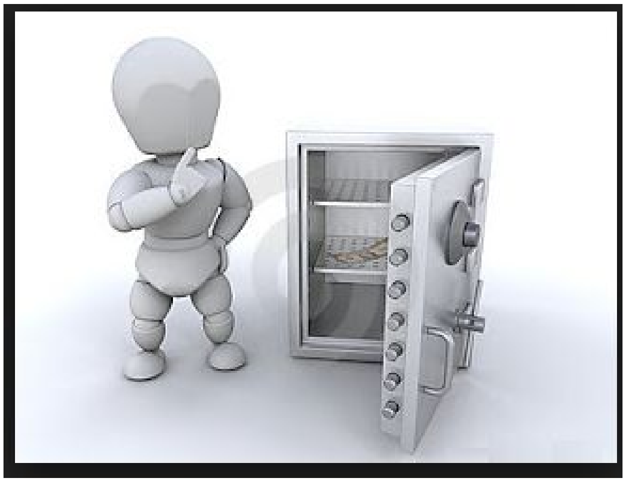
Chapter -12B

UFT & Database



“A new world to call home”

UFT can access Databases by using ActiveX Data Objects (ADO). ADO is a Microsoft technology. ADO can be used from any programming language that complies with the Component Object Model (COM). So, effectively, VBScript, VBA, Visual C++, Visual J++ can access data by using ADO. ADO allows a developer to write programs that access data without knowing how the database is implemented; developers must be aware of the database for connection only. Before success of ADO, developer previously used Remote Data Objects (RDO) and Data Access Objects (DAO)

Let's first discuss about the database world

Database is a collection of f ile that comprise user's data. A database management system is a set of software programs that allows users to create, edit and update data in database files, and store and retrieve data from those database files. Data in a database can be added, deleted, changed, sorted or searched all using a DBMS.

Structured Query Language (SQL) is query language that is used to perform operations using relational databases. SQL can be used to retrieve information from related tables in a database or to select, modify and retrieve information from specific rows and columns in one or more tables.

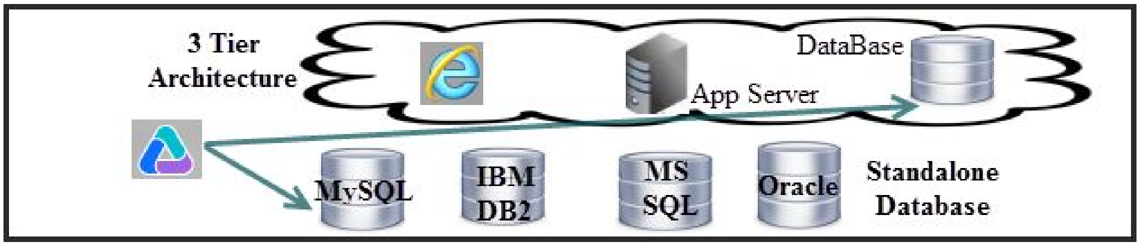
```
SELECT Name, Address FROM Employee WHERE salary > 2000
```

Every relational database supports SQL and its own proprietary language for example Oracle supports SQL and PL/SQL.

Few famous DBMS system:

Database	**Description**
MySQL	Free. One of the most influential and widely spread open source database It can be accessed over the Web as well. Extensively used with PHP.
Microsoft SQL Server	Designed to create web, enterprise, and desktop database systems.
Microsoft Office Access	MS Access can be used by small businesses to design special customized desktop systems for manipulating the creation and management of data.
PostgreSQL	Relational DBMS that many web application developers prefer as the back-end data management component. Advantages: open source community support, very low deployment cost, and easy administration
Oracle	One of the leading commercial SQL relational database management systems.
IBM DB2	IBM Relational DBMS system. Similar to other leading RDBMS

UFT can access any database with the help of ADO. It can access the application database or standalone database. We require to make connection with database by using connection string.



