

Tech Store

DB Design

Group No: 2

Student Name	Student Number
Dana Saleh Al-Malki	2006346
Areej Abdullah Suleman	1916786
Anfal Sultan Al-shihri	2005034
Layan Turki Zaafarani	2005070

Contents

PART I: Analysis	6
1 Problem Definition and Data Requirements	6
1.1 Problem Description	6
1.2 Data Requirements	6
1.3 Business Rules	7
1.4 Intended Output of the System	8
PART II: DB DESIGN	9
2 ER Diagram Design	9
2.1 ER Diagram	9
2.2 Design of Business Rules	10
3 ER-to-logical Schema Mapping	12
3.1 Mapping of Regular Entity Types	12
3.2 Mapping of Weak Entity Types	13
3.3 Mapping of Binary 1-1 Relationship Types	14
3.4 Mapping of Binary 1-N Relationship Types	15
3.5 Mapping of Binary M-N Relationship Types	17
3.6 Mapping of Multivalued Attributes	17
3.7 Mapping of N-ary Relationship Types	17
3.9 Schema Diagram	18
4 Normalization	19
4.1 First Normal Form	19
4.2 Second Normal Form	19
4.3 Third Normal Form	22
5 Final DB Schema Diagram	22
PART III: Implementation	24
6 Table Creation Script	24
6.1 <STORE> TABLE	24
6.2 <EMPLOYEE> TABLE	25
6.3 EMPLOYEE SPECIALIZATIONS	26
6.3.1 <TECHNICIAN> TABLE	26

6.3.2 <DELIVERY_PERSON> TABLE	26
6.4 <CUSTOMER> TABLE	27
6.5 <DELIVERY> TABLE	28
6.6 <MEMBERSHIP> TABLE	28
6.7 <SERVICES> TABLE	29
6.8 <DEVICE> TABLE	29
6.9 <PURCHASES> TABLE	30
6.10 <SPARE_PIECES> TABLE	31
6.11 <SUPPLIER> TABLE	32
6.12 <WARRANTY> TABLE	32
6.13 <DEGREE> TABLE	33
6.14 <ORDERS> TABLE	33
7 Constraints Script	35
8 Queries and Transactions	39
8.1 <Total Cost>	39
8.2 <Discount>	40
8.4 <opened store>	41
8.5 <due today>	42
8.6 <Order for delivery person>	43
8.7 <Services>	43
8.7 <Customers>	44
8.6 Update Example	45
APPENDIX	47

PART I: Analysis

1 Problem Definition and Data Requirements

1.1 Problem Description

These days technology has been by our side making our lives easier and saving us time. Through technology, we are able to create a quality database design to satisfy the needs of a tech store that sells electronic devices and equipment and provides different kinds of services online and offline. In this project, we aim to live up to the expectations of our customers/workers and ease their demands.

1.2 Data Requirements

Entities and their attributes:

- 1. Employee:**
 - o Primary key: id.
 - o name.
 - o Date of Birth.
 - o Shift time
 - o Salary
 - o Hours worked
- 2. Technician:** subclass of Employee inheriting all its attributes with addition of:
 - o degrees
- 3. Delivery Person:** another subclass of Employee inheriting all its attributes with addition of:
 - o driver's licence number.
- 4. Customer:**
 - o Primary key: phone number.
 - o Name
 - o Date of birth
 - o Address
- 5. Device:**
 - o Primary key: Device serial number.
 - o Name
 - o Company
 - o Type
 - o Color
 - o Edition

- o Price
- o Storage
- 6. Supplier:**
 - o Primary key: id.
 - o Name
- 7. Membership:**
 - o Primary key: id.
 - o Total points
 - o Expiration date
- 8. Spare piece:**
 - o primary key: id.
 - o Type
 - o Price
 - o Compatibility(Edditi, name, company)
- 9. Store:**
 - o primary key: Store number.
 - o Working hours
 - o Online store pick-up booth
- 10. Delivery:**
 - o primary key: order number.
 - o Estimated date
 - o shipping fee
 - o Package weight
- 11. Warranty:** weak entity
 - o primary key: device serial number and type.
 - o expiration date
 - o Cost
- 12. Service:**
 - o primary key: id
 - o Type (1- replacing the battery, 2- replacing the screen, 3- new device set-up, 4- diagnostics)
 - o Cost
 - o Duration

1.3 Business Rules

- Each employee may have a supervisor, and each employee could be a supervisor.
- Each employee works in a store, and each store must have employees.
- Each store has a manager, each manager manages one store.
- Employee types are Technician and Delivery Person. An employee can be of one type or neither, but not both.
- Only one customer can order a particular device with a particular delivery.

- Each delivery is delivered by one Delivery Person, a Delivery Person can deliver multiple deliveries.
- Each customer can have one membership, each membership must belong to only one customer.
- a customer can purchase one or more devices, a device could be purchased by only one customer.
- Each device could include 0 or more warranties, each warranty is included in one device only.
- Each supplier can supply multiple devices, each device of a certain company is supplied by one supplier.
- Each supplier can supply multiple spare pieces, each spare piece is supplied by one supplier.
- A technician can perform multiple services, a service is performed by one technician.
- Each customer can request multiple services, each service is requested by one customer.

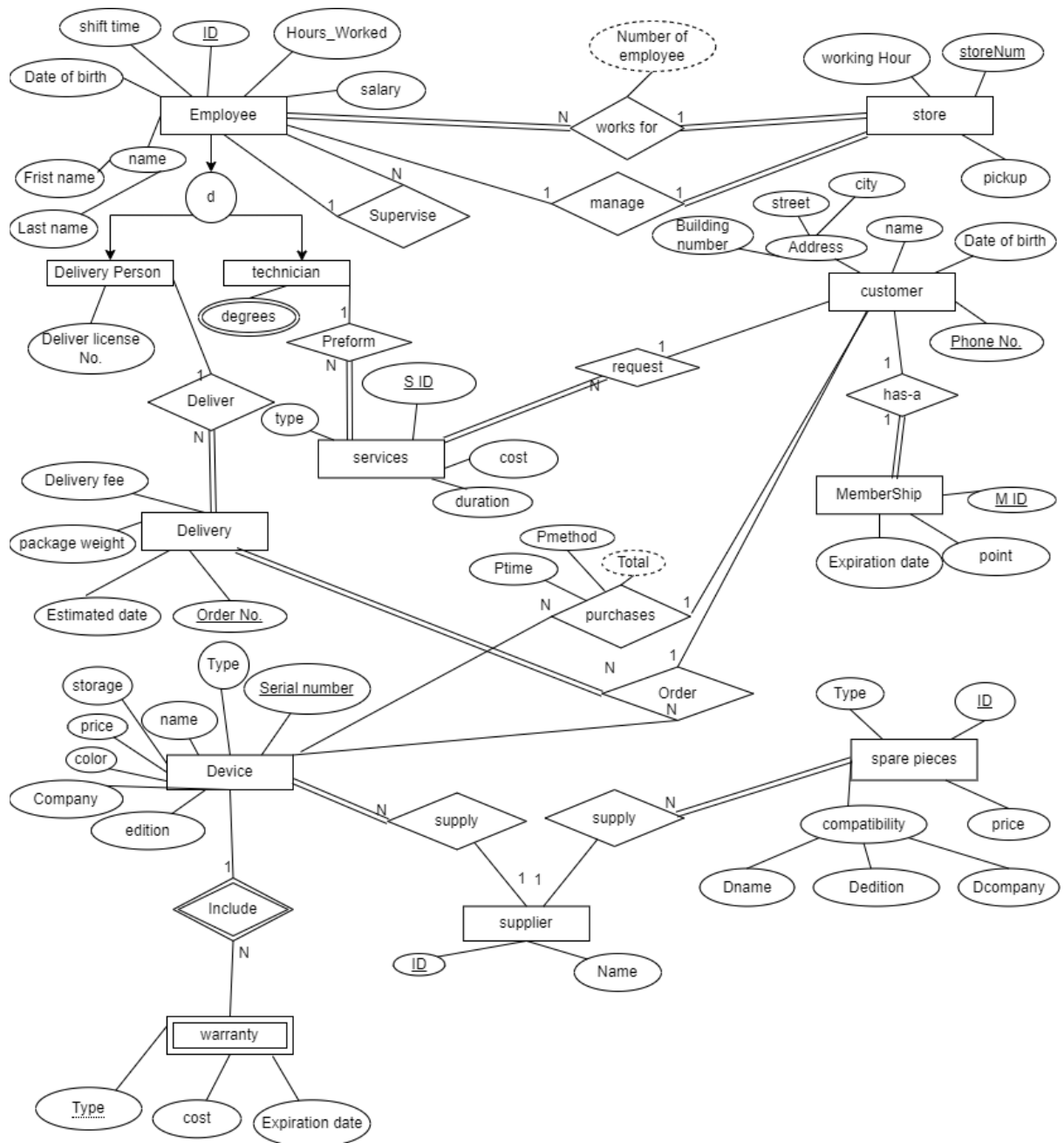
1.4 Intended Output of the System

- Calculate the total cost with the additional services.
- Calculate the discount amount using points from a customer membership.
- Check availability of a device or a spare piece across all stores.
- Renew customers' membership by updating the expiration date and incrementing the version attribute.
- Retrieve all stores that are open at a specific time.
- Retrieve all deliveries due today along with their assigned Delivery Person's employee ID.
- Retrieve a table containing all registered orders' numbers and the customer's phone number and the customer's address for a specific Delivery Person.
- Retrieve all services with pick-up dates that are within a week along with their designated Technician.
- Retrieve all customers who either purchased or ordered a device that includes an active warranty.

