Basic Tasks

1.

a. Candidate Key

Definition: A Candidate Key is an attribute group in a relational schema whose value uniquely identifies a tuple (i.e., a row in a table) in the relation. If any attribute is removed from this group, it no longer possesses this property. In other words, a Candidate Key is an attribute or attribute group in a relational table that can uniquely determine a single record.

b. Composite Key

Definition: A Composite Key is a Candidate Key composed of two or more attributes. These attributes, when combined, uniquely identify a tuple in the relation, but no single attribute alone can achieve this.

c. Foreign Key

Definition: A Foreign Key is a column or a combination of columns in a table whose value must correspond to the value of the primary key or a unique key in another table. Foreign Keys are used to establish and maintain data integrity and associative relationships between two tables.

d. Functional Dependency

Definition: Functional Dependency refers to the situation in a relational database where the value of one attribute (or attribute set) uniquely determines the value of another attribute (or attribute set). Simply put, the value of one attribute determines the value of another.

2.

Entity Integrity

Definition: Entity Integrity requires that each tuple (i.e., each row in a table) in a relation must be unique in the attributes that constitute the primary key, and the primary key attribute values cannot be NULL. This ensures that each row in the table can be uniquely identified, thereby avoiding data redundancy and inconsistency.

Referential Integrity

Definition: Referential Integrity is a constraint between tables in a relational database that requires the values of foreign key columns in the foreign key table to be one of the values of the primary key columns in the primary key table, or NULL. This ensures that the associative relationships between tables are correct, preventing the occurrence of data inconsistency.

User-Defined Integrity

Definition: User-Defined Integrity refers to the constraint conditions specific to a particular relational database, which reflect the semantic requirements that the data involved in a particular application must meet. These constraint conditions are not defined by the relational model itself but are set according to the specific needs of users.

3.

Table 1: The genre of 008 in Film table is missing. **Entity Integrity**

Table 2: The quantity of Aziz is characters but not numbers. **User-Defined Integrity**

Table 3: The artist of A King of Magic is missing. **Entity Integrity**

Medium Tasks

4.

a. Insert New Project: Exception: Attempting to insert a new project (e.g., ProjectCode=PRC30, ProjectTitle=Skills Matrix, etc.) without providing complete employee information or related fields may result in incomplete data or formatting errors.

b. Delete Project: Exception: Deleting a project (e.g., PRC10) while there is still related employee information under that project may lead to data loss or inconsistencies.

c. Employee Movement: Exception: If an employee (e.g., J Kirk) is moved to a different department but the corresponding department information is not updated in the table, it may cause data inconsistencies.

d. 1NF:
Data Redundancy: When a field within a table can be further decomposed into smaller, indivisible basic data items, it breaches 1NF. This results in data redundancy as identical data recurs across multiple records.

Update Anomaly: Due to data redundancy, altering a data item might necessitate updating multiple records concurrently, not only complicating maintenance but also potentially leading to data discrepancies from inconsistent updates.

Insertion Anomaly: If a record encompasses a composite data item (i.e., one that can be further decomposed) and part of this composite item's information is not yet available, the record cannot be inserted since databases mandate that all fields be non-null.

Deletion Anomaly: Removing a record containing a composite data item may cause the loss of other information linked to that composite item, as other records might also rely on parts of its information.

e. 2NF

Partial Functional Dependency: When non-primary key fields in a table partially depend on the primary key (i.e., they depend on only a portion of the primary key), it violates 2NF. This leads to data redundancy and update anomalies.

Update Anomaly: Owing to partial functional dependency, modifying a data item could affect multiple records. For instance, if the primary key comprises multiple attributes and a non-primary key field depends on just part of it, changing that part of the primary key might require updating the non-primary key field in numerous records.

Insertion Anomaly: Akin to 1NF, if part of the primary key's information is not yet available, non-primary key fields reliant on that part cannot be inserted.

f. 3NF

Transitive Functional Dependency: When non-primary key fields in a table indirectly depend on the primary key (i.e., they depend on another non-primary key field, which itself depends on the primary key), it breaches 3NF. This gives rise to data redundancy and update anomalies.

Update Anomaly: Due to transitive functional dependency, updating a data item might necessitate updating other multiple records linked through the transitive dependency chain, escalating maintenance complexity and potentially resulting in data inconsistency.

Data Redundancy: Similar to 1NF and 2NF, violating 3NF also leads to data redundancy. Redundant data not only consumes additional storage but may also cause data errors from inconsistent updates.

5.

Primary Key:

Order No. :This will uniquely identify each order

Foreign Key:

Acc.No. :This could be a foreign key referencing the Account table, which would hold customer account information

Primary Key:

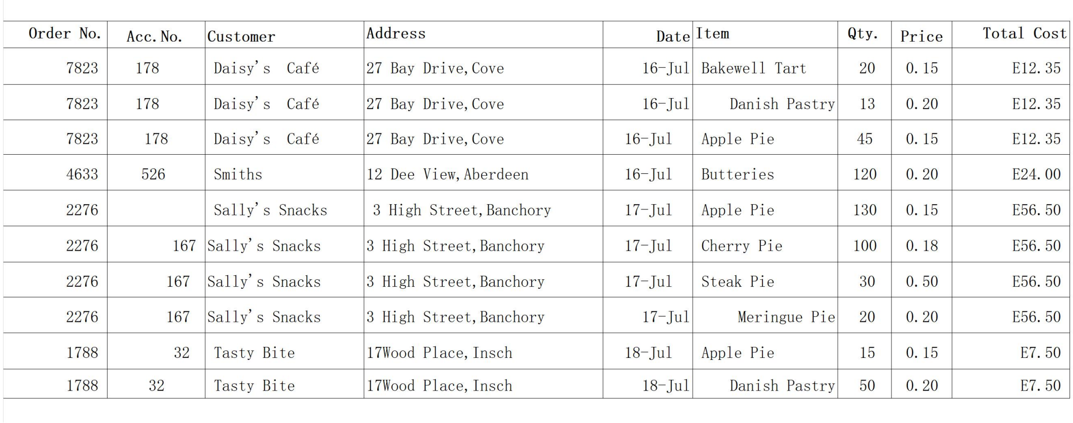
Composite key consisting of Order No. and Item

Foreign Key:

Order No. : This is a foreign key referencing the Orders table, indicating which order the detail is associated with

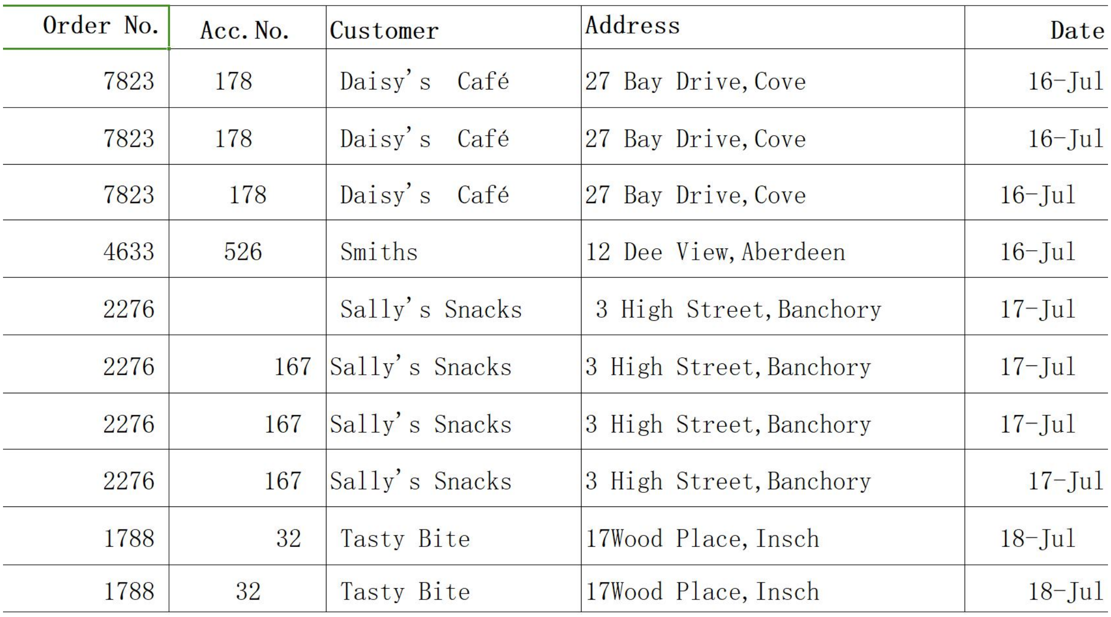
a. The same data appears repeatedly in multiple records, resulting in data redundancy.

b.

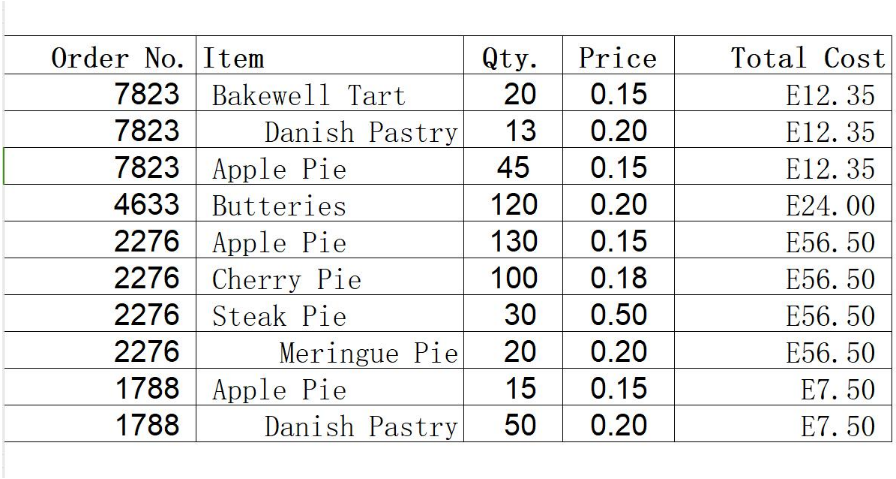


Order No.	Acc.No.	Customer	Address	Date	Item	Qty.	Price	Total Cost
7823	178	Daisy's Café	27 Bay Drive,Cove	16-Jul	Bakewell Tart	20	0.15	E12.35
7823	178	Daisy's Café	27 Bay Drive,Cove	16-Jul	Danish Pastry	13	0.20	E12.35
7823	178	Daisy's Café	27 Bay Drive,Cove	16-Jul	Apple Pie	45	0.15	E12.35
4633	526	Smiths	12 Dee View,Aberdeen	16-Jul	Butteries	120	0.20	E24.00
2276		Sally's Snacks	3 High Street,Banchory	17-Jul	Apple Pie	130	0.15	E56.50
2276	167	Sally's Snacks	3 High Street,Banchory	17-Jul	Cherry Pie	100	0.18	E56.50
2276	167	Sally's Snacks	3 High Street,Banchory	17-Jul	Steak Pie	30	0.50	E56.50
2276	167	Sally's Snacks	3 High Street,Banchory	17-Jul	Meringue Pie	20	0.20	E56.50
1788	32	Tasty Bite	17Wood Place,Insch	18-Jul	Apple Pie	15	0.15	E7.50
1788	32	Tasty Bite	17Wood Place,Insch	18-Jul	Danish Pastry	50	0.20	E7.50

c.



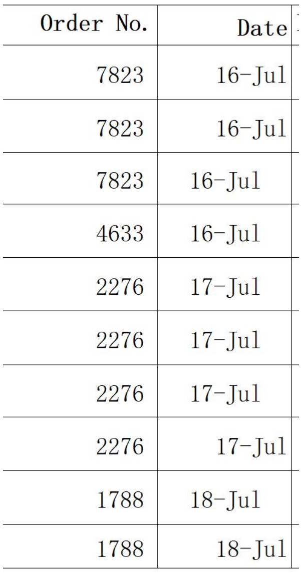
Order No.	Acc.No.	Customer	Address	Date
7823	178	Daisy's Café	27 Bay Drive,Cove	16-Jul
7823	178	Daisy's Café	27 Bay Drive,Cove	16-Jul
7823	178	Daisy's Café	27 Bay Drive,Cove	16-Jul
4633	526	Smiths	12 Dee View,Aberdeen	16-Jul
2276		Sally's Snacks	3 High Street,Banchory	17-Jul
2276	167	Sally's Snacks	3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