STEPS to Accessing Database by ADO

These are the following steps to access database:

1. Create an ADO connection to a database
2. Open the database connection
3. Create an ADO recordset
4. Open the recordset
5. Extract the data from the recordset
6. Close the recordset
7. Close the connection

Exercise – Connect to the database

STEP1: Copy the UFT database

☝ *Navigate to UFT home folder (C:\Program Files (x86)\HP\Unified Functional Testing\samples) | folder Samples | copy flight32.mdb | paste in "C:\Test"*

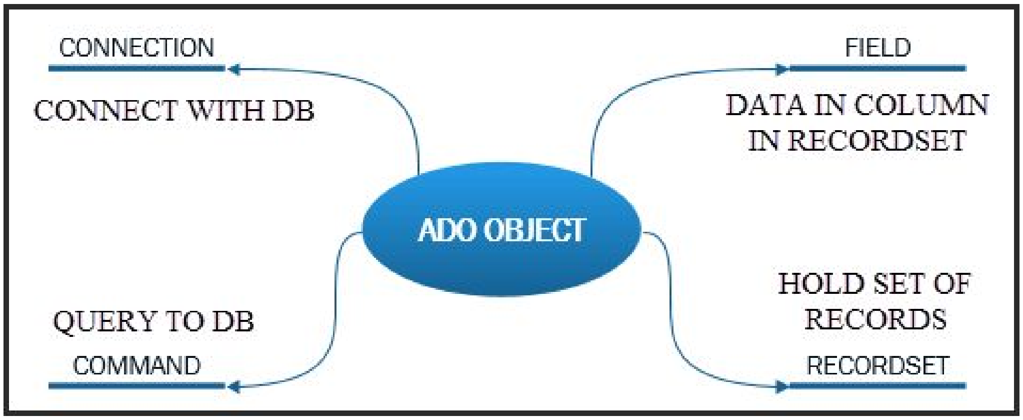
STEP2: Make the connection with database and verify connection

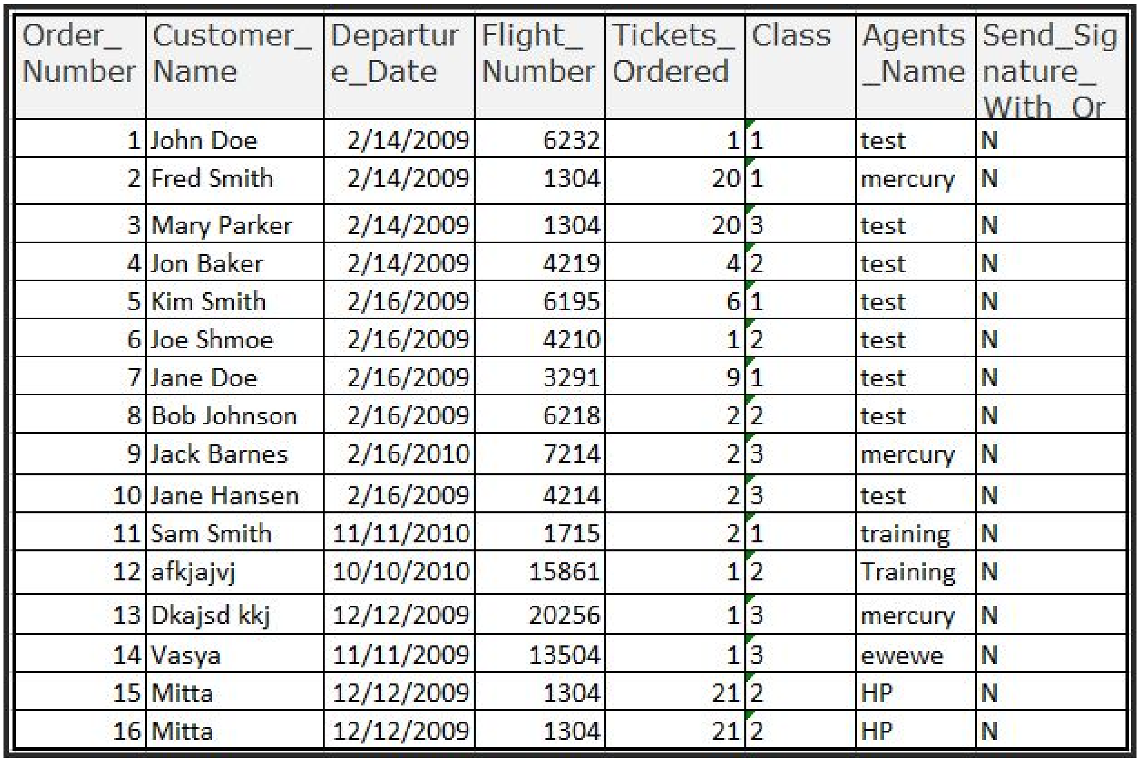
☝ *Navigate to UFT | Write the following code | Run*

```
Dim oCon
Set oCon =CreateObject("ADODB.Connection")
oCon.Provider="Microsoft.Jet.OLEDB.4.0"
oCon.Open "c:\Test\flight32.mdb"
Msgbox conn.State  ' 1 = connection is open
oCon.Close
Msgbox conn.State  ' 0 = connection is close
Set oCon = nothing
```

