

UNIVERSITY OF
WESTMINSTER



INFORMATICS
INSTITUTE OF
TECHNOLOGY

5COSC020W DATABASE SYSTEMS
COURSEWORK (2022/2023)

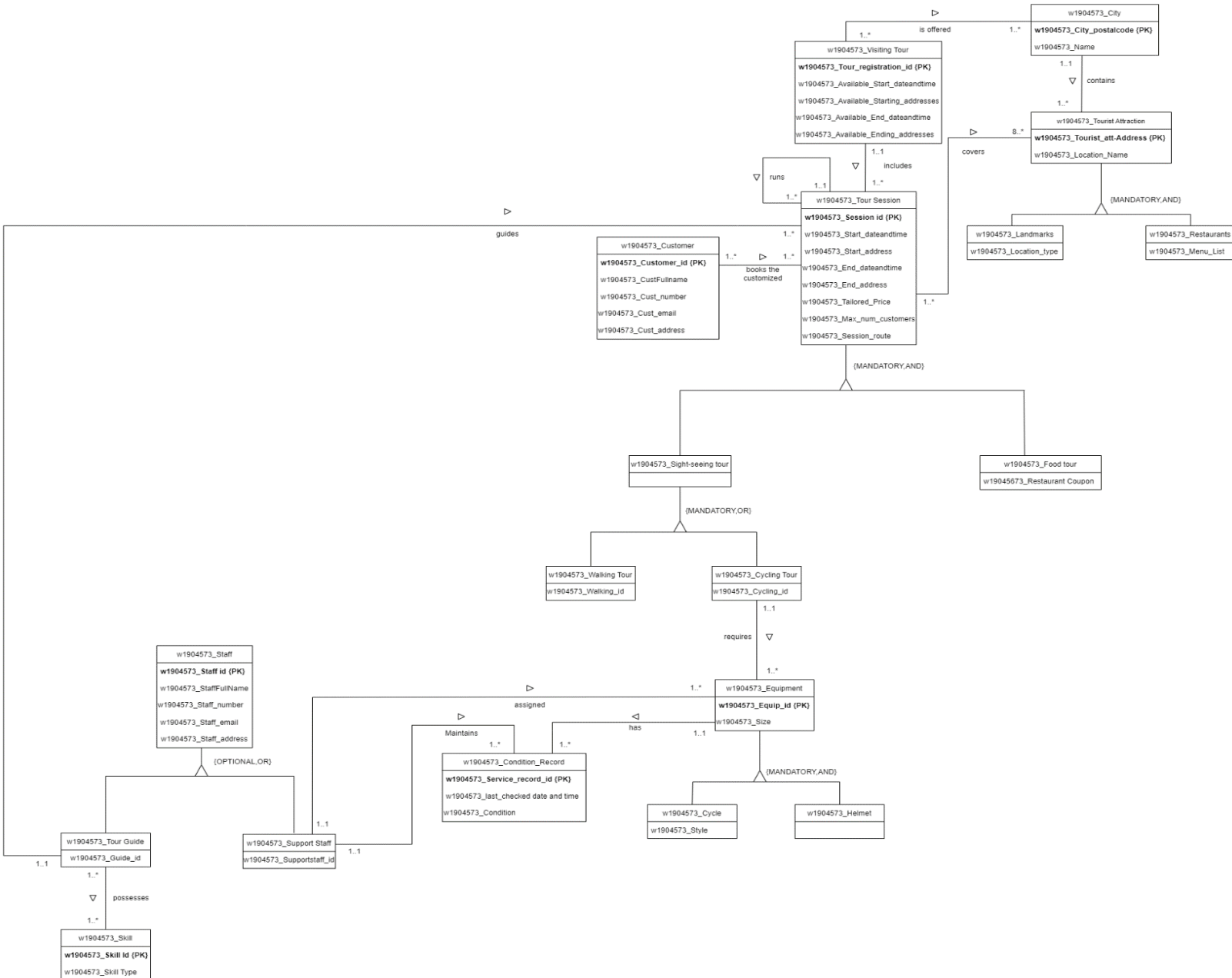
Part A & Part B

FINAL REPORT

Abhinav Ratnathurai
w1904573
20211507

Tutorial Group – A

Question 01



ASSUMPTIONS

- I assume that the tour guides should have some kind of skill, so I added a skilled entity that says that the tour guide must have at least one skill or many skills.
- I assume that one piece of equipment can have either 1 or many condition records as the equipment might get serviced more than one time, so it might have many condition records.
- I assume the recursive relationship on the tour session, as the project brief says that the same tour could be run several times, so I added 1 tour session at minimum and maximum could run at least 1 time or many times.