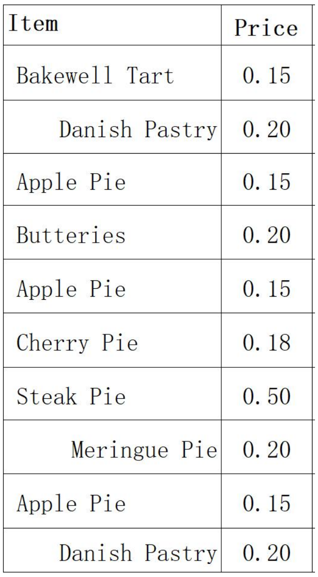
Order No.	Item	Qty.	Price	Total Cost
7823	Bakewell Tart	20	0.15	E12.35
7823	Danish Pastry	13	0.20	E12.35
7823	Apple Pie	45	0.15	E12.35
4633	Butteries	120	0.20	E24.00
2276	Apple Pie	130	0.15	E56.50
2276	Cherry Pie	100	0.18	E56.50
2276	Steak Pie	30	0.50	E56.50
2276	Meringue Pie	20	0.20	E56.50
1788	Apple Pie	15	0.15	E7.50
1788	Danish Pastry	50	0.20	E7.50

d.

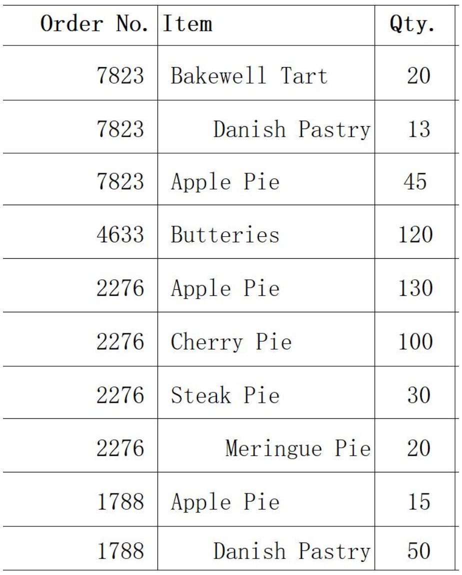
Acc. No.	Customer	Address
178	Daisy's Café	27 Bay Drive, Cove
178	Daisy's Café	27 Bay Drive, Cove
178	Daisy's Café	27 Bay Drive, Cove
526	Smiths	12 Dee View, Aberdeen
	Sally's Snacks	3 High Street, Banchory
167	Sally's Snacks	3 High Street, Banchory
167	Sally's Snacks	3 High Street, Banchory
167	Sally's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



Order No.	Date
7823	16-Jul
7823	16-Jul
7823	16-Jul
4633	16-Jul
2276	17-Jul
2276	17-Jul
2276	17-Jul
2276	17-Jul
1788	18-Jul
1788	18-Jul



Item	Price
Bakewell Tart	0.15
Danish Pastry	0.20
Apple Pie	0.15
Butteries	0.20
Apple Pie	0.15
Cherry Pie	0.18
Steak Pie	0.50
Meringue Pie	0.20
Apple Pie	0.15
Danish Pastry	0.20



Order No.	Item	Qty.
7823	Bakewell Tart	20
7823	Danish Pastry	13
7823	Apple Pie	45
4633	Butteries	120
2276	Apple Pie	130
2276	Cherry Pie	100
2276	Steak Pie	30
2276	Meringue Pie	20
1788	Apple Pie	15
1788	Danish Pastry	50

6.

a.

Patient (PatientNo, Surname, FirstNname)

Admission (PatientNo, Admitted, Discharged, Ward)

Doctor (DoctorNo, Surname, FirstName, Ward)

Ward (Ward, WardName, DoctorNo-InCharge)

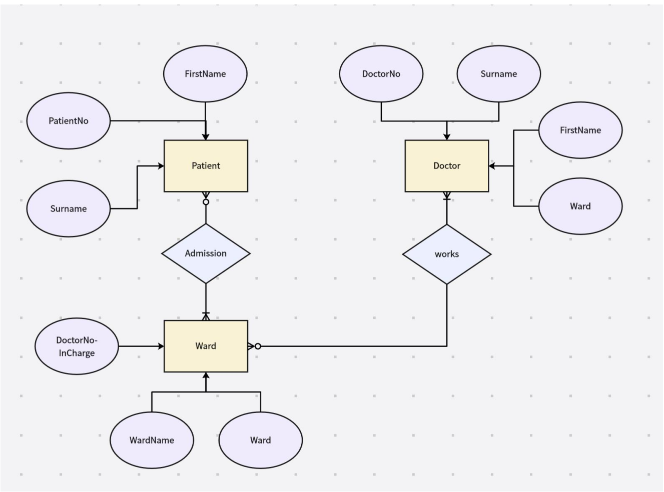
b.

In the "Admission" table, PatientNo is a foreign key that references PatientNo in the "Patient" table.

In the "Admission" table, Ward is a foreign key that references Ward in the "Ward" table.

In the “Doctor” table, DoctorNo is a foreign key that references DoctorNo-InCharge in the “Ward” table.

c.



Advanced Tasks

7.