PART II: DB DESIGN

2 ER Diagram Design

2.1 ER Diagram



2.2 Design of Business Rules

Business Rule	Design Decisions	Justification (if any)
Each employee may have a supervisor, and each employee could be a supervisor.	Partial participation on both sides of supervise relationship. one-to-many constraint for supervisor and supervisee respectively.	
Each employee work in a store, each store must have employees	Total participation on both sides of work for relationship. many-to-one constraint for Employee and store respectively.	
Each store has a manager, each manager manages one store.	Relationship manage with total participation on the store side and partial participation on the employee side.	
Types of employees are Technician and Delivery Person. an employee can be of one type or neither, but not both.	Subclasses technician, and delivery Person. partial specialization displayed by a single line from superclass to subclasses. disjointness constraint.	The partial specialization is because a technician and a delivery person are two job types that are distinguished by degree and Driver's Licence number respectively, while other employees are stored in the database as regular employees even if they are assigned a particular job in real life.
Only one customer can order a particular device with a particular delivery.	ternary relationship order with one-many-many constraint for customer, device, and delivery, respectively.	
Each delivery is delivered by one Delivery Person, a Delivery Person can deliver multiple deliveries.	Relationship Deliver with total participation on the delivery side and partial participation on the Delivery person side. one-to-many constraint for Delivery person and Delivery respectively.	

Each customer can have one membership, each membership must belong to one customer.	Relationship Has-a with total participation on the membership side and partial participation on the customer side. one-to-one constraint.	The partial participation on the customer side is because although signing up for a membership is free, it is still optional.
A customer can purchase one or more devices, a device could be purchased by only one customer.	Partial participation on both sides of Purchases relationship many-to-one constraint for Device and customer respectively.	
Each device could include 0 or more warranties, each warranty is included in one device only.	Relationship Includes with total participation on the warranty side and partial participation on the Device side. one-to-many constraint for Device and Warranty respectively.	
Each supplier can supply multiple devices, each device is supplied by one supplier.	Relationship Supply with total participation on the Device side and partial participation on the Supplier side. one-to-many constraint for Supplier and Device respectively.	
Each supplier can supply multiple spare pieces, each spare piece is supplied by one supplier.	Relationship Supply with total participation on the Spare_Pieces side and partial participation on the Supplier side. one-to-many constraint for Supplier and Spare_Pieces respectively.	
A technician can perform multiple services, a service must be performed by one technician.	Relationship Perform with total participation on the Service side and partial participation on the Technician side. one-to-many constraint for Technician and Service respectively.	

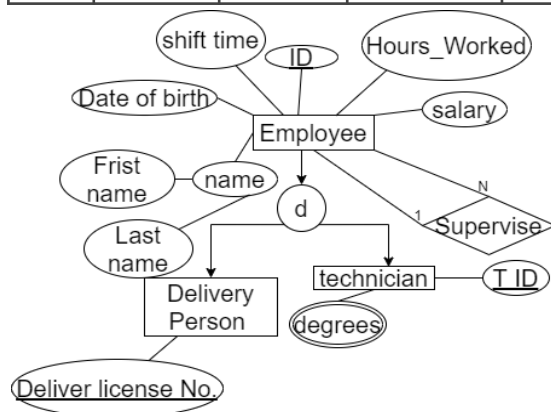
Each customer can request multiple services, each service is requested by one customer.	Relationship Request with total participation on the Service side and partial participation on the Customer side. one-to-many constraint for Customer and Service respectively.	
---	---	--

3 ER-to-logical Schema Mapping

3.1 Mapping of Regular Entity Types

Employee:

<u>ID</u>	Fname	Lname	Date of birth	Hours_worked	Shift time	salary
-----------	-------	-------	---------------	--------------	------------	--------



Technician is a subclass of Employee:

Technician:

<u>EID</u>

FK EID → Employee ID

Delivery person is a subclass of Employee:

Delivery person:

driver_licenseNo	<u>EID</u>
------------------	------------

FK EID → Employee ID

Store:

working_hours	<u>storeNum</u>	pick_up
---------------	-----------------	---------

Customer:

<u>Phone number</u>	name	Date of birth	apt_number	City	street	Building number
---------------------	------	---------------	------------	------	--------	-----------------

Delivery:

<u>orderNo.</u>	estimated_date	delivery_fee	package_weight
-----------------	----------------	--------------	----------------

Membership:

<u>MID</u>	expiration_date	points
------------	-----------------	--------

Services:

<u>SID</u>	type	duration	cost
------------	------	----------	------

Device:

<u>Serial_number</u>	name	type	company	color	price	storage	edition
----------------------	------	------	---------	-------	-------	---------	---------

Spare pieces:

<u>ID</u>	Dname	edition	Dcompany	Price
-----------	-------	---------	----------	-------

Supplier:

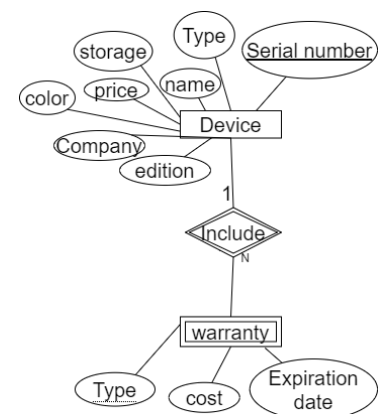
<u>ID</u>	name
-----------	------

3.2 Mapping of Weak Entity Types

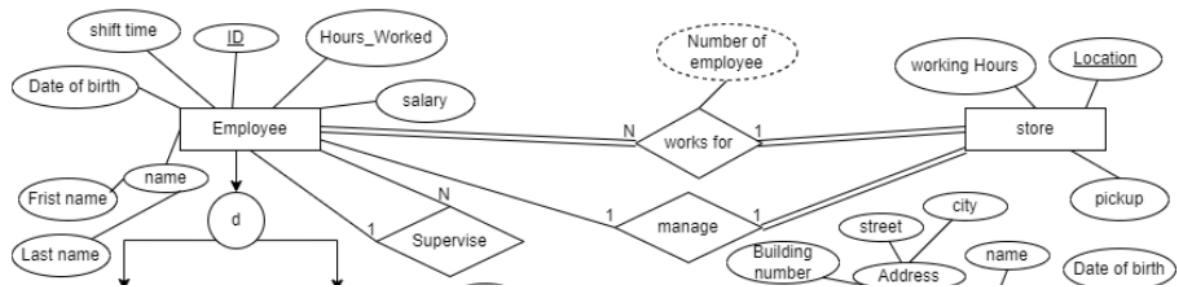
Warranty:

<u>DSerialNo</u>	<u>type</u>	cost	expiration_date
------------------	-------------	------	-----------------

FK DSerialNo. → Device serial_number.