- I added Available start date & time, Available end date & time, Available starting addresses, and Available ending addresses, assuming Tourmato company has several visiting tours which has several dates, times, and addresses for the customer to customize and the customized final session is included as a tour session which is booked by the customer. Therefore, from visiting the touring entity, a tour session entity is created which is booked by the customer and could be run several times due to the above-mentioned recursive relationship.
- Since both, cycle and helmet are given to the customers in the cycling session, I assume that 1 cycling session would get minimum 1 equipment(including helmet & cycle) and a maximum of many pieces of equipment (Depending on the number of customers on the cycling session).
- I assume {OPTIONAL, OR} for the specialization entities from the general entity called Staff. This is because there could be other staff working in the Tourmato company who might have not been mentioned in the project brief. Eg- Manager, Supervisor, etc.
- I bring out two specialized entities, which are called, “Sight-seeing tours” & “Food sessions”. I added {MANDATORY, AND}, so the customer can choose either the sight-seeing tour sessions or food tour sessions, or have the possibility to choose both of them. Then I have specialized the “Sight-seeing tour” specialized entity into two separations called the “Walking tour” and the “Cycling tour”, and added {MANDATORY, OR}, where the Users can select either the Walking tour or the Cycling tour only.

Question 02

Entity Name	Brief Explanation
w1904573_Visiting Tour	Tourmato company offers exciting “off-the-beaten-track” visiting tours across Europe.
w1904573_City	Visiting tours are offered across several cities across Europe.
w1904573_Tour session	When a tour is scheduled, it is known as a “Tour session”.
w1904573_Tourist Attraction	Tourmato contains multiple touristic attractions considered worth visiting.
w1904573_Customer	Tourmato takes groups of customers around European cities to get them to see them and experience their unique atmospheres.
w1904573_Equipment	A cycling session requires the use of equipment that needs to be provided by Tourmato.
w1904573_Staff	Tourmato relies on highly trained employees with specialized roles, who are known as Staff.
w1904573_Skill	Tour guides should have the skill to narrate key facts on the history, geography, architecture, ecology, or gastronomy of each visited attraction.
w1904573_Condition Record	Support staff maintains a condition record to assure the equipment is kept in a good condition.
w1904573_Landmarks	Landmarks are simply relevant locations in the city that can be viewed, such as monuments, buildings,

	statues, squares, streets, parks, places of worship and so many more.
w1904573_Restaurants	Offer interesting typical foods (dishes and/or drinks) to be sampled to allow people to experience the local culinary delicacies.
w1904573_Sight-seeing tour	Contains both the Walking session and Cycling session which could be chosen by the customer.
w1904573_Food tour	Allow customers to stop at different restaurants and sample selected foods i.e., several dishes and/or drinks.
w1904573_Walking tour	Customers visit the city by walking from one attraction to another.
w1904573_Cycling tour	Customers ride from one attraction to another on a bicycle.
w1904573_Cycle	The cycle is provided by Tourmato for the cycling tour session.
w1904573_Helmet	The helmet is provided by Tourmato for the cycling tour session.
w1904573_Support Staff	Support staff plays a key role by ensuring the strict maintenance of all equipment used by Tourmato.
w1904573_Tour Guide	Tour guides naturally lead the tour sessions.