a. Functional Dependencies in Table 1 and Table 2

Table 1: Catering

Order No → Account No, Customer, Address, Date, Item, Quantity, Item Price

Account No → Customer, Address

Customer → Address

Date → Item, Quantity, Item Price

Item → Quantity, Item Price

Table 2: Student Records

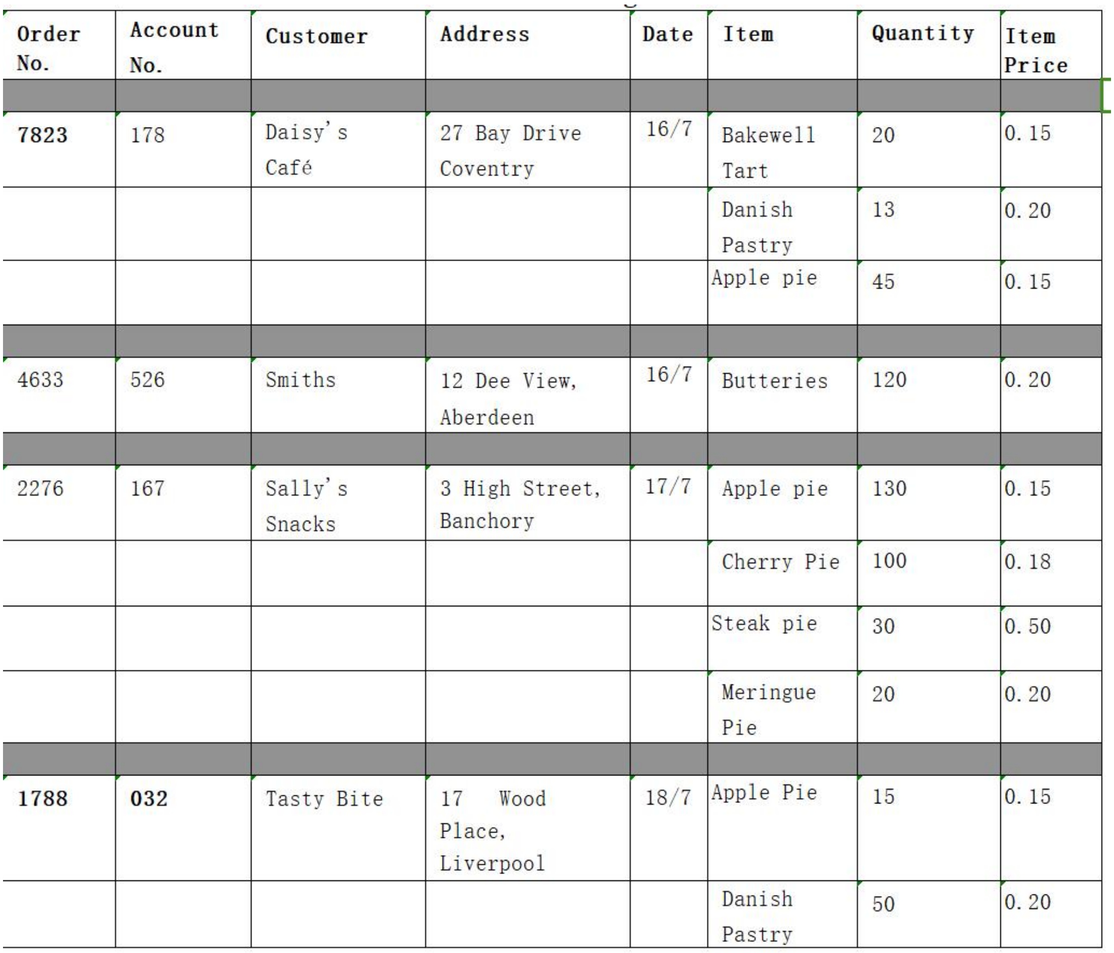
Student No → Name, Course, Course Duration, Module No, Module Name, Lecturer

Course → Course Duration

Module No → Module Name, Lecturer

Lecturer → Module Name

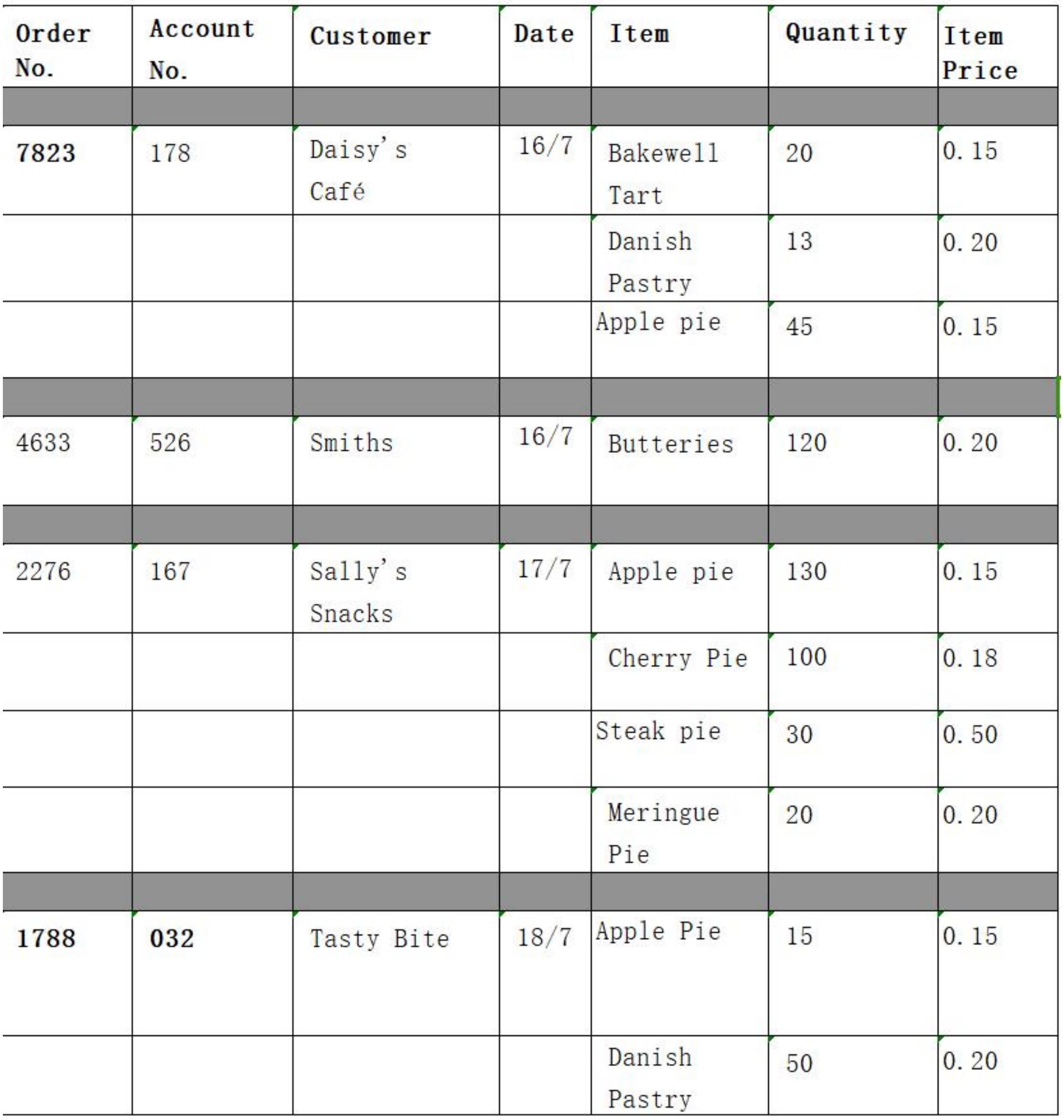
b. Table 1: 1NF



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
					Danish Pastry	13	0.20
					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
					Cherry Pie	100	0.18
					Steak pie	30	0.50
					Meringue Pie	20	0.20
1788	**032**	Tasty Bite	17 Wood Place, Liverpool	18/7	Apple Pie	15	0.15
					Danish Pastry	50	0.20

