INFORMATICS
INSTITUTE OF
TECHNOLOGY

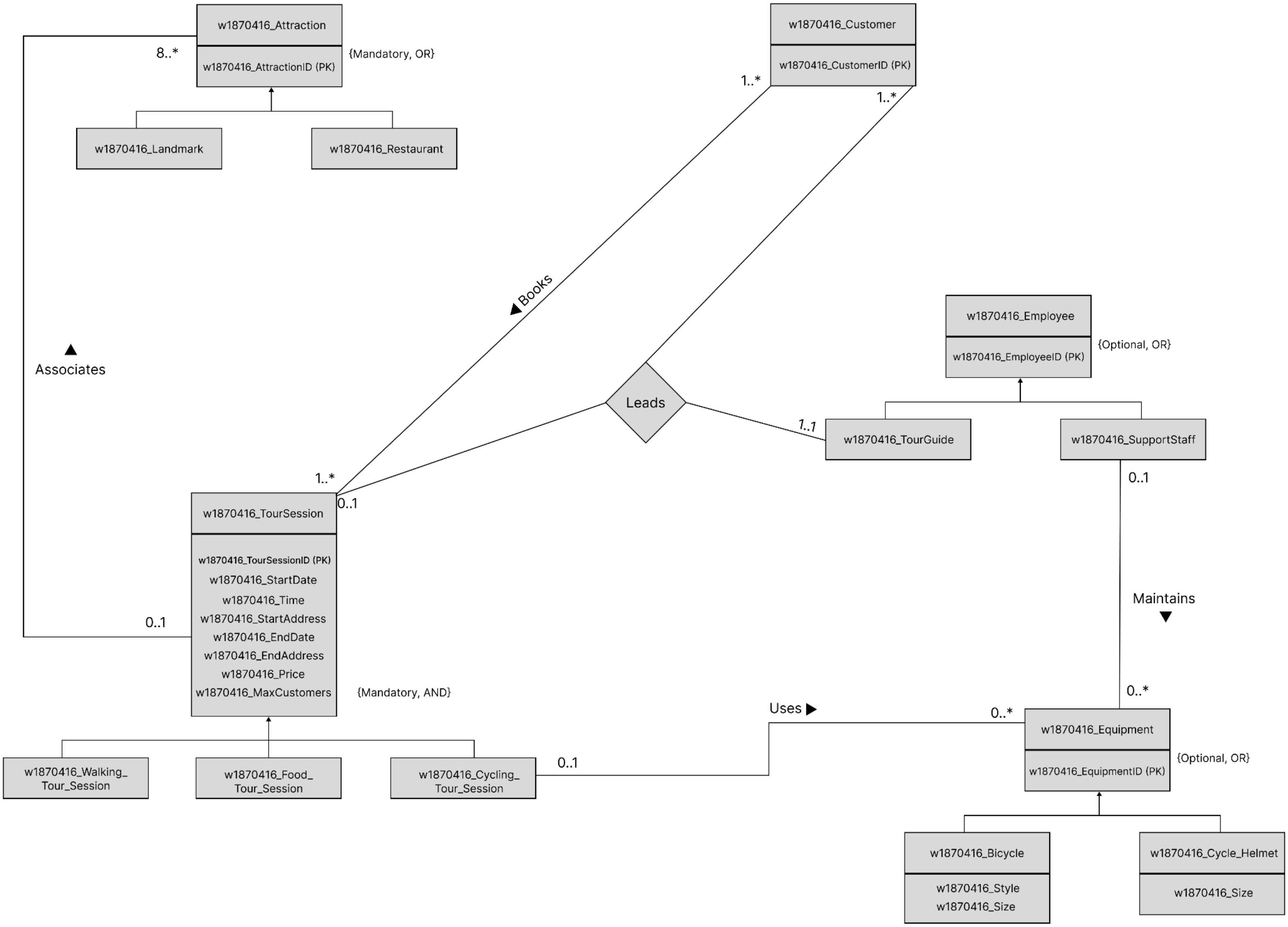
UNIVERSITY OF
WESTMINSTER

5COSC020C.1 Database Systems Coursework

Name: Kisara Fernando
IIT Number: 20210586
UoW Number: w1870416

Content

(Part A) Question 01



Assumptions:

- Attraction is a general entity. And it’s been linked with other entities as well. Therefore, a unique identifier is required.
- TourSessionID acts as the unique identifier of records under this entity.
- Customer is an important entity that gets linked to other entities. Therefore a unique identifier is required.
- Employee is a generalized entity. It requires a unique identifier to refer to records.
- Equipment is also a generalized entity. A primary key to uniquely identify records is needed

(Part A) Question 02

Entity	Brief Description
Customer	Tourmato takes customers around European cities. Therefore, the customer is an entity.
Attraction	Each city covered by Tourmato contains multiple tourist attractions. And tourist attractions fall under two categories. Therefore, an attraction is an entity.
Landmark	Landmark is a specialized entity that falls under the general entity of attractions. It's stated that, attractions have two categories
Restaurant	Restaurant is a specialized entity that falls under the general entity of attractions. It's stated that, attractions have two categories
Tour Session	Describes a tour session with its associated properties.
Walking Tour Session	Each tour session has 3 categories and walking tour session is one of them.
Cycling Tour Session	Each tour session has 3 categories and cycling tour session is one of them.
Food Tour Session	Each tour session has 3 categories and food tour session is one of them.
Equipment	Each cycling tour session requires the use of equipment
Bicycle	Equipment has 2 main categories.Bicycle is one of them
Cycle Helmet	Equipment has 2 main categories.Cycle helmet is one of them
Employee	Tourmato has highly trained employees
Tour Guide	Tour guides are one of the specialized roles in Touramato
Support staff	Support staff is one of the specialized roles in Touramato