General Entity	Specialized Entity	Brief Explanation
w1904573_Touristic Attraction	w1904573_Landmarks, w1904573_Restaurants	These attractions fall under two categories: tourist landmarks and restaurants. {MANDATORY, AND} is used to show whether one of the two options could be chosen, or both options could be chosen.
w1904573_Tour Session	w1904573_Sight-seeing tour, w1904573_Food tour	The tour session is divided into a food session and a group walking and cycling session which is called a Sightseeing session. {MANDATORY, AND} is used to show whether one of the two options could be chosen, or both options could be chosen.
w1904573_Sight-seeing tour	w1904573_Walking tour, w1904573_Cycling tour	A sightseeing tour specialized entity is considered a general entity here because this entity is separated into two other sessions called the Walking tour and Cycling tour. {MANDATORY, OR} is used here to specify that either a Walking or Cycling tour could be chosen.
w1904573_Equipment	w1904573_Helmet, w1904573_Cycle	Tourmato offers a helmet and a cycle for Cycling session customers. {MANDATORY, AND} is used here to specify both the helmet and cycle could be chosen at the same time.
w1904573_Staff	w1904573_The tour guide, w1904573_Support Staff	Tourmato has tour guides and support staff in their company. Since there can be other members of

		staff in another post which are not mentioned in the project brief, {OPTIONAL, OR} is added here.
--	--	---

Question 03

Entity	Multiplicities	Relationship	Multiplicities	Entity	Justification (4)
w1904573_Visiting Tour	1..1	includes	1.. *	w1904573_Tour Session	<ul style="list-style-type: none"> One Visiting tour includes at least one Tour session. One Visiting tour includes at most many Tour sessions. One Tour session is included in One Visiting tour at least. One Tour session is included in One Visiting tour at most.
w1904573_Visiting Tour	1.. *	is offered	1.. *	w1904573_City	<ul style="list-style-type: none"> One Visiting tour is offered with at least one City. One Visiting tour is offered with at most many Cities. One City is offered to at least one Visiting tour. One City is offered to at most many Visiting tours.
w1904573_City	1..1	contains	1.. *	w1904573_Tourist Attraction	<ul style="list-style-type: none"> One city contains at least one tourist attraction. One city contains at most many tourist attractions. One Tourist attraction is contained under at least one city. One Tourist attraction is contained under at most one city.

w1904573_Tour session	1.. *	covers with	8.. *	w1904573_Tourist Attraction	<ul style="list-style-type: none"> • One Tour session covers at least 8 tourist attractions. • One Tour session covers at most many tourist attractions. • One Tourist attraction is included in at least 1 Tour session. • One Tourist attraction is included in at most many Tour sessions.
w1904573_Customer	1.. *	Books the customized	1.. *	w1904573_Tour session	<ul style="list-style-type: none"> • One Customer books at least one tour session. • One Customer books at most many tour sessions. • One Tour session is booked by at least one customer. • One Tour session is booked by at most many customers.
w1904573_Cycling tour	1..1	requires	1.. *	w1904573_Equipment	<ul style="list-style-type: none"> • One Cycling tour requires at least one Equipment. • One Cycling tour requires at most many Equipment. • One Equipment is handed to at least one Cycling tour • One Equipment is handed to at most one Cycling tour.
w1904573_Support Staff	1..1	assigned	1.. *	w1904573_Equipment	<ul style="list-style-type: none"> • One Support Staff is assigned at least one equipment. • One Support Staff is assigned at most many equipment.

					<ul style="list-style-type: none"> • One Equipment is assigned to at least one support staff. • One Equipment is assigned to at most one support staff.
w1904573_Support Staff	1..1	Maintains	1.. *	w1904573_Condition Record	<ul style="list-style-type: none"> • One Support Staff maintains at least one condition record. • One Support Staff maintains at most many condition records. • One Condition record is maintained by at least one staff. • One Condition record is maintained by at most one staff.
w1904573_Equipment	1..1	has	1.. *	w1904573_Condition Record	<ul style="list-style-type: none"> • One Equipment has at least one condition record. • One Equipment has at most many condition records. • One Condition record is maintained for at least one Equipment. • One Condition record is maintained for at most one Equipment.
w1904573_Tour Guide	1.. *	possesses	1.. *	w1904573_Skill	<ul style="list-style-type: none"> • One Tour Guide possesses at least one skill. • One Tour Guide possesses at most many skills. • One skill is possessed by at

					<p>least one tour guide.</p> <ul style="list-style-type: none"> One Tour Guide is possessed by at most many tour guides.
w1904573_Tour Guide	1..1	guides	1.. *	w1904573_Tour Session	<ul style="list-style-type: none"> One Tour Guide guides at least one Tour session. One Tour Guide guides at most many Tour sessions. One Tour session is guided by at least one tour guide. One Tour session is guided by at most one tour guide.
w1904573_Tour Session	1..1	runs	1.. *	w1904573_Tour Session (Number of times it runs)	<ul style="list-style-type: none"> Minimum One Tour session runs at least once. Minimum One Tour session runs at most many times Maximum One Tour session run once at least once. Maximum One Tour session run at most many times.