Concept of Exercise – Connect to the database

In STEP1, we are copying the database of UFT flight reservation sample application. This is MS Access database. You can also install MS Access and create your own database. The current MS Access database extension is .accdb instead of .mdb (after windows7). This database has many tables including "Orders" table.
In STEP2, we are creating an ADODB object. This object enables to connect the database, query and manipulate the database. ADODB supports many objects for example Connection, Command, Recordset and field. We will explore these objects later in chapter.





Order_Number	Customer_Name	Departure_Date	Flight_Number	Tickets_Ordered	Class	Agents_Name	Send_Signature_With_Or
1	John Doe	2/14/2009	6232	1	1	test	N
2	Fred Smith	2/14/2009	1304	20	1	mercury	N
3	Mary Parker	2/14/2009	1304	20	3	test	N
4	Jon Baker	2/14/2009	4219	4	2	test	N
5	Kim Smith	2/16/2009	6195	6	1	test	N
6	Joe Shmoe	2/16/2009	4210	1	2	test	N
7	Jane Doe	2/16/2009	3291	9	1	test	N
8	Bob Johnson	2/16/2009	6218	2	2	test	N
9	Jack Barnes	2/16/2010	7214	2	3	mercury	N
10	Jane Hansen	2/16/2009	4214	2	3	test	N
11	Sam Smith	11/11/2010	1715	2	1	training	N
12	afkjajvj	10/10/2010	15861	1	2	Training	N
13	Dkajsd kkj	12/12/2009	20256	1	3	mercury	N
14	Vasya	11/11/2009	13504	1	3	ewewe	N
15	Mitta	12/12/2009	1304	21	2	HP	N
16	Mitta	12/12/2009	1304	21	2	HP	N

Exercise: What you think about the following queries?
Select * from Orders where Flight_number > 6500
Select * from Orders where customer_name like ‘%mi’ And agents_name =”test”
Select sum (tickets_ordered) from Orders group by agents_name
Select count (*) from Orders

Key Learning of exercise: UFT can access various datasource (database, excel etc.) by Microsoft ADO. Connection object can connect database and recordset object can hold the records. The connection for different datasource requires different parameters for connection string.

Exercise – Connect to the database – END

Exercise – Get data from database

STEP1: Make connection with database

☝ *Navigate to UFT | File | New | “RecordSet” test | Write following code*

```
Dim oCon
Set oCon =CreateObject("ADODB.Connection")
oCon.Provider="Microsoft.Jet.OLEDB.4.0"
oCon.Open "c:\Test\flight32.mdb"
```

STEP2: Query the database and hold the record in recordset

☝ *Navigate to “RecordSet” test in UFT | append following code | Run*

```
set oRes=CreateObject("ADODB.recordset")
oRes.Open "Select * from orders", conn
Do while Not oRes.EOF
 print oRes.fields(0) & oRes.fields(1) & oRes.fields(2)
 print oRes.fields(3) & oRes.fields(4) & oRes.fields(5)
 print oRes.fields(6) & oRes.fields(7)
i = i +1
oRes.MoveNext
Loop
oRes.close
conn.Close
Set oRes = nothing
Set conn = nothing
```

STEP3: Send record in excel sheet

☝ *Navigate to “C:\Test” | create oreder.xlsx file | navigate to “RecordSendInExcel” test in UFT | Write following code | Run*