Generalized Entity	Specialized Entity	Brief Explanation
Attraction	Landmark	It's stated that attractions fall under 2 categories and landmarks are one of the specialized entities
Attraction	Restaurant	It's stated that attractions fall under 2 categories and restaurants are one of the specialized entities
Tour Session	Walking Tour Session	Tour sessions have 3 main types. One is walking tour sessions
Tour Session	Cycling Tour Session	Tour sessions have 3 main types. One is cycling tour sessions
Tour Session	Food Tour Session	Tour sessions have 3 main types. One is a food tour session which can be a walking or a cycling session as well.
Equipment	Bicycle	Tourmato lends two main types of equipment. Bicycles are one of those equipment which is used on cycling tour sessions
Equipment	Cycle helmet	Tourmato lends two main types of equipment. Cycle helmets are one of those equipment which is used on cycling tour sessions
Employee	Tour guide	Tourmaro highly relies on specialized roles. There are 2 roles and tour guides are one those
Employee	Support Staff	Tourmaro highly relies on specialized roles. There are 2 roles and support staff is one those

(Part A) Question 03

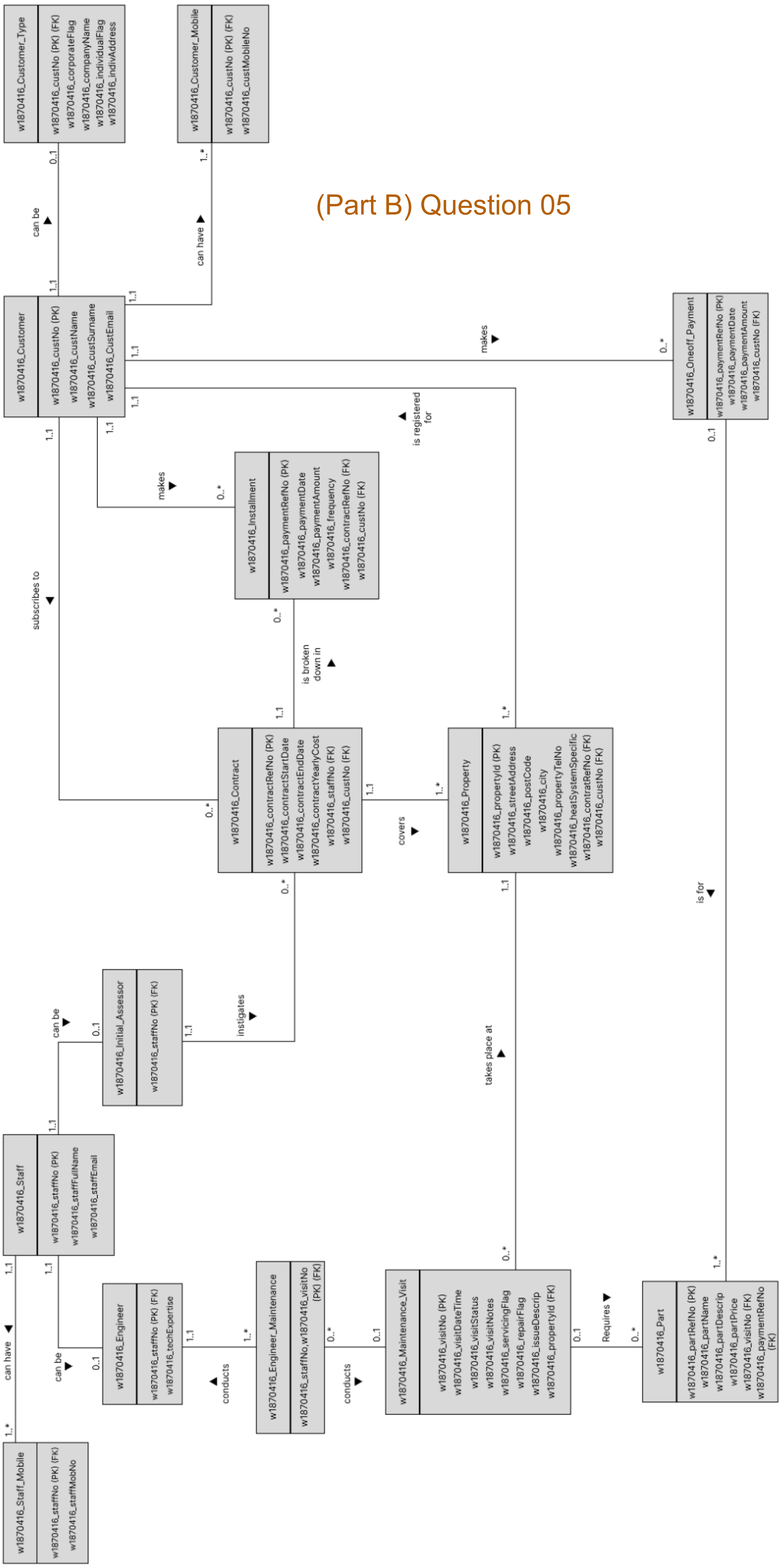
Entity Name	Multiplicity	Relationship	Multiplicity	Entity Name	Brief justification
Attraction	8..*	Associates	0..1	Tour Session	● A tour session can have minimum of 8 attractions ● A tour session can have many attractions ● A tourist attraction may not be associated any tour session ● A tourist attraction may be associated to 1 tour session
Customer	1..*	Books	1..*	Tour Session	● A customer can book 1 tour session ● A customer can book many tour sessions ● A tour session can be booked by 1 customer ● A tour session can be booked by many customers
Cycling Tour Session	0..1	Uses	0..*	Equipment	● A cycling session may not use any equipment ● A cycling session may use many equipment ● Equipment may not used by any cycling session ● Many equipment may be used by a cycling session
Tour Guide	1..1	Leads	0..1	Tour Session	● Tour guide leads a tour session ● A tour session may not be lead by a tour guide ● A tour session may lead by a tour guide

Tour Guide	1..1	Leads	1..*	Customer	● A tour guide leads a customer ● A tour guide leads many customers ● A customer is lead by a tour guide ● Many customers are lead by a tour guide

