# Application name: Appointment scheduling application for appliances repair

# Course No: CST2355

# Mengying Liu

# **Description of application**

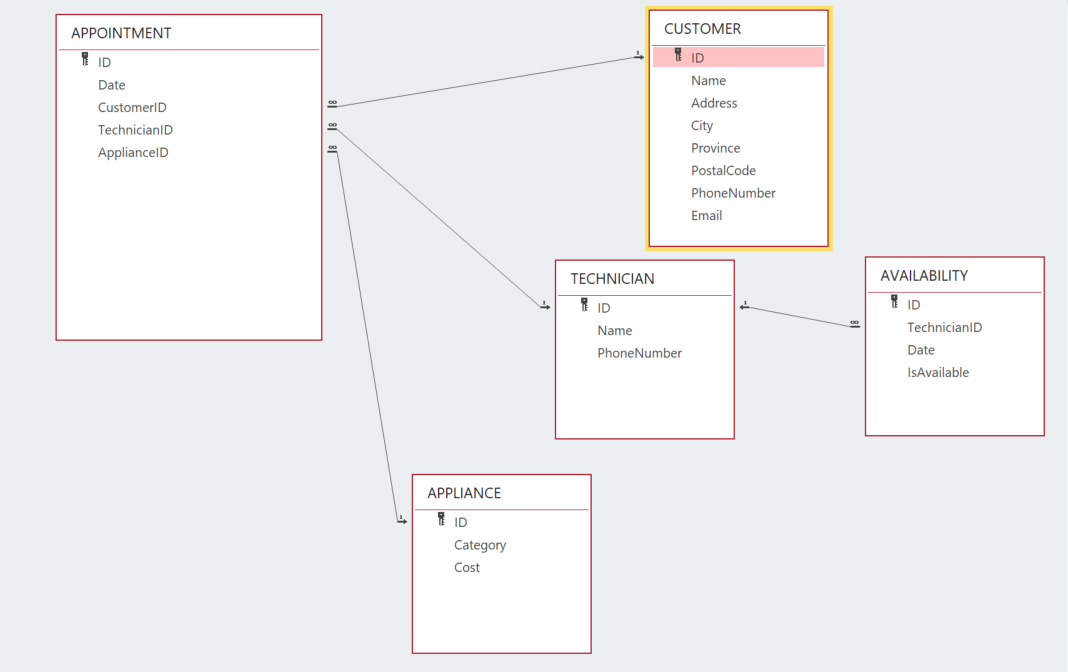
## **The purpose of the application**

### This application is designed to help users manage appointments, as well as customers and appliance catalog.

## **Which apps are being used**

### MS Access and SQL Server

## **Image of Access of Relationship Model**



# **Instructions (For Users)**

## **Section 1 – how to use the front-end application (Access)**

### Purpose: This instruction is written to guide front-end users to navigate and manage the database via MS Access.

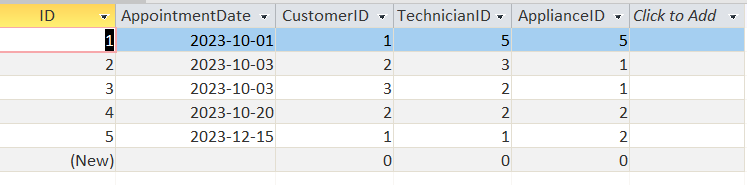
**1. Tables**

There are 5 tables created to record and manage the database:

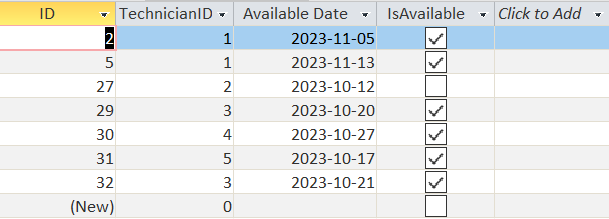
**APPLIANCE**: It is an appliance catalog, which stores all catalog names and repair costs.users are able to view or edit under datasheet view.



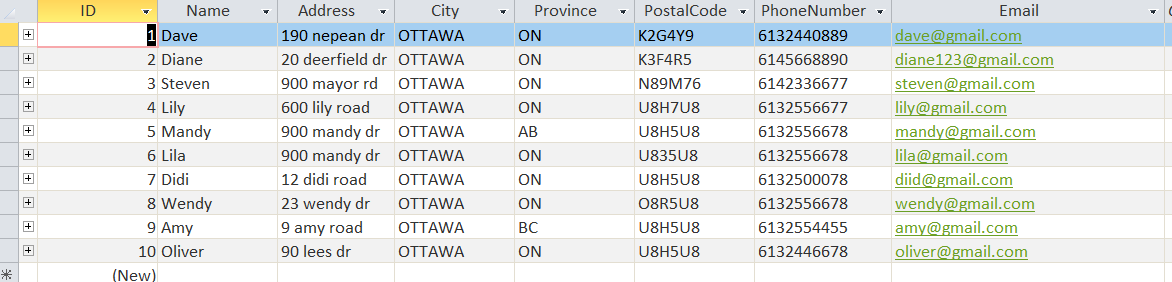
**APPOINTMENT**: This table records appliance repair appointments, including the date and time, customer ID, and technician assigned.



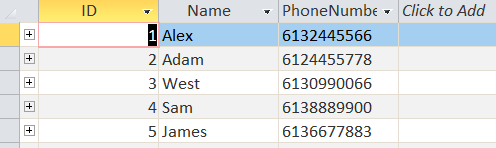
**AVAILABILITY**: This table tracks technician availability on specific dates, using a Boolean field to indicate whether a technician is available for appointments on a given day.