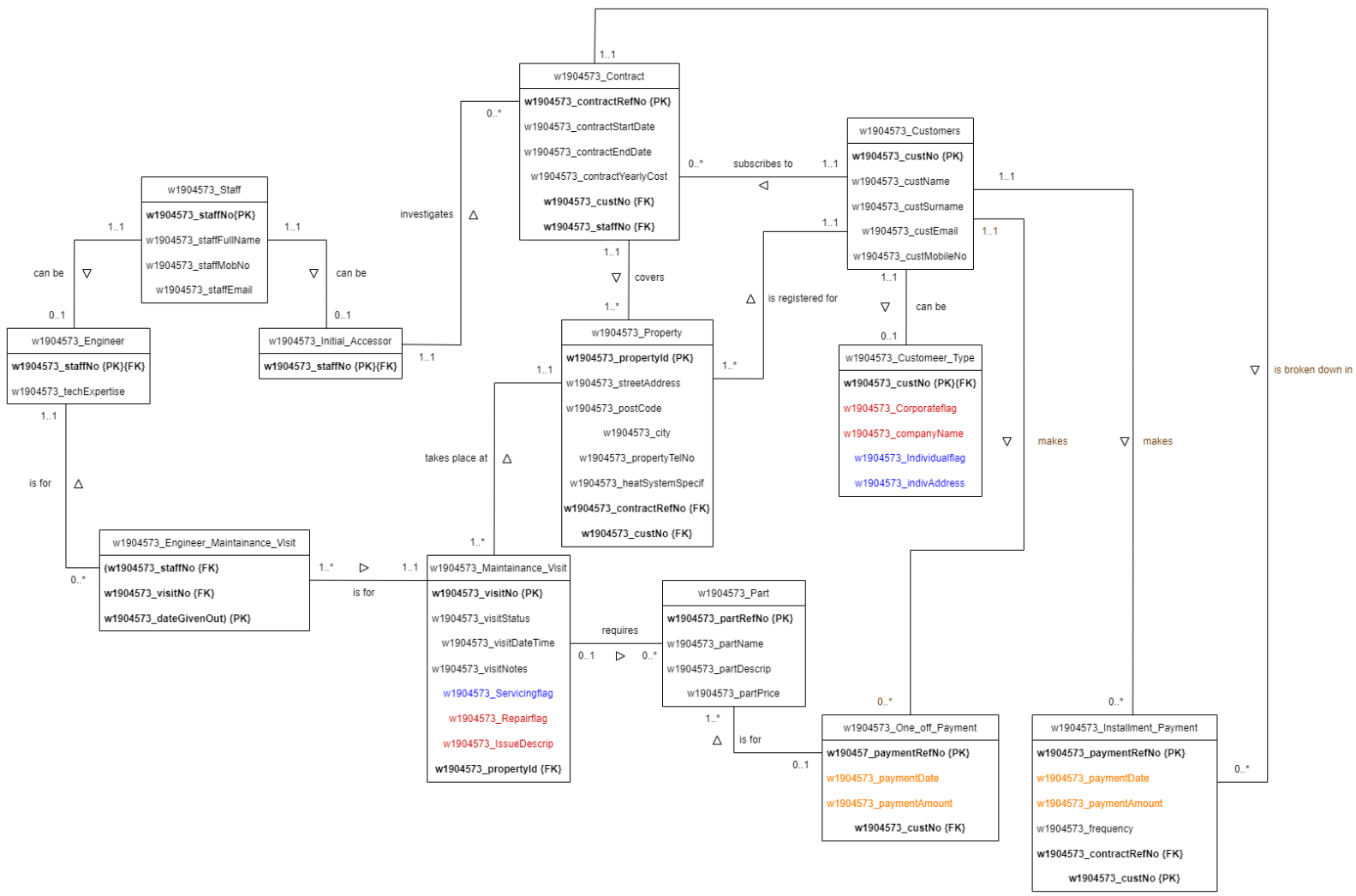
Question 04

Entity Name	Attributes for this entity (Including PK)	Brief Explanation
w1904573_Visiting Tour	w1904573_Tour_registration_id {PK}	<ul style="list-style-type: none"> Tour Registration ID is the primary key which would be unique and separate for individuals.
	w1904573_Available_Start_dateandtime	<ul style="list-style-type: none"> The available start date and time is given for the customer to customize from the available time slots and date slots.