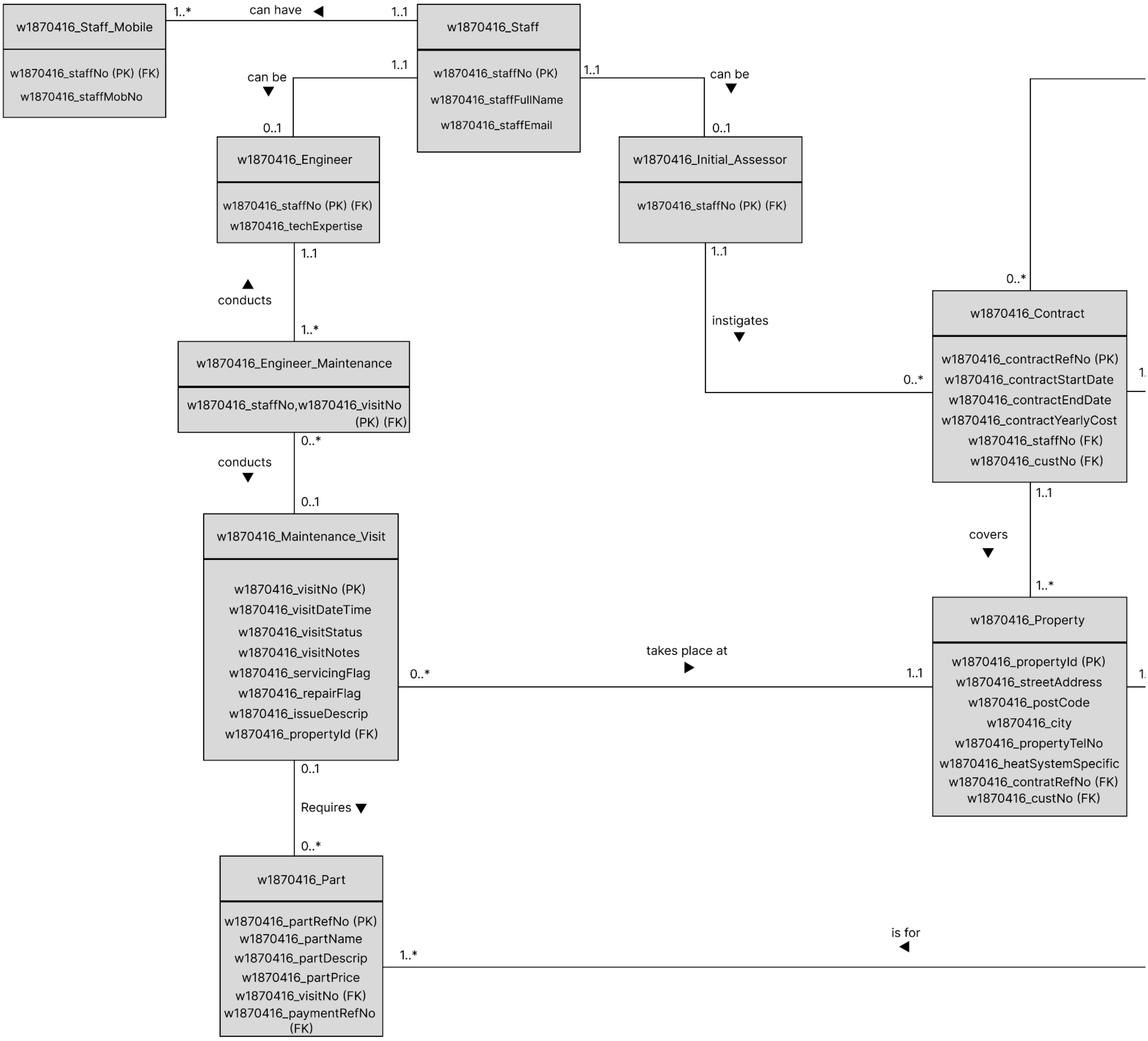
(Part A) Question 04

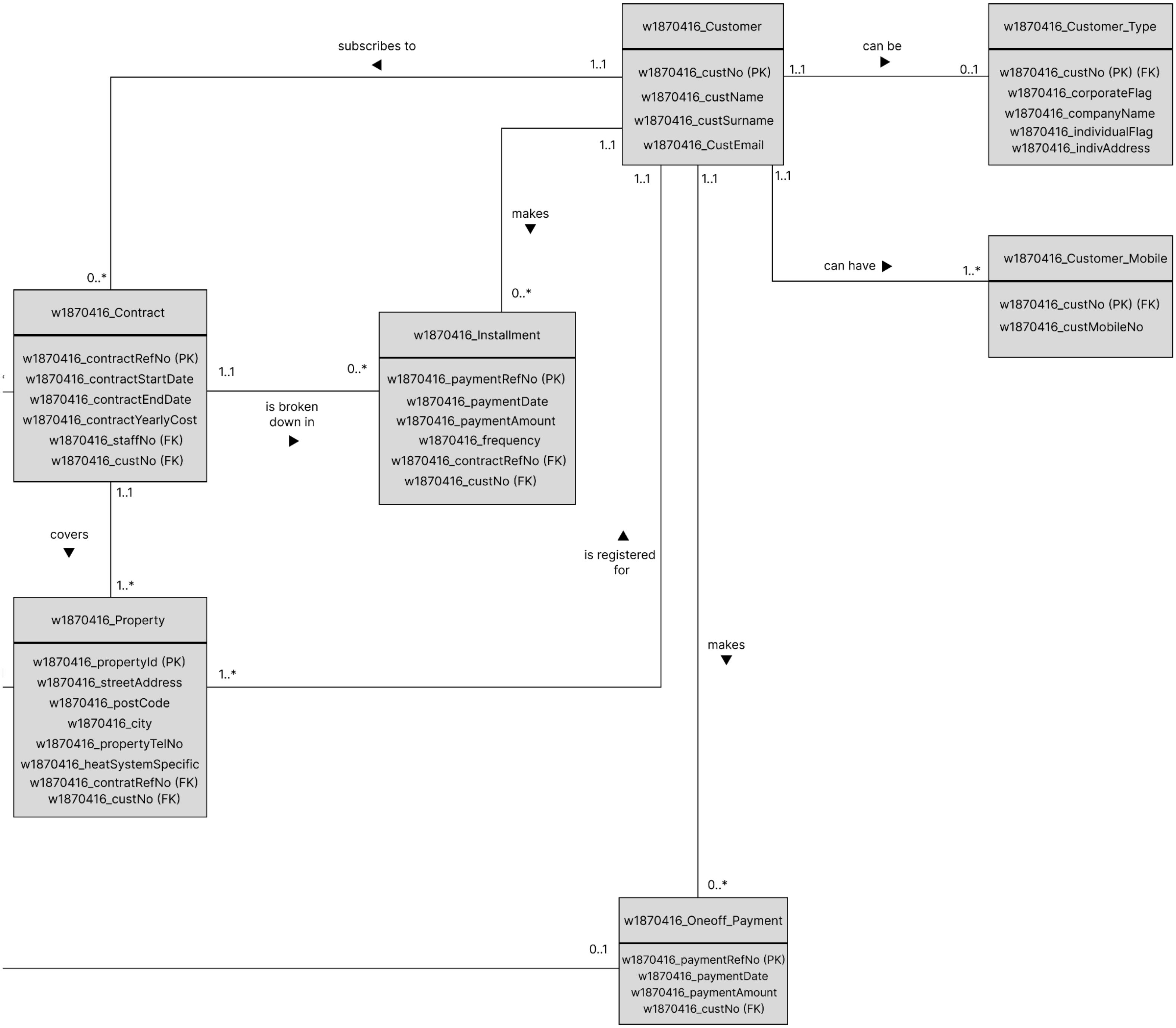
Entity Name	Attributes	Brief Explanation
Attraction	AttractionID (PK)	Attraction is a general entity. And it's been linked with other entities as well. Therefore, a unique identifier is required.
TourSession	TourSessionID (PK) StartDate Time StartAddress EndDate EndAddress Price MaxCustomers	TourSessionID acts as the unique identifier of records under this entity. Other attributes were stated in the case study.
Customer	CustomerID (PK)	Customer is an important entity that gets linked to other entities. Therefore a unique identifier is required.
Employee	EmployeeID (PK)	Employee is a generalized entity. It requires a unique identifier to refer to records.
Equipment	EquipmentID (PK)	Equipment is also a generalized entity. A primary key to uniquely identify records is needed
Bicycle	Style Size	Being a specialized entity, equipmentID is inherited by equipment entity while the style and size attributes are only valid to bicycle entity
Cycle Helmet	Size	Being a specialized entity, equipmentID is inherited by equipment entity while the size attributes are only valid to cycle helmet entity

(Part B) Question 05



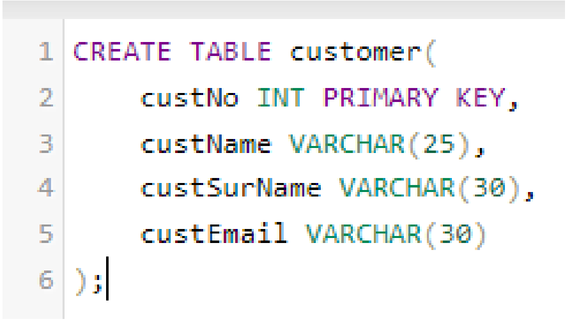
Additional Screenshots




w1870416_Customer
w1870416_custNo (PK)