**CUSTOMER:**This table records customer’s name, address, phone and email informations.



**TECHNICIAN:** This table records both name and phone numbers of technicians.



**2. Forms**

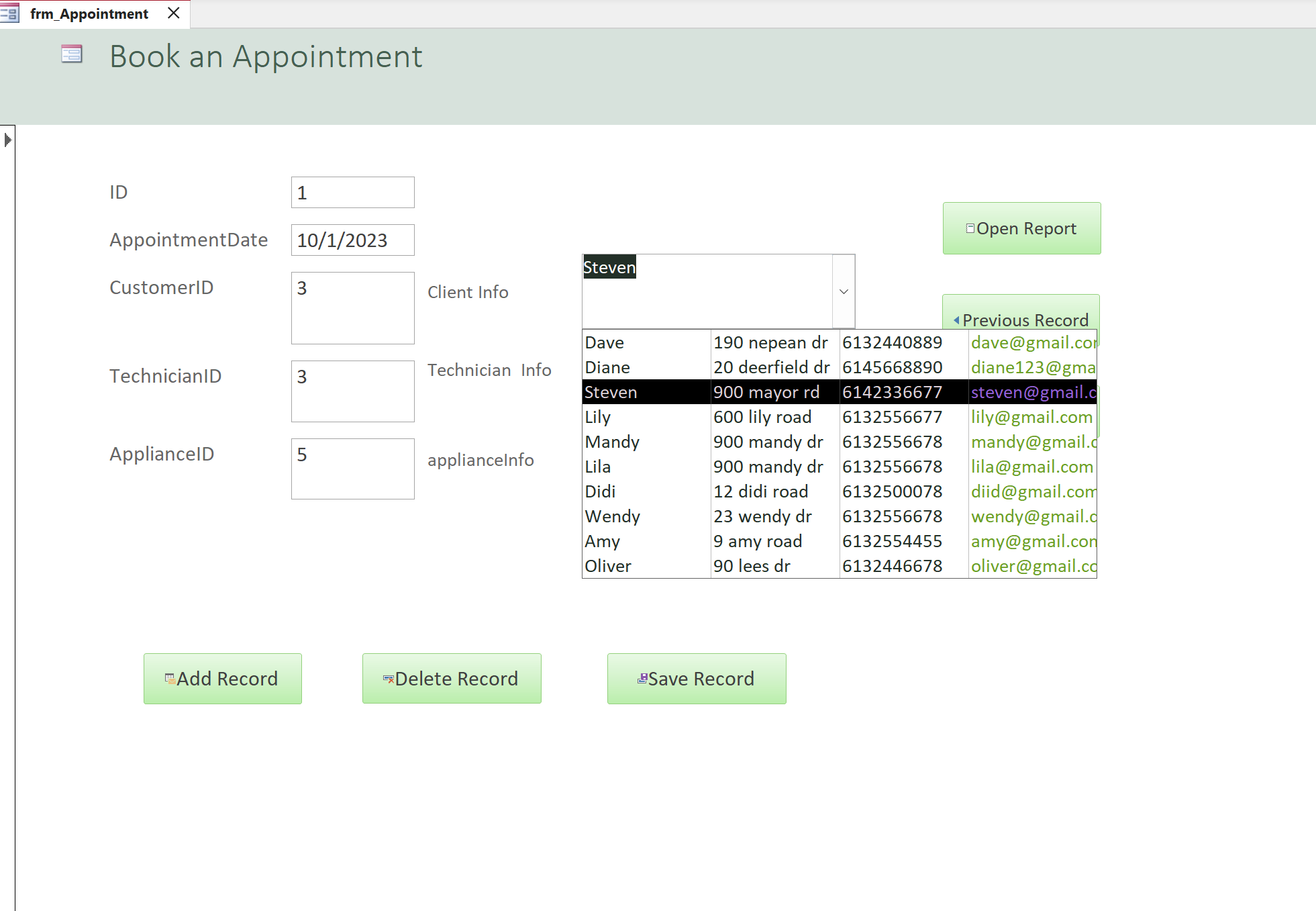
By using forms, users can easily manage corresponding tables, which is adding, updating, or deleting the tables.

**frm\_ApplianceUpdate:** It allows users to navigate each record easily by clicking back and forward arrow buttons. It also helps with adding new records, saving updates and deleting records by clicking delete and save buttons. It also has a shortcut button on the top right to generate a report for the appliance.