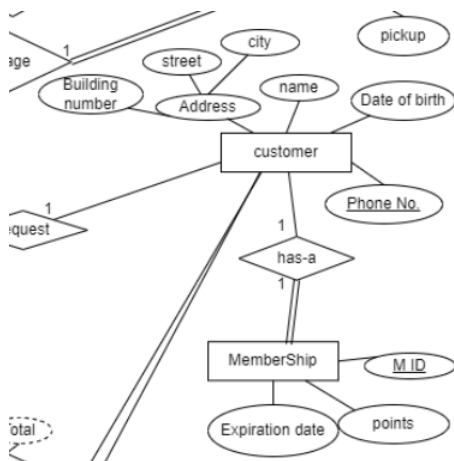
3.3 Mapping of Binary 1-1 Relationship Types



Store:

working_hours	<u>storeNum</u>	mgr_ID	pick_up
---------------	-----------------	--------	---------

FK mgr_ID → Employee ID.

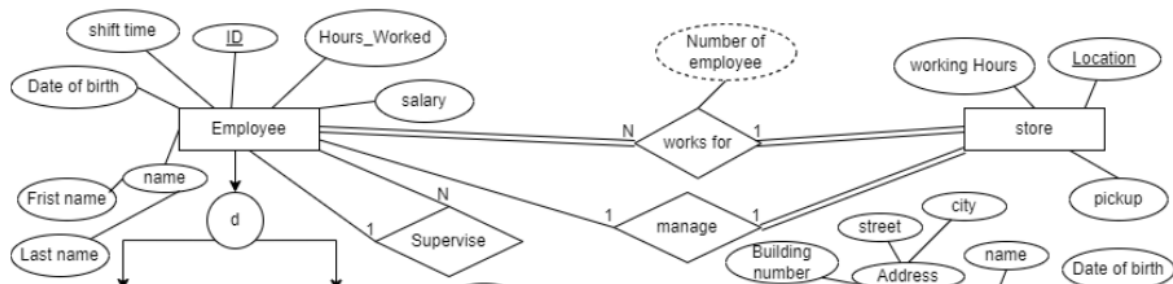


Membership:

CphoneNo	<u>MID</u>	expiration_date	points
----------	------------	-----------------	--------

FK CphoneNo. → Customer phone_number

3.4 Mapping of Binary 1-N Relationship Types



Employee:

<u>ID</u>	Fname	Lname	Date of birth	hours worked	Shift time	salary	store_loc	super_ID
-----------	-------	-------	---------------	--------------	------------	--------	-----------	----------

FK store_loc → Store storeNum.

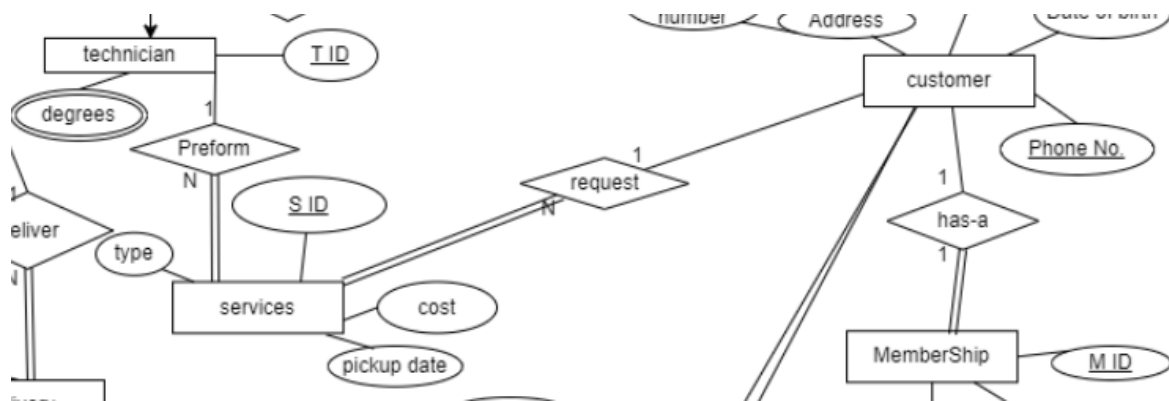
FK super_ID → Employee ID.



Delivery:

<u>orderNo</u>	estimated_date	delivery_fee	package_weight	driver_licenseNo
----------------	----------------	--------------	----------------	------------------

FK driver_licenseNo → Delivery person driver_licenseNo

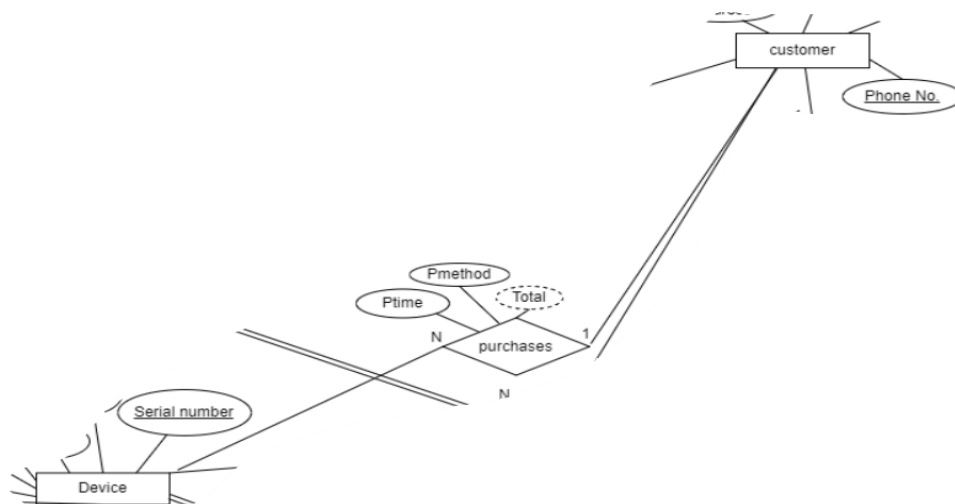


Services:

<u>SID</u>	type	duration	cost	EID	CPhoneNo
------------	------	----------	------	-----	----------

FK EID → Technician EID

FK CPhoneNo → Customer phone_number



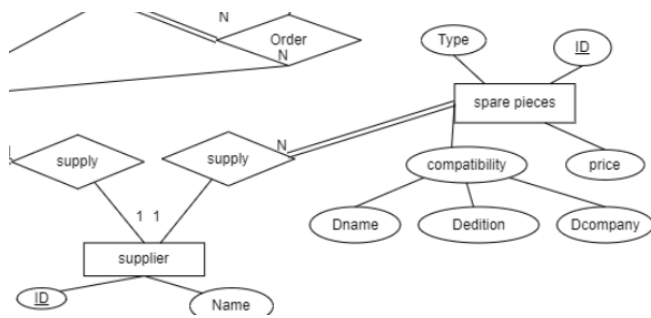
Purchases:

Unlike other mapping in this subsection, for the **purchases** relation we opted to use a Relationship relation (cross referencing relation) to avoid as many null values as possible since the Relationship is partial on both sides.

<u>DSerial number</u>	CPhone	p_method	p_time
-----------------------	--------	----------	--------

FK CPhone → Customer phone_number

FK Dserial_number → Device Serial_number



Spare pieces:

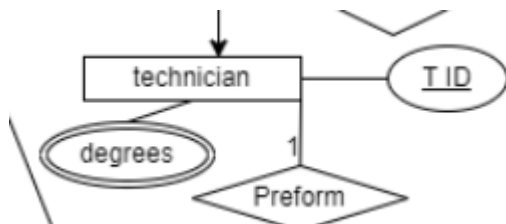
<u>ID</u>	Dname	Dedition	Dcompany	Price	sup_ID
-----------	-------	----------	----------	-------	--------

FK sup_ID → Supplier ID.

3.5 Mapping of Binary M-N Relationship Types

No changes in the schema because there are no M-N relationships.