w1870416_custName
w1870416_custSurname
w1870416_CustEmail
w1870416_Customer_Type
w1870416_custNo (PK) (FK)
w1870416_corporateFlag
w1870416_companyName
w1870416_individualFlag
w1870416_indivAddress
w1870416_Customer_Mobile
w1870416_custNo (PK) (FK)
w1870416_custMobileNo
w1870416_Contract
w1870416_contractRefNo (PK)
w1870416_contractStartDate
w1870416_contractEndDate
w1870416_contractYearlyCost
w1870416_staffNo (FK)
w1870416_custNo (FK)
w1870416_Installment
w1870416_paymentRefNo (PK)
w1870416_paymentDate
w1870416_paymentAmount
w1870416_frequency
w1870416_contractRefNo (FK)
w1870416_custNo (FK)
w1870416_Property
w1870416_propertyId (PK)
w1870416_streetAddress
w1870416_postCode
w1870416_city
w1870416_propertyTelNo
w1870416_heatSystemSpecific
w1870416_contratRefNo (FK)
w1870416_custNo (FK)
w1870416_Oneoff_Payment
w1870416_paymentRefNo (PK)
w1870416_paymentDate
w1870416_paymentAmount
w1870416_custNo (FK)
subscribes to ◀
1..1
0..*
can be ▶
1..1
0..1
makes ▼
1..1
0..*
can have ▶
1..1
1..*
is broken down in ▶
1..1
0..*
covers ▼
1..1
1..*
▲ is registered for
1..1
1..*
makes ▼
1..1
0..*
0..1

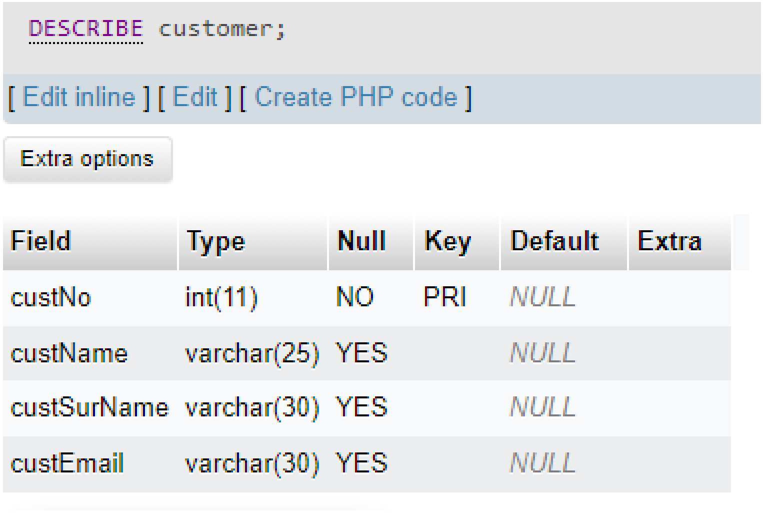
(Part B) Question 06

Creating The Customer Table:



```
CREATE TABLE customer(
    custNo INT PRIMARY KEY,
    custName VARCHAR(25),
    custSurName VARCHAR(30),
    custEmail VARCHAR(30)
);
```