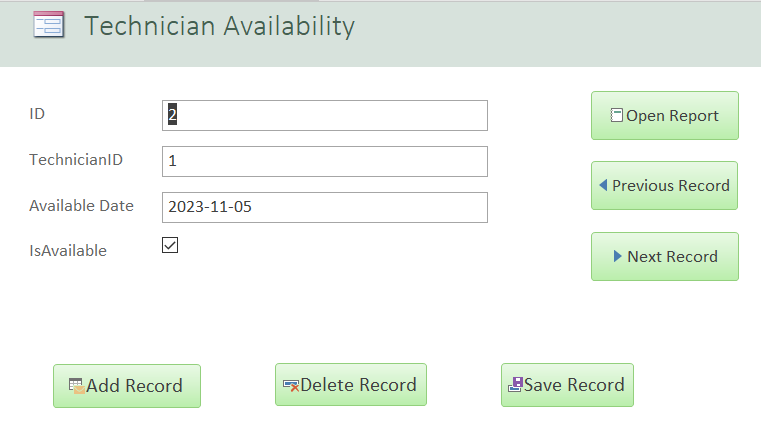
A screenshot of a computer

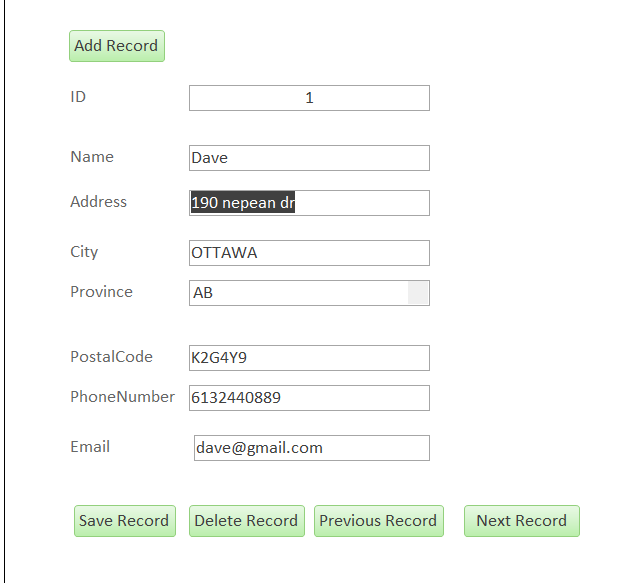
Description automatically generated

**frm\_Appointment:** This form allows users to navigate through and manage appliance repair appointment records by using back and forward arrow buttons. Users can add, delete, and save appointment records, providing an efficient way to update the schedule. Additionally, the form includes a button for generating a report, facilitating the creation of detailed reports on appliance repair appointments.

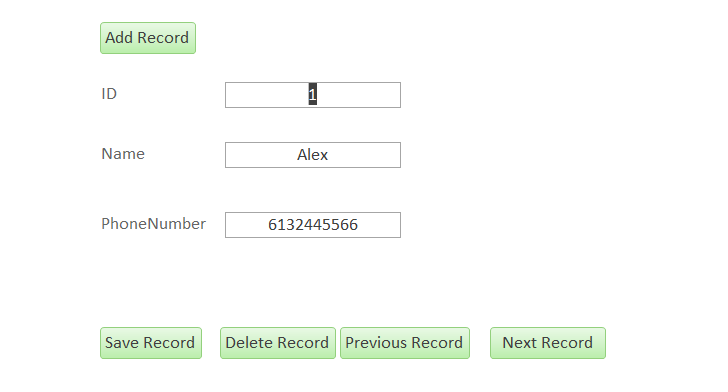
****

**frm\_Availability:** This form provides users with the ability to navigate through technician availability records using back and forward arrow buttons. Users can add, delete, and save technician availability records, ensuring accurate management of their schedules. The form also features a dedicated button for generating availability reports, simplifying the process of tracking and reporting on technician work schedules.

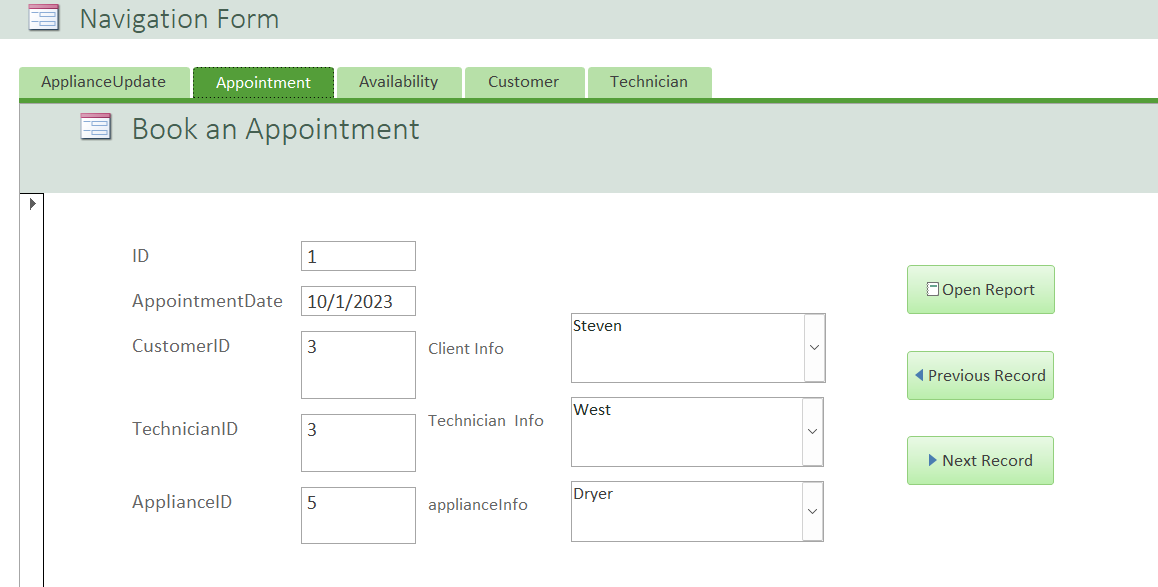
****

**frm\_Customer:** It allows users to navigate each record easily by clicking previous record and next record buttons. It also helps with adding new records, saving, deleting records by clicking delete record and save record buttons.