	w1904573_Available_Starting_addresses	<ul style="list-style-type: none"> User could customize from where his customized session could start from the start place slots given.
	w1904573_Available_End_dateandtime	<ul style="list-style-type: none"> The available end date and time is given for the customer to customize from the available time slots and date slots.
	w1904573_Available_Ending_addresses	<ul style="list-style-type: none"> User could customize from where his customized session could end from the end place slots given.
w1904573_Tour Session	w1904573_Session id {PK}	<ul style="list-style-type: none"> A unique ID for the customers who book the specific tour session from the visiting tour sessions.
	w1904573_Start_dateandtime	<ul style="list-style-type: none"> Customized Specific Start date and time for the tour session.
	w1904573_Start_address	<ul style="list-style-type: none"> Customized Specific Start address for the tour session
	w1904573_End_dateandtime	<ul style="list-style-type: none"> Customized Specific End date and time for the tour session
	w1904573_End_address	<ul style="list-style-type: none"> Customized Specific End address for the tour session
	w1904573_Tailored_Price	<ul style="list-style-type: none"> The Tailored price for the booked customized tour session
	w1904573_Max_num_customers	<ul style="list-style-type: none"> The maximum number of customers are present in one tour session
	w1904573_Session_route	<ul style="list-style-type: none"> The customized session route for the tour session
w1904573_City	w1904573_City_postalcode {PK}	<ul style="list-style-type: none"> The city postal code is a unique code that is used as a primary key
	w1904573_Name	<ul style="list-style-type: none"> The name of each city
w1904573_Tourist Attraction	w1904573_Tourist_att-Address {PK} w1904573_Location_Name	<ul style="list-style-type: none"> Tourist attraction address is a unique primary key The location name of each tourist attraction
w1904573_Landmarks	w1904573_Location_type	<ul style="list-style-type: none"> The type of the location.
w1904573_Restaurants	w1904573_Menu_List	<ul style="list-style-type: none"> The menu list of each restaurant's foods.
w1904573_Customer	w1904573_Customer_id {PK}	<ul style="list-style-type: none"> Customer id which is a unique primary key used to identify each customer.
	w1904573_CustFullname	<ul style="list-style-type: none"> Full Name of Customer
	w1904573_Cust_number	<ul style="list-style-type: none"> Mobile number of Customer
	w1904573_Cust_email	<ul style="list-style-type: none"> Customer email address














	w1904573_Cust_address	<ul style="list-style-type: none"> Customer home/residential address.
w1904573_Sight-seeing tour	---	<ul style="list-style-type: none"> -----
w1904573_Food tour	w1904573_Restaurant Coupon	<ul style="list-style-type: none"> Restaurant offer coupons to attend several restaurants and taste their foods.
w1904573_Walking Tour	w1904573_Walking_id	<ul style="list-style-type: none"> An id that is the same for all the walking tour session customers.
w1904573_Cycling Tour	w1904573_Cycling_id	<ul style="list-style-type: none"> An id that is the same for all the cycling tour session customers.
w1904573_Equipment	w1904573_Equip_id {PK} w1904573_Size	<ul style="list-style-type: none"> A unique Equipment id for each customer in the cycling session. Size of the equipment given to the cycling session customers.
w1904573_Cycle	w1904573_Style	<ul style="list-style-type: none"> Different styles of Cycles.
w1904573_Helmet	---	<ul style="list-style-type: none"> -----
w1904573_Condition_Record	w1904573_Service_record_id {PK}	<ul style="list-style-type: none"> Service record ID is a unique primary key given to each piece of equipment to ensure its condition record.
	w1904573_last_checked date and time	<ul style="list-style-type: none"> Checking the last checked date and time of the specific equipment.
	w1904573_Condition	<ul style="list-style-type: none"> Condition of the specific equipment in the condition record management system.
w1904573_Staff	w1904573_Staff id {PK}	<ul style="list-style-type: none"> Staff ID is a unique ID given.
	w1904573_StaffFullName	<ul style="list-style-type: none"> The Full name of the staff.
	w1904573_Staff_number	<ul style="list-style-type: none"> The Staff's mobile number
	w1904573_Staff_email	<ul style="list-style-type: none"> The Staff's email address.
	w1904573_Staff_address	<ul style="list-style-type: none"> The Staff's home/residential address.
w1904573_Support Staff	w1904573_Supportstaff_id	<ul style="list-style-type: none"> An ID, which is the same for all the support staff.
w1904573_Tour Guide	w1904573_Guide_id	<ul style="list-style-type: none"> An ID, which is the same for the tour guides.
w1904573_Skill	w1904573_Skill Id {PK}	<ul style="list-style-type: none"> A unique skill ID
	w1904573_Skill Type	<ul style="list-style-type: none"> The type of skill the tour guide possesses.