Before executing the query

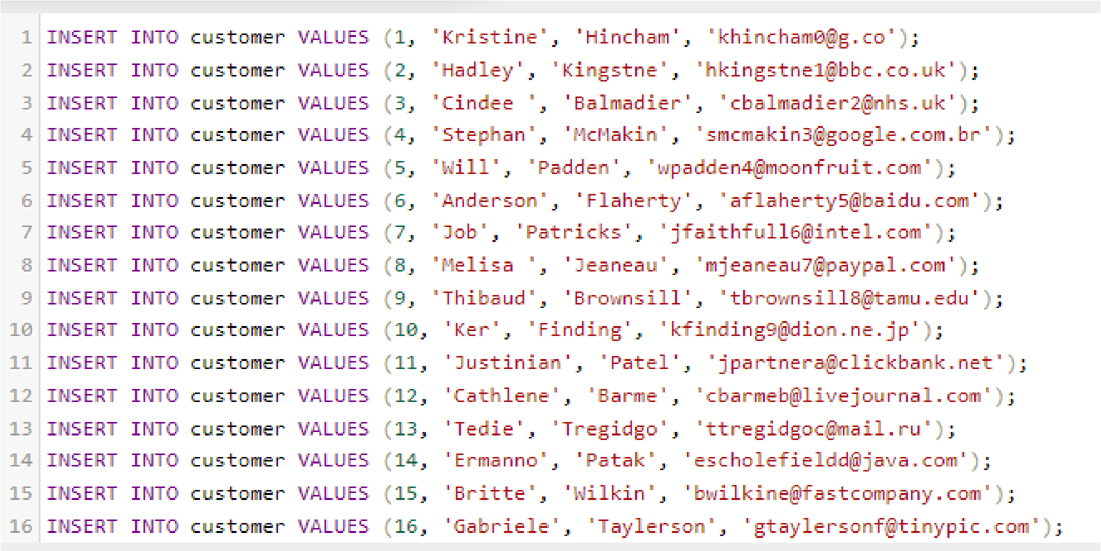


```
DESCRIBE customer;
```

Field	Type	Null	Key	Default	Extra
custNo	int(11)	NO	PRI	*NULL*	
custName	varchar(25)	YES		*NULL*	
custSurName	varchar(30)	YES		*NULL*	
custEmail	varchar(30)	YES		*NULL*	