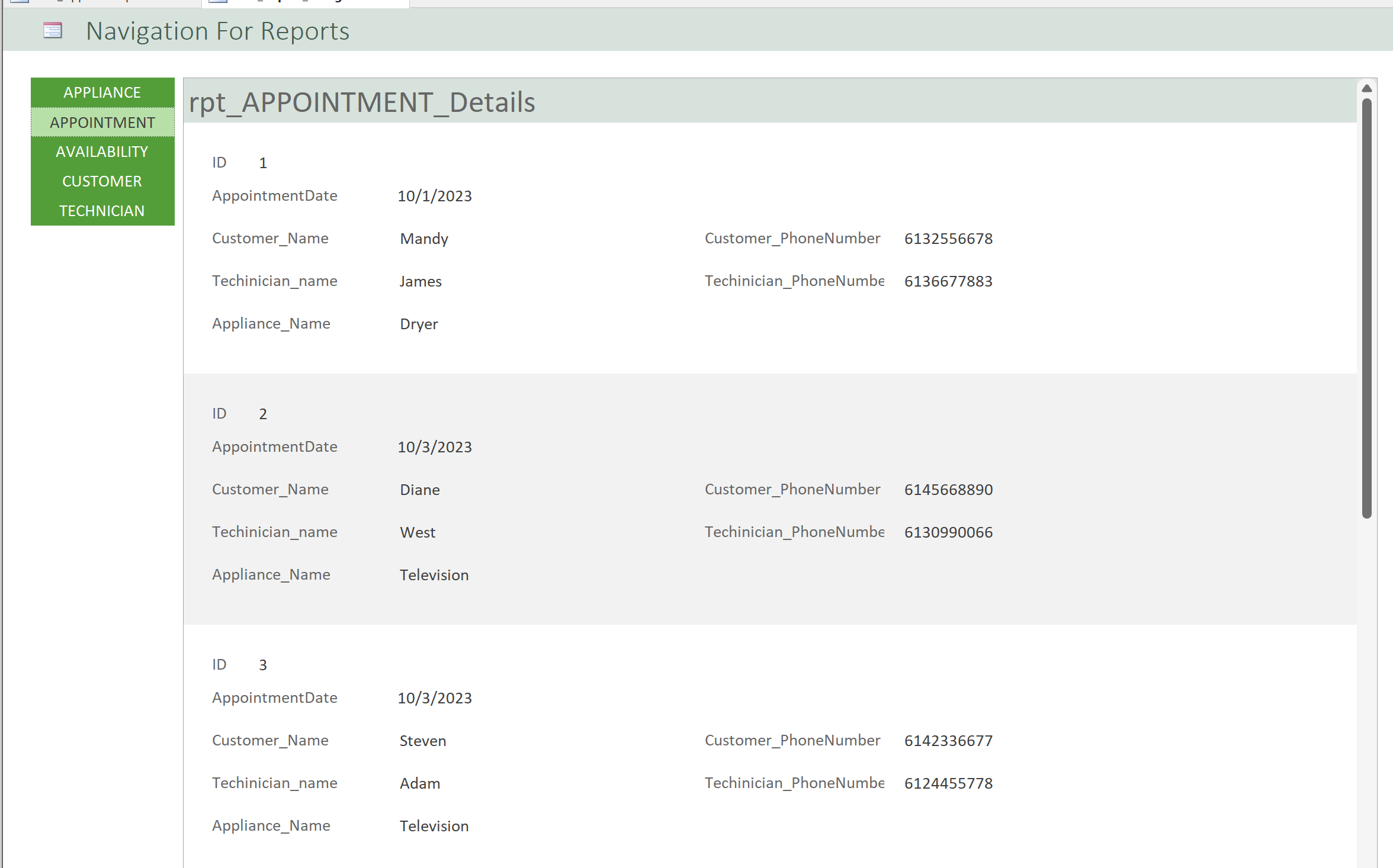
**frm\_Technician:** It allows users to navigate each record easily by clicking previous record and next record buttons. It also helps with adding new records, saving, deleting records by clicking delete record and save record buttons.



**frm\_Navigation:** It allows users to navigate all forms in one place, using the navigation buttons on top, users can easily review what form they are looking to editing.

