1. Define the following terms:

a. Candidate Key

A candidate key is a combination of one or more attributes that can uniquely identify each record in a relation. Each relation can have one or more candidate keys, but only one is chosen as the primary key.

b. Composite Key

A composite key is a key that consists of two or more attributes combined to uniquely identify each record in a relation. Composite keys are used when no single attribute can uniquely identify the records.

c. Foreign Key

A foreign key is one or more columns in a table that references the primary key of another table. Foreign keys are used to establish and enforce relationships between tables, ensuring data integrity.

d. Functional Dependency

Functional dependency refers to a relationship where one attribute (or a set of attributes) uniquely determines another attribute (or set of attributes) within a relation. It can be expressed as: if attribute set A functionally determines attribute set B, it is denoted as $A \rightarrow B$.

2. Identify and define the three integrity rules/constraints in the relational model:

1. **Entity Integrity**

   The entity integrity rule states that each table must have a primary key, and that primary key's value cannot be NULL. This ensures that each row in the table is unique and identifiable.

2. **Referential Integrity**

   The referential integrity rule ensures that the foreign key values must match primary key values in the referenced table. If a foreign key value does not exist in the parent table, it leads to data inconsistency.

3. **User-defined Integrity**

   User-defined integrity rules are specific constraints set according to business requirements. These can limit specific attributes (e.g., prices cannot be negative) or enforce complex conditions on the data.

3. Identify any violations of relational integrity rules in the following tables where the primary key is underlined:

Table 1: Film

Integrity Check:

Entity Integrity: If we assume the combination of and is the primary key, there is an issue with the row containing where is , as quantities should be numeric.SupplierPartNoCd12QuantityABC

Referential Integrity: There are no foreign keys in this table, so this cannot be checked.

User-defined Integrity: No specific user-defined constraints are present, but should be a positive number.Quantity

Table 3: Song

Integrity Check:

Entity Integrity: If and are the composite primary key, the row for is missing the , violating entity integrity.SongArtistA Kind of MagicArtist

Referential Integrity: There are no foreign keys in this table, so this cannot be checked.

User-defined Integrity: No obvious constraints are present, but if is expected to be continuous, there may be an issue with gaps in the sequence.Position

Table Director

Integrity Check:

Entity Integrity: serves as the primary key, and each value is unique and not null, satisfying entity integrity.directorNo

Referential Integrity: There are no foreign keys in this table, so this cannot be checked.

User-defined Integrity: No obvious constraints are present.

1. Insertion, Deletion, and Modification Anomalies

a. Insertion Anomaly

Operation: The management wishes to insert a new project into the table:

ProjectCode = PRC30

ProjectTitle = Skills Matrix

ProjectManager = M. Uhura

ProjectBudget = 20000

Anomaly Explanation:

An insertion anomaly occurs when certain attributes cannot be inserted into the database without the presence of other data. In this case, if the new project has no associated employees yet, it leads to the challenge of having incomplete data. For instance, there will be no entries for , , , , and . This could cause data integrity issues since all related information must be populated to maintain consistent records across the database.PRC30EmployeeNoEmployeeNameDepartmentNoDepartmentNameHourlyRate

b. Deletion Anomaly

Operation: The project ended abruptly and was deleted from the table.PRC10

Anomaly Explanation:

A deletion anomaly occurs when the removal of data results in the unintended loss of additional data. If is deleted, any associated employee information (e.g., , , etc.) will also be removed from the table. This is problematic if there are no other records of or any other employee in the database, leading to loss of historical employment data, which could be essential for reporting or auditing purposes.PRC10J. KirkL. JonesJ. Kirk

c. Modification Anomaly

Operation: Due to her outstanding performance, was moved from department (IT) to department (HR).J. KirkL004L009

Anomaly Explanation:

A modification anomaly arises when updating a piece of information requires multiple changes in the database, which can lead to inconsistencies if not all instances are updated. If is listed under multiple projects with different hourly rates and departments, changing her department in one place and forgetting to update it in another can result in inconsistent data across the table. If has multiple entries under different projects, forgetting to change all records means

some records will incorrectly reflect her old department, leading to data integrity issues.J. KirkJ. Kirk

2. Normalization to Different Normal Forms

d. First Normal Form (1NF)

Requirement: Each column should contain atomic values with no repeating groups.

The given table is already in 1NF because:

Each field contains only atomic values (e.g., , , etc., have single values).ProjCodeProjTitle

There are no repeating groups of columns.

