NORMALIZATION

SEMESTER PROJECT

EMAN FATIMA
BACHELOR OF INFORMATION TCHNOLOGY | GOVT GRADUATE COLLEGE FOR BOYS CIVIL LINES , SHEIKHUPPURA

Database Design and Implementation in MySQL

- Entity Relationship Diagram
- Relational schema\table schema
- Normalization
- Implementation in MySQL

PROLEM STATEMENT

Electronic Product Service System:

This system supports sales of electronic products like laptops and smartphones. Products come with basic or extended warranties. When a customer initiates a service request, the system logs the device, issue, technician assigned, repair history, and whether replacement parts were used. It also flags serial numbers of products that frequently fail for quality control. The system integrates manufacturer recall data to auto-notify customers affected.

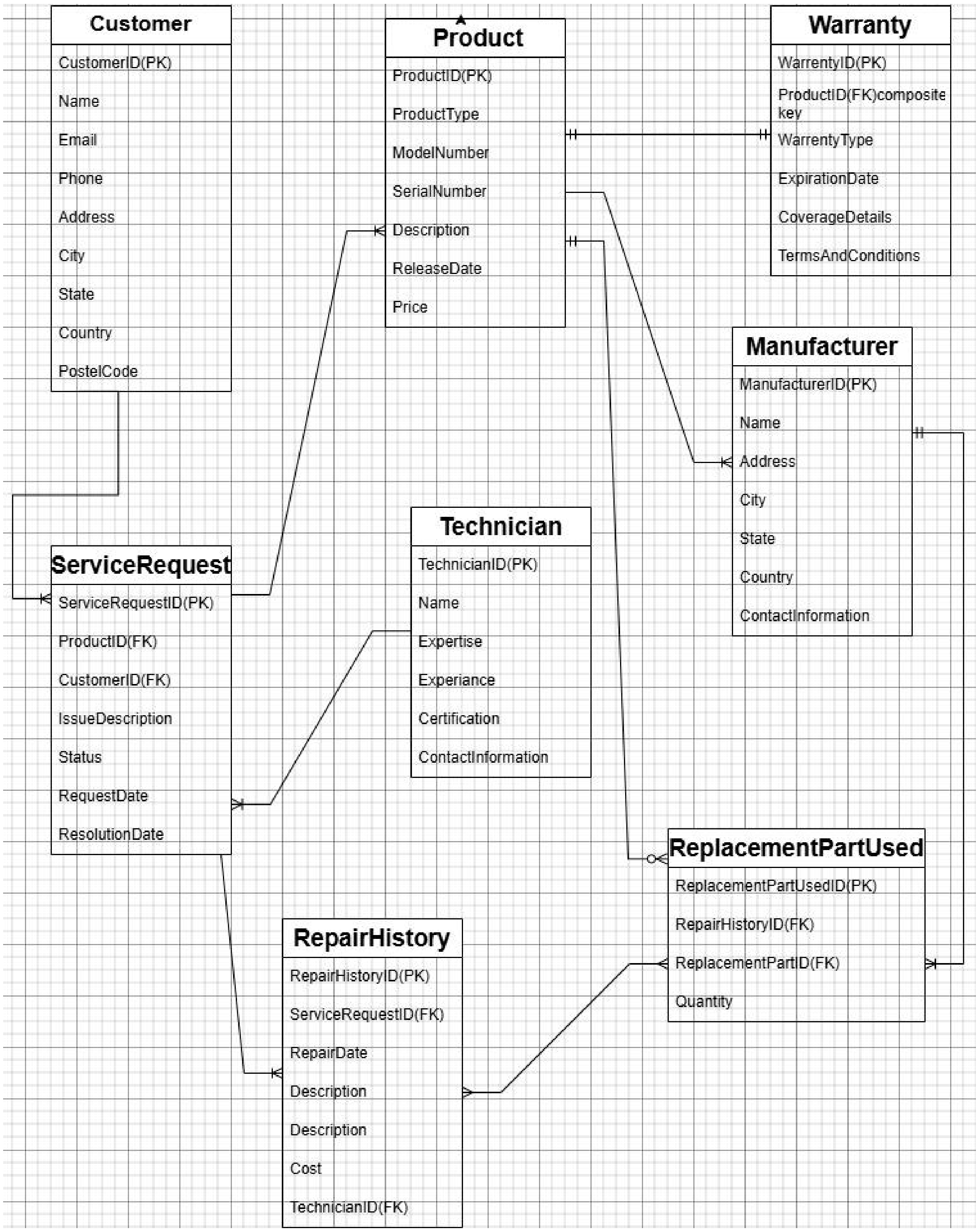
Entity Relationship Diagram

ERD stands for Entity-Relationship Diagram. It's a visual representation of the structure of a database, showing the relationships between entities (tables) and their attributes.