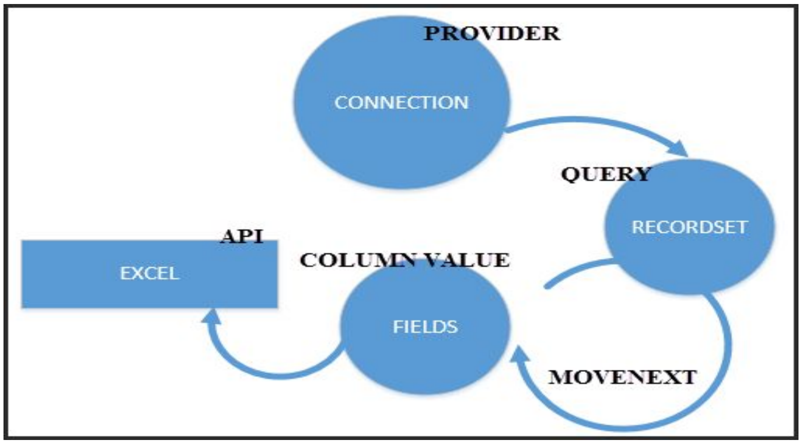
```
set conn=CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:\Test\flight32.mdb"

set oRes=CreateObject("ADODB.recordset")
oRes.Open "Select * from orders", conn
count =1

Set objExcel=CreateObject("Excel.Application")
objExcel.Visible=True
Set objWBook=objExcel.Workbooks.Open("C:\Test\orders.xlsx")
Set objSheet=objWBook.ActiveSheet

i = 1
Do while Not oRes.EOF

 objSheet.Cells(i,1) = oRes.fields(0)
 objSheet.Cells(i,2) = oRes.fields(1)
 objSheet.Cells(i,3) = oRes.fields(2)
 objSheet.Cells(i,4) = oRes.fields(3)
 objSheet.Cells(i,5) = oRes.fields(4)
 objSheet.Cells(i,6) = oRes.fields(5)
 objSheet.Cells(i,7) = oRes.fields(6)
 objSheet.Cells(i,8) = oRes.fields(7)

i = i +1
oRes.MoveNext
Loop
objWBook.Save
objWBook.Close
objExcel.Quit
oRes.close
conn.Close
Set oRes = nothing
Set conn = nothing
Set objSheet=Nothing
Set objExcel=Nothing
```

Concept of Exercise – Get data from database

In STEP1, we are creating ADODB connection object. This connection object need to know about the database and provider. Provider enables the connection with ADODB and database. In STEP2, We are creating another ADODB object that can hold the recordset from database. The Recordset object has the open method that takes two arguments 1. SQL query 2. Connection object. After recordset.open statement, we get the query results in it. You can think it like the values in excel sheet filled with rows and columns and it ends when recodset.EOF becomes true. Recordset support fields object. Fields is cell value in the recordset. We can access cell in recordset by index or table column name.

Fields(0)	Fields(1)	Fields(2)	Fields(3)	Fields(4)	Fields(5)	Fields(6)	Fields(7)
1	John Doe	39858.74	6232	1	1	test	N

In STEP3, we are making connection with excel by excel API and sending all the recordset values into excel. This step demonstrate that we can device any logic on the dataset after getting hold on the recordset. We can use it for database checkpoint, sending data to 3rd party application, store data in datatable or write logic in test script.



Now the first point first. How do we know that which provider we need to make the correct database connection? Do all database connects with same provider? Is there any other way to make connection? This is the most important aspect of database connection with UFT.

When we open connection object, we attempt to connect datasource either by ODBC (open source) or OLE DB (Microsoft technology). State property returns whether connection successfully established or not. Execute method can run SQL or stored procedure of the connection object. If any data return by query then it stores in recordset object. Following are few important methods and properties.

Method	Description	Properties	Description
Open	Opens a connection to a data store.	**ConnectionString**	Contains the information used to establish a connection to a data store.
Close	Closes a connection and any dependent objects	**ConnectionTimeout**	Wait while establishing a connection before terminating the attempt
Execute	Executes SQL statement or stored procedure	**CommandTimeout**	Wait while executing a command before terminating the attempt
CommitTrans	Saves any changes and ends the current transaction.	**State**	Indicates whether a connection is currently open, closed, or connecting.
RollbackTrans	Cancels any changes made during the current transaction	**Provider**	Indicates the name of the provider used by the connection.