Table 1: Project-Employee Data in 1NF

ProjCode	ProjTitle	ProjManager	ProjBudget	EmpNo	EmpName	DeptNo	DeptName	HrlyRate
PRC10	Payroll System	M Scott	24500	S10001	J Kirk	L004	IT	£22.00
PRC10	Payroll System	M Scott	24500	S10030	L Jones	L023	Pensions	£18.50
PRC10	Payroll System	M Scott	24500	S21010	P Lewis	L004	IT	£21.00
PRC45	Pension System	L McCoy	17400	S10010	B Jones	L004	IT	£21.75
PRC45	Pension System	L McCoy	17400	S10001	J Kirk	L004	IT	£18.00
PRC45	Pension System	L McCoy	17400	S31002	T Gilbert	L028	Database	£25.50

ProjCode	ProjTitle	ProjManager	ProjBudget	EmpNo	EmpName	DeptNo	DeptName	HrlyRate
PRC45	Pension System	L McCoy	17400	S13210	W Richards	L008	Salary	£17.00
PRC64	CRM System	P Chekov	12250	S31002	T Gilbert	L028	Database	£23.25
PRC64	CRM System	P Chekov	12250	S21010	P Lewis	L004	IT	£17.50
PRC64	CRM System	P Chekov	12250	S10034	B James	L009	HR	£16.50

e. Second Normal Form (2NF)

Requirement: Eliminate partial dependencies, ensuring all non-key attributes fully depend on the primary key.

Analysis: The primary key for this table is a composite key consisting of and . However, some non-key attributes depend only on (like , , and).ProjCodeEmpNoProjCodeProjTitleProjManagerProjBudget

Normalization Process:

Create a separate table for project details to eliminate partial dependencies.

Table 2: Project Table (2NF)

ProjCode	ProjTitle	ProjManager	ProjBudget
PRC10	Payroll System	M Scott	24500
PRC45	Pension System	L McCoy	17400
PRC64	CRM System	P Chekov	12250

Table 3: Employee Table (2NF)

EmpNo	EmpName	DeptNo	HrlyRate
S10001	J Kirk	L004	£22.00
S10030	L Jones	L023	£18.50
S21010	P Lewis	L004	£21.00
S10010	B Jones	L004	£21.75
S10001	J Kirk	L004	£18.00
S31002	T Gilbert	L028	£25.50
S13210	W Richards	L008	£17.00
S31002	T Gilbert	L028	£23.25
S21010	P Lewis	L004	£17.50
S10034	B James	L009	£16.50

f. Third Normal Form (3NF)

Requirement: Eliminate transitive dependencies, ensuring that non-key attributes do not depend on other non-key attributes.

Analysis: In the , and have a dependency (as depends on), which can lead to redundancy.Employee TableDeptNoDeptNameDeptNameDeptNo

Normalization Process:

Create a separate table for department information.

Table 4: Department Table (3NF)

DeptNo	DeptName
L004	IT
L023	Pensions
L028	Database
L008	Salary
L009	HR

Final Structure After Normalization

Project Table: ProjCode, ProjTitle, ProjManager, ProjBudget

Employee Table: EmpNo, EmpName, DeptNo, HrlyRate

Department Table: DeptNo, DeptName

Summary of Normalization Process

1. **1NF**: The initial table is converted to 1NF by ensuring all fields have atomic values.
2. **2NF**: The table is split into separate tables to eliminate partial dependencies on composite keys.
3. **3NF**: Additional separation occurs to remove transitive dependencies, thus achieving a structure that minimizes redundancy and maintains data integrity.

5. Normalization Process

a. Identify Anomalies

In the provided dataset, several anomalies can be identified:

1. **Insertion Anomaly**:

   If the bakery wants to add a new customer who has not yet placed any orders, they cannot do so because the is necessary for the table. This creates an issue when trying to insert customer data without an associated order.Order No

2. **Update Anomaly**:

   If a customer's address changes (e.g., Daisy's Café), every instance of that customer in the dataset must be updated. If any instance is missed during the update, it could lead to inconsistent data across the records.

3. **Deletion Anomaly**:

   If a customer decides to delete an order (e.g., Order No 7823), and that order is the only one associated with a particular customer, deleting it will result in the loss of all information about that customer, including their address.

b. Deriving First Normal Form (1NF)

To achieve **First Normal Form (1NF)**, we need to ensure that all attributes in the table contain atomic values, meaning each cell should hold a single value, and there should be no repeating groups.

Here's how the unnormalized table looks in 1NF:

Order No	Acc. No	Customer	Address	Date	Item	Qty	Price	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16-Jul	Bakewell Tart	20	0.15	£12.35
7823	178	Daisy's Café	27 Bay Drive, Cove	16-Jul	Danish Pastry	13	0.20	£12.35
7823	178	Daisy's Café	27 Bay Drive, Cove	16-Jul	Apple Pie	45	0.15	£12.35
4633	526	Smiths	12 Dee View, Aberdeen	16-Jul	Butteries	120	0.20	£24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17-Jul	Apple Pie	130	0.15	£56.50
2276	167	Sally's Snacks	3 High Street, Banchory	17-Jul	Cherry Pie	100	0.18	£56.50
2276	167	Sally's Snacks	3 High Street, Banchory	17-Jul	Steak Pie	30	0.50	£56.50
2276	167	Sally's Snacks	3 High Street, Banchory	17-Jul	Meringue Pie	20	0.20	£56.50
1788	32	Tasty Bite	17 Wood Place, Insch	18-Jul	Apple Pie	15	0.15	£7.50
1788	32	Tasty Bite	17 Wood Place, Insch	18-Jul	Danish Pastry	50	0.20	£7.50

c. Deriving Second Normal Form (2NF)

Second Normal Form (2NF) requires that the table is in 1NF and that all non-key attributes are fully functionally dependent on the primary key. In this case, the composite key will consist of and .Order NoItem

To achieve 2NF, we need to split the original table into two tables: one for orders and another for order items.