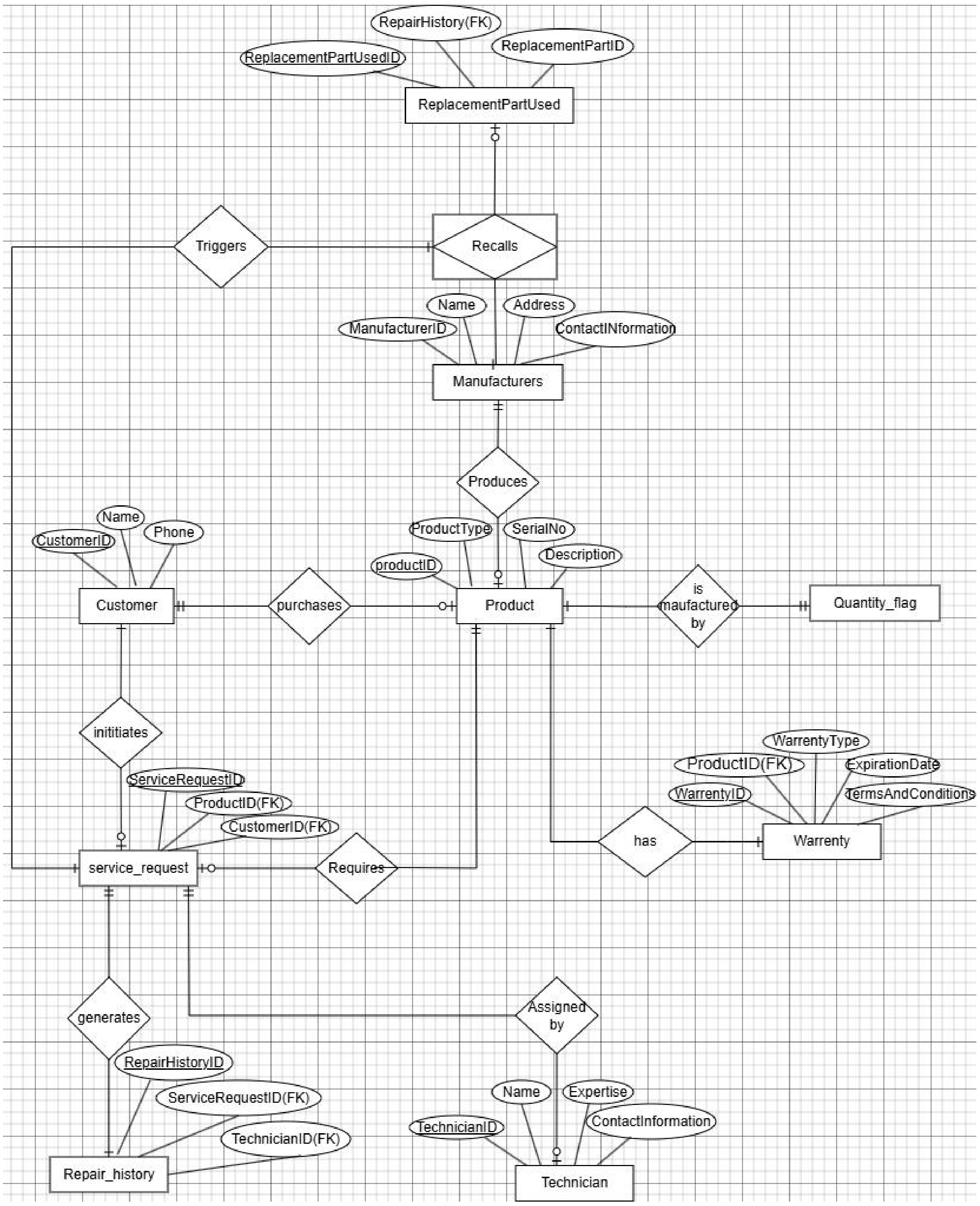
Entity relationship diagram is of 2 type:

- Crow Feet Model
- Chen Model

Crow's Feet Model



Chen Model



RepairHistory(FK)
ReplacementPartID
ReplacementPartUsedID
ReplacementPartUsed
Triggers
Recalls
Name
Address
ManufacturerID
ContactINformation
Manufacturers
Produces
Name
CustomerID
Phone
ProductType
SerialNo
Description
productID
Customer
purchases
Product
is maufactured by
Quantity_flag
inititiates
WarrentyType
ProductID(FK)
ExpirationDate
WarrentyID
TermsAndConditions
ServiceRequestID
ProductID(FK)
CustomerID(FK)
has
Warrenty
service_request
Requires
generates
Assigned by
RepairHistoryID
ServiceRequestID(FK)
TechnicianID(FK)
Name
Expertise
ContactInformation
TechnicianID
Repair_history
Technician

Relational schema\table schema

Here's the relational schema using this notation:

Customer (CustomerID, Name, Email, Phone, Address, PostalCode)

Product (ProductID, ProductType, ModelNumber, SerialNumber, Description, ReleaseDate, Price, ManufacturerID)

Technician (TechnicianID, Name, Expertise, ContactInformation)

Manufacturer (ManufacturerID, Name, Address, ContactInformation)

Warranty (WarrantyID, ProductID, WarrantyType, CoverageDetails)

ServiceRequest (ServiceRequestID,ProductID ,CustomerID , IssueDescription, RequestDate, TechnicianID)

RepairHistory (RepairHistoryID, ServiceRequestID , RepairDate, Description, Cost, TechnicianID)

ReplacementPartUsed (ReplacementPartUsedID ,RepairHistoryID , ReplacementPartID, Quantity)

Note: Assuming a ReplacementPart table exists.