Question 05








Question 06

Customer Table

<div>← T →</div>				custNo	custName	custSurname	custEmail	custMobileNo
<input type="checkbox"/>	 Edit	 Copy	 Delete	341	Albert	Pattek	albert.p@gmail.com	209-200-3739
<input type="checkbox"/>	 Edit	 Copy	 Delete	342	Freddie	Patterson	freddie.p@gmail.com	209-202-8997
<input type="checkbox"/>	 Edit	 Copy	 Delete	343	Sharon	Williams	sharon.w@gmail.com	209-205-3634
<input type="checkbox"/>	 Edit	 Copy	 Delete	344	Robert	Patrick	robert.p@gmail.com	209-206-4949
<input type="checkbox"/>	 Edit	 Copy	 Delete	345	Daniel	Mac	danielmac@gmail.com	209-210-9776

One-off-payment

				paymentRefNo	paymentDate	paymentAmount	custNo
<input type="checkbox"/>	 Edit	 Copy	 Delete	98340	2022-04-08	120	343
<input type="checkbox"/>	 Edit	 Copy	 Delete	98657	2022-05-23	80	345
<input type="checkbox"/>	 Edit	 Copy	 Delete	98765	2022-06-20	75	341
<input type="checkbox"/>	 Edit	 Copy	 Delete	98812	2022-08-22	80	342
<input type="checkbox"/>	 Edit	 Copy	 Delete	98915	2022-09-17	70	344

```

DROP TABLE IF EXISTS Customers;
DROP TABLE IF EXISTS One_off_payments;

CREATE TABLE Customers
(
    custNo          INTEGER not null,
    custName        VARCHAR(20) not null,
    custSurname     VARCHAR(20) not null,
    custEmail       VARCHAR(50) not null unique,
    custMobileNo    VARCHAR(15) not null,
    constraint      c_cid_pk PRIMARY KEY (custNo)
);

CREATE TABLE One_off_payments
(
    paymentRefNo    INTEGER not null,
    paymentDate     DATE not null,
    paymentAmount   INTEGER not null,
    custNo          INTEGER not null,
    constraint      off_payid_pk PRIMARY KEY (paymentRefNo),
    constraint      c_offid_fk FOREIGN KEY (custNo)
    references      Customers(custNo)
);

INSERT INTO
Customers (custNo,custName,custSurname,custEmail,custMobileNo)
VALUES
(341, 'Albert', 'Pattek', 'albert.p@gmail.com', '209-200-3739'),
(342, 'Freddie', 'Patterson', 'freddie.p@gmail.com', '209-202-8997'),
(343, 'Sharon', 'Williams', 'sharon.w@gmail.com', '209-205-3634'),
(344, 'Robert', 'Patrick', 'robert.p@gmail.com', '209-206-4949'),
(345, 'Daniel', 'Mac', 'danielmac@gmail.com', '209-210-9776');

INSERT INTO
One_off_payments (paymentRefNo,paymentDate,paymentAmount,custNo)
VALUES
(098765, '2022-06-20', 75, 341),
(098812, '2022-08-22', 80, 342),
(098340, '2022-04-08', 120, 343),
(098915, '2022-09-17', 70, 344),
(098657, '2022-05-23', 80, 345);