Table 1: Orders

Order No	Acc. No	Customer	Address	Date
7823	178	Daisy's Café	27 Bay Drive, Cove	16-Jul
4633	526	Smiths	12 Dee View, Aberdeen	16-Jul
2276	167	Sally's Snacks	3 High Street, Banchory	17-Jul
1788	32	Tasty Bite	17 Wood Place, Insch	18-Jul

Table 2: Order_Items

Order No	Item	Qty	Price	Total Cost
7823	Bakewell Tart	20	0.15	£12.35
7823	Danish Pastry	13	0.20	£12.35
7823	Apple Pie	45	0.15	£12.35
4633	Butteries	120	0.20	£24.00
2276	Apple Pie	130	0.15	£56.50
2276	Cherry Pie	100	0.18	£56.50
2276	Steak Pie	30	0.50	£56.50
2276	Meringue Pie	20	0.20	£56.50
1788	Apple Pie	15	0.15	£7.50
1788	Danish Pastry	50	0.20	£7.50

d. Deriving Third Normal Form (3NF)

To achieve **Third Normal Form (3NF)**, the tables must be in 2NF, and there should be no transitive dependencies. This means that non-key attributes should not depend on other non-key attributes.

In our case, the and information can be separated into another table, as they do not depend on the order items.CustomerAddress

Table 3: Customers

Acc. No	Customer	Address
178	Daisy's Café	27 Bay Drive, Cove
526	Smiths	12 Dee View, Aberdeen
167	Sally's Snacks	3 High Street, Banchory
32	Tasty Bite	17 Wood Place, Insch

Table 1: Orders (updated)

Order No	Acc. No	Date
7823	178	16-Jul
4633	526	16-Jul
2276	167	17-Jul
1788	32	18-Jul

Table 2: Order_Items (remains unchanged)

Order No	Item	Qty	Price	Total Cost
7823	Bakewell Tart	20	0.15	£12.35
7823	Danish Pastry	13	0.20	£12.35
7823	Apple Pie	45	0.15	£12.35
4633	Butteries	120	0.20	£24.00

Order No	Item	Qty	Price	Total Cost
2276	Apple Pie	130	0.15	£56.50
2276	Cherry Pie	100	0.18	£56.50
2276	Steak Pie	30	0.50	£56.50
2276	Meringue Pie	20	0.20	£56.50
1788	Apple Pie	15	0.15	£7.50
1788	Danish Pastry	50	0.20	£7.50

6

a. Schema for Each Table

1. Patient Table

Column Name	Data Type	Description
PatientNo	VARCHAR(10)	Patient ID (Primary Key)
Surname	VARCHAR(50)	Last Name
FirstName	VARCHAR(50)	First Name

2. Admission Table

Column Name	Data Type	Description
PatientNo	VARCHAR(10)	Patient ID (Foreign Key)
Admitted	DATE	Admission Date
Discharged	DATE	Discharge Date
Ward	CHAR(1)	Ward (Foreign Key)

3. Doctor Table

Column Name	Data Type	Description
DoctorNo	INT	Doctor ID (Primary Key)

Column Name	Data Type	Description
Surname	VARCHAR(50)	Last Name
FirstName	VARCHAR(50)	First Name
Ward	CHAR(1)	Ward ID

4. Ward Table

Column Name	Data Type	Description
Ward	CHAR(1)	Ward ID (Primary Key)
WardName	VARCHAR(50)	Ward Name
DoctorNo	INT	In-Charge Doctor ID (Foreign Key)

b. Identify the Relationships Between the Tables

1. **Patient and Admission Tables**:
    - **Relationship**: A patient can have multiple admission records.
    - **Foreign Key**: references .Admission.PatientNoPatient.PatientNo
2. **Doctor and Ward Tables**:

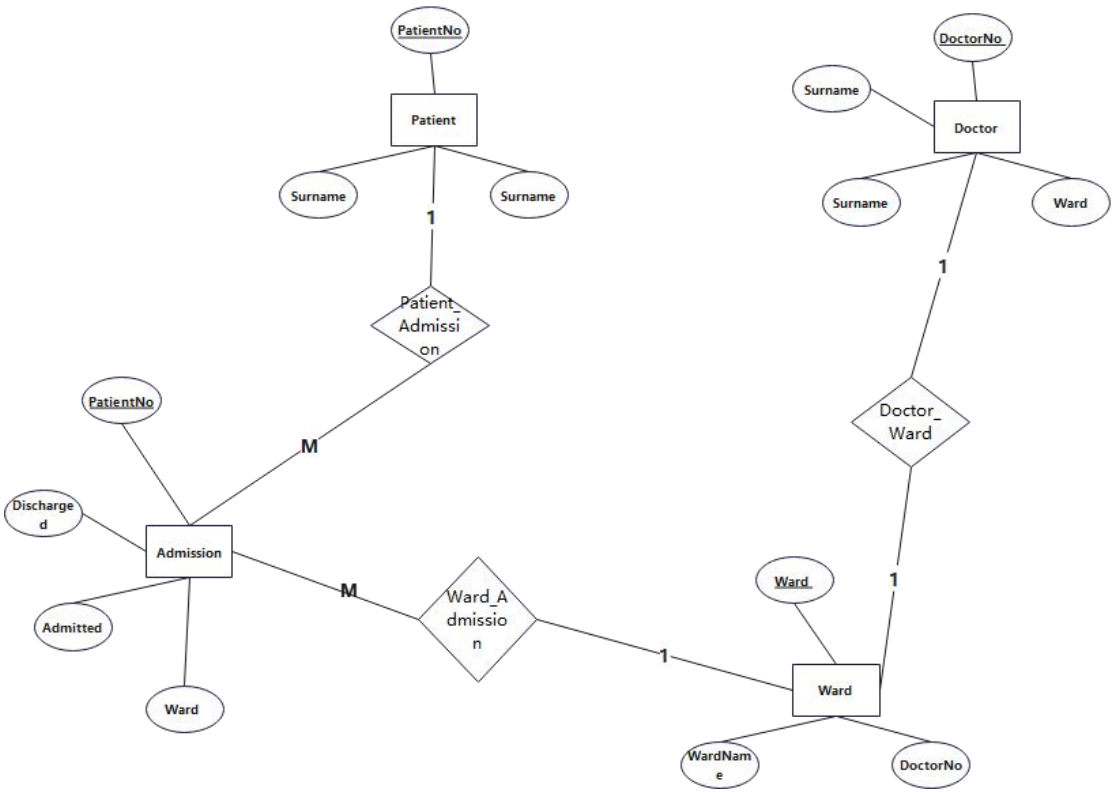
    Relationship: A doctor is responsible for one ward.

    Foreign Key: references .Ward.DoctorNoDoctor.DoctorNo

3. **Ward and Admission Tables**:

    Relationship: Each admission is associated with a specific ward.

    Foreign Key: references .Admission.WardWard.Ward



7.

Table 1: Catering

a. Identify the Functional Dependencies

1. **Order No → Account No, Customer, Address, Date**

   Each Order Number uniquely determines the Account Number, Customer name, Address, and Date.

2. **Order No, Item → Quantity, Item Price**

   Each Item in an Order Number is associated with its Quantity and Item Price.

b. Normalization Process

1NF (First Normal Form)

Remove repeating groups and ensure all fields are atomic.

Catering 1NF Table:

Order No	Account No	Customer	Address	Date	Item	Quantity	Item Price
7823	178	Daisy's Café	27 Bay Drive, Coventry	16/7	Bakewell Tart	20	0.15
7823	178	Daisy's Café	27 Bay Drive, Coventry	16/7	Danish Pastry	13	0.20
7823	178	Daisy's Café	27 Bay Drive, Coventry	16/7	Apple Pie	45	0.15
4633	526	Smiths	12 Dee View, Aberdeen	16/7	Butteries	120	0.20
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	Apple Pie	130	0.15
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	Cherry Pie	100	0.18
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	Steak Pie	30	0.50
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	Meringue Pie	20	0.20
1788	032	Tasty Bite	17 Wood Place, Liverpool	18/7	Apple Pie	15	0.15
1788	032	Tasty Bite	17 Wood Place, Liverpool	18/7	Danish Pastry	50	0.20

2NF (Second Normal Form)

- Remove partial dependencies to ensure all non-key attributes are fully functionally dependent on the primary key.

Catering 2NF Tables:

1. **Orders Table**

Order No	Account No	Customer	Address	Date
7823	178	Daisy's Café	27 Bay Drive, Coventry	16/7
4633	526	Smiths	12 Dee View, Aberdeen	16/7
2276	167	Sally's Snacks	3 High Street, Banchory	17/7
1788	032	Tasty Bite	17 Wood Place, Liverpool	18/7

2. **Items Table**

Order No	Item	Quantity	Item Price
7823	Bakewell Tart	20	0.15
7823	Danish Pastry	13	0.20
7823	Apple Pie	45	0.15
4633	Butteries	120	0.20
2276	Apple Pie	130	0.15
2276	Cherry Pie	100	0.18
2276	Steak Pie	30	0.50
2276	Meringue Pie	20	0.20
1788	Apple Pie	15	0.15
1788	Danish Pastry	50	0.20

3NF (Third Normal Form)

Remove transitive dependencies to ensure all non-key attributes depend only on the primary key.