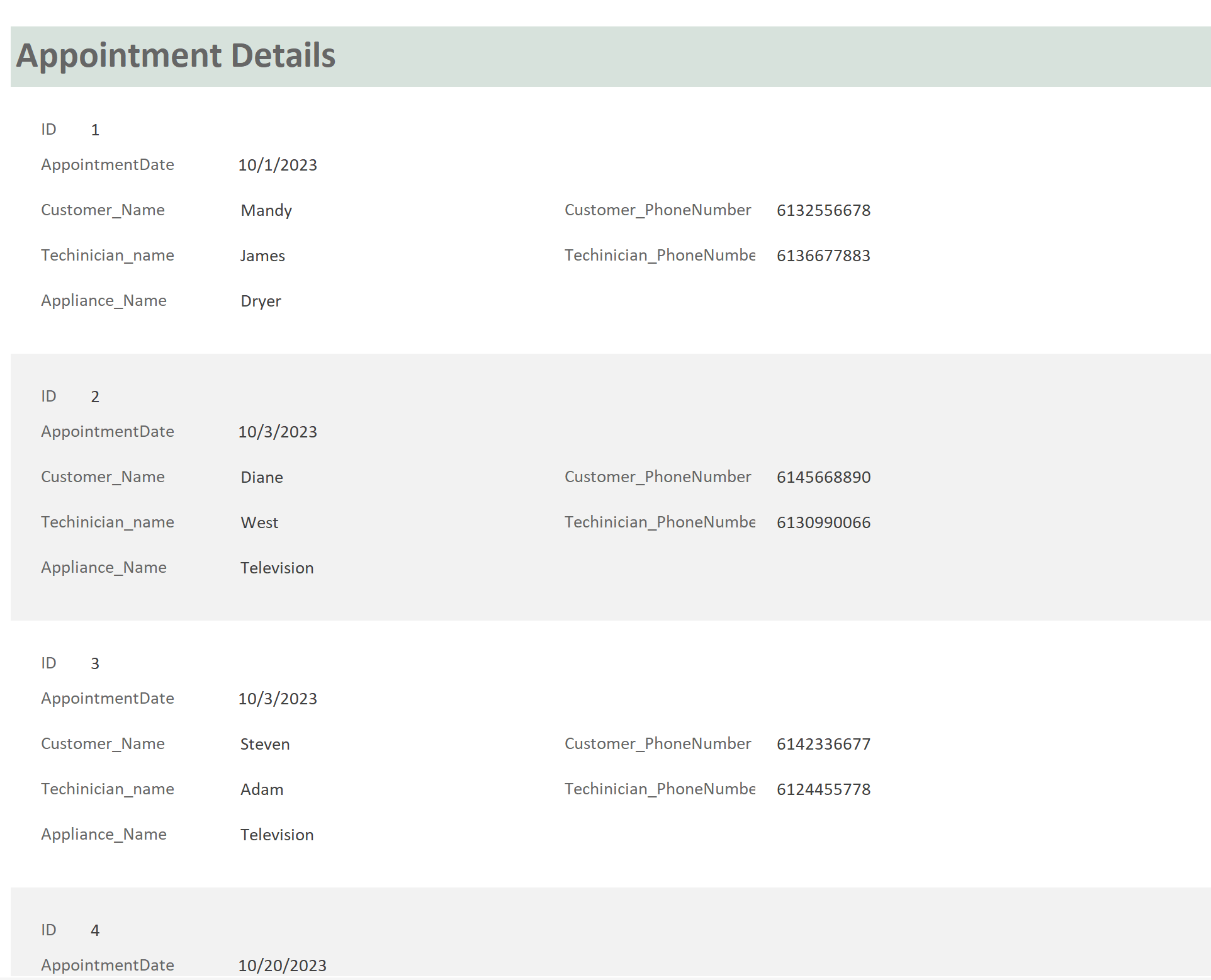


**frm\_Report\_Navigation:** It allows users to navigate all reports in one place, using the navigation buttons on the left side, users can easily review each report.

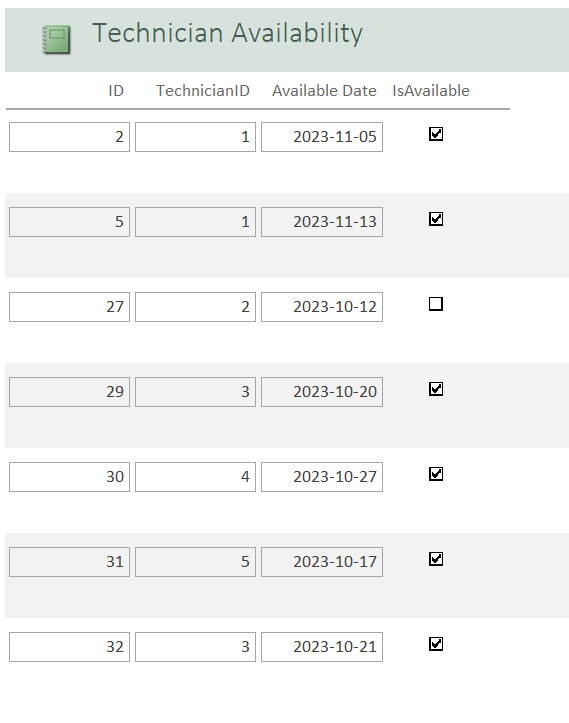
****

**3. Reports**

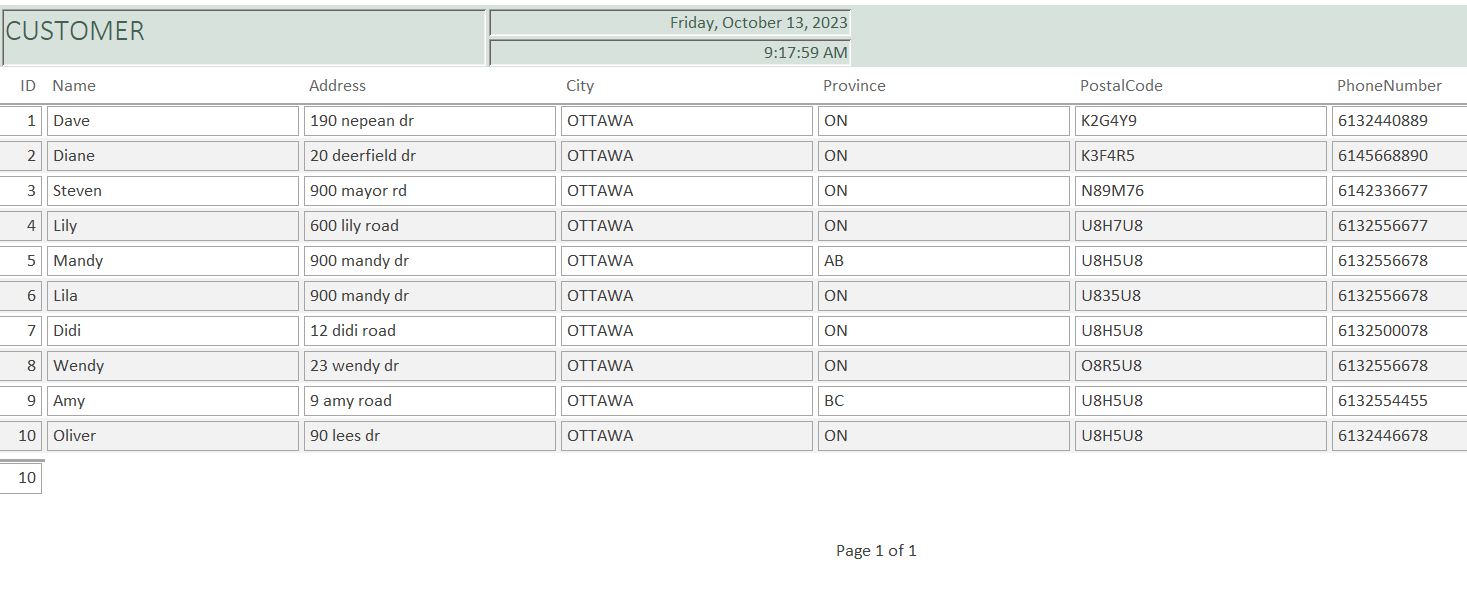
**rpt\_Appointment\_Details:** This report presents a detailed summary of appliance repair appointments, offering valuable insights into scheduled repair tasks, this Report uses underlying query instead of table, which joins 4 tables together to give users a more meaningful report.