Purchase (PurchaseID,CustomerID , ProductID , PurchaseDate)

Recall (RecallID,ProductID , ServiceRequestID , RecallDate, NotificationSent)

Explanation of the Notation:

- **TableName:** The name of the table (entity).
- **Underlined Attribute(s):** Indicates the primary key of the table. If there are multiple underlined attributes, it's a composite primary key.
- **Double underlined Attribute(s) :**Shows foreign key relationships. The attribute(s) in the current table are foreign keys that reference the primary key(s) in the specified referenced table.
- **Other Attributes:** These are the remaining non-key attributes of the table.

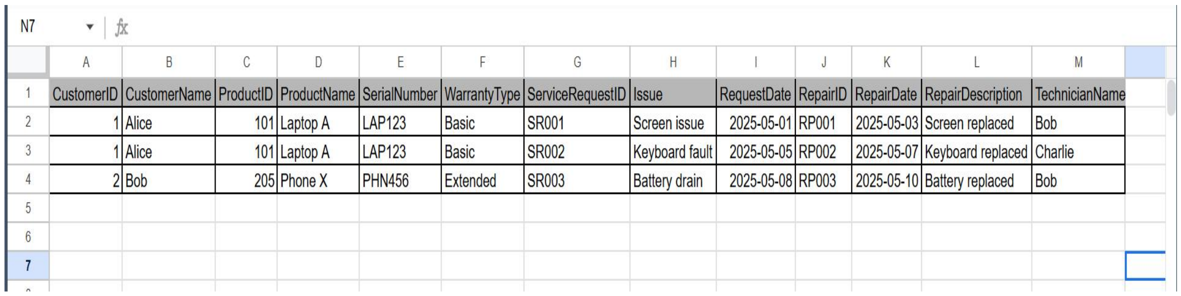
Normalization

Normalization is a process of organizing data in a database to minimize redundancy and improve data integrity. The main objective of database normalization is to eliminate redundant data, minimize data modification errors, and simplify the query process.

- Every table has primary key.
- Other keys depend upon primary key.
- Field must contain atomic values.

Conceptual Unnormalized Table:

(CustomerID, CustomerNameProductID, ProductName, SerialNumber, WarrantyType, ServiceRequestID, Issue, RequestDate, RepairID, RepairDate, RepairDescription,



CustomerID	CustomerName	ProductID	ProductName	SerialNumber	WarrantyType	ServiceRequestID	Issue	RequestDate	RepairID	RepairDate	RepairDescription	TechnicianName
1	Alice	101	Laptop A	LAP123	Basic	SR001	Screen issue	2025-05-01	RP001	2025-05-03	Screen replaced	Bob
1	Alice	101	Laptop A	LAP123	Basic	SR002	Keyboard fault	2025-05-05	RP002	2025-05-07	Keyboard replaced	Charlie
2	Bob	205	Phone X	PHN456	Extended	SR003	Battery drain	2025-05-08	RP003	2025-05-10	Battery replaced	Bob

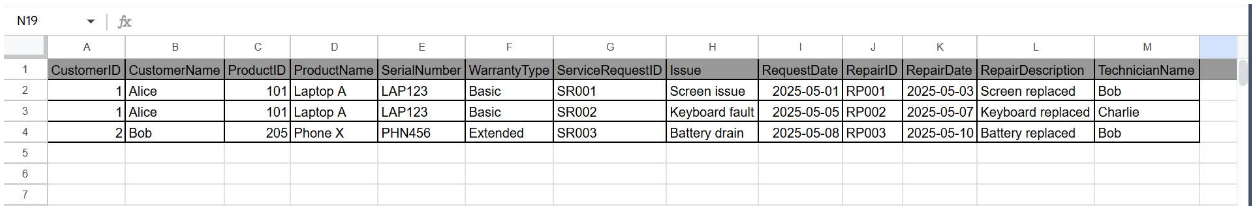
TechnicianName)

1st Normal Form(1NF):

Each table cell contains a single value. No repeating groups or arrays in a single column. 1NF eliminates data redundancy and improves data integrity by organizing data into well-structured tables. In 1NF there must be a primary key.

Each attribute contains only atomic (indivisible) values, and there are no repeating groups of columns.

(CustomerID(PK) , CustomerName, ProductID(PK) , ProductName, SerialNumber, WarrantyType, ServiceRequestID, Issue, RequestDate, RepairID(PK), RepairDate, RepairDescription, TechnicianName)