Catering 3NF Tables:

1. **Orders Table** (remains the same)

Order No	Account No	Customer	Address	Date
7823	178	Daisy's Café	27 Bay Drive, Coventry	16/7
4633	526	Smiths	12 Dee View, Aberdeen	16/7
2276	167	Sally's Snacks	3 High Street, Banchory	17/7
1788	032	Tasty Bite	17 Wood Place, Liverpool	18/7

2. **Items Table** (remains the same)

Order No	Item	Quantity	Item Price

Order No	Item	Quantity	Item Price
7823	Bakewell Tart	20	0.15
7823	Danish Pastry	13	0.20
7823	Apple Pie	45	0.15
4633	Butteries	120	0.20
2276	Apple Pie	130	0.15
2276	Cherry Pie	100	0.18
2276	Steak Pie	30	0.50
2276	Meringue Pie	20	0.20
1788	Apple Pie	15	0.15
1788	Danish Pastry	50	0.20

Table 2: Student Records

a. Identify the Functional Dependencies

1. **Student No → Name, Course, Course Duration**

   Each Student Number uniquely determines the Student's Name, Course, and Course Duration.

2. **Course → Module No, Module Name, Lecturer**

   Each Course corresponds to multiple Modules and their Lecturers.

3. **Module No → Module Name, Lecturer**

   Each Module Number uniquely determines its Module Name and Lecturer.

b. Normalization Process

1NF (First Normal Form)

Remove repeating groups and ensure all fields are atomic.

Student Records 1NF Table:

Student No	Name	Course	Course Duration	Module No	Module Name	Lecturer
1002	Salif Keita	G701	4	COF104	Java	Asimov
1002	Salif Keita	G701	4	COF118	Distributed Systems	Patel
1005	Emma Wilson	G504	3	COF105	Computer Architecture	Zidane
1005	Emma Wilson	G504	3	COF118	Distributed Systems	Patel
1005	Emma Wilson	G504	3	COF120	Operating Systems	Brando
1006	Hong Wang	G701	4	COF111	Networks	Austin
1006	Hong Wang	G701	4	COF105	Computer Architecture	Zidane
1010	Kiri Anahera	G722	2	COF111	Networks	Austin
1010	Kiri Anahera	G722	2	COF105	Computer Architecture	Zidane

2NF (Second Normal Form)

Remove partial dependencies to ensure all non-key attributes are fully functionally dependent on the primary key.

Student Records 2NF Tables:

1. **Students Table**

Student No	Name	Course	Course Duration
1002	Salif Keita	G701	4
1005	Emma Wilson	G504	3
1006	Hong Wang	G701	4
1010	Kiri Anahera	G722	2

2. **Modules Table**

Course	Module No	Module Name	Lecturer
G701	COF104	Java	Asimov
G701	COF118	Distributed Systems	Patel
G504	COF105	Computer Architecture	Zidane

Course	Module No	Module Name	Lecturer
G504	COF118	Distributed Systems	Patel
G504	COF120	Operating Systems	Brando
G701	COF111	Networks	Austin
G722	COF111	Networks	Austin
G701	COF105	Computer Architecture	Zidane

3NF (Third Normal Form)

Remove transitive dependencies to ensure all non-key attributes depend only on the primary key.

Student Records 3NF Tables:

1. **Students Table** (remains the same)

Student No	Name	Course	Course Duration
1002	Salif Keita	G701	4
1005	Emma Wilson	G504	3
1006	Hong Wang	G701	4
1010	Kiri Anahera	G722	2

2. **Modules Table** (remains the same)

Course	Module No	Module Name	Lecturer
G701	COF104	Java	Asimov
G701	COF118	Distributed Systems	Patel
G504	COF105	Computer Architecture	Zidane
G504	COF118	Distributed Systems	Patel
G504	COF120	Operating Systems	Brando
G701	COF111	Networks	Austin
G722	COF111	Networks	Austin
G701	COF105	Computer Architecture	Zidane

8.

Part A: Identify Functional Dependencies

From the provided dataset, I can identify the following functional dependencies:

1. **Report ID → Reporting Period**, **Branch Code**, **Branch Name**

   (Each Report ID corresponds to a specific Reporting Period, Branch Code, and Branch Name.)

2. **Branch Code → Branch Name**

   (Each Branch Code corresponds to a specific Branch Name.)

3. **Car Plate Nr → Car Type**

   (Each Car Plate Number corresponds to a specific Car Type.)

4. **Bill Nr → Bill Date**, **Penalty**, **Final Bill**

   (Each Bill Number corresponds to a specific Bill Date, Penalty, and Final Bill amount.)