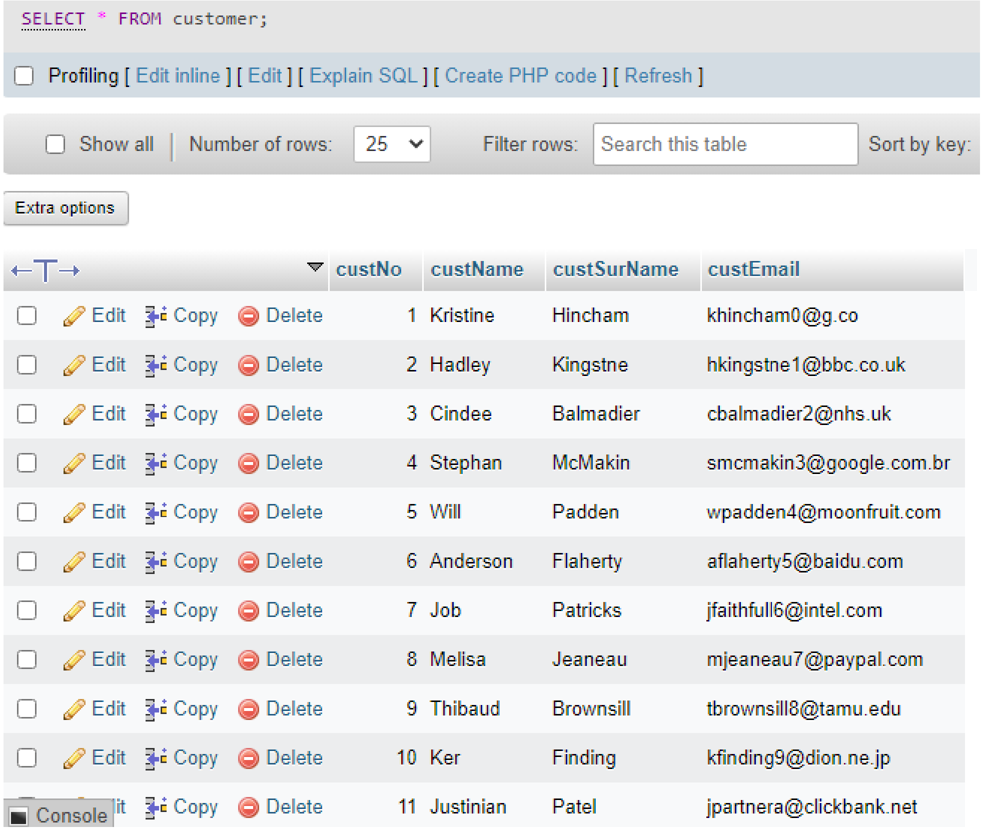
After executing the query

Populating The Customer Table:



```sql
INSERT INTO customer VALUES (1, 'Kristine', 'Hincham', 'khincham0@g.co');
INSERT INTO customer VALUES (2, 'Hadley', 'Kingstne', 'hkingstne1@bbc.co.uk');
INSERT INTO customer VALUES (3, 'Cindee ', 'Balmadier', 'cbalmadier2@nhs.uk');
INSERT INTO customer VALUES (4, 'Stephan', 'McMakin', 'smcmakin3@google.com.br');
INSERT INTO customer VALUES (5, 'Will', 'Padden', 'wpadden4@moonfruit.com');
INSERT INTO customer VALUES (6, 'Anderson', 'Flaherty', 'aflaherty5@baidu.com');
INSERT INTO customer VALUES (7, 'Job', 'Patricks', 'jfaithfull6@intel.com');
INSERT INTO customer VALUES (8, 'Melisa ', 'Jeaneau', 'mjeaneau7@paypal.com');
INSERT INTO customer VALUES (9, 'Thibaud', 'Brownsill', 'tbrownsill8@tamu.edu');
INSERT INTO customer VALUES (10, 'Ker', 'Finding', 'kfinding9@dion.ne.jp');
INSERT INTO customer VALUES (11, 'Justinian', 'Patel', 'jpartnera@clickbank.net');
INSERT INTO customer VALUES (12, 'Cathlene', 'Barme', 'cbarmeb@livejournal.com');
INSERT INTO customer VALUES (13, 'Tedie', 'Tregidgo', 'ttregidgoc@mail.ru');
INSERT INTO customer VALUES (14, 'Ermanno', 'Patak', 'escholefieldd@java.com');
INSERT INTO customer VALUES (15, 'Britte', 'Wilkin', 'bwilkine@fastcompany.com');
INSERT INTO customer VALUES (16, 'Gabriele', 'Taylerson', 'gtaylersonf@tinypic.com');
```

Before executing the query



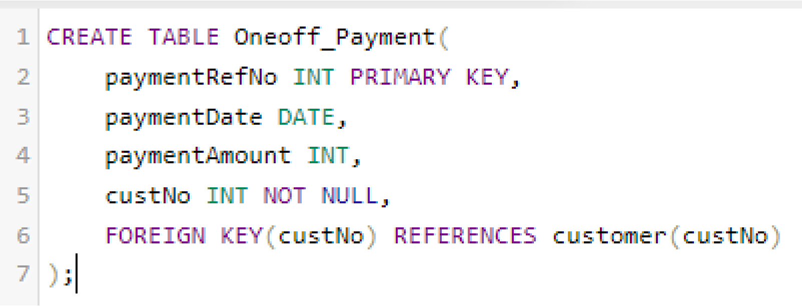
```sql
SELECT * FROM customer;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key:
Extra options

←T→				custNo	custName	custSurName	custEmail
□	Edit	Copy	Delete	1	Kristine	Hincham	khincham0@g.co
□	Edit	Copy	Delete	2	Hadley	Kingstne	hkingstne1@bbc.co.uk
□	Edit	Copy	Delete	3	Cindee	Balmadier	cbalmadier2@nhs.uk
□	Edit	Copy	Delete	4	Stephan	McMakin	smcmakin3@google.com.br
□	Edit	Copy	Delete	5	Will	Padden	wpadden4@moonfruit.com
□	Edit	Copy	Delete	6	Anderson	Flaherty	aflaherty5@baidu.com
□	Edit	Copy	Delete	7	Job	Patricks	jfaithfull6@intel.com
□	Edit	Copy	Delete	8	Melisa	Jeaneau	mjeaneau7@paypal.com
□	Edit	Copy	Delete	9	Thibaud	Brownsill	tbrownsill8@tamu.edu
□	Edit	Copy	Delete	10	Ker	Finding	kfinding9@dion.ne.jp
Console	it	Copy	Delete	11	Justinian	Patel	jpartnera@clickbank.net

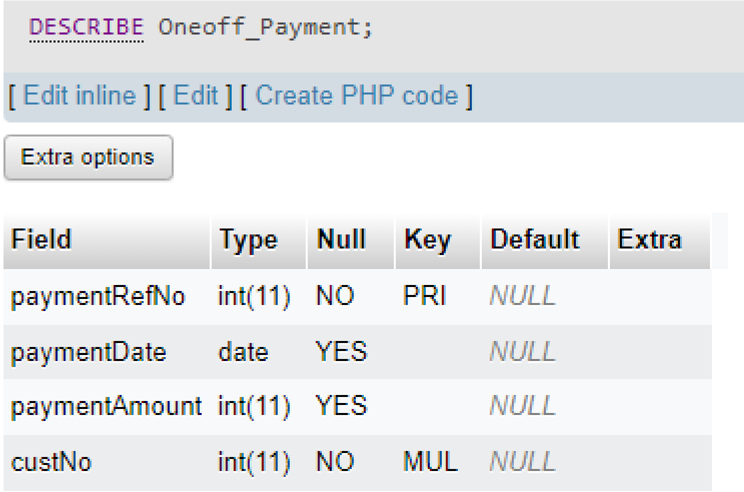
After executing the query

Creating The Oneoff Payment Table:



```
CREATE TABLE Oneoff_Payment(
    paymentRefNo INT PRIMARY KEY,
    paymentDate DATE,
    paymentAmount INT,
    custNo INT NOT NULL,
    FOREIGN KEY(custNo) REFERENCES customer(custNo)
);
```

Before executing the query

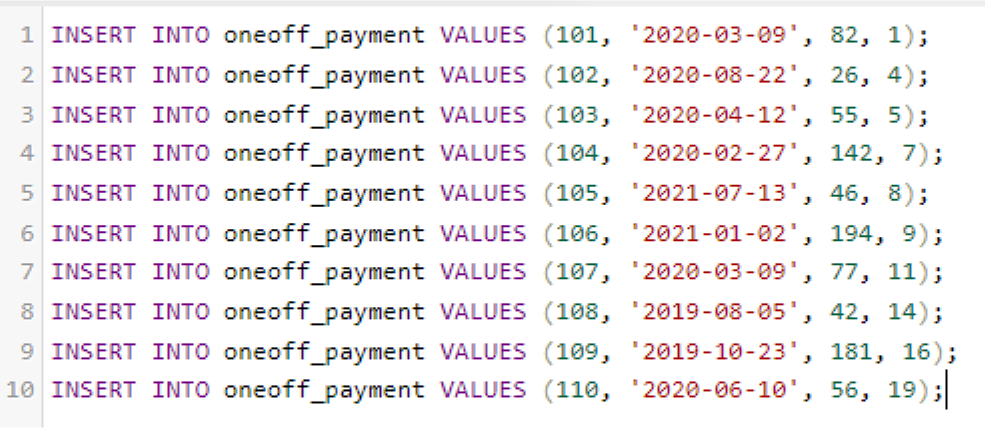