3.6 Mapping of Multivalued Attributes

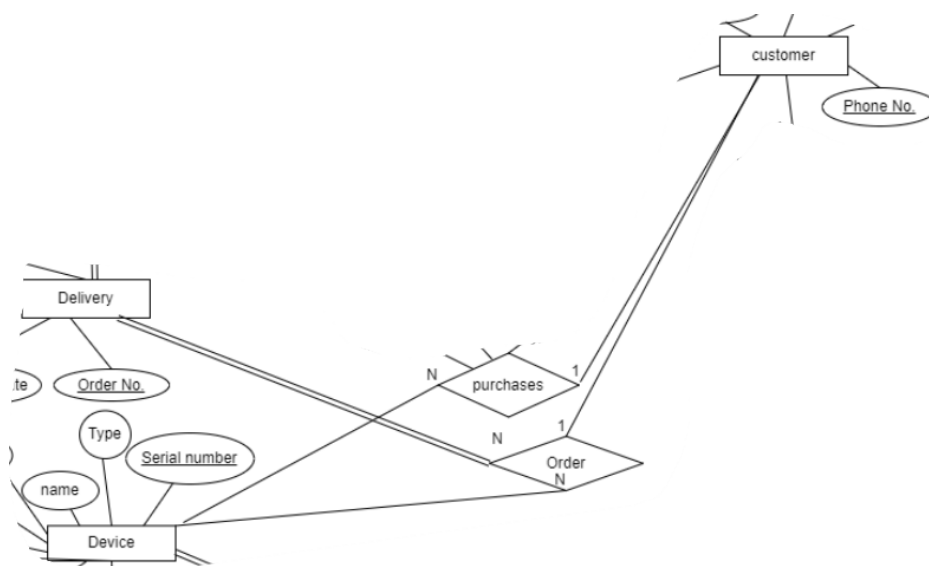


Degrees:

<u>EID</u>	<u>Tdegree</u>
------------	----------------

FK EID → Technician EID.

3.7 Mapping of N-ary Relationship Types



Order:

<u>orderNo</u>	<u>DSerialNo</u>	<u>CPhoneNo</u>
----------------	------------------	-----------------

FK orderNo → Delivery orderNo

FK DSerialNo → Device serial_number

FK CPhoneNo → Customer phone_number

3.9 Schema Diagram

Employee (ID, Fname, Lname, Date_of_birth, Hours_worked, Shift_time, salary, storeNum, super_ID).

FK storeNum → Store storeNum.

FK super_ID → Employee ID.

Technician (EID).

FK EID → Employee ID.

Delivery person (dirver_licenseNo, EID).

FK EID → Employee ID.

Store (working_hours, storeNum, mgr_ID, pick_up).

FK mgr_ID → Employee ID.

Customer (Phone_number, name, Date_of_birth, City, street, Building_number).

Delivery (orderNo., estimated_date, delivery_date, package_weight, EID)

FK EID → Delivery person EID.

Membership (CphoneNo, MID, expiration_date, points).

FK CphoneNo → Customer phone_number

Services (SID, type, duration, cost, EID, CphoneNo).

FK EID → Technician EID.

FK CPhoneNo → Customer phone_number.

Device (Serial_number, name, type, company, color, price, storage, edition, sup_ID).

FK CPhone → Customer phone_number.

FK sup_ID → Supplier ID.

Purchases (DSerial_number, CPhone, p_method, p_time).

FK CPhone → Customer phone_number.

FK Dserial_number → Device Serial_number.

Spare pieces (ID, Dname, Dedition, Dcompany, Price, sup_ID).

FK sup_ID → Supplier ID.

Supplier (ID, name).

Warranty (DSerialNo, type, cost, expiration_date).

FK DSerialNo → Device serial_number.

Degrees (EID, Tdegree).

FK EID → Technician EID.

Order (orderNo, DSerialNo, CPhoneNo).

FK orderNo → Delivery orderNo

FK DSerialNo → Device serial_number

FK CPhoneNo → Customer phone_number

4 Normalization

4.1 First Normal Form

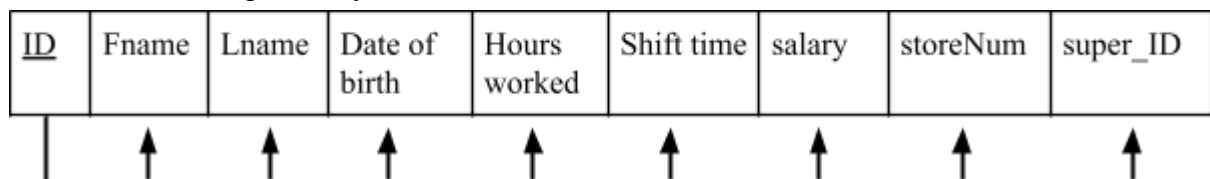
The relation Schema is in 1NF as all attributes' values are atomic, meaning that there are no composite attributes, multivalued attributes, or nested relations.

4.2 Second Normal Form

Disallows partial dependency. Therefore, we will study the Functional Dependencies(FD) of each relation in the schema and determine whether it's a Full FD or a Partial FD.

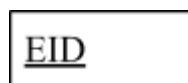
Employee:

Full Functional Dependency



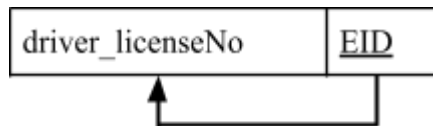
Technician:

Full Functional Dependency



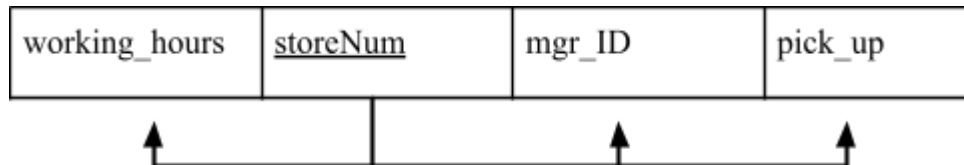
Delivery Person:

Full Functional Dependency



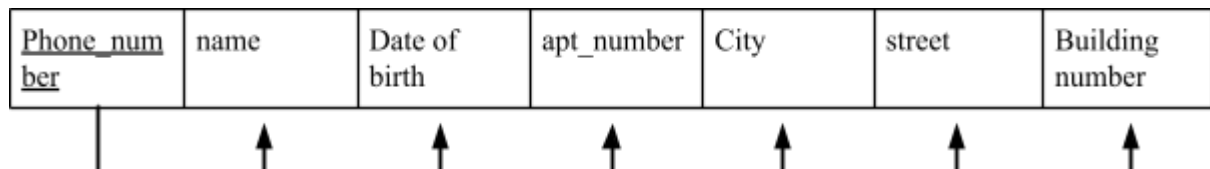
Store:

Full Functional Dependency



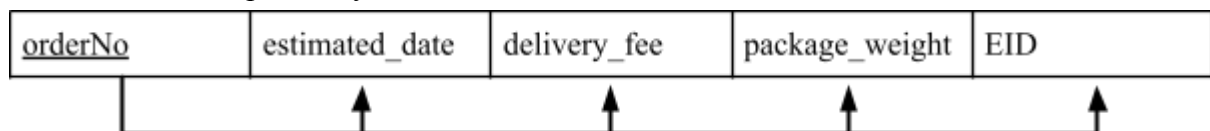
Customer:

Full Functional Dependency



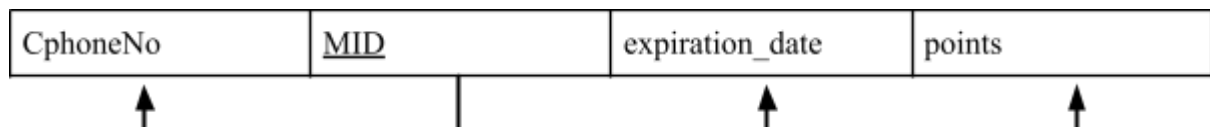
Delivery:

Full Functional Dependency



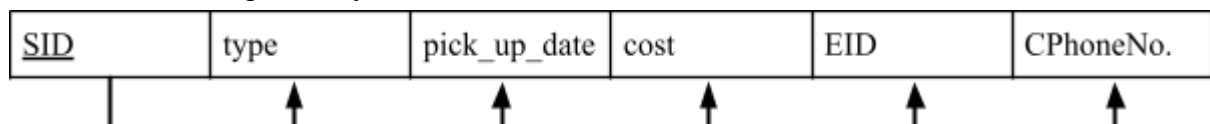
Membership:

Full Functional Dependency



Services:

Full Functional Dependency



Device:

Full Functional Dependency

<u>Serial_number</u>	name	type	company	color	price	storage	edition	sup_ID
----------------------	------	------	---------	-------	-------	---------	---------	--------

Purchases:

Full Functional Dependency

<u>DSerial_number</u>	CPhone	p_method	p_time
-----------------------	--------	----------	--------

Spare pieces:

Full Functional Dependency

<u>ID</u>	Dname	Dedition	Dcompany	Price	sup_ID
-----------	-------	----------	----------	-------	--------

Supplier:

Full Functional Dependency

<u>ID</u>	name
-----------	------

Warranty:

Full Functional Dependency.

The cost depends on both the type of the warranty and the device serial number, because the type is the percentage of the device price that will determine the final cost of the warranty.