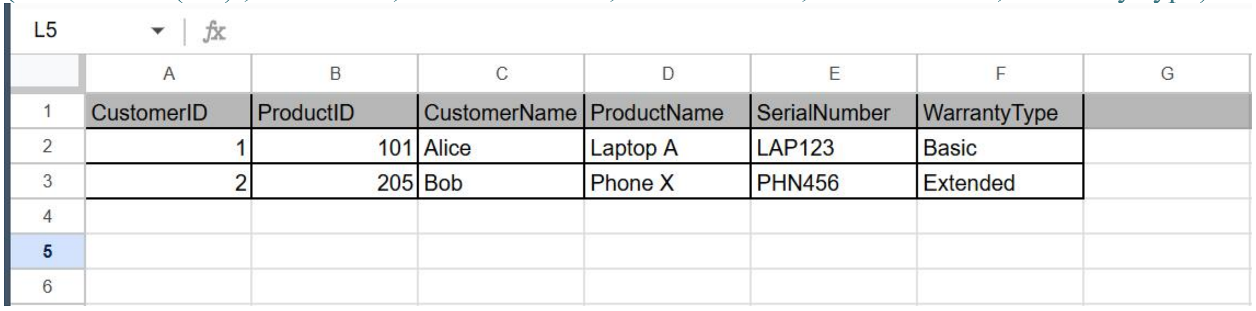
CustomerID	CustomerName	ProductID	ProductName	SerialNumber	WarrantyType	ServiceRequestID	Issue	RequestDate	RepairID	RepairDate	RepairDescription	TechnicianName
1	Alice	101	Laptop A	LAP123	Basic	SR001	Screen issue	2025-05-01	RP001	2025-05-03	Screen replaced	Bob
1	Alice	101	Laptop A	LAP123	Basic	SR002	Keyboard fault	2025-05-05	RP002	2025-05-07	Keyboard replaced	Charlie
2	Bob	205	Phone X	PHN456	Extended	SR003	Battery drain	2025-05-08	RP003	2025-05-10	Battery replaced	Bob

2nd Normal Form(2NF):

Second Normal Form (2NF) is a level of database normalization that builds on First Normal Form (1NF). A table is in 2NF if:
There is no partial dependency of any column on a composite primary key.
All non-key attributes depend on the entire primary key.
All non-key attributes are fully functionally dependent on the entire primary key in above schema.

CustomerProductWarranty Table:

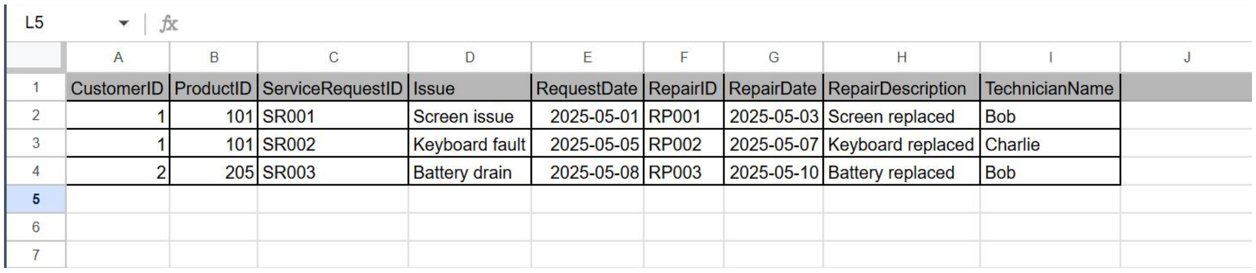
(CustomerID(PK) , ProductID, CustomerName, ProductName, SerialNumber, WarrantyType)



CustomerID	ProductID	CustomerName	ProductName	SerialNumber	WarrantyType
1	101	Alice	Laptop A	LAP123	Basic
2	205	Bob	Phone X	PHN456	Extended

ServiceRequestRepair Table:

(CustomerID(PK) , ProductID, ServiceRequestID, Issue, RequestDate, RepairID, RepairDate, RepairDescription, TechnicianName)



CustomerID	ProductID	ServiceRequestID	Issue	RequestDate	RepairID	RepairDate	RepairDescription	TechnicianName
1	101	SR001	Screen issue	2025-05-01	RP001	2025-05-03	Screen replaced	Bob
1	101	SR002	Keyboard fault	2025-05-05	RP002	2025-05-07	Keyboard replaced	Charlie
2	205	SR003	Battery drain	2025-05-08	RP003	2025-05-10	Battery replaced	Bob

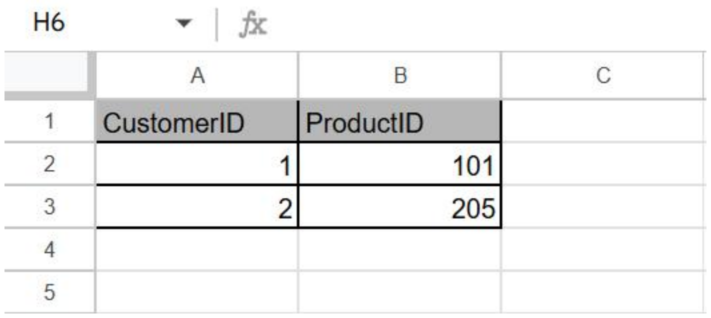
3rd Normal Form(3NF):

Third Normal Form (3NF) in database normalization eliminates transitive dependencies, ensuring that each non-key attribute in a table depends only on the primary key, not on other non-key attributes. It builds upon First Normal Form (1NF) and Second Normal Form (2NF).

3NF eliminates indirect dependencies, ensuring that each non-key attribute depends directly on the primary key.