5. **Supervisor ID → Supervisor Name**

   (Each Supervisor ID corresponds to a specific Supervisor Name.)

Part B: Complete the Normalization Process

Step 1: First Normal Form (1NF)

To satisfy 1NF, I need to ensure all entries are atomic (no repeating groups). The dataset can be structured as follows:

Report ID	Reporting Period	Branch Code	Branch Name	Car Plate Nr	Car Type	Bill Nr	Bill Date	Penalty	Final Bill	Supervisor ID	Supervisor Name
7686	January 2021 to March 2021	876734	Walsall	DS4049	SUV	166651	18.01.2021	50	1050	102	David Brown
7686	January 2021 to March 2021	876734	Walsall	DL3434	Sports Car	123111	19.02.2021	0	500	102	David Brown
7686	January 2021 to March 2021	876734	Walsall	OP9817	SUV	561909	06.03.2021	0	480	102	David Brown
7686	January 2021 to March 2021	876734	Walsall	SJ7182	Hatchback	565690	29.01.2021	0	680	102	David Brown
1056	October 2021 to December 2021	100023	Coventry	BN9745	SUV	128976	10.10.2021	0	710	871	Anna Smith
1056	October 2021 to December 2021	100023	Coventry	LA5142	Sedan	511899	25.11.2021	20	1500	871	Anna Smith
1056	October 2021 to December 2021	100023	Coventry	CB0098	Sports Car	141421	03.12.2021	0	850	871	Anna Smith
1056	October 2021 to Decem	100023	Coventry	ZX7222	Coupe	514879	29.10.2021	0	1250	871	Anna Smith

Report ID	Reporting Period	Branch Code	Branch Name	Car Plate Nr	Car Type	Bill Nr	Bill Date	Penalty	Final Bill	Supervisor ID	Supervisor Name
	ber 2021										
1056	October 2021 to December 2021	100023	Coventry	DL3434	Sports Car	771100	16.11.2021	20	300	871	Anna Smith
4981	January 2022 to March 2022	456109	Leamington Spa	PO8123	SUV	675912	06.01.2022	50	350	149	John Cruise
4981	January 2022 to March 2022	456109	Leamington Spa	IU7878	Hatchback	991762	08.02.2022	0	950	149	John Cruise
4832	January 2022 to March 2022	981256	Warwick	NM8787	Sports Car	876234	14.02.2022	0	350	823	James Doherty
7002	July 2021 to September 2021	981256	Warwick	OP9817	SUV	110054	19.07.2021	100	1400	823	James Doherty
7002	July 2021 to September 2021	981256	Warwick	NM8787	Sports Car	378123	12.08.2021	20	450	823	James Doherty
7002	July 2021 to September 2021	981256	Warwick	VC1111	Sedan	808051	18.09.2021	0	670	823	James Doherty
7002	July 2021 to Septem	981256	Warwick	FG7100	Hatchback	100023	21.07.2021	0	1030	823	James Doherty

Report ID	Reporting Period	Branch Code	Branch Name	Car Plate Nr	Car Type	Bill Nr	Bill Date	Penalty	Final Bill	Supervisor ID	Supervisor Name
	ber 2021										
7002	July 2021 to September 2021	981256	Warwick	RE6000	Sedan	611554	27.08.2021	50	520	823	James Doherty
3121	April 2021 to June 2021	555901	Wolverhampton	TR6199	SUV	888712	10.04.2021	0	490	111	Catherine Johnson
3121	April 2021 to June 2021	555901	Wolverhampton	DL3434	Sports Car	343412	28.05.2021	20	1230	111	Catherine Johnson
3121	April 2021 to June 2021	555901	Wolverhamoton	BP9111	Coupe	222678	04.06.2021	0	1680	111	Catherine Johnson

Step 2: Second Normal Form (2NF)

To achieve 2NF, I must eliminate partial dependencies. This means every non-key attribute must depend on the entire primary key.

Decomposition:

1. **Reports Table** (Primary Key: Report ID)

Report ID	Reporting Period	Branch Code	Supervisor ID
7686	January 2021 to March 2021	876734	102
1056	October 2021 to December 2021	100023	871
4981	January 2022 to March 2022	456109	149
7002	July 2021 to September 2021	981256	823

Report ID	Reporting Period	Branch Code	Supervisor ID
3121	April 2021 to June 2021	555901	111

2. **Bill Details Table** (Primary Key: Composite Key of Report ID and Bill Nr)