```

```
1 SELECT c.custSurname, c.custEmail, payment.paymentRefNo, payment.paymentDate, payment.paymentAmount
2 FROM Customers c JOIN one_off_payments payment
3 ON c.custNo = payment.custNo
4 WHERE (c.custSurname LIKE 'pat%'
5 AND payment.paymentAmount <= 80);
```

✔ Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

```
SELECT c.custSurname, c.custEmail, payment.paymentRefNo, payment.paymentDate, payment.paymentAmount FROM Customers c JOIN one_off_payments
payment ON c.custNo = payment.custNo WHERE (c.custSurname LIKE 'pat%' AND payment.paymentAmount <= 80);
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all

Number of rows:

25

 ▾

Filter rows:

Search this table

Extra options

custSurname	custEmail	paymentRefNo	paymentDate	paymentAmount
Pattek	albert.p@gmail.com	98765	2022-06-20	75
Patterson	freddie.p@gmail.com	98812	2022-08-22	80
Patrick	robert.p@gmail.com	98915	2022-09-17	70

Question 07

Criteria	MySQL	MongoDB
PERFORMANCE	<p>MySQL is made to allow for quick joins between many correctly indexed tables. Most of the data in MongoDB documents are contained in a single document and normally follow a hierarchical data model, therefore there is no need for joins across several documents. However, MongoDB supports joins using the \$ lookup function.</p>	<p>In contrast to MySQL, which mandates row-by-row data entry, MongoDB is designed for write efficiency and has a special insertMany() API for swiftly entering data. MongoDB prioritizes speed above transaction safety.</p> <p>As the number of records grows, MongoDB's execution time is substantially faster than MySQL's. As a result, when there are more records, MongoDB operates more quickly than MySQL. MongoDB is preferred in terms of performance. (Aboutorabi et al., 2015)</p>
SECURITY	<p><u>SECURITY MODEL</u> Access to just certain commands, such as CREATE, UPDATE, DELETE, etc., is granted to users under MySQL's privilege-based security paradigm. Therefore, depending on the user type, certain privileges may be defined.</p> <p><u>INJECTIONS</u> SQL injections, which are straightforward attacks that insert malicious code into SQL statements via web page output, can affect MySQL.</p> <p><u>LOGGING</u> The transaction support, comprehensive logging capabilities by default, and rollback support of MySQL all contribute to data integrity.</p> <p><u>ACCESS CONTROLS</u> Role-based access control and discretionary access control (REVOKE & GRANT) commands are just two of the access control methods that MySQL supports.</p> <p><u>INTEGRITY MODEL</u> MySQL takes advantage of the ACID (Atomic, Consistent, Isolated, Durable) principle. If a relational database cannot accomplish any of these four goals, it is not seen to be reliable. Database</p>	<p><u>SECURITY MODEL</u> To make sure that the data is only available to the intended user, MongoDB offers TLS and SSL for encryption.</p> <p><u>INJECTIONS</u> Due to the use of an interpretable language like JavaScript, MongoDB is not completely immune to injection mistakes even if it is not sensitive to SQL injections (as was explained above).</p> <p><u>LOGGING</u> Complete logging is not enabled by default in MongoDB. The operating system and application layers both incorporate extra logging.</p> <p><u>ACCESS CONTROLS</u> MongoDB only offers role-based access control. it is not turned on by default. It provides several pre-built roles that give a set of privileges typically required in databases.</p> <p><u>INTEGRITY MODEL</u> The BASE (Basic Availability, Security, and Economic) model of ultimate consistency and soft states. Now that MongoDB 4.0 has been released, document support for ACID transactions is possible. Giving a</p>

	administrators use a variety of techniques to ensure ACID, including write-ahead logging (WAL), shadow paging, and two-phase commit protocol. (Sahib Singh, 2019)	consistent view of the data through transactions and snapshot isolation while demanding all-or-nothing execution and ensuring data integrity. (Sahib Singh, 2019)