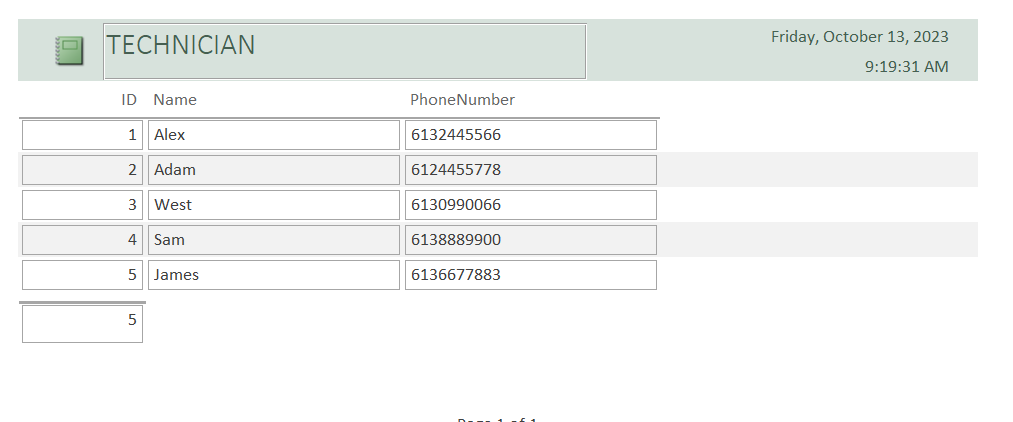
**rpt\_Availability:** This report provides an overview of technician availability, aiding in the efficient management of technician work schedules.



**rpt\_CUSTOMER:** This report provides an overview of customers contact information including name, phone number, address and email address.



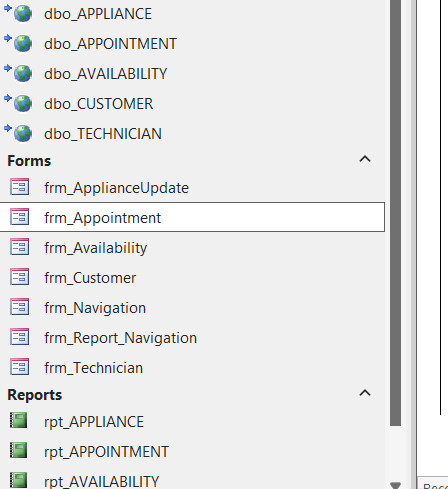
**rpt\_TECHNICIAN:** This report provides an overview of technicians' information including name, phone number.



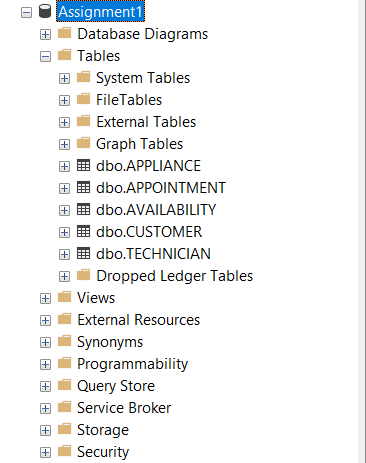
**Section 2 - how to use the back-end application (SQL Server)**

Once access and SQL Server is connected. Both front-end users and back-end users can access and manipulate the same tables, any saved changes will be updated accordingly to each end.

Here is shown from MS ACCESS:



Here is shown from SQL server:

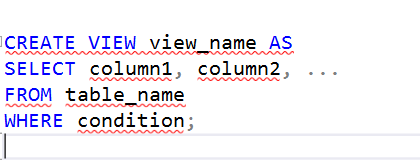


There are some essential and powerful features in SQL servers. This instruction will walk thru them individually.

### 1, views

Views are considered as frequently used queries saved within the database. Views can access and manipulate tables depending on the needs. Also, certain users may not have access to Tables directly, but they can access certain views instead.

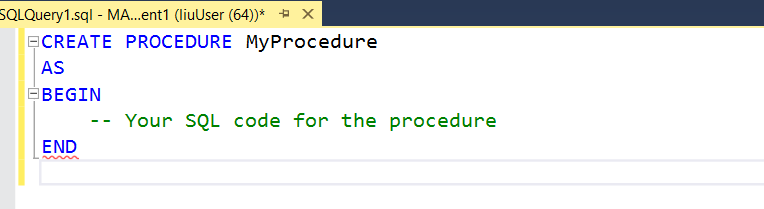
In SQL Server, the basic syntax for creating a view is:



### 2. Stored procedures:

Besides system-stored procedures, users can also create customized procedures for quicker access for specific groups.