```
DESCRIBE Oneoff_Payment;
```

Field	Type	Null	Key	Default	Extra
paymentRefNo	int(11)	NO	PRI	*NULL*	
paymentDate	date	YES		*NULL*	
paymentAmount	int(11)	YES		*NULL*	
custNo	int(11)	NO	MUL	*NULL*	

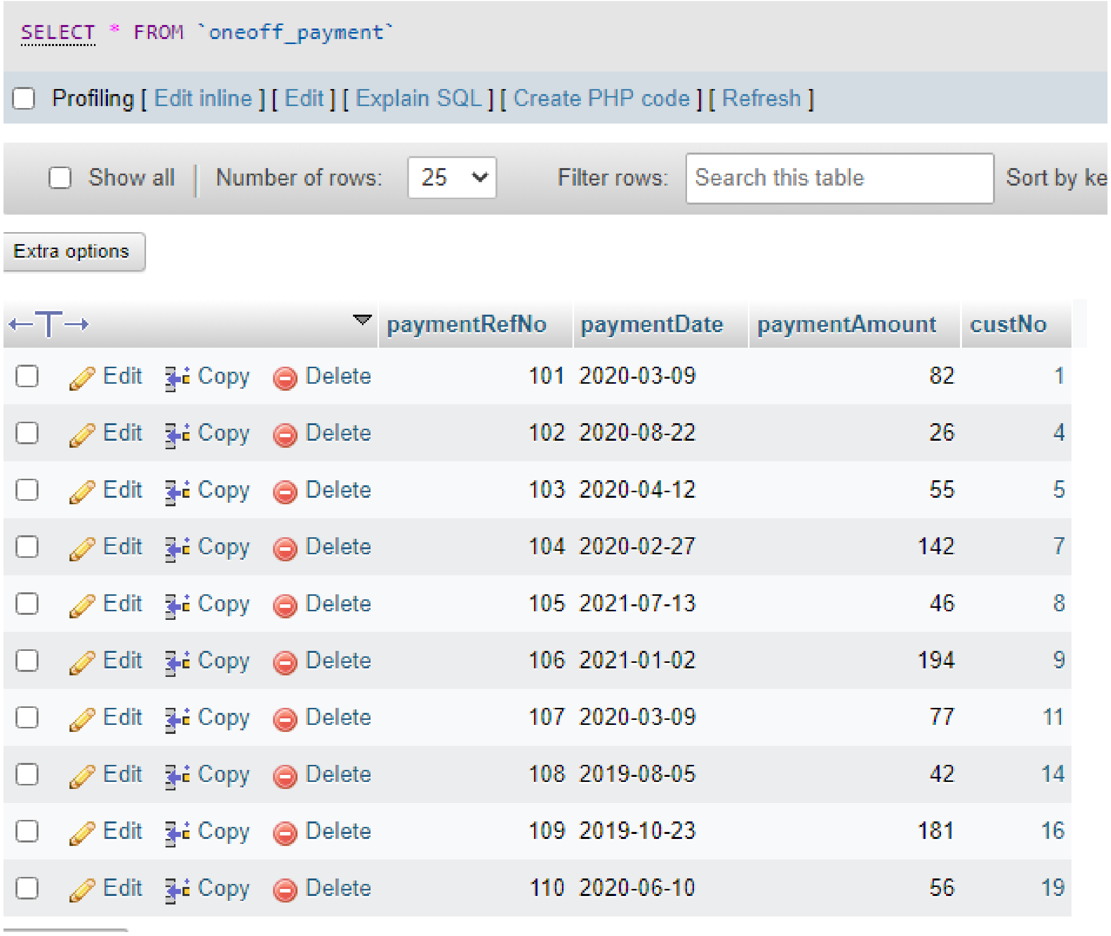
After executing the query

Populating The Oneoff Payment Table:



```sql
INSERT INTO oneoff_payment VALUES (101, '2020-03-09', 82, 1);
INSERT INTO oneoff_payment VALUES (102, '2020-08-22', 26, 4);
INSERT INTO oneoff_payment VALUES (103, '2020-04-12', 55, 5);
INSERT INTO oneoff_payment VALUES (104, '2020-02-27', 142, 7);
INSERT INTO oneoff_payment VALUES (105, '2021-07-13', 46, 8);
INSERT INTO oneoff_payment VALUES (106, '2021-01-02', 194, 9);
INSERT INTO oneoff_payment VALUES (107, '2020-03-09', 77, 11);
INSERT INTO oneoff_payment VALUES (108, '2019-08-05', 42, 14);
INSERT INTO oneoff_payment VALUES (109, '2019-10-23', 181, 16);
INSERT INTO oneoff_payment VALUES (110, '2020-06-10', 56, 19);
```

Before executing the query