2NF: In 2NF, the table should be in 1NF, and all non-key attributes should be fully functionally dependent on the primary key. Since the primary key (Order No) already uniquely determines all other attributes, Table 1 is already in 2NF.

3NF:

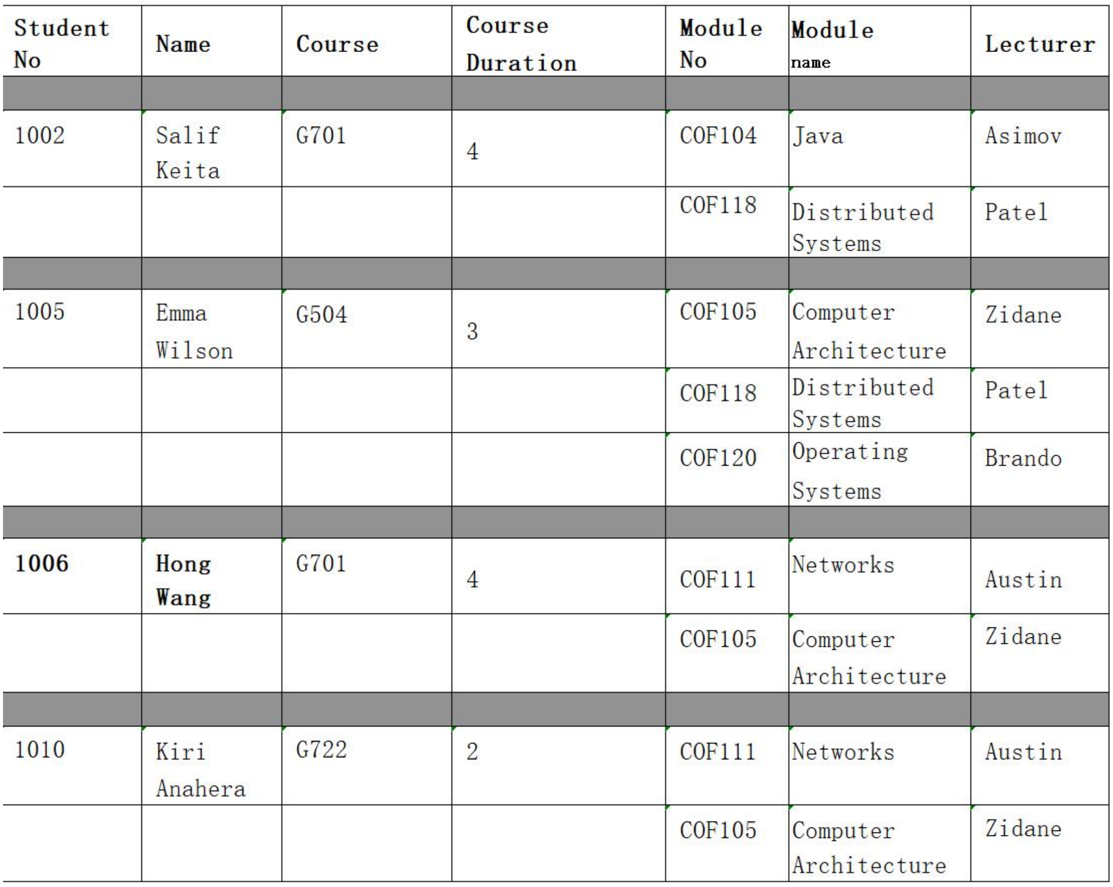


Order No.	**Account No.**	**Customer**	**Date**	**Item**	**Quantity**	**Item Price**
7823	178	Daisy's Café	16/7	Bakewell Tart	20	0.15
				Danish Pastry	13	0.20
				Apple pie	45	0.15
4633	526	Smiths	16/7	Butteries	120	0.20
2276	167	Sally's Snacks	17/7	Apple pie	130	0.15
				Cherry Pie	100	0.18
				Steak pie	30	0.50
				Meringue Pie	20	0.20
1788	**032**	Tasty Bite	18/7	Apple Pie	15	0.15
				Danish Pastry	50	0.20

Customer	Address
Daisy’s Café	27 Bay Drive Coventry
Smiths	12 Dee View, Aberdeen
Sally’s Snacks	3 High Street, Banchory
Tasty Bite	17 Wood Place, Liverpool

Table2:

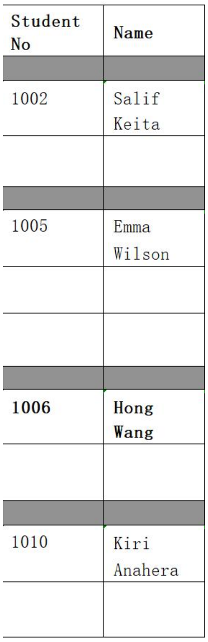
1NF:



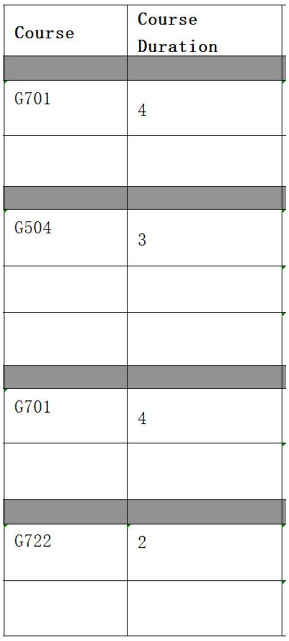
Student No	Name	Course	Course Duration	Module No	Module name	Lecturer
1002	Salif Keita	G701	4	COF104	Java	Asimov
				COF118	Distributed Systems	Patel
1005	Emma Wilson	G504	3	COF105	Computer Architecture	Zidane
				COF118	Distributed Systems	Patel
				COF120	Operating Systems	Brando
1006	**Hong Wang**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

2NF: In 2NF, the table should be in 1NF, and all non-key attributes should be fully functionally dependent on the primary key. Since the primary key (Student No) already uniquely determines all other attributes, Table 2 is already in 2NF.

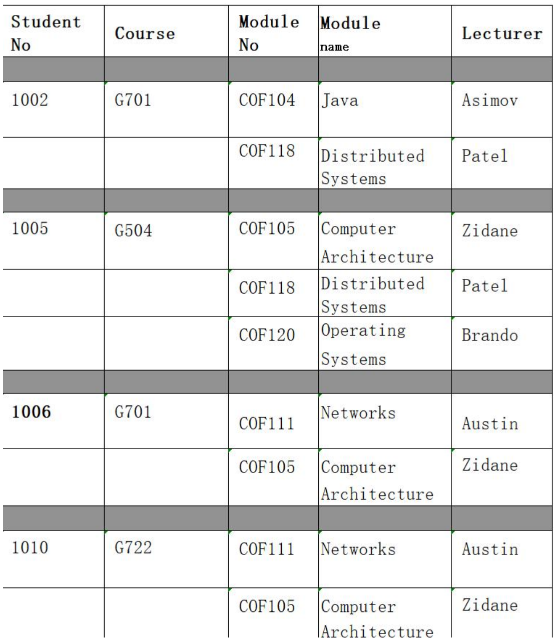
3NF:



Student No	Name
1002	Salif Keita
1005	Emma Wilson
1006	**Hong Wang**
1010	Kiri Anahera



Course	Course Duration
G701	4
G504	3
G701	4
G722	2



Student No	Course	Module No	Module name	Lecturer
1002	G701	COF104	Java	Asimov
		COF118	Distributed Systems	Patel
1005	G504	COF105	Computer Architecture	Zidane
		COF118	Distributed Systems	Patel
		COF120	Operating Systems	Brando
1006	G701	COF111	Networks	Austin
		COF105	Computer Architecture	Zidane
1010	G722	COF111	Networks	Austin
		COF105	Computer Architecture	Zidane

8.

a.

BranchCode → BranchName

CarPlateNr → CarType

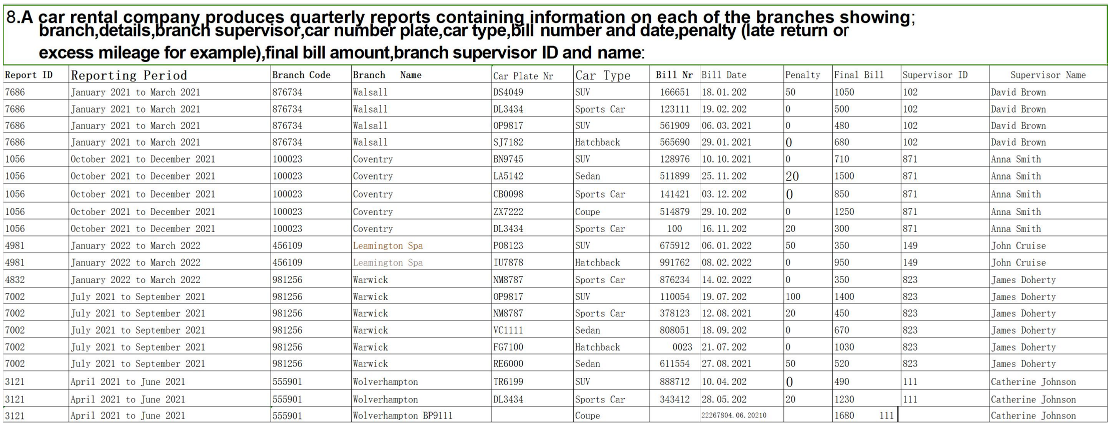
BillNr → BillDate

Penalty → FinalBill

SupervisorID → SupervisorName

b.

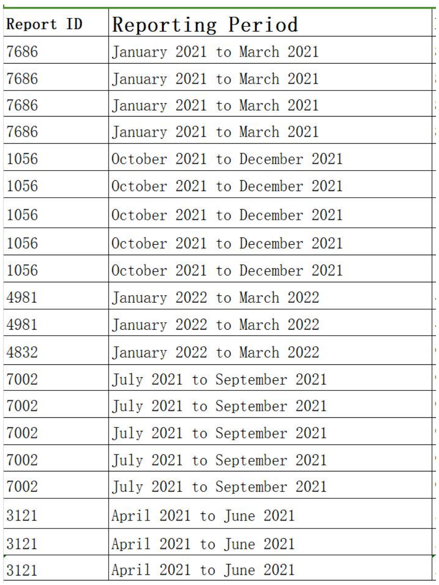
1NF



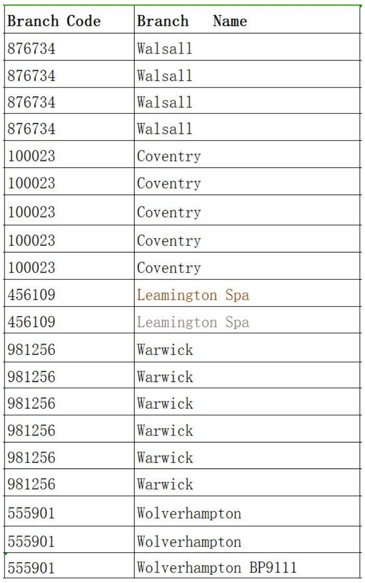
8.A car rental company produces quarterly reports containing information on each of the branches showing; branch,details,branch supervisor,car number plate,car type,bill number and date,penalty (late return or excess mileage for example),final bill amount,branch supervisor ID and name:

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	18.01.202	50	1050	102	David Brown
7686	January 2021 to March 2021	876734	Walsall	DL3434	Sports Car	123111	19.02.202	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	25.11.202	20	1500	871	Anna Smith
1056	October 2021 to December 2021	100023	Coventry	CB0098	Sports Car	141421	03.12.202	0	850	871	Anna Smith
1056	October 2021 to December 2021	100023	Coventry	ZX7222	Coupe	514879	29.10.202	0	1250	871	Anna Smith
1056	October 2021 to December 2021	100023	Coventry	DL3434	Sports Car	100	16.11.202	20	300	871	Anna Smith
4981	January 2022 to March 2022	456109	Leamington Spa	P08123	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	19.07.202	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	18.09.202	0	670	823	James Doherty
7002	July 2021 to September 2021	981256	Warwick	FG7100	Hatchback	0023	21.07.202	0	1030	823	James Doherty
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	10.04.202	0	490	111	Catherine Johnson
3121	April 2021 to June 2021	555901	Wolverhampton	DL3434	Sports Car	343412	28.05.202	20	1230	111	Catherine Johnson
3121	April 2021 to June 2021	555901	Wolverhampton BP9111		Coupe		22267804.06.20210		1680 111		Catherine Johnson

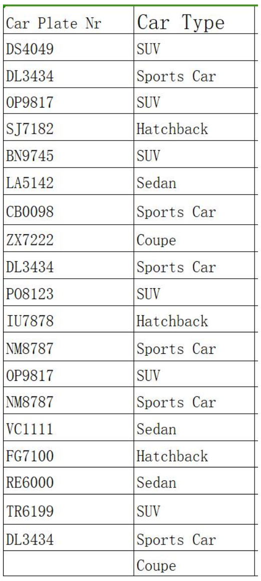
2NF：



Report ID	Reporting Period
7686	January 2021 to March 2021
7686	January 2021 to March 2021
7686	January 2021 to March 2021
7686	January 2021 to March 2021
1056	October 2021 to December 2021
1056	October 2021 to December 2021
1056	October 2021 to December 2021
1056	October 2021 to December 2021
1056	October 2021 to December 2021
4981	January 2022 to March 2022
4981	January 2022 to March 2022
4832	January 2022 to March 2022
7002	July 2021 to September 2021
7002	July 2021 to September 2021
7002	July 2021 to September 2021
7002	July 2021 to September 2021
7002	July 2021 to September 2021
3121	April 2021 to June 2021
3121	April 2021 to June 2021
3121	April 2021 to June 2021



Branch Code	Branch Name
876734	Walsall
876734	Walsall
876734	Walsall
876734	Walsall
100023	Coventry
100023	Coventry
100023	Coventry
100023	Coventry
100023	Coventry
456109	Leamington Spa
456109	Leamington Spa
981256	Warwick
981256	Warwick
981256	Warwick
981256	Warwick
981256	Warwick
981256	Warwick
555901	Wolverhampton
555901	Wolverhampton
555901	Wolverhampton BP9111



Car Plate Nr	Car Type
DS4049	SUV
DL3434	Sports Car
OP9817	SUV
SJ7182	Hatchback
BN9745	SUV
LA5142	Sedan
CB0098	Sports Car
ZX7222	Coupe
DL3434	Sports Car
P08123	SUV
IU7878	Hatchback
NM8787	Sports Car
OP9817	SUV
NM8787	Sports Car
VC1111	Sedan
FG7100	Hatchback
RE6000	Sedan
TR6199	SUV
DL3434	Sports Car
	Coupe

Bill Nr	Bill Date
166651	18. 01. 202
123111	19. 02. 202
561909	06. 03. 2021
565690	29. 01. 2021
128976	10. 10. 2021
511899	25. 11. 202
141421	03. 12. 202
514879	29. 10. 202
100	16. 11. 202
675912	06. 01. 2022
991762	08. 02. 2022
876234	14. 02. 2022
110054	19. 07. 202
378123	12. 08. 2021
808051	18. 09. 202
0023	21. 07. 202
611554	27. 08. 2021
888712	10. 04. 202
343412	28. 05. 202
	22267804. 06. 20210

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

Report ID	**Branch Code**	Car Plate Nr	**Bill Nr**	Penalty	Final Bill	Supervisor ID
7686	876734	DS4049	166651	50	1050	102
7686	876734	DL3434	123111	0	500	102
7686	876734	OP9817	561909	0	480	102
7686	876734	SJ7182	565690	0	680	102
1056	100023	BN9745	128976	0	710	871
1056	100023	LA5142	511899	20	1500	871
1056	100023	CB0098	141421	0	850	871
1056	100023	ZX7222	514879	0	1250	871
1056	100023	DL3434	100	20	300	871
4981	456109	P08123	675912	50	350	149
4981	456109	IU7878	991762	0	950	149
4832	981256	NM8787	876234	0	350	823
7002	981256	OP9817	110054	100	1400	823
7002	981256	NM8787	378123	20	450	823
7002	981256	VC1111	808051	0	670	823
7002	981256	FG7100	0023	0	1030	823
7002	981256	RE6000	611554	50	520	823
3121	555901	TR6199	888712	0	490	111
3121	555901	DL3434	343412	20	1230	111
3121	555901				1680 111	

3NF is the same as 2NF

9.

a. Database Development Life Cycle and Models for M70 Scenario

1. Requirements Analysis:

Gather and understand the requirements from the M70 scenario.

Identify the main entities: Boat, Engineer, Contractor, Task, and Service.

2. Conceptual Design:

Develop a high-level conceptual model representing the main entities and their relationships.

3. Logical Design:

Translate the conceptual model into a logical model.

Define tables and their relationships.

Logical Model:

Boat (BoatID, BoatName, EquipmentType)

Engineer (EngineerID, EngineerName, Specialization)

Contractor (ContractorID, ContractorName, Skill)

Task (TaskID, TaskName)

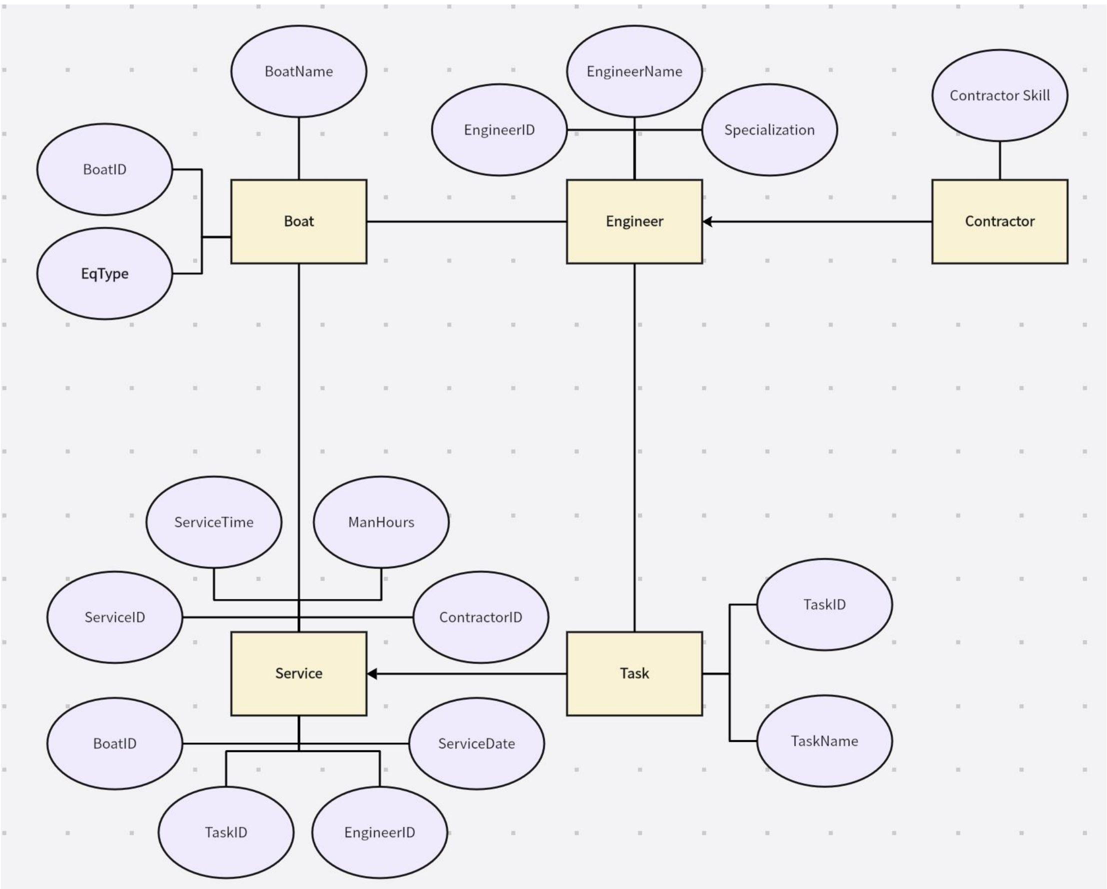
Service (ServiceID, BoatID, TaskID, EngineerID, ContractorID (nullable), ServiceDate, ServiceTime, ManHours)

4.Physical Model:

SQL:

```
CREATE TABLE Boat (
        BoatID INT PRIMARY KEY,
        BoatName VARCHAR(100),
        EquipmentType VARCHAR(100)
);

CREATE TABLE Engineer (
        EngineerID INT PRIMARY KEY,
        EngineerName VARCHAR(100),
        Specialization VARCHAR(100)
);

CREATE TABLE Contractor (
        ContractorID INT PRIMARY KEY,
        ContractorName VARCHAR(100),
        Skill VARCHAR(100)
);

CREATE TABLE Task (
        TaskID INT PRIMARY KEY,
        TaskName VARCHAR(100)
);

CREATE TABLE Service (
	ServiceID INT PRIMARY KEY,
	BoatID INT,
	TaskID INT,
	EngineerID INT,
	ContractorID INT,
	ServiceDate DATE,
	ServiceTime TIME,
	ManHours DECIMAL(5,2),
	FOREIGN KEY (BoatID) REFERENCES Boat(BoatID),
	FOREIGN KEY (TaskID) REFERENCES Task(TaskID),
	FOREIGN KEY (EngineerID) REFERENCES Engineer(EngineerID),
	FOREIGN KEY (ContractorID) REFERENCES Contractor(ContractorID) ON DELETE SET NULL
);
```

b.



c. Functional Dependencies

For the given tables, the functional dependencies are:

Boat: BoatID -> BoatName, EquipmentType

Engineer: EngineerID -> EngineerName, Specialization
Contractor: ContractorID -> ContractorName, Skill
Task: TaskID -> TaskName
Service:
ServiceID -> BoatID, TaskID, EngineerID, ContractorID, ServiceDate, ServiceTime, ManHours
(BoatID, TaskID, EngineerID, ServiceDate) -> ServiceID (Candidate Key, assuming no two services can have the same combination of these attributes)

d. Test Data Population

SQL

```
INSERT INTO Boat (BoatID, BoatName, EquipmentType) VALUES
(1, 'Boat A', 'Type 1'),
(2, 'Boat B', 'Type 2');

INSERT INTO Engineer (EngineerID, EngineerName, Specialization) VALUES
(1, 'John Doe', 'Mechanical'),
(2, 'Jane Smith', 'Electrical');

INSERT INTO Contractor (ContractorID, ContractorName, Skill) VALUES
(1, 'Contractor A', 'Software'),
(2, 'Contractor B', 'Special Repair');

INSERT INTO Task (TaskID, TaskName) VALUES
(1, 'Service'),
(2, 'Software Upgrade'),
(3, 'Repair'),
(4, 'Safety Inspection'),
(5, 'Other');

INSERT INTO Service (ServiceID, BoatID, TaskID, EngineerID, ContractorID, ServiceDate, ServiceTime, ManHours) VALUES
(1, 1, 1, 1, NULL, '2023-10-01', '09:00:00', 2.5),
(2, 2, 2, 2, 1, '2023-10-02', '10:00:00', 3.0),
(3, 1, 3, 1, 2, '2023-10-03', '11:00:00', 4.0),
(4, 2, 4, 2, NULL, '2023-10-04', '12:00:00', 1.5),
(5, 1, 5, 1, NULL, '2023-10-05', '13:00:00', 2.0);
```

e. Normalization

The tables are already in 3NF (Third Normal Form) as:

The database tables do not have any non-key attributes that are partially dependent on the primary key.

There are no transitive dependencies (i.e., non-key attributes depending on other non-key attributes).

Thus, no further normalization is required. The design ensures minimal redundancy and dependency.