SCALABILITY (WORKLOAD) Scalability is required to prevent your database from collapsing under an increased workload.	MySQL also uses partitioning and sharding to scale writing, however, these are challenging to implement. MySQL Cluster enables sharding automatically, and tables are partitioned among nodes so that the database may scale horizontally. To obtain the data, SQL and NoSQL APIs can be utilized. Because sharding is transparent, queries may be conducted on any node. Because MySQL is built on an active/active, multi-master architecture, updates may be made by any node and are immediately visible to all other clients using the cluster. MongoDB's scalability is larger than that of MySQL. It is also easier to develop in MongoDB than in MySQL because MongoDB handles most of the work, but MySQL is a popular choice for high-performance RDBMS because it preserves data integrity even when scaling and sharding. (Pandey, Rachit, 2020)	In MongoDB, a shard key is used to divide data into shards; data is specified in 64 MB chunks and is the smallest logical block of data that can be sharded. A shard can carry a data replica set and can have numerous parts. MongoDB decides which A replica set is preserved on a particular shard. Users and programs have no way of knowing which data is stored in which shard. MongoDB can therefore scale horizontally while simultaneously securing the data. When you look at a single shard to see what's within, you'll see a random fraction of the data. The application has no notion which data replica is kept in which shard or partition, and the user has no control over it. As a result, you (the client) must always connect to MongoDB and enable it to discover the data you want. Because MongoDB handles the bulk of the work, its scalability is larger than that of MySQL. It is also easier to develop because MongoDB handles most of the work. (Pandey, Rachit, 2020)
DATA CONSISTENCY	Because they meet the ACID (Atomicity, Consistency, Isolation, and Durability) requirements, traditional relational databases are frequently vertically scalable and provide high data consistency. Almost every strategy considered consistency. As previously indicated, scalability is an issue because all requests are handled by a single powerful system. (P. Filip and L. Čegan, 2020)	NoSQL databases like MongoDB, on the other hand, are built for scalability, resulting in poor consistency. This technique is known as a BASE (Basically Available, Soft state, and Eventually consistent). Some current NoSQL database systems are also quite consistent. Strong consistency necessitates replica synchronization, which slows down the entire system. (P. Filip and L. Čegan, 2020)
SCHEMA	The MySQL system schema is used. It has tables that store the data required by the active MySQL server. The data dictionary tables for storing database item metadata and system tables for general operating purposes make up the MySQL schema.	MongoDB stores data in collections that do not have an imposed schema. Because of its flexibility in data storage, it is ideal for developers who are not database professionals but require a database to aid in the development of their products. This is a significant advantage over MySQL since you need to understand normalization, referential integrity, and relational database design before you can make the most of a relational database.

		Teams creating applications that may not need all the security safeguards offered by relational databases can take use of MongoDB's flexible developer interface and ability to store documents with a variety of schemas, including unstructured data sets.
--	--	--

References

Aboutorabi, Seyyed & Rezapour, Mehdi & Moradi, Milad & Ghadiri, Nasser. (2015). Performance evaluation of SQL and MongoDB databases for big e-commerce data. Available from https://www.researchgate.net/publication/281629941_Performance_evaluation_of_SQL_and_MongoDB_databases_for_big_e-commerce_data. [Accessed 7th November 2022]

P. Filip and L. Čegan. 2020. Comparison of MySQL and MongoDB with focus on performance. International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS). Available from <https://ieeexplore.ieee.org/document/9354307>. [Accessed 8th November 2022]

Pandey, Rachit. (2020). Performance Benchmarking and Comparison of Cloud-Based Databases MongoDB (NoSQL) Vs MySQL (Relational) using YCSB. Available from https://www.researchgate.net/profile/Rachit-Pandey-3/publication/344047197_Performance_Benchmarking_and_Comparison_of_Cloud-Based_Databases_MongoDB_NoSQL_Vs_MySQL_Relational_using_YCSB/links/5f4fd2cc458515e96d256d72/Performance-Benchmarking-and-Comparison-of-Cloud-Based-Databases-MongoDB-NoSQL-Vs-MySQL-Relational-using-YCSB.pdf. [Accessed 6th November 2022]

Sahib Singh. (2019). Security Analysis of MongoDB. Available from <https://infonomics-society.org/wp-content/uploads/Security-Analysis-of-MongoDB.pdf>. [Accessed 8th November 2022]

THE END