Report ID	Bill Nr	Car Plate Nr	Car Type	Bill Date	Penalty	Final Bill
7686	166651	DS4049	SUV	18.01.2021	50	1050
7686	123111	DL3434	Sports Car	19.02.2021	0	500
7686	561909	OP9817	SUV	06.03.2021	0	480
7686	565690	SJ7182	Hatchback	29.01.2021	0	680
1056	128976	BN9745	SUV	10.10.2021	0	710
1056	511899	LA5142	Sedan	25.11.2021	20	1500
1056	141421	CB0098	Sports Car	03.12.2021	0	850
1056	514879	ZX7222	Coupe	29.10.2021	0	1250
1056	771100	DL3434	Sports Car	16.11.2021	20	300
4981	675912	PO8123	SUV	06.01.2022	50	350
4981	991762	IU7878	Hatchback	08.02.2022	0	950
4832	876234	NM8787	Sports Car	14.02.2022	0	350
7002	110054	OP9817	SUV	19.07.2021	100	1400
7002	378123	NM8787	Sports Car	12.08.2021	20	450
7002	808051	VC1111	Sedan	18.09.2021	0	670
7002	100023	FG7100	Hatchback	21.07.2021	0	1030
7002	611554	RE6000	Sedan	27.08.2021	50	520
3121	888712	TR6199	SUV	10.04.2021	0	490
3121	343412	DL3434	Sports Car	28.05.2021	20	1230
3121	222678	BP9111	Coupe	04.06.2021	0	1680

3. **Branches Table** (Primary Key: Branch Code)

Branch Code	Branch Name
876734	Walsall
100023	Coventry
456109	Leamington Spa
981256	Warwick
555901	Wolverhampton

4. **Supervisors Table** (Primary Key: Supervisor ID)

Supervisor ID	Supervisor Name
102	David Brown
871	Anna Smith
149	John Cruise
823	James Doherty
111	Catherine Johnson

Step 3: Third Normal Form (3NF)

In 3NF, I must remove transitive dependencies. The decomposition already satisfies this condition since all non-key attributes depend only on primary keys in their respective tables.

Summary of Tables After Normalization

1. **Reports Table**

   Contains Report ID, Reporting Period, Branch Code, and Supervisor ID.

2. **Bill Details Table**

   Contains Bill Details linked to Report ID, including Bill Number, Car Plate Number, Car Type, Bill Date, Penalty, and Final Bill.

3. **Branches Table**

   Contains Branch Codes and their corresponding Branch Names.

4. **Supervisors Table**

   Contains Supervisor IDs and their corresponding names.

1. Database Development Lifecycle

a. Logical Model

Entities:

Engineer

Boat

Contractor

Maintenance Task

Service Record

Attributes:

Engineer

Engineer_ID (Primary Key)

Name

Skills

Phone

Boat

Boat_ID (Primary Key)

Equipment_Type

Service_Time

Contractor

Contractor_ID (Primary Key)

Name

Specialization

Phone

Maintenance Task

Task_ID (Primary Key)

Task_Type (ENUM type: Software Upgrade, Repair, Safety Inspection, Other)

Service Record

Service_ID (Primary Key)

Boat_ID (Foreign Key referencing Boat)

Engineer_ID (Foreign Key referencing Engineer)

Contractor_ID (Foreign Key referencing Contractor)

Task_ID (Foreign Key referencing Maintenance Task)

Service_Date

Man_Hours

b. Physical Model

Table Structure and Data Types:

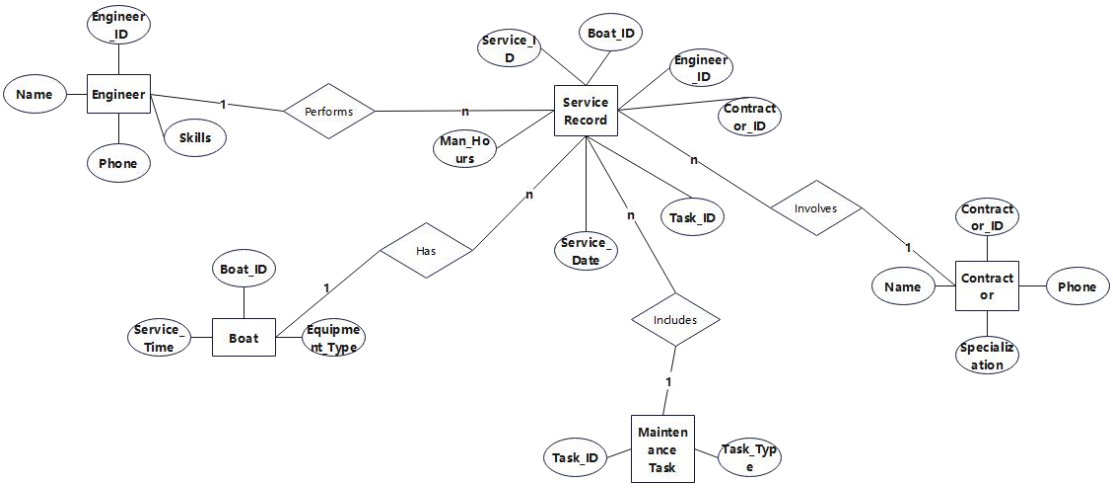
```
-- Engineer Table
CREATE TABLE Engineer (
    Engineer_ID INT PRIMARY KEY,
    Name VARCHAR(100),
    Skills VARCHAR(255),
    Phone VARCHAR(15)
);
```

```
-- Boat Table
CREATE TABLE Boat (
    Boat_ID INT PRIMARY KEY,
    Equipment_Type VARCHAR(100),
    Service_Time DATETIME
);

-- Contractor Table
CREATE TABLE Contractor (
    Contractor_ID INT PRIMARY KEY,
    Name VARCHAR(100),
    Specialization VARCHAR(100),
    Phone VARCHAR(15)
);

-- Maintenance Task Table
CREATE TABLE Maintenance_Task (
    Task_ID INT PRIMARY KEY,
    Task_Type ENUM('Software Upgrade', 'Repair', 'Safety Inspection', 'Other')
);

-- Service Record Table
CREATE TABLE Service_Record (
    Service_ID INT PRIMARY KEY,
    Boat_ID INT,
    Engineer_ID INT,
    Contractor_ID INT,
    Task_ID INT,
    Service_Date DATETIME,
    Man_Hours DECIMAL(5, 2),
    FOREIGN KEY (Boat_ID) REFERENCES Boat(Boat_ID),
    FOREIGN KEY (Engineer_ID) REFERENCES Engineer(Engineer_ID),
    FOREIGN KEY (Contractor_ID) REFERENCES Contractor(Contractor_ID),
    FOREIGN KEY (Task_ID) REFERENCES Maintenance_Task(Task_ID)
);
```