The basic syntax is :



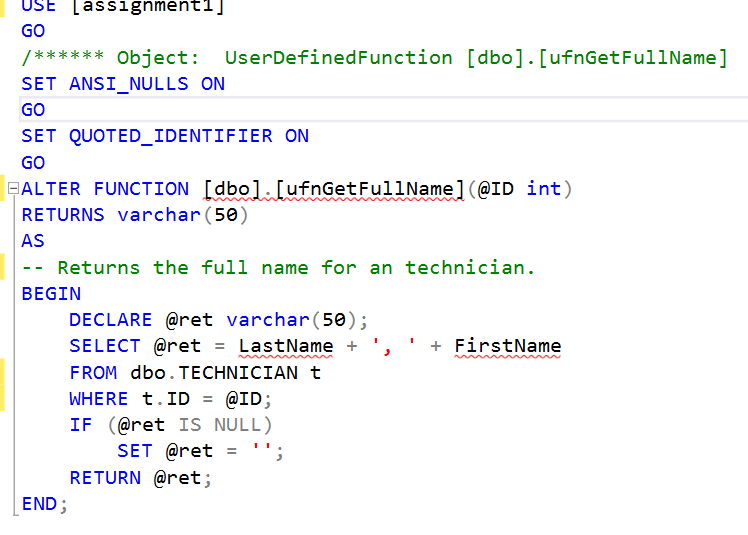
Users can verify all stored procedures by using this query:



### 3. User-defined functions:

Besides system-stored functions, users can also add user-defined functions for future references.

An example of creating a function to get the full name is shown as below:

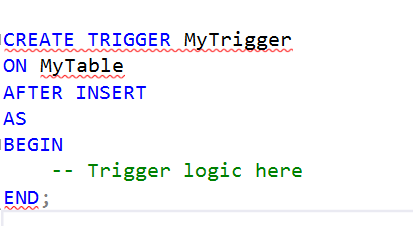


Once created, the user can access the functions by calling their function names with paraments while execute the queries.

### 4. Triggers

Triggers are typically associated with a specific table and are automatically executed in response to a particular event, such as an INSERT, UPDATE, or DELETE operation on that table. They are used to enforce data integrity, audit changes, or perform other actions based on database events.

To create a trigger in SQL Server, you would typically use a SQL script to define the trigger within the context of the database, like this:



### 5. Security and roles management

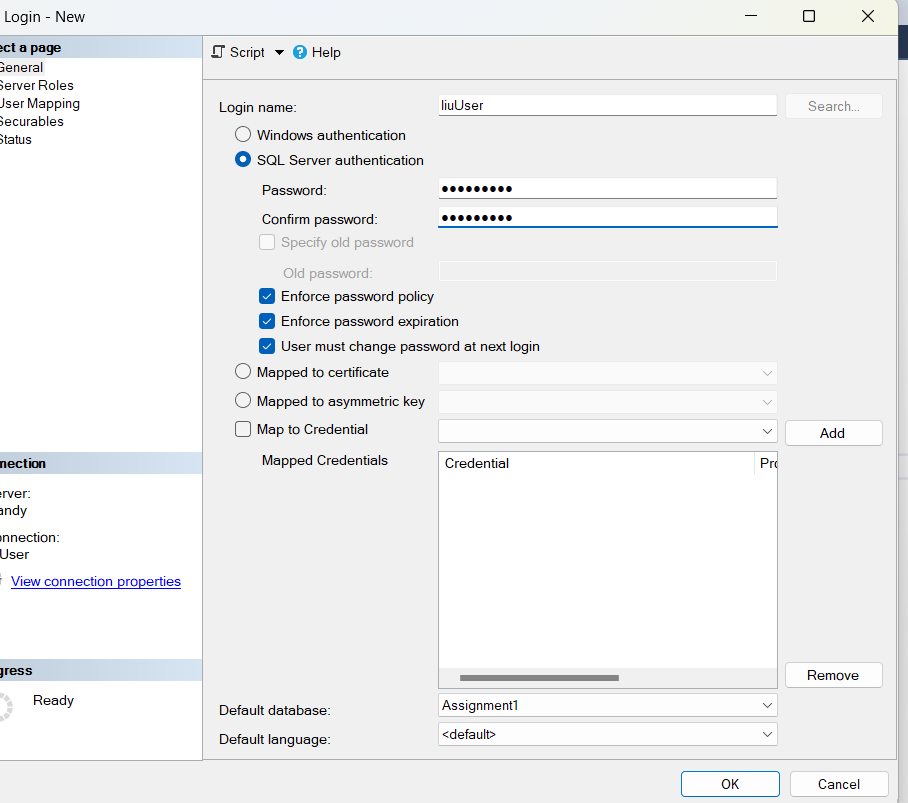
Creating different users for a database in a database management system, provides several benefits, including security, access control, and data isolation.

How to implement different user logins:

1. We need to create a new user account for our applications to use when they connect.

2. Open the Security dialog to create a new login.

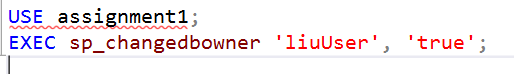
3. In the dialogue for New login, select the General page and fill in the following fields. Make sure you set the default database and unselect the password expiry fields.



4. If successful, you will see your new login in the list.

5. Now, while still connected as the Windows login user, run the following SQL commands to change the ownership of the database assignment1.