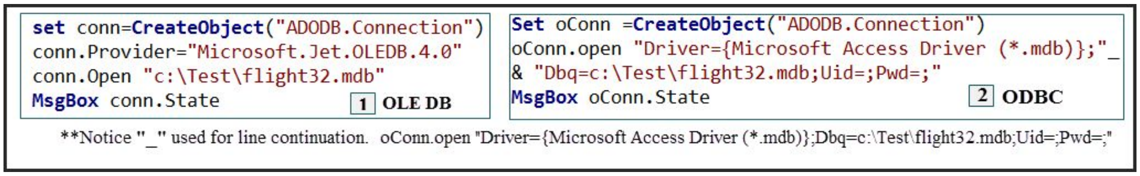
Connection.Open method takes following parameter. Connection string contains the crucial information about the database.

```
connection.Open ConnectionString, UserID, Password, Options
```

If you are making connection by OLE DB then connection will look something like this
“Provider=MySQLProv;Data;Source=mydb;User Id=myUsername;Password=myPassword;

If you are making connection by ODBC then connection will look something like this

‘MySQL connection
"DRIVER={MySQL ODBC 3.51 Driver};SERVER=localhost;DATABASE=newdb;user id=root ; password=admin"
‘’SQL Server connection
“DRIVER={SQL SERVER}; Server= localhost; DATABASE= newdb;uid=sa; pwd=”



```
set conn=CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open "c:\Test\flight32.mdb"
MsgBox conn.State
```

1 OLE DB

```
Set oConn =CreateObject("ADODB.Connection")
oConn.open "Driver={Microsoft Access Driver (*.mdb)};"_
& "Dbq=c:\Test\flight32.mdb;Uid=;Pwd=;"
MsgBox oConn.State
```

2 ODBC

**Notice "_" used for line continuation. oConn.open "Driver={Microsoft Access Driver (*.mdb)};Dbq=c:\Test\flight32.mdb;Uid=;Pwd=;"

There are various provider for different data stores for example SQL Server, Oracle etc. In same way there are various driver versions to connect to the database. It is your design consideration to choose the way to connect the data source. Here we need to understand that we can connect to Excel, file system and other data store in same way as database. Just we need to find the correct provider or ODBC driver. The details of ODBC or providers are out of scope of this book though you should have correct driver installed on your machine to run the code. We will demonstrate common examples to make connection with Excel and database.

We are interested in database records. So, the most important usability of ADO object is recordset. Recordset always start pointing the first result in recordset. We can navigate in recordset by MoveNext method. There are other methods to move for example MoveLast, MoveFirst and MovePrevious. These are very helpful methods. We can iterate the recordset until recordset.EOF i.e. “End of File” and process the desired information. Recordset can take the SQL query otherwise we can execute SQL query (or stored procedure) by execute method of ADODB command object

```
Set objcommand = CreateObject(“ADODB.COMMAND”)
Set objcommand.CommandText="SELECT * FROM Orders"
objCommand.Execute
```

Key Learning of exercise: ADODB supports various object to get data from various sources including database, excel, flat file etc. We can make the valid connection with data source and run the desired sql query.

Exercise – Get data from database – END

```
con.open "Driver={Microsoft Access Driver (*.mdb)};Dbq=C:\mydatabase.mdb;Uid=Admin;Pwd=;"
```

```
con.open"Driver={SQL Server};server=MySqlServer;uid=MyUserName;pwd=MyPassword;database=pubs"

con.open "Driver={Microsoft ODBC for Oracle};Server=QTPWorld; Uid=your_username;Pwd=your_password;"

con.open"Driver={MySQL ODBC 3.51 Driver};Server=localhost;Database=myDB;User=Uname;Password=Pwd;Option=3;"

con.open "DRIVER={Microsoft Excel Driver (*.xls)};DBQ=C:\TestStatus.xls;Readonly=True"

con.open"Driver={SYBASE SYSTEM 11};Srvr=myServerAddress;Uid=Uname;Pwd=Pwd;Database=myDataBase;"
```

Exercise – Get data from Excel

STEP1: Create a MS Excel with data

☝ *Navigate to “C:\Test” | Create Orders1.xlsx | Copy all records from Order.xls to Orders1.xlsx*

STEP2: Make connection with MS Excel and get the recordset

☝ *Navigate to UFT | File | New | “ExcelTest” | Write following questions | Run*

```
1   Set oConn = CreateObject("ADODB.Connection")
2   strFileName="C:\Test\orders1.xlsx"
3   oConn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" _
3B              & "Data Source=" & strFileName & ";" _
3C              & "Extended Properties='Excel 8.0;HDR=YES;IMEX=1';"
4   Set oRes = CreateObject("ADODB.Recordset")
5   oRes.Open "select *  FROM  [Sheet1$] ", oConn
6   Do until oRes.EOF
7   print oRes.fields(0) & oRes.fields(1) & oRes.fields(2)
8   print oRes.fields(3) & oRes.fields(4) & oRes.fields(5)
9   print oRes.fields(6) & oRes.fields(7)
10  oRes.MoveNext
11  Loop
12   'You can write many statement in single line by using ":"
13  oRes.Close: oConn.Close:  Set oRes = Nothing: Set oConn = Nothing
```