Ex: device price: 3500 SR, type: 0.15 → cost = 3500*0.15 = 525 SR.

<u>DSerialNo</u>	<u>type</u>	cost	expiration_date
------------------	-------------	------	-----------------

Degrees:

Full Functional Dependency

<u>EID</u>	<u>Tdegree</u>
------------	----------------

Order:

Full Functional Dependency

<u>orderNo</u>	<u>DSerialNo</u>	<u>CPhoneNo</u>
----------------	------------------	-----------------

The Schema is in Second Normal Form as there are no Partial Dependencies.

4.3 Third Normal Form

Looking at the schema and FDs in section 4.2, we can conclude that the schema is in the 3rd normal form because:

1. All attributes depend on the key only.
2. There is no transitive dependency.

5 Final DB Schema Diagram

Employee (ID, Fname, Lname, Date_of_birth, Hours_worked, Shift_time, salary, storeNum, super_ID).

FK storeNum → Store storeNum

FK super_ID → Employee ID.

Technician (EID).

FK EID → Employee ID.

Delivery person (dirver_licenseNo., EID).

FK EID → Employee ID.

Store (working_hours, storeNum, mgr_ID, pick_up).

FK mgr_ID → Employee ID.

Customer (Phone_number, name, Date_of_birth, City, street, Building_number).

Delivery (orderNo., estimated_date, delivery_fee, package_weight, EID)

FK EID → Delivery person EID.

Membership (CphoneNo., MID, expiration_date, points).

FK CphoneNo. → Customer phone_number

Services (SID, type, duration, cost, EID, CphoneNo.).

FK EID → Technician EID.

FK CPhoneNo. → Customer phone_number.

Device (Serial_number, name, type, company, color, price, storage, edition, sup_ID).

FK CPhone → Customer phone_number.

FK sup_ID → Supplier ID.

Purchases (DSerial_number, CPhone, p_method, p_time).

FK CPhone → Customer phone_number.

FK Dserial_number → Device Serial_number.

Spare pieces (ID, Dname, Dedition, Dcompany, Price, sup_ID).

FK sup_ID → Supplier ID.

Supplier (ID, name).

Warranty (DSerialNo., type, cost, expiration_date).

FK DSerialNo. → Device serial_number.

Degrees (EID, Tdegree).

FK EID → Technician EID.

Order (orderNo., DSerialNo., CPhoneNo.).

FK orderNo → Delivery orderNo.

FK DSerialNo. → Device serial_number.

FK CPhoneNo. → Customer phone_number.

No changes have been made.

PART III: Implementation

6 Table Creation Script

6.1 <STORE> TABLE

```
CREATE TABLE STORE (  
  
storeNum VARCHAR(2) NOT NULL PRIMARY KEY,  
  
working_Hours VARCHAR(50),  
  
mgr_ID NUMBER(10) UNIQUE,  
  
Pick_up VARCHAR(3));  
  
CREATE TABLE STORE  
(  
    storeNum VARCHAR(2) NOT NULL PRIMARY KEY,  
    working_Hours VARCHAR(50),  
    mgr_ID Number(10),  
    Pick_up VARCHAR(3)  
)
```

Table created.

6.2 <EMPLOYEE> TABLE

```
CREATE TABLE EMPLOYEE (  
  ID NUMBER(10) NOT NULL PRIMARY KEY,  
  Fname VARCHAR(20) NOT NULL,  
  Lname VARCHAR(20) NOT NULL,  
  Date_of_birth DATE,  
  Hours_worked NUMBER(4,2),  
  Shift_time VARCHAR(50),  
  Salary NUMBER(10,2),  
  storeNum VARCHAR(2) NOT NULL,  
  Super_ID NUMBER(10));
```

```
CREATE TABLE EMPLOYEE (  
  ID NUMBER(10) NOT NULL PRIMARY KEY,  
  Fname VARCHAR(20) NOT NULL,  
  Lname VARCHAR(20) NOT NULL,  
  Date_of_birth DATE,  
  Hours_worked NUMBER(4,2),  
  Shift_time VARCHAR(50),  
  Salary NUMBER(10,2),  
  storeNum VARCHAR(2) NOT NULL,  
  Super_ID NUMBER(10))
```

Table created.

6.3 EMPLOYEE SPECIALIZATIONS

6.3.1 <TECHNICIAN> TABLE

```
CREATE TABLE TECHNICIAN (  
EID NUMBER(10) NOT NULL PRIMARY KEY);
```

```
CREATE TABLE TECHNICIAN (  
EID NUMBER(10) NOT NULL PRIMARY KEY)
```

Table created.

6.3.2 <DELIVERY_PERSON> TABLE

```
CREATE TABLE DELIVERY_PERSON (  
EID NUMBER(10) NOT NULL PRIMARY KEY,  
driver_licenseNo VARCHAR(12) NOT NULL UNIQUE);
```

```
CREATE TABLE DELIVERY_PERSON (  
EID NUMBER(10) NOT NULL PRIMARY KEY,  
driver_licenseNo VARCHAR(12) NOT NULL UNIQUE)
```

Table created.

6.4 <CUSTOMER> TABLE

```
CREATE TABLE CUSTOMER (  
    Phone_number NUMBER(12) NOT NULL PRIMARY KEY,  
    Name VARCHAR(50),  
    Date_of_birth DATE,  
    City VARCHAR(50),  
    Street NUMBER(5),  
    Building_number NUMBER(5));
```

```
CREATE TABLE CUSTOMER (  
    Phone_number NUMBER(12) NOT NULL PRIMARY KEY,  
    Name VARCHAR(50),  
    Date_of_birth DATE,  
    City VARCHAR(50),  
    Street NUMBER(5),  
    Building_number NUMBER(5))
```

Table created.

6.5 <DELIVERY> TABLE

```
CREATE TABLE DELIVERY (  
  
orderNo  NUMBER(10)  NOT NULL PRIMARY KEY,  
  
Estimated_date  DATE,  
  
Delivery_fee  NUMBER(2,1)  CHECK(Delivery_fee >= 15 AND 35 <=  
Delivery_fee),  
  
Package_weight  NUMBER(2,2)  CHECK(Package_weight <= 50),  
  
EID  NUMBER(10) );  
  
CREATE TABLE DELIVERY (  
orderNo  NUMBER(10)  NOT NULL PRIMARY KEY,  
Estimated_date  DATE,  
Delivery_fee  NUMBER(2,1)  CHECK(Delivery_fee >= 15 AND 35 <= Delivery_fee),  
Package_weight  NUMBER(2,2)  CHECK(Package_weight <= 50),  
EID  NUMBER(10) )
```

Table created.

6.6 <MEMBERSHIP> TABLE

```
CREATE TABLE MEMBERSHIP (  
  
MID  NUMBER(10)  NOT NULL PRIMARY KEY,  
  
CphoneNo  NUMBER(12)  UNIQUE,  
  
expiration_date  DATE,  
  
points  NUMBER(5));
```

```
CREATE TABLE MEMBERSHIP
(
    MID NUMBER(10) NOT NULL PRIMARY KEY,
    CphoneNo NUMBER(12),
    expiration_date DATE,
    points NUMBER(5)
)
```

Table created.

6.7 <SERVICES> TABLE

```
CREATE TABLE SERVICES (
    SID NUMBER(10) NOT NULL PRIMARY KEY,
    serviceType VARCHAR(50),
    duration NUMBER(2),
    cost NUMBER(4,2),
    EID NUMBER(10),
    CphoneNo NUMBER(12));
```

```
CREATE TABLE SERVICES
(
    SID NUMBER(10) NOT NULL PRIMARY KEY,
    serviceType VARCHAR(50),
    duration NUMBER(2),
    cost NUMBER(4),
    EID NUMBER(10),
    CphoneNo NUMBER(12)
)
```

Table created.

6.8 <DEVICE> TABLE

```
CREATE TABLE DEVICE (
```

```
Serial_number VARCHAR(20) NOT NULL PRIMARY KEY,  
name VARCHAR(50),  
company VARCHAR(50),  
color VARCHAR(20),  
price NUMBER(10,2),  
storage NUMBER(4),  
Edition NUMBER(10),  
sup_ID NUMBER(10) NOT NULL);
```

```
CREATE TABLE DEVICE (  
Serial_number VARCHAR(20) NOT NULL PRIMARY KEY,  
name VARCHAR(50),  
company VARCHAR(50),  
color VARCHAR(20),  
price NUMBER(10,2),  
storage NUMBER(4),  
Edition NUMBER(10),  
sup_ID NUMBER(10) NOT NULL)
```

Table created.