CustomerProduct Table:

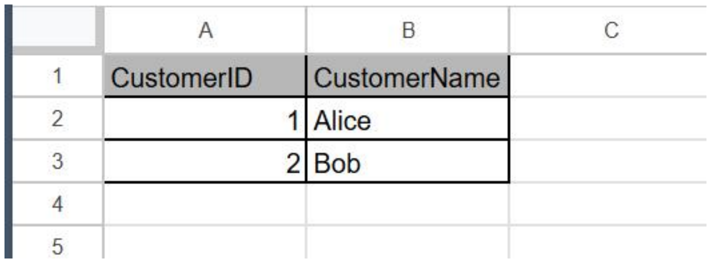
(CustomerID(PK) , ProductID)



CustomerID	ProductID
1	101
2	205

Customer Details Table:

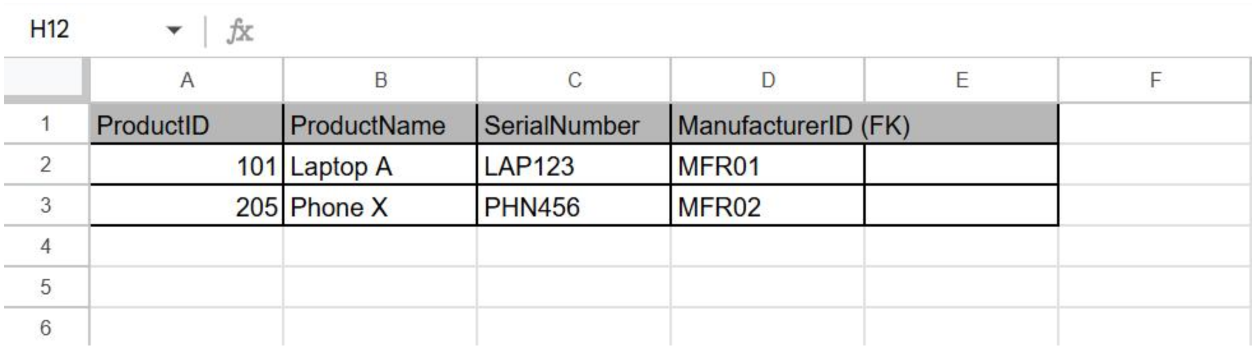
(CustomerID(PK), CustomerName)



CustomerID	CustomerName
1	Alice
2	Bob

Product Details Table:

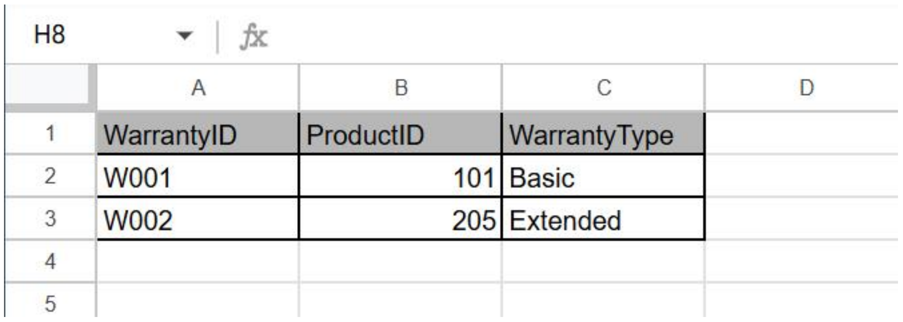
(ProductID(PK), ProductName, SerialNumber, ManufacturerID (FK))



ProductID	ProductName	SerialNumber	ManufacturerID (FK)	
101	Laptop A	LAP123	MFR01	
205	Phone X	PHN456	MFR02	

Warranty Product Table:

(WarrantyID(PK), ProductID, WarrantyType)

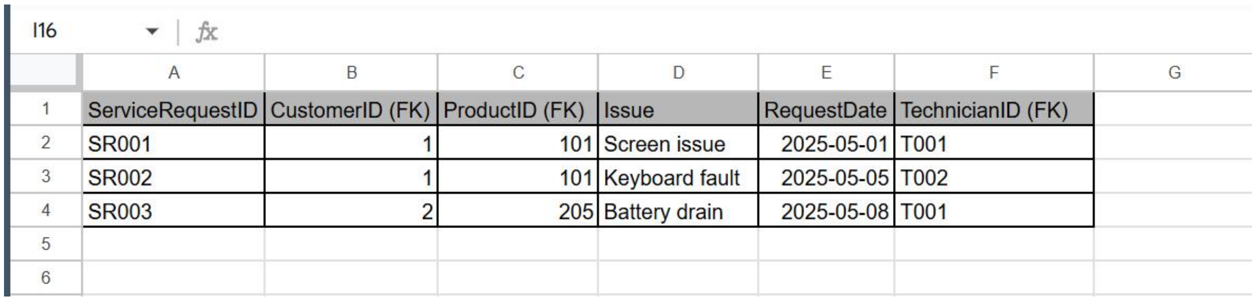


H8

WarrantyID	ProductID	WarrantyType
W001	101	Basic
W002	205	Extended

ServiceRequestIssue Table:

(ServiceRequestID(PK), CustomerID (FK), ProductID (FK), Issue, ,RequestDate TechnicianID (FK))