```sql
SELECT * FROM `oneoff_payment`
```

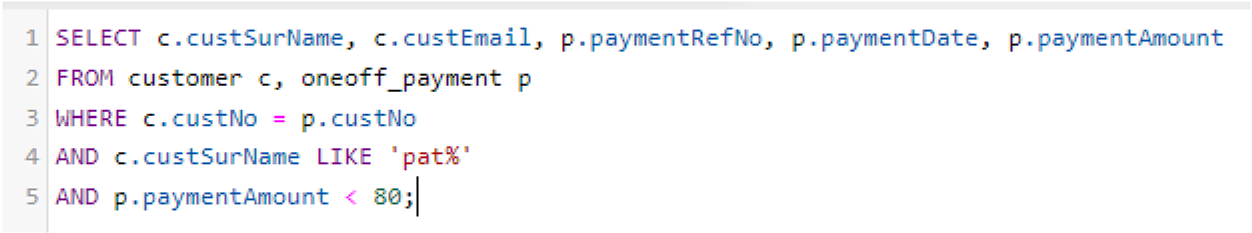
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by ke
Extra options

←T→				paymentRefNo	paymentDate	paymentAmount	custNo
□	Edit	Copy	Delete	101	2020-03-09	82	1
□	Edit	Copy	Delete	102	2020-08-22	26	4
□	Edit	Copy	Delete	103	2020-04-12	55	5
□	Edit	Copy	Delete	104	2020-02-27	142	7
□	Edit	Copy	Delete	105	2021-07-13	46	8
□	Edit	Copy	Delete	106	2021-01-02	194	9
□	Edit	Copy	Delete	107	2020-03-09	77	11
□	Edit	Copy	Delete	108	2019-08-05	42	14
□	Edit	Copy	Delete	109	2019-10-23	181	16
□	Edit	Copy	Delete	110	2020-06-10	56	19

After executing the query

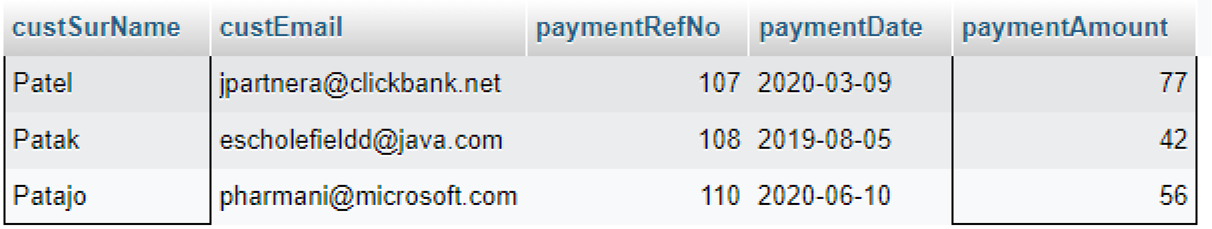
Query to get the relevant output:

```
SELECT c.custSurName, c.custEmail, p.paymentRefNo, p.paymentDate, p.paymentAmount
FROM customer c, oneoff_payment p
WHERE c.custNo = p.custNo
AND c.custSurName LIKE 'pat%'
AND p.paymentAmount < 80;
```



```
SELECT c.custSurName, c.custEmail, p.paymentRefNo, p.paymentDate, p.paymentAmount
FROM customer c, oneoff_payment p
WHERE c.custNo = p.custNo
AND c.custSurName LIKE 'pat%'
AND p.paymentAmount < 80;
```

Before executing the query



custSurName	custEmail	paymentRefNo	paymentDate	paymentAmount
Patel	jpartnera@clickbank.net	107	2020-03-09	77
Patak	escholefieldd@java.com	108	2019-08-05	42
Patajo	pharmani@microsoft.com	110	2020-06-10	56

After executing the query

(Part B) Question 07

	MongoDB	**MySQL**
Schema flexibility	Gives the ability to modify the design in accordance with requirements, particularly for Big Data applications. You can quickly integrate and store many data types using it, and you can dynamically change the structure without experiencing any downtime. Because it uses a non-relational database system, it is possible to store numerous documents in a collection even if there is no relationship between them. It makes use of optionally schemated JSON-like (BSON) documents. However, it lacks transactions and joins, thus depending on how the app accesses data, you frequently need schema optimization.	Before putting data alongside rows and columns in MySQL, you must explicitly declare columns and tables. Each field in this table consists of a row and a column. This indicates that, unlike MongoDB, data storage does not offer you a lot of freedom. Additionally, the deployment and development processes will be delayed. However, MySQL is the best if your applications have a set schema. Without repeatedly modifying the design or wasting effort on it, it will provide improved data consistency. However, once more, MongoDB may be a superior choice for you if your needs change.
Performance	Due to the fact that it holds a lot of unstructured data, it is relatively faster than MySQL. MongoDB facilitates faster data read or write operations by storing data for an object in a single document. With objects, its efficiency is considerably greater because of JSON-like (BSON) object storage. Performance for writes is prioritized in MongoDb. By using a hierarchical data model, it does away with the necessity for joins.	Due to its tabular data store format, performance is comparatively slower when compared to MongoDB. The need to read and write data across several tables when changing or extracting data makes servers work harder and reduces their performance. requires a specific data structure in order to add data to the database, and designing a suitable schema becomes challenging when dealing with complicated data.
Security	MongoDb uses a flexible set of permissions along with the	Privilege-based access controls exist in MySQL. supports similar

	well-known role-based access control approach. TLS (Transport Layer Security) is used for all communication, and by utilizing a master key, it is possible to write encrypted documents to MongoDB data collections, achieving data at rest encryption. This prevents unauthorized users or malicious attackers from accessing and using your database.	authentication approaches and the same encryption features as MongoDB. In order to increase data access security, MySQL now supports dual passwords. Because of its rigorous architecture and structure, which improves data consistency and durability, MySQL is thought to be more secure
Scalability	Scaling the database is quite simple. Enables the configuration of a sharded cluster so that a replica set can be configured for a part of the database (a group of MongoDB servers that hold the same data, ensuring high availability and disaster recovery). Data is spread out among a large number of servers in these sharded clusters. As a result, MongoDB can expand horizontally. Best choice for e-commerce sites and CMS systems.	MySQL is capable of vertical scaling and read replica addition. More resources are added to the existing database server when scaling vertically. There can be a maximum of five read replicas added at once. Only read operations are performed on read replicas. Writing-intensive apps have problems after adding read replicas. Although MySQL supports multi-main replication, its implementation is relatively limited in comparison to the features offered by MongoDB
Data Storage	Stores data in a JSON format known as binary json (BSON).It is easy to use CRUD functions without any hassle in MongoDB.	Stores data in tables (rows and columns). This rigid structure is more difficult to change compared to MongoDB.

References

Comparing the differences - mongodb vs mysql (no date) *MongoDB*. Available at: https://www.mongodb.com/compare/mongodb-mysql (Accessed: December 11, 2022).

MongoDB vs mysql: Which is the better database management system? (2022) *Kinsta*®. Available at: https://kinsta.com/blog/mongodb-vs-mysql/#:~:text=But%20if%20you%20have%20a,MongoDB%20offers%20more%20schema%20flexibility. (Accessed: December 11, 2022).