ER Diagram (text representation):

Engineer (Engineer_ID) ---< Service_Record (Service_ID)

Boat (Boat_ID) ---< Service_Record (Service_ID)

Contractor (Contractor_ID) ---< Service_Record (Service_ID)

Maintenance_Task (Task_ID) ---< Service_Record (Service_ID)

c. Identify Functional Dependencies

The following are the functional dependencies for each table:

Engineer:

Engineer_ID → Name, Skills, Phone

Boat:

Boat_ID → Equipment_Type, Service_Time

Contractor:

Contractor_ID → Name, Specialization, Phone

Maintenance Task:

Task_ID → Task_Type

Service Record:

Service_ID → Boat_ID, Engineer_ID, Contractor_ID, Task_ID,

Service_Date, Man_Hours

d. Populate the Developed Relations with Test (Fictitious) Data

Here are the SQL statements to populate the tables with test data:

```
-- Engineer data
INSERT INTO Engineer (Engineer_ID, Name, Skills, Phone) VALUES
(1, 'John Doe', 'Electrical, Mechanical', '123-456-7890'),
(2, 'Jane Smith', 'Software, Mechanical', '098-765-4321'),
(3, 'Mike Brown', 'Repair, Safety Inspection', '456-789-0123'),
(4, 'Anna White', 'Software Upgrade', '321-654-0987'),
(5, 'Tom Green', 'Mechanical, Safety Inspection', '654-321-9870'),
(6, 'Emily Blue', 'Electrical', '789-123-4567'),
(7, 'Robert Black', 'Mechanical', '135-792-4680');


-- Boat data
```

```
INSERT INTO Boat (Boat_ID, Equipment_Type, Service_Time) VALUES
(1, 'Engine', '2024-10-01 09:00:00'),
(2, 'Hull', '2024-10-02 10:00:00'),
(3, 'Propeller', '2024-10-03 11:00:00');

-- Contractor data
INSERT INTO Contractor (Contractor_ID, Name, Specialization, Phone) VALUES
(1, 'Bob Johnson', 'Electrical', '321-654-0987'),
(2, 'Alice Brown', 'Mechanical', '789-012-3456'),
(3, 'Charlie White', 'Safety Inspection', '456-123-7890');

-- Maintenance Task data
INSERT INTO Maintenance_Task (Task_ID, Task_Type) VALUES
(1, 'Software Upgrade'),
(2, 'Repair'),
(3, 'Safety Inspection'),
(4, 'Other');

-- Service Record data
INSERT INTO Service_Record (Service_ID, Boat_ID, Engineer_ID, Contractor_ID,
Task_ID, Service_Date, Man_Hours) VALUES
(1, 1, 1, 1, 1, '2024-10-01 09:00:00', 4.5),
```

(2, 2, 2, 2, 2, '2024-10-02 10:00:00', 3.0),

(3, 3, 3, 3, 3, '2024-10-03 11:00:00', 5.0);

e. Complete Normalization to Minimize Redundancy and Dependency in the Database Relations

First Normal Form (1NF): Ensure each table has atomic values and no repeating groups. All the table structures meet 1NF.

Second Normal Form (2NF): Ensure all non-key attributes are fully functionally dependent on the primary key, eliminating partial dependencies. All tables have attributes appropriately assigned to primary keys.

Third Normal Form (3NF): Ensure there are no transitive dependencies; non-key attributes must not depend on other non-key attributes. The design above has eliminated any transitive dependencies.

The final model has been normalized effectively to reduce redundancy and dependencies.