Concept of Exercise – Get data from Excel

In STEP1, we are creating ordres1.xlsx and writing data into it. In STEP2, we are taking data from MS Excel by using ADO objects. As we know, MS Excel has a robust API (remember CreateObject(Excel.Applictaion)?) but we can access records by ADO as well.
Line-3 opens the database connection by OLE DB (not ODBC way) “Microsoft.Jet.OLEDB.4.0;” provider. The “_” used for line continuation otherwise we can write Line 3, 3B & 3C in a single line.
Line-5 opens the recordset with query and excel connection. Query contains [Sheet1$] as table name.So, effectively we can get the data from any excel sheet by changing the sheet name.
Line-7 prints the record by using “fields” method. Remember fields method either take index (0, 1.. not “0”,”1”) or column name of table.
Line-10 is important to navigate in recordset otherwise we will keep reading the first row of recordset.
Line-13 is also very crucial. Closing and releasing memory is one of the most important thing to work with 3rd party application objects. They must be in correct order for example recordset cannot close after closing connection object. One more thing in this line is that we can separate multiple statement in VBScript by using “:”. You cannot write Dim i i=10 in vbscript. You must use new line to write i=10. But If you use “:” then you can write statement in same line i.e. dim i : i=10

Character	VBS Interpreter	Example
:	Statement termination	oRes.Close: oConn.Close: Set oRes = Nothing:
_	Line continuation	oConn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" _ & "Data Source=" & strFileName & ";" _
&	variable/string concatenation	

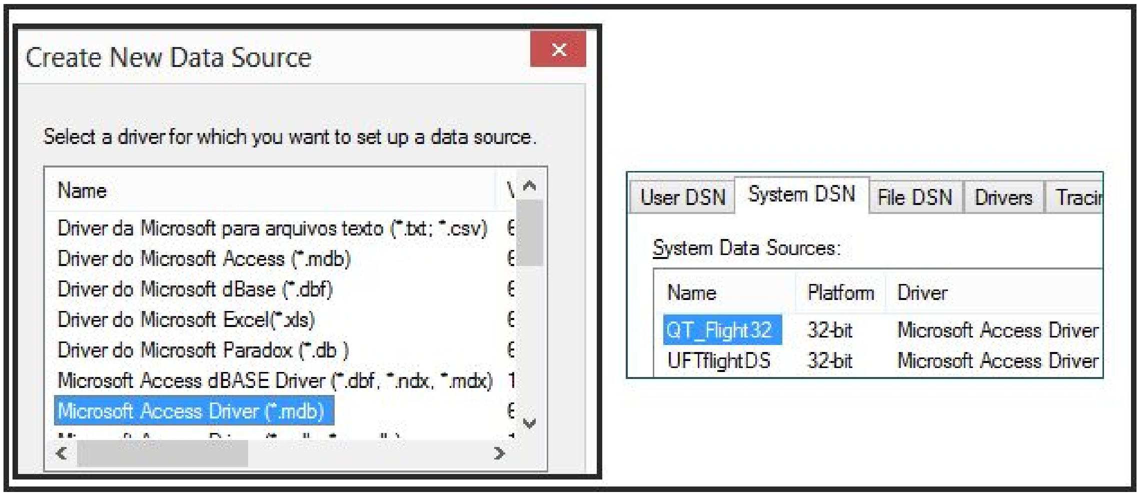
Key Learning of exercise: ADO (ActiveX Data Object) provides versatile API to access various kind of data sources. Majorly it used for databases but it can pave the way to access other applications as well.

Exercise – Get data from Excel – END

Exercise – Get data from database without programming

STEP1: Make connection with database

Navigate to Control Panel | System and Security\Administrative Tools | ODBC Data Sources | System DSN | Add | Select Microsoft Access Driver (.mdb) driver | Finish | “UFTflightDS” Data source name | Select database | “C:\Test” | Select “flight32.mdb” | OK | OK | OK*



Create New Data Source

Select a driver for which you want to set up a data source.

Name
Driver da Microsoft para arquivos texto (*.txt; *.csv)
Driver do Microsoft Access (*.mdb)
Driver