6.9 <PURCHASES> TABLE

```
CREATE TABLE PURCHASES (  
Dserial_number VARCHAR(12) NOT NULL PRIMARY KEY,  
CPhoneNo NUMBER(12) NOT NULL,  
p_method VARCHAR(20),  
p_time TIMESTAMP);
```

```
CREATE TABLE PURCHASES
(
    Dserial_number VARCHAR(12) NOT NULL PRIMARY KEY,
    CPhoneNo NUMBER(12),
    p_method VARCHAR(20),
    p_time TIMESTAMP
)
```

Table created.

6.10 <SPARE_PIECES> TABLE

```
CREATE TABLE SPARE_PIECES (
    ID NUMBER(10) NOT NULL PRIMARY KEY,
    Dname VARCHAR(50),
    Dedition VARCHAR(4),
    Dcompany VARCHAR(20),
    Price NUMBER(10,2),
    sup_ID NUMBER(10));
```

```
CREATE TABLE SPARE_PIECES (  
ID NUMBER(10) NOT NULL PRIMARY KEY,  
Dname VARCHAR(50),  
Dedition VARCHAR(4),  
Dcompany VARCHAR(20),  
Price NUMBER(10,2),  
sup_ID NUMBER(10) NOT NULL)
```

Table created.

6.11 <SUPPLIER> TABLE

```
CREATE TABLE SUPPLIER (  
  
ID NUMBER(10) NOT NULL PRIMARY KEY,  
  
name VARCHAR(50));
```

```
CREATE TABLE SUPPLIER (  
ID NUMBER(10) NOT NULL PRIMARY KEY,  
name VARCHAR(50) )
```

Table created.

6.12 <WARRANTY> TABLE

```
CREATE TABLE WARRANTY (  
  
Dserial_number varchar(12) NOT NULL PRIMARY KEY,  
  
type NUMBER(2,2) CHECK(type < 1 AND type > 0),  
  
Cost NUMBER(10,2),  
  
expiration_date DATE);
```

```
CREATE TABLE WARRANTY (  
Dserial_number varchar(12) NOT NULL PRIMARY KEY,  
type NUMBER(2,2) CHECK(type < 1 AND type > 0),  
Cost NUMBER(10,2),  
expiration_date DATE)
```

Table created.

6.13 <DEGREE> TABLE

```
CREATE TABLE DEGREE (  
  
EID NUMBER(10) NOT NULL PRIMARY KEY,  
  
Tdegree VARCHAR(50));
```

```
CREATE TABLE DEGREE (  
EID NUMBER(10) NOT NULL PRIMARY KEY,  
Tdegree VARCHAR(50) UNIQUE )
```

Table created.

6.14 <ORDERS> TABLE

```
CREATE TABLE ORDERS (  
  
orderNo NUMBER(10) NOT NULL,  
  
Dserial_number VARCHAR(20) NOT NULL,  
  
CPhoneNo NUMBER(12) NOT NULL,  
  
PRIMARY KEY(orderNo,Dserial_number, CPhoneNo));
```

```
CREATE TABLE ORDERS
(
    orderNo NUMBER(10) NOT NULL,
    Dserial_number VARCHAR(20) NOT NULL,
    CPhoneNo NUMBER(12) NOT NULL ,
    PRIMARY KEY(orderNo,Dserial_number, CPhoneNo)
)
```

Table created.

7 Constraints Script

Business Rule	SQL Script	Table
Each employee may have a supervisor, and each employee could be a supervisor.	<pre>Super_ID NUMBER(10) ALTER TABLE EMPLOYEE ADD CONSTRAINT super_fk FOREIGN KEY(Super_ID) REFERENCES EMPLOYEE (ID) ON DELETE SET NULL;</pre>	EMPLOYEE
Each employee works in a store, and each store must have employees.	<pre>storeNum VARCHAR(2) NOT NULL ALTER TABLE EMPLOYEE ADD CONSTRAINT stNum_fk FOREIGN KEY(storeNum) REFERENCES STORE(storeNum) ON DELETE SET NULL;</pre>	EMPLOYEE
Each store has a manager, each manager manages one store.	<pre>ALTER TABLE STORE ADD CONSTRAINT mng_fk FOREIGN KEY (mgr_ID) REFERENCES EMPLOYEE (ID) ON DELETE SET NULL;</pre>	STORE
Types of employees are Technician and Delivery Person.	<pre>ALTER TABLE TECHNICIAN ADD CONSTRAINT eID_fk FOREIGN KEY (EID) REFERENCES EMPLOYEE (ID) ON DELETE CASCADE; ALTER TABLE DELIVERY_PERSON ADD CONSTRAINT eIdDe1_fk FOREIGN KEY</pre>	TECHNICIAN & DELIVERY_PERSON

	(EID) REFERENCES EMPLOYEE (ID) ON DELETE CASCADE;	
Only one customer can order a particular device with a particular delivery.	PRIMARY KEY(orderNo,Dserial_number, CPhoneNo) ALTER TABLE ORDERS ADD CONSTRAINT oNum_fk FOREIGN KEY (orderNo) REFERENCES DELIVERY (orderNo) ON DELETE CASCADE; ALTER TABLE ORDERS ADD CONSTRAINT orDNum_fk FOREIGN KEY (Dserial_number) REFERENCES DEVICE (Serial_number) ON DELETE CASCADE; ALTER TABLE ORDERS ADD CONSTRAINT orCNum FOREIGN KEY (CphoneNo) REFERENCES CUSTOMER (Phone_number) ON DELETE CASCADE;	ORDERS
Each delivery is delivered by one Delivery Person, a Delivery Person can deliver multiple deliveries.	orderNo NUMBER(10) NOT NULL PRIMARY KEY ALTER TABLE DELIVERY ADD CONSTRAINT eIDDe12_fk FOREIGN KEY(EID) REFERENCES DELIVERY_PERSON(EID) ON DELETE SET NULL;	DELIVERY
Each customer can have one membership, each membership must belong to only one customer.	CphoneNo NUMBER(10) UNIQUE ALTER TABLE MEMBERSHIP ADD CONSTRAINT mem_fk FOREIGN KEY (CphoneNo) REFERENCES CUSTOMER (Phone_number) ON DELETE CASCADE;	MEMBERSHIP

a customer can purchase one or more devices, a device could be purchased by only one customer.	<pre> ALTER TABLE PURCHASES ADD CONSTRAINT pPH_fk FOREIGN KEY (CphoneNo) REFERENCES CUSTOMER (Phone_number) ON DELETE CASCADE; ALTER TABLE PURCHASES ADD CONSTRAINT pDN_fk FOREIGN KEY (Dserial_number) REFERENCES DEVICE (Serial_number) ON DELETE CASCADE; </pre>	PURCHASES
Each device could include 0 or more warranties, each warranty is included in one device only.	<pre> Dserial_number VARCHAR(12) NOT NULL PRIMARY KEY ALTER TABLE WARRANTY ADD CONSTRAINT warr_fk FOREIGN KEY (Dserial_number) REFERENCES DEVICE (Serial_number) ON DELETE CASCADE; </pre>	WARRANTY
Each supplier can supply multiple devices, each device is supplied by one supplier.	<pre> ALTER TABLE DEVICE ADD CONSTRAINT dev_fk FOREIGN KEY (sup_ID) REFERENCES SUPPLIER (ID) ON DELETE SET NULL; </pre>	DEVICE
Each supplier can supply multiple spare pieces, each spare piece is supplied by one supplier.	<pre> ALTER TABLE SPARE_PIECES ADD CONSTRAINT spID_fk FOREIGN KEY (sup_ID) REFERENCES SUPPLIER (ID) ON DELETE SET NULL; </pre>	SPARE_PIECES

A technician can perform multiple services, a service is performed by one technician.	ALTER TABLE SERVICES ADD CONSTRAINT srID_fk FOREIGN KEY (EID) REFERENCES TECHNICIAN (EID) ON DELETE SET NULL;	SERVICES
Each customer can request multiple services, each service is requested by one customer.	ALTER TABLE SERVICES ADD CONSTRAINT srPH_fk FOREIGN KEY (CphoneNo) REFERENCES CUSTOMER (Phone_number) ON DELETE CASCADE;	SERVICES
Each Technician can have multiple degrees	ALTER TABLE DEGREE ADD CONSTRAINT deg_fk FOREIGN KEY (EID) REFERENCES TECHNICIAN (EID) ON DELETE CASCADE;	DEGREE