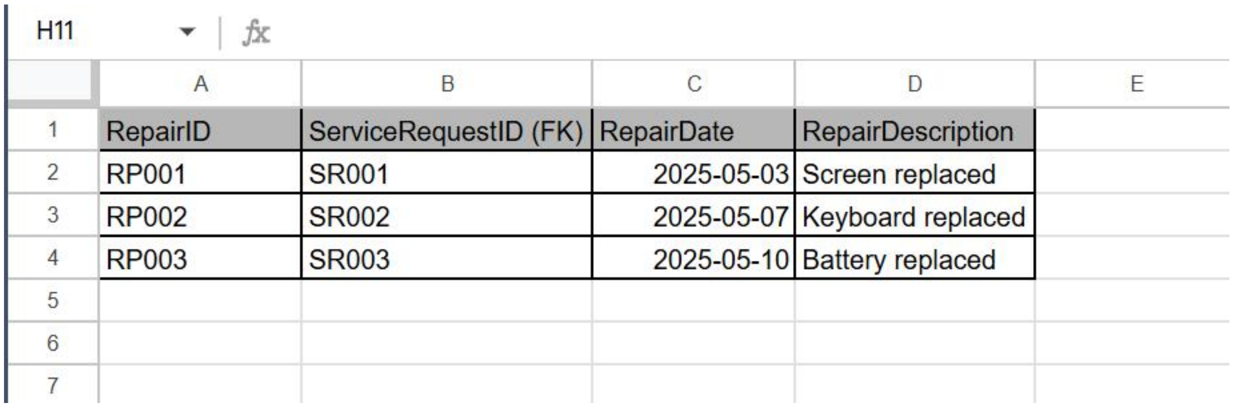


I16

ServiceRequestID	CustomerID (FK)	ProductID (FK)	Issue	RequestDate	TechnicianID (FK)
SR001	1	101	Screen issue	2025-05-01	T001
SR002	1	101	Keyboard fault	2025-05-05	T002
SR003	2	205	Battery drain	2025-05-08	T001

RepairLog Table:

(RepairID(PK), ServiceRequestID (FK), RepairDate, RepairDescription)

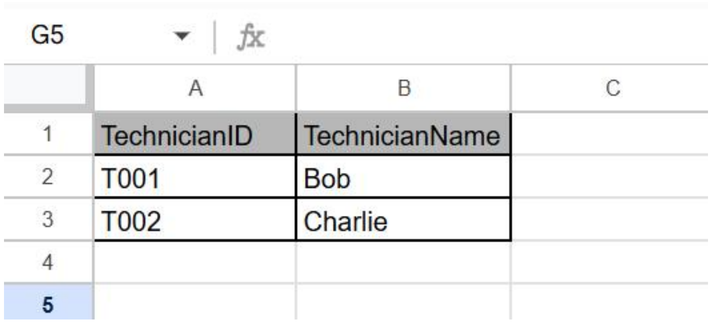


H11

RepairID	ServiceRequestID (FK)	RepairDate	RepairDescription
RP001	SR001	2025-05-03	Screen replaced
RP002	SR002	2025-05-07	Keyboard replaced
RP003	SR003	2025-05-10	Battery replaced

TechnicianName Table:

(TechnicianID(PK), TechnicianName)



	A	B	C
1	TechnicianID	TechnicianName	
2	T001	Bob	
3	T002	Charlie	
4			
5			

COMMON COMMANDS:

1. To see existing databases:

show databases;

2. To Create a database:

CREATE DATABASE database_name;

3. To use the database:

use database_name;

4. To see existing TABLES:

show tables;

5. To Create a tables:

create table table_name(attribute_id_1 type(domain) primary key, attribute_2 type(domain),…,n);

6. To describe table:

describe table_name;

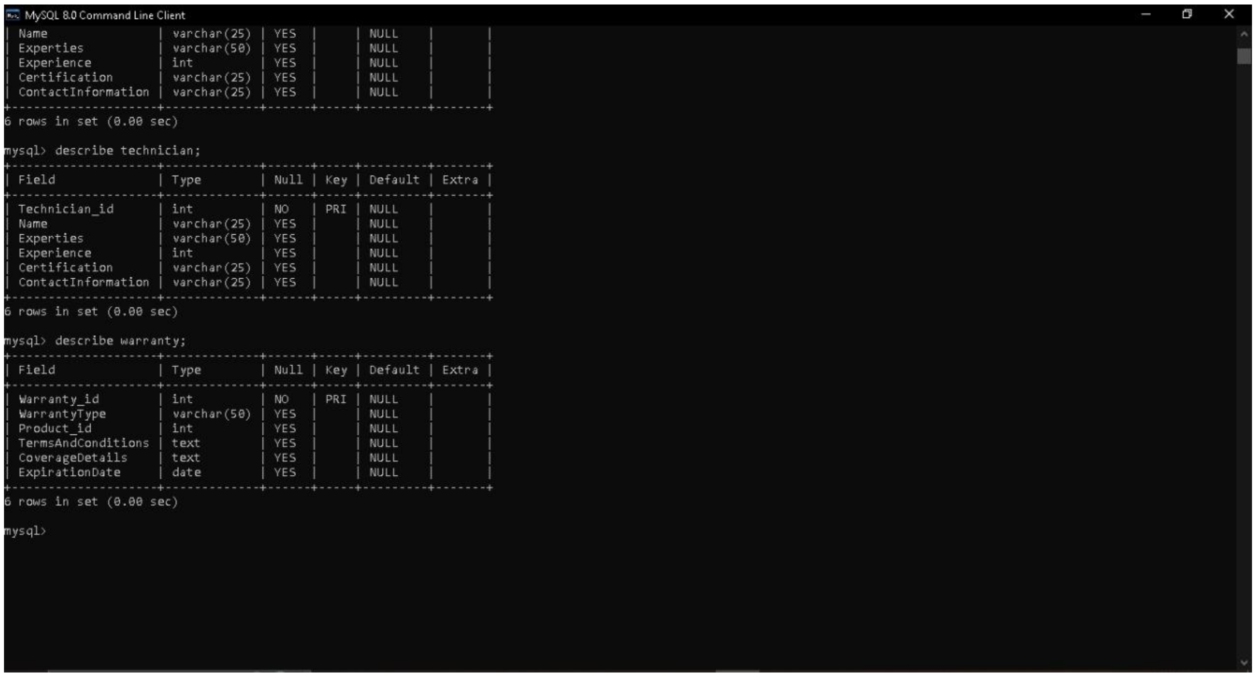
7. To see values from table:

select * from table_name;

8. To add foreign key in existing table:

alter table table_name ADD constraint fk_table_name foreign key (attribute) references ref_table(ref_attribute);

IMPLEMENTATION IN MYSQL

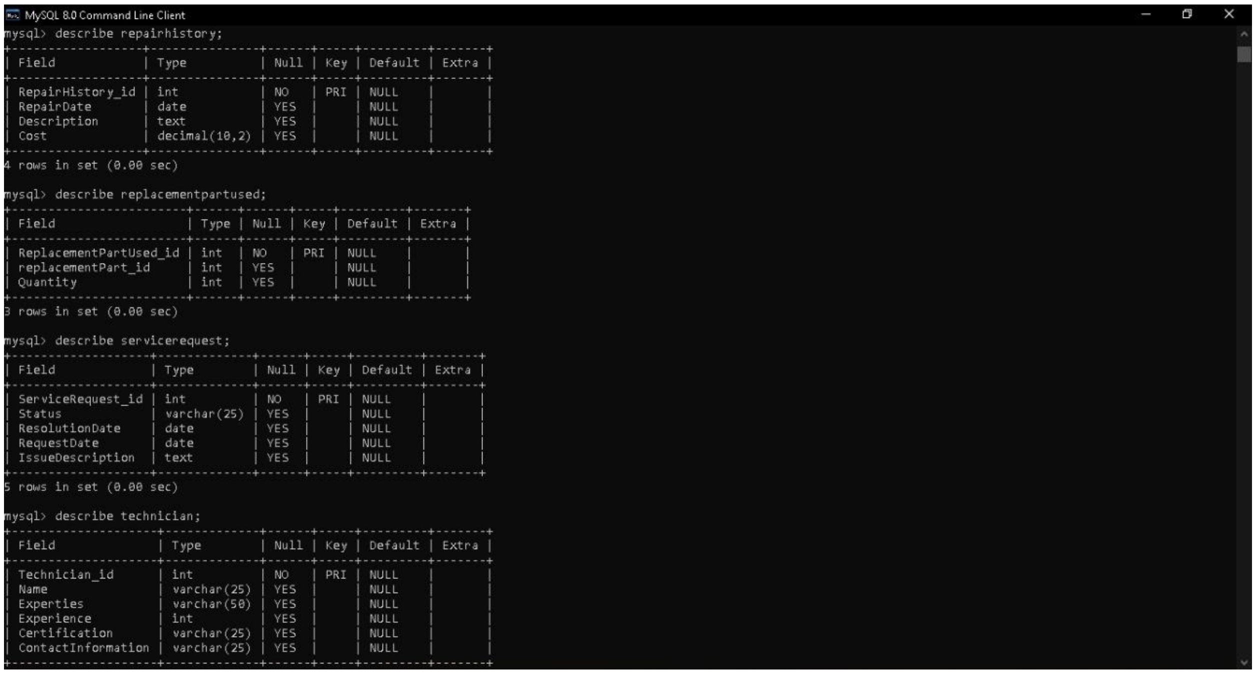


```
| Name              | varchar(25) | YES  |     | NULL    |       |
| Experties         | varchar(50) | YES  |     | NULL    |       |
| Experience        | int         | YES  |     | NULL    |       |
| Certification     | varchar(25) | YES  |     | NULL    |       |
| ContactInformation | varchar(25) | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> describe technician;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| Technician_id     | int         | NO   | PRI | NULL    |       |
| Name              | varchar(25) | YES  |     | NULL    |       |
| Experties         | varchar(50) | YES  |     | NULL    |       |
| Experience        | int         | YES  |     | NULL    |       |
| Certification     | varchar(25) | YES  |     | NULL    |       |
| ContactInformation | varchar(25) | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> describe warranty;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| Warranty_id       | int         | NO   | PRI | NULL    |       |
| WarrantyType      | varchar(50) | YES  |     | NULL    |       |
| Product_id        | int         | YES  |     | NULL    |       |
| TermsAndConditions | text        | YES  |     | NULL    |       |
| CoverageDetails   | text        | YES  |     | NULL    |       |
| ExpirationDate    | date        | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql>
```