6. Change the owner using a query if needed:



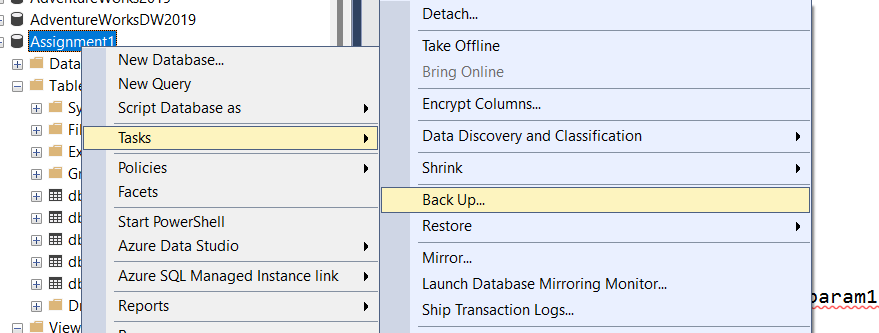
# Section3 – How to backup and restore database

Backup

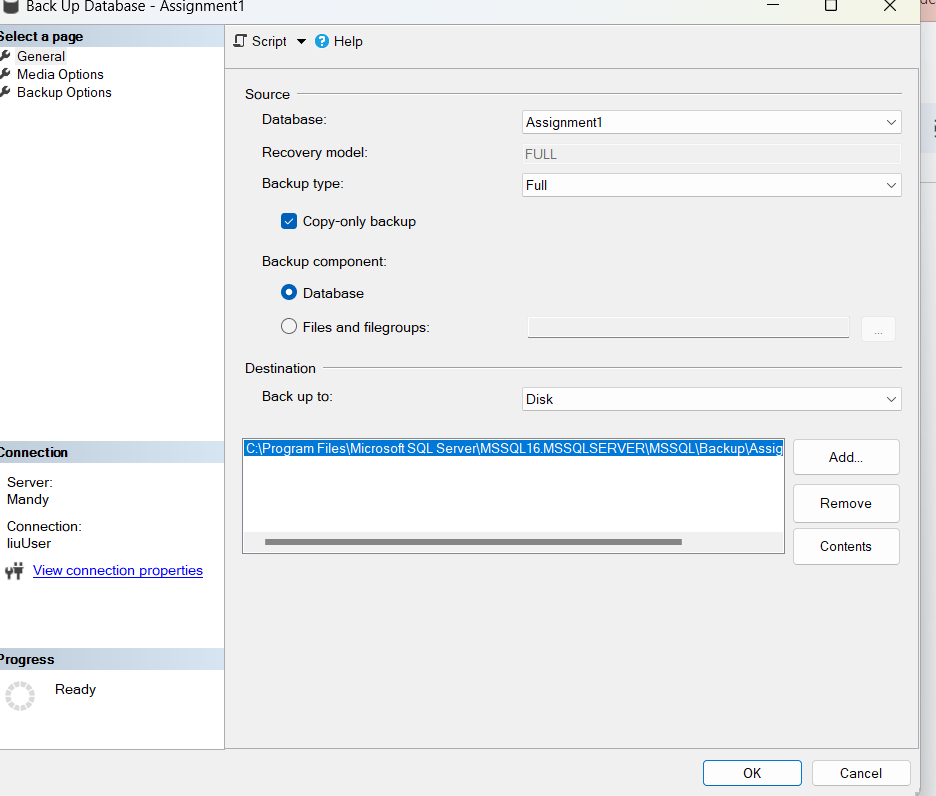
## Steps:

1,Create a backup of the “assignment1” database.

2.Select assignment1 > Tasks > Back Up …



3. Make sure you select the “Copy Only” option.

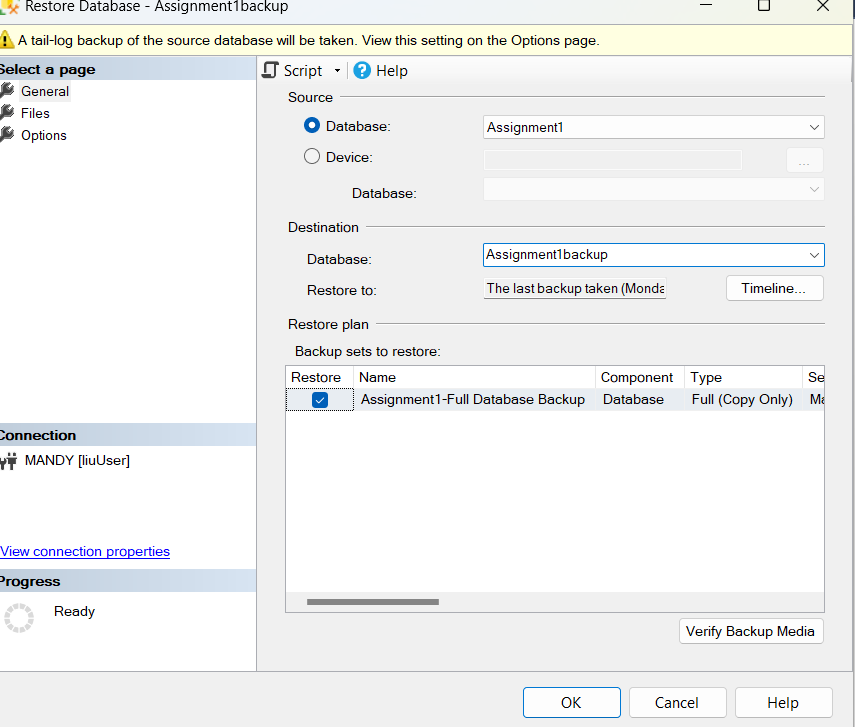


Restore

## steps:

1. Select Databases > Restore Database …

2. Provide the source database as “assignment1”. Then a list of backups will appear. Select the destination as “assignment1backup”.



3. Once restored, you will also need to change the owner, use a query as shown below:

