoW ID | : | w1810872 |
| Student Name | : | Chithara Jithmanthi Karunasekera |
| Tutorial Group | : | Group C |
| Due Date | : | 6th December 2021 |

**Module: 5COSC020C – Database Systems**

**Module Leader: Mr. Ragu Sivaraman**

**INDIVIDUAL COURSEWORK - Part A + B**

**Informatics Institute of Technology**

**Department of Computing**

**(B.Eng.) in Software Engineering**

Contents

[Table of figures 2](#_Toc89523189)

[Part A 3](#_Toc89523190)

[1. Conceptual ERD for ArchipelagoCrazy (Part – A). 3](#_Toc89523191)

[2. Data dictionary to document entities for ArchipelagoCrazy 4](#_Toc89523192)

[(Part–A). 4](#_Toc89523193)

[3. Data Dictionary to document relationships and multiplicities for ArchipelagoCrazy (Part – A). 6](#_Toc89523194)

[3.1. Binary Relationships. 6](#_Toc89523195)

[3.2. Turnery Relationships. 11](#_Toc89523196)

[4. Data dictionary to document attributes and primary keys for each entity for ArchipelagoCrazy (Part – A). 13](#_Toc89523197)

[Part B 16](#_Toc89523198)

[5. LOGICAL ERD for SoundStuff (Part B). 16](#_Toc89523199)

[6. SQL queries to create tables, populate tables and retieving required output (Part B). 17](#_Toc89523200)

[6.1. SQL Queries to create tables. 17](#_Toc89523201)

[6.2. SQL Queries to populate tables. 19](#_Toc89523202)

[6.3. Studio table and Equipment table after populating 21](#_Toc89523203)

[6.4. SQL Query to retrieve studios in London, along with makes, series, and models of equipment that cost more than £125 a day to hire. 22](#_Toc89523204)

[(Cost per day displayed to confirm output). 22](#_Toc89523205)

[7. Comparative analysis table (Part B). 23](#_Toc89523206)

[8. References 26](#_Toc89523207)

# Table of figures

[Figure 1: Create table Studio 17](file:///D:\IIT\2021-2022%20Level%205\DB%20Systems\Course%20Work\Part%20B\Part-A+B.docx#_Toc89520814)

[Figure 2: Successfully created Studio table 17](#_Toc89520815)

[Figure 3: Create table Equipment 18](#_Toc89520816)

[Figure 4: Successfully created table Equipment 18](#_Toc89520817)

[Figure 5: Populating studio table with data 19](#_Toc89520818)

[Figure 6: Successfully inserted data to Studio table 19](#_Toc89520819)

[Figure 7: Populate Equipment table with data 20](#_Toc89520820)

[Figure 8: Successfully inserted data to Equipment table 20](#_Toc89520821)

[Figure 9: Studio table with sample data 21](#_Toc89520822)

[Figure 10: Equipment table with sample data 21](#_Toc89520823)

[Figure 11: SQL Query for required output 22](#_Toc89520824)

# Part A

# 1. Conceptual ERD for ArchipelagoCrazy (Part – A).

Diagram, schematic

Description automatically generated

# 2. Data dictionary to document entities for ArchipelagoCrazy

# (Part–A).

|  |  |
| --- | --- |
| *Entity name* | *Description* |
| w1810872\_Island | General term describing all islands with attractive sites including mainland, that belong to ArchipelagoCrazy. |
| w1810872\_Boat | General term describing all types of boats that belong to ArchipelagoCrazy. |
| w1810872\_Customer | General term that describes all customers of ArchipelagoCrazy. |
| w1810872\_Booking | General term that describes bookings placed for services offered by ArchipelagoCrazy. |
| w1810872\_Adventure | General term that describes all adventures organised by ArchipelagoCrazy. |
| w1810872\_Employee | General term describing all staff members employed by ArchipelagoCrazy. |
| w1810872\_BoatService | General term describing all types of services done to boats. |
| w1810872\_Meal | General term describing all meals organized by ArchipelagoCrazy. |

|  |  |  |
| --- | --- | --- |
| *General entity* | *Specialized entity* | *Explanation* |
| w1810872\_Boat | w1810872\_LargMotorizedBoat | A specialized term that describes large boats which can handle higher number of customers. |
| w1810872\_PaddlePropelledBoat | A specialized term that describes a boat which can be used for marine exploration. |
| w1810872\_SmallMotorizedBoat | A specialized term that describes small boat which can be used for local beach taxi service. |
| w1810872\_Adventure | w1810872\_SeaAdventure | A specialised term that describes adventures related to sea. |
| w1810872\_LandAdventure | A specialised term that describes adventures related to land. |
| w1810872\_SeaAdventure | w1810872\_SeaCrossing | A specialised term describing the sea adventure of crossing between one or more islands. |
| w1810872\_BeachTaxiService | A specialised term describing the sea adventure of moving between beaches. |
| w1810872\_MarineExploration | The specialised term that describes the sea adventure of exploring islands, accessing remote bays and creeks. |
| w1810872\_LandAdventure | w1810872\_VisitLandmark | A specialised term that describes the land adventure of visiting landmark. |
| w1810872\_WalkingTour | A specialised term that describes the land adventure of a walking tour with a guid. |
| w1810872\_Emplyee | w1810872\_Guid | A specialised term that describes the tour guid who guides customers as an employee |
| w1810872\_SeaCrewMember | A specialised term for members of the sea crew who organises and manages boat service as an employee. |
| w1810872\_BoatMechanic | A specialised term that describes boat mechanics who services boats as an employee. |

# 3. Data Dictionary to document relationships and multiplicities for ArchipelagoCrazy (Part – A).

## 3.1. Binary Relationships.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Entity name* | *Multiplicity* | *Relationship* | *Multiplicity* | *Entity name* | *Justifications for the multiplicity*  *(4 statements for each*  *relationship)* |
| w1810872\_Island | 1..1 | has | 1..\* | w1810872\_Adventure | An island must have at least one adventure |
| An island may have many adventures |
| An adventure must belong to at least one island |
| An adventure must belong to one and only one island |
| w1810872\_Customer | 1..1 | places | 1..\* | w1810872\_Booking | One customer must place at least one booking |
| One customer may place many bookings |
| One booking must belong to at least one customer |
| One booking must belong to one and only one customer |
| w1810872\_Boat |  | carries |  | w1810872\_Customer | A boat may or may not carry a customer to an island. |
| A boat may carry many customers to one islands. |
| A customer must use at least one boat to visit an island. |
| A customer must use one and only one boat to visit an island. |
| w1810872\_Booking | 0..\* | consists of | 1..\* | w1810872\_Adventure | One booking consists of at least one adventure |
| One booking may consist of many adventures |
| One adventure may or may not be booked. |
| One adventure may or may be booked by many. |
| w1810872\_LargeMotorizedBoat | 0..1 | carries | 0..\* | w1810872\_Customer | A large motorized boat may or may not carry any customer |
| A large motorized boat may carry many customers. |
| A customer may or may not be carried by any large motorized boat. |
| A customer must be carried by one and only one large motorized boat. |
| w1810872\_PaddlePropelledBoat | 0..1 | carries | 1..2 | w1810872\_Customer | A paddle boat must carry at least one customer |
| A paddle boat can carry a maximum of two customers |
| A customer may or may not be carried by a paddle boat |
| A customer may be carried by one and only one paddle boat |
| w1810872\_SmallerMotorizedBoat | 0..1 | carries | 0..8 | w1810872\_Customer | A small motorized boat may or may not carry any customer. |
| A small motorized boat may carry a maximum of eight customers. |
| A customer may or may not be carried by any small motorized boat. |
| A customer may be carried by one and only one small motorized boat. |
| w1810872\_Guid | 1..1 | runs | 0..1 | w1810872\_LandAdventure | One guide may or may not run any land adventure. |
| One guide may run one and only one land adventure |
| One land adventure must be run by at least one guide |
| One land adventure must be run by one and only one guide |
| w1810872\_SeaCrewMember | 1..\* | manages | 0..\* | w1810872\_SeaAdventuere | One member of sea crew may or may not manage any sea adventure |
| One sea crew member may manage many sea adventures |
| One sea adventure must be managed by at least one member of sea crew |
| One sea adventure may be managed by many sea crew members |
| w1810872\_SeaCrossing | 0..1 | uses | 1..1 | w1810872\_LargeMotorizedBoat | One sea crossing must use at least one large motorized boat. |
| One sea crossing must use one and only one large motorized boat. |
| One large motorized boat may or may not be used for an sea crossing. |
| One large motorized boat may be used for one and only one sea crossing. |
| w1810872\_BrachTaxiService | 0..1 | uses | 1..1 | w1810872\_SmallerMotorizedBoat | One beach taxi service must use at least one smaller motorized boat. |
| One beach taxi service may use one and only one smaller motorized boat. |
| One smaller motorized boat may or may not be used by any beach taxi service. |
| One smaller motorized boat may be used for one and only one beach taxi service. |
| w1810872\_MarineExploration | 0..1 | uses | 1..1 | w1810872\_PaddlePropelledBoat | One marine exploration must use at least one paddle boat |
| One marine exploration may use one and only one paddle boat |
| One paddle boat may or may not be used by any marine exploration |
| One paddle boat may be used by one and only one paddle boat |
| w1810872\_SeaCrossing | 1..1 | provides | 0..\* | w1810872\_Meal | One sea crossing may or may not provide a meal. |
| One sea crossing may provide many meals. |
| One meal must be provided by at least one sea crossing |
| One meal may be provided by one and only one sea crossing |

## 3.2. Turnery Relationships.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Entity name* | *Multiplicity* | *Relationship* | *Multiplicity* | *Entity name* | *Justifications for the multiplicity* |
| w1810872\_Boat | 1..1 | carries | 0..\*  1..1 | w1810872\_Customer  w1810872\_Island | A boat may or may not carry any customer to at least one island. |
| A boat may carry many customers to one and only one island. |
| A customer may be carried to at least one island by at least one boat |
| A customer may be carried to one and only one island by one and only one boat |
| An island may or may not be carried customers by at least one boat. |
| An island may be carried many customers by one and only one boat. |
| w1810872\_BoatMechanic | 0..\* | provides | 0..\*  0..\* | w1810872\_BoatService  w1810872\_Boat | One Boat mechanic may or may not provide any boat service to any boat. |
| One boat mechanic may provide many services to many boats. |
| One boat service may or may not be provided to any boat by any boat mechanic. |
| One boat service may be provided to many boats by many boat mechanics. |
| One boat may or may not be provided any service by any boat mechanic. |
| One boat may be provided many services by many boat mechanics. |

# 4. Data dictionary to document attributes and primary keys for each entity for ArchipelagoCrazy (Part – A).

|  |  |  |
| --- | --- | --- |
| *Entity name* | *Attribute for this entity* | *justification* |
| w1810872\_Island | w1810872\_islandId {PK}  w1810872\_islandName  w1810872\_location | To uniquely identify every island  Ever island has a name  Every island has its own geographic location |
| w1810872\_Boat | w1810872\_boatId {PK}  w1810872\_condition  w1810872\_boatType  w1810872\_capacity | Uniquely identifies a boat  About the condition status of the boat  Type of the boat  Capacity of each boat |
| w1810872\_LargeMotorizedBoat | w1810872\_currentLocation | Current location where the large, motorized boat is |
| w1810872\_PaddlePropelledBoat | w1810872\_locationToVisit | Location it must visit if available |
| w1810872\_SmallerMotorizedBoat | w1810872\_currentBeach | Current beach it is located in |
| w1810872\_Customer | w1810872\_customerId {PK}  w1810872\_customerName  w1810872\_fName  w1810872\_lName  w1810872\_emailAddress | Uniquely identifies a customer  First name of customer  Last name of customer  E-mail address of customer |
| w1810872\_Booking | w1810872\_bookingId {PK}  w1810872\_bookingDate  w1810872\_duration | Each booking has a number to uniquely identify  Date the booking was made  Time duration taken for a particular event booked |
| w1810872\_Adventure | w1810872\_adventureId {PK}  w1810872\_adventureType  w1810872\_timeDuration | Each adventure has a unique ID number  Type of adventure  Time consuming to complete adventure |
| w1810872\_SeaAdventure | w1810872\_requirement  w1810872\_numberOfGuests | Requirements for the sea adventure  Number of guests participating in an adventure |
| w1810872\_SeaCrossing | w1810872\_crossingLocation  w1810872\_mealRequired | To which island is the sea crossing is for  Whether a meal is required or not |
| w1810872\_BeachTaxiService |  |  |
| w1810872\_MarineExploration |  |  |
| w1810872\_LandAdventure | w1810872\_description | Short description about the land adventure |
| w1810872\_VisitLandmark |  |  |
| w1810872\_WalkTour |  |  |
| w1810872\_Employee | w1810872\_emplyeeId {PK}  w1810872\_name  w1810872\_fName  w1810872\_lName  w1810872\_Address  w1810872\_street  w1810872\_city  w1810872\_postcode  w1810872\_tellNo[1..3] | Each employee has an ID to be uniquely identified  First name of employee  Last name of employee  Street name of address of employee  City of employee’s address  Postcode of address  Telephone number of each employee |
| w1810872\_Guid | w1810872\_specializedArea | Specialise area of study regarding various landmarks |
| w1810872\_SeaCrewMember | w1810872\_licenseNumber | Diving/ Swimming license number of each sea crew member |
| w1810872\_BoatMechanic |  |  |
| w1810872\_BoatService | w1810872\_serviceId {PK}  w1810872\_serviceName  w1810872\_serviceType | To uniquely identify each service  Name of service  Type of service based on various parts of boat |
| w1810872\_Meal | w1810872\_mealId {PK}  w1810872\_description  w1810872\_veg | Uniquely identifies each meal  Short description about what is included in the meal  Is the meal vegetarian or non-vegetarian |

# Part B

# 5. LOGICAL ERD for SoundStuff (Part B).

Diagram

Description automatically generated

# 6. SQL queries to create tables, populate tables and retrieving required output (Part B).

## 6.1. SQL Queries to create tables.

Graphical user interface, text, application, chat or text message

Description automatically generatedCreate table Studio.

Figure : Create table Studio

Successfully created table studio.

Graphical user interface, text, application

Description automatically generated

Figure : Successfully created Studio table

Create table Equipment

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure : Create table Equipment

Successfully created table equipment.

Graphical user interface, application, Word

Description automatically generated

Figure : Successfully created table Equipment

## 6.2. SQL Queries to populate tables.

Populating Studio table.

Graphical user interface, text

Description automatically generated

Figure : Populating studio table with data

Successfully inserted data to Studio table.

Graphical user interface, application

Description automatically generated with medium confidence

Figure : Successfully inserted data to Studio table

Populate Equipment table with data.

Graphical user interface, text, application

Description automatically generatedFigure : Populate Equipment table with data

Successfully inserted data to equipment table.

Graphical user interface

Description automatically generated with medium confidence

Figure : Successfully inserted data to Equipment table

## 6.3. Studio table and Equipment table after populating

Graphical user interface

Description automatically generated with medium confidence

Figure : Studio table with sample data

A screenshot of a computer

Description automatically generated with low confidence

Figure : Equipment table with sample data

# 6.4. SQL Query to retrieve studios in London, along with makes, series, and models of equipment that cost more than £125 a day to hire.

# (Cost per day displayed to confirm output).

SQL Query for the required output

Graphical user interface, text, application, email

Description automatically generated

Figure : SQL Query for required output

Output based on required conditions.

Graphical user interface

Description automatically generated with medium confidence

# 7. Comparative analysis table (Part B).

|  |  |  |
| --- | --- | --- |
|  | **Relational Databases** | **NoSQL databases** |
| **schemas** | In relational databased as a Fixed schema is used, it is required to it is required to define the schema before adding data. Changing schema in this type of database could be extremely expensive and require time consuming service interruptions. | Allowed to use unstructured data as Dynamic schemas are used. Therefore, NoSQL database can be applied without having to define the schema first. It is easy to change data as requirements change. |
| **data consistency** | High consistency (Follows acid properties such as Atomicity, Isolation, Durability). Not patrician tolerance. Provisioned to single server. | Eventually consistent. Database with BASE consistency mode. Prefers availability over consistency of replicated data at write time. |
| **storage** | Stores data according to a specific schema. Has fixed rows and columns. Data is stored in a tabular structure. | Can be stored in any structure, by providing a way to update that data when changing structure. Data is stored in Document based, graph based, as key-value pairs, or as wide -column stores |
| **performance** | Vertically scalable, but expensive. | Allows to work without any schema with unstructured data. Less expensive. |
| **workload** | Flexible, Low redundancy, easy to backup data, and simple diester recovery as some facts that make the workload more efficient though its costly. | Availability over consistency, key-value pairs and flexible schemas make the work much easier with an efficient cost. |
| **infrastructures** | Also known as SQL databases. Has a tabular structure, with strict pre-defined schema. Vertically scaled and more expensive. | Document based, graph databases, key-value pairs or wide column stores are known as non-relational databases. Horizontally scaled and more cost-effective. |
| **security** | Role-based security, encrypted communications, and support for row and field access control, as well as access control through user-level permissions on stored procedures, are all integrated components of relational database security. | Authentication and encryption are practically non-existent in NoSQL databases or are extremely poor when implemented. External encryption techniques such as LDAP, Kerberos, and others are not supported. The data files do not have encryption functionality. Both the client and the servers have weak authentication. |
| **Scaling** | Vertically scalable. As relational databases use single database to host databases it is required to have bigger expensive servers. | Cheap commodity servers can be used in NoSQL databases as the capacity can be added by scaling horizontally. |
| **Data structure** | Table based as relational database was introduced during a time that data was mostly structured and defined by their relationships. | Document based/ graph based/ as key-value pairs/ or as wide -column stores. As mow data is much more complex NoSQL databases are designed to handle complex, unstructured data  Ex: text, social media posts, phots, email |
| **Relationships** | Database connections are associations between tables that are built by retrieving data using join statements. On each side of the relationship, there can only be one record in each table. Each primary key value corresponds to either none or one entry in the corresponding table. | Can store relationship data. Stored differently than in relational databases. Relational data don’t split between tables. NoSQL databases may have modelling relationship data. |
| **Examples** | Microsoft SQL server, Oracle databases, MySQL, and IBM DB2 | MongoDB, CouchBase, Cassandra, HBase, Redis, Riak, Neo4J |

# 8. References

* Oracle.com. 2021. *What is a relational database?*. [online] Available at: <https://www.oracle.com/database/what-is-a-relational-database/> [Accessed 25 November 2021].
* Docs.microsoft.com. 2021. *Relational vs. NoSQL data*. [online] Available at: <https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data> [Accessed 25 November 2021].
* MongoDB. 2021. *NoSQL vs Relational Databases*. [online] Available at: <https://www.mongodb.com/scale/nosql-vs-relational-databases> [Accessed 25 November 2021].
* Education, I., 2021. *relational-databases*. [online] Ibm.com. Available at: <https://www.ibm.com/cloud/learn/relational-databases> [Accessed 25 November 2021].
* Connolly, T. and Begg, C., n.d. *Database systems* A Practical Approach to Design, Implementation, and Management.
* Dindoliwala, V. and Morena, R., 2018. Comparative Study of Integrity Constraints, Storage and Profile Management of Relational and Non-Relational Database using MongoDB and Oracle. *International Journal of Computer Sciences and Engineering*, 6(7), pp.831-837.