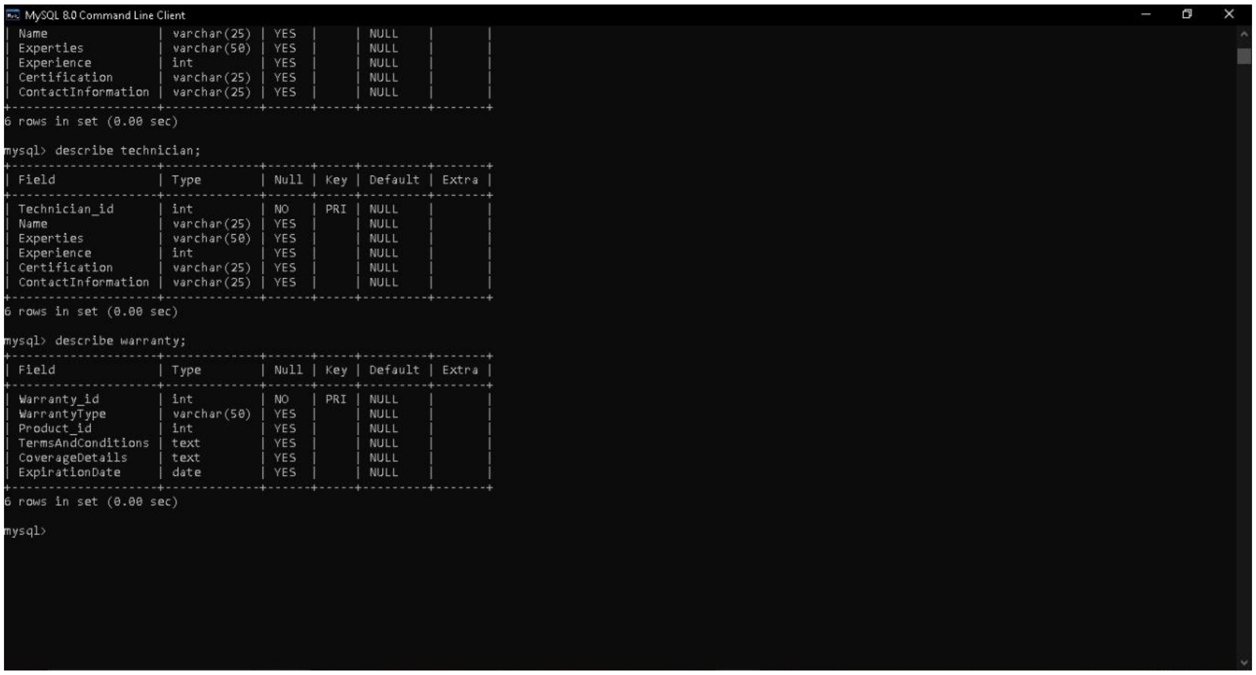


```
mysql> describe repairhistory;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| RepairHistory_id | int           | NO   | PRI | NULL    |       |
| RepairDate       | date          | YES  |     | NULL    |       |
| Description      | text          | YES  |     | NULL    |       |
| Cost             | decimal(10,2) | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> describe replacementpartused;
+------------------------+------+------+-----+---------+-------+
| Field                  | Type | Null | Key | Default | Extra |
+------------------------+------+------+-----+---------+-------+
| ReplacementPartUsed_id | int  | NO   | PRI | NULL    |       |
| replacementPart_id     | int  | YES  |     | NULL    |       |
| Quantity               | int  | YES  |     | NULL    |       |
+------------------------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> describe servicerequest;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| ServiceRequest_id | int         | NO   | PRI | NULL    |       |
| Status            | varchar(25) | YES  |     | NULL    |       |
| ResolutionDate    | date        | YES  |     | NULL    |       |
| RequestDate       | date        | YES  |     | NULL    |       |
| IssueDescription  | text        | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> describe technician;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| Technician_id     | int         | NO   | PRI | NULL    |       |
| Name              | varchar(25) | YES  |     | NULL    |       |
| Experties         | varchar(50) | YES  |     | NULL    |       |
| Experience        | int         | YES  |     | NULL    |       |
| Certification     | varchar(25) | YES  |     | NULL    |       |
| ContactInformation | varchar(25) | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
```



MySQL 8.0 Command Line Client

```
| Name              | varchar(25) | YES  |     | NULL    |       |
| Experties         | varchar(50) | YES  |     | NULL    |       |
| Experience        | int         | YES  |     | NULL    |       |
| Certification     | varchar(25) | YES  |     | NULL    |       |
| ContactInformation | varchar(25) | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> describe technician;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| Technician_id     | int         | NO   | PRI | NULL    |       |
| Name              | varchar(25) | YES  |     | NULL    |       |
| Experties         | varchar(50) | YES  |     | NULL    |       |
| Experience        | int         | YES  |     | NULL    |       |
| Certification     | varchar(25) | YES  |     | NULL    |       |
| ContactInformation | varchar(25) | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> describe warranty;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| Warranty_id       | int         | NO   | PRI | NULL    |       |
| WarrantyType      | varchar(50) | YES  |     | NULL    |       |
| Product_id        | int         | YES  |     | NULL    |       |
| TermsAndConditions | text       | YES  |     | NULL    |       |
| CoverageDetails   | text        | YES  |     | NULL    |       |
| ExpirationDate    | date        | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql>
```