8 Queries and Transactions

8.1 <Total Cost>

Query in Natural Language (English)

This query will retrieve the total cost for the order with the additional services

SQL Script

```
SELECT
    o.orderNO
```

```

,r.Delivery_fee
,v.Serial_number
,v.price
,v.price+(SELECT s.cost FROM SERVICES as s )
FROM
DELIVERY as r
,DEVICE as v
,ORDERS as o

```

Caption of the First Five Rows of the Output

8.2 <Discount>

Query in Natural Language (English)

This Query calculate discount for the customer based on the membership points

SQL Script

```

SELECT
    d.price
    ,d.price-(SELECT m.points FROM MEMBERSHIP as m)/100
    as PriceAfterDiscount
FROM
    DEVICE as d
WHERE EXISTS
    (SELECT
        o. CphoneNo
        ,c. Phone_number
    FROM
        ORDERS as o
        ,CUSTOMER as c
    WHERE
        o. CphoneNo=c. Phone_number );

```

Caption of the First Five Rows of the Output

8.3 < Availability >

Query in Natural Language (English)

This query will check the availability for a device or a spare piece across all stores.

SQL Script

```
SELECT  *  
FROM    DEVICE  
GROUP BY DEVICE.name;
```

Caption of the First Five Rows of the Output

8.4 <opened store>

Query in Natural Language (English)

This query will retrieve all stores that are open at certain time

SQL Script

```
SELECT working_Hours, storeNum  
FROM STORE
```

Caption of the First Five Rows of the Output

8.5 <due today>

Query in Natural Language (English)

This query will return the deliveries are due today with assigned Delivery Person's employee ID.

SQL Script

```
SELECT  
    d. orderNo  
    ,d.Estimated_date  
    ,d. Delivery_fee  
    ,d.EID,v.price  
    ,p. driver_licenseNo
```

```

        ,e.Fname,e.Lname
FROM
    DELIVERY as d
    , DELIVERY_PERSON as p
    ,EMPLOYEE as e
    ,DEVICE as v
WHERE
    Estimated_date =GETDATE() ;

```

Caption of the First Five Rows of the Output

8.6 <Order for delivery person>

Query in Natural Language (English)

This query retrieve table of all orders for specific Delivery person

SQL Script

```

SELECT
    d.orderNo
    , d.Estimated_date
    , d.Delivery_fee
    , d.Package_weight
FROM
    DELIVERY as d
    ,DELIVERY_PERSON as p
WHERE
    d.EID=p.EID ;

```


Caption of the First Five Rows of the Output

8.7 <Services>

Query in Natural Language (English)

The Services query will retrieve all services with pick-up dates that are within a week along with their designated Technician.

SQL Script

```
SELECT
    s.SID,s.EID
    ,s.duration
    ,t.EID
    ,e.Fname
    ,e.Lname
FROM
    SERVICES as s
    , TECHNICIAN as t
    ,EMPLOYEE as e
WHERE
    s.duration<8 AND
    s.EID = t.EID AND e.ID= t.EID
;
```

Caption of the First Five Rows of the Output

8.7 <Customers>

Query in Natural Language (English)

The customer query will return all customers who either purchased or ordered a device that includes an active warranty .

SQL Script

```
SELECT
    c.Name,
    c.Phone_number,
    o.CPhoneNo,
    o.Dserial_number,
    d.Serial_number,
    w.Dserial_number
FROM
    CUSTOMER as c,
    OREDERS as o,
    DEVICE as d,
    WARRANTY as w
WHERE
    c.Phone_number = o.CPhoneNo
    AND o.Dserial_number = d.Serial_number
    AND d.Serial_number = w.Dserial_number;
```

Caption of the First Five Rows of the Output

8.6 Update Example

Update in Natural Language (English)

Renew customers' membership by updating the expiration date and incrementing the points of the customer.

SQL Script

```
UPDATE
    MEMBERSHIP
SET
    points=points +1 ;

UPDATE
    MEMBERSHIP
SET
    expiration_date= expiration_date+1 ;
```

Caption of the Output

APPENDIX

<STORE> TABLE

SQL Script

```
SELECT  
  *  
FROM  
  STORE;
```

Caption of the Output

STORENUM	WORKING_HOURS	MGR_ID	PICK_UP
1	10 am - 11 pm	1038679572	YES
2	7 am - 10 pm	1094786733	YES
3	10 am - 11 pm	1082436826	NO
4	10 am - 11 pm	1038327567	YES
5	12 pm - 2 am	1077256342	NO
6	10 am - 11 pm	1083673456	NO
7	12 pm - 2 am	1085863673	YES
8	10 am - 11 pm	1028463432	NO
9	7 am - 10 pm	1038247871	NO
10	10 am - 11 pm	1083653226	NO

[Download CSV](#)

10 rows selected.

<EMPLOYEE> TABLE

SQL Script

```
SELECT
*
FROM
EMPLOYEE;
```

Caption of the Output

ID	FNAME	LNAME	DATE_OF_BIRTH	HOURS_WORKED	SHIFT_TIME	SALARY	STORENUM	SUPER_ID
1077256342	Danah	Al-malki	15-OCT-97	40	3 pm - 10 pm	10000	5	-
1078374763	Waad	Al-harthi	13-JUL-00	22	7 pm - 2 am	7000	5	-
1083653226	Nicolas	Romero	17-MAY-89	45	5 pm - 11 pm	10000	10	-
1082436826	Saleh	Al-malki	09-FEB-88	29	10 am - 2 pm	10000	3	-
1094786733	Anfal	Al-shihri	04-DEC-98	23	8 am - 2 pm	10000	2	-
1083647338	Oliver	Barnes	13-AUG-90	17	5 pm - 11 pm	6000	1	-
1085863673	Layan	Zaafarani	04-JUN-00	12	9 pm - 2 am	10000	7	-
1083673456	Salem	Abudawood	19-APR-95	37	10 am - 3 pm	10000	6	-
1038679572	Malak	Al-thbiti	10-APR-00	40	10 am - 4 pm	10000	1	-
1082637445	Rana	Al-selami	17-SEP-01	22	10 am - 4 pm	7000	8	-
1028463432	Keiji	Akaashi	31-DEC-99	41	1 pm - 9 pm	10000	8	-
1074354720	Shatha	Al-jehani	29-JUL-95	20	3 pm - 10 pm	6000	9	-
1038327567	Areej	Suleman	12-MAR-97	19	10 am - 4 pm	10000	4	-
1038247871	Rayan	Al-sharif	24-FEB-96	29	7 am - 12 pm	10000	9	-
1083687428	Faisal	Al-zahrani	11-SEP-99	31	10 am - 4 pm	8000	10	-
1086463421	Dania	Azzhar	13-MAR-97	37	3 pm - 10 pm	9000	3	-
1094782468	Khaled	Al-harbi	01-JAN-01	13	9 pm - 2 am	6000	7	1085863673
1083744545	Nice	Rodrigues	11-SEP-97	14	7 am - 1 pm	6000	5	1077256342
1097244794	Ahmed	Al-qarni	18-JAN-95	24	9 pm - 2 am	7000	5	1077256342
1099484889	Hajar	Arrmaih	11-MAY-97	38	5 pm - 11 pm	8000	4	1038327567

[Download CSV](#)

20 rows selected.

EMPLOYEE SPECIALIZATIONS

<TECHNICIAN> TABLE

SQL Script

SELECT

*

FROM

TECHNICIAN;

Caption of the Output

EID
1038247871
1038327567
1038679572
1077256342
1082637445
1083653226
1083673456
1085863673
1086463421
1094786733

[Download CSV](#)

10 rows selected.

<DELIVERY_PERSON> TABLE

SQL Script

SELECT

*

FROM

DELIVERY;

Caption of the Output

EID	DRIVER_LICENSENO
1078374763	SU1078374763
1082436826	SU1082436826
1094782468	SU1094782468
1083647338	SU1083647338
1083744545	SU1083744545
1097244794	SU1097244794
1099484889	SU1099484889
1028463432	SU1028463432
1074354720	SU1074354720
1083687428	SU1083687428

Download CSV

10 rows selected.

<CUSTOMER> TABLE

SQL Script

SELECT

*

FROM

CUSTOMER;

Caption of the Output

PHONE_NUMBER	NAME	DATE_OF_BIRTH	CITY	STREET	BUILDING_NUMBER
966504131109	Ghaida	16-MAY-00	Jeddah	15	187
966566076744	Jehan	-	Jeddah	18	-
966502178372	Hoor	-	Riyadh	44	423
966563892300	Rawad Walid	-	Makkah	1	216
966544538329	Lama	17-FEB-03	Jeddah	-	-
966502217175	Fahad	18-OCT-96	Makkah	-	-
966535550989	Majed	-	Riyadh	5	231
966542075918	Mohamed	30-NOV-01	Riyadh	-	-
966559811576	Saleha	11-JUL-89	Riyadh	54	0
966541541429	Yazan	-	Makkah	56	124

Download CSV

10 rows selected.

<DELIVERY> TABLE

SQL Script

SELECT

*

FROM

DELIVERY;

Caption of the Output

ORDERNO	ESTIMATED_DATE	DELIVERY_FEE	PACKAGE_WEIGHT	EID
2065645894	01-JUL-22	22	32	1078374763
2074389910	12-DEC-22	15	2	1094782468
2022904789	08-JUN-22	17	19	1083744545
2010978949	18-AUG-22	29	13	1082436826
2008438799	13-JAN-23	25	49	1083647338
2098984738	28-JAN-23	31	41	1083687428
2011839049	28-MAR-23	18	22	1028463432
2078480090	23-AUG-22	29	4	1097244794
2089473976	02-OCT-22	23	18	1099484889
2000188488	12-NOV-22	21	21	1074354720

[Download CSV](#)

10 rows selected.

<MEMBERSHIP> TABLE

SQL Script

SELECT

*

FROM

MEMBERSHIP;

Caption of the Output

MID	CPHONENO	EXPIRATION_DATE	POINTS
3057488384	966502178372	03-JUL-22	50
3078746210	966542075918	05-NOV-22	837
3028487849	966541541429	30-DEC-22	2223
3009847397	966544538329	13-AUG-22	94873
3018737372	966535550989	10-JUL-22	983
3048939870	966566076744	22-JUN-22	10003
3030074839	966502217175	03-FEB-23	4984
3022237832	966559811576	23-JAN-23	87322
3023884986	966563892300	16-AUG-22	3459
3019199584	966504131109	14-SEP-22	3

Download CSV

10 rows selected.

<SERVICES> TABLE

SQL Script

```
SELECT
*
FROM
SERVICES;
```

Caption of the Output

SID	SERVICETYPE	DURATION	COST	EID	CPHONENO
4068958490	Replacing the battery	10	300	1085863673	966559811576
4022928590	Replacing the screen	12	400	1038679572	966504131109
4090094985	New device set-up	7	75	1077256342	966541541429
4097883920	Diagnostics	14	100	1038327567	966566076744
4019837859	Diagnostics	10	100	1083653226	966542075918
4039489509	New device set-up	8	75	1094786733	966502178372
4009699590	Replacing the screen	9	400	1082637445	966535550989
4085909850	New device set-up	13	75	1077256342	966563892300
4036377838	Replacing the battery	11	300	1085863673	966502217175
4011999194	Replacing the screen	7	400	1038679572	966544538329

Download CSV

10 rows selected.

<DEVICE> TABLE

SQL Script

SELECT

*

FROM

DEVICE;

Caption of the Output

SERIAL_NUMBER	NAME	COMPANY	COLOR	PRICE	STORAGE	EDITION	SUP_ID
JXZZJ9E6JLAD	MPad tablet	Maple	Red	1999	256	7	2947920471
5S7E9WCS4FHE	Neptun laptop	Galaxy	Blue	4999	1024	4	2174479874
XHNBAAKJ34LE	MPhone phone	Maple	White	2999	128	9	3859826359
6SZLPHVB7S4P	FloatPods earpods	Air	Black	999	0	2	7294026368
EZQ7GJTRLJE4	Super PC	Unusual	Black	7999	2048	1	7394728304
YVN5DWDUZA75	See Monitor	Through our eyes	Gray	1299	0	3	7248592047
9ZXPE77KE9SD	Wave Mouse	Sea	White	499	0	10	2952457643
TRAEA37E7AET	StarPhone phone	Galaxy	Yellow	2099	64	3	9302856359
WYLU9M5WN36M	L3 PC	Loyeen	Purple	9999	2048	3	2943425661
UEUF35CTK3S5	Realization Keyboard	Through our eyes	White	599	0	5	6543893266

[Download CSV](#)

10 rows selected.

<PURCHASES> TABLE

SQL Script

SELECT

*

FROM

PURCHASES;

Caption of the Output

DSERIAL_NUMBER	CPHONENO	P_METHOD	P_TIME
JXZZJ9E6JLAD	966502178372	card	21-FEB-22 02.44.11.000000 PM
YVN5DWDUZA75	966563892300	cash	14-APR-22 09.10.44.000000 AM
5S7E9WCS4FHE	966566076744	card	13-JAN-22 10.23.11.000000 PM
TRAE37E7AET	966559811576	cash	09-MAY-22 08.04.45.000000 PM
XHNBAAKJ34LE	966502217175	card	08-DEC-22 07.04.45.000000 PM
6SZLPHVB7S4P	966544538329	card	17-MAR-22 08.59.59.000000 PM
WYLU9M5WN36M	966504131109	cash	21-OCT-22 07.06.56.000000 AM
EZQ7GJTRLJE4	966541541429	card	03-MAR-22 03.45.04.000000 PM
UEUF35CTK3S5	966535550989	cash	09-MAR-22 06.12.58.000000 PM
9ZXPE77KE9SD	966542075918	card	27-JAN-22 08.44.33.000000 AM

Download CSV

10 rows selected.

<SUPPLIER> TABLE

SQL Script

SELECT

*

FROM

SUPPLIER;

Caption of the Output

ID	NAME
2947920471	Toei Animation
2952457643	Madhouse
9302856359	A-1 Pictures
7248592047	Bones
7294026368	White Fox
2174479874	Wit Studio
3859826359	Kyoto Animation
2943425661	Studio Pierrot
7394728304	P.A Works
6543893266	Egg Firm

Download CSV

10 rows selected.

<WARRANTY> TABLE

SQL Script

SELECT

*

FROM

WARRANTY;

Caption of the Output

DSERIAL_NUMBER	TYPE	COST	EXPIRATION_DATE
JXZZJ9E6JLAD	.15	299.85	18-OCT-23
5S7E9WCS4FHE	.2	999.8	28-APR-33
XHNBAAKJ34LE	.3	899.7	19-FEB-25
6SZLPHVB7S4P	.15	149.85	26-MAR-30
EZQ7GJTRLJE4	.2	1599.8	08-APR-23
YVN5DWDUZA75	.3	389.7	22-JUL-24
9ZXPE77KE9SD	.2	99.8	19-FEB-25
TRAEA37E7AET	.15	74.85	26-MAR-30
WYLU9M5WN36M	.2	1999.8	08-APR-26
UEUF35CTK3S5	.3	179.7	22-JUL-23

[Download CSV](#)

10 rows selected.

<DEGREE> TABLE

SQL Script

SELECT

*

FROM

DEGREE;

Caption of the Output

EID	TDEGREE
1077256342	A.A.S - A.S.E.T - Bachelor of Technology
1083653226	A.A.S
1094786733	A.S.E.T
1085863673	A.A.S
1083673456	A.A.S
1038679572	A.S.E.T - A.A.S
1082637445	A.A.S - Bachelor of Technology
1038327567	Bachelor of Technology
1038247871	A.S.E.T
1086463421	A.A.S

Download CSV

10 rows selected.

<ORDERS> TABLE

SQL Script

SELECT

*

FROM

ORDERS;

Caption of the Output

ORDERNO	DSERIAL_NUMBER	CPHONENO
2000188488	UEUF35CTK3S5	966541541429
2008438799	EZQ7GJTRLJE4	966544538329
2010978949	6SZLPHVB7S4P	966563892300
2011839049	9ZXPE77KE9SD	966535550989
2022904789	XHNBAAKJ34LE	966502178372
2065645894	JXZZJ9E6JLAD	966504131109
2074389910	5S7E9WCS4FHE	966566076744
2078480090	TRAEA37E7AET	966542075918
2089473976	WYLU9M5WN36M	966559811576
2098984738	YVN5DWDUZA75	966502217175

[Download CSV](#)

10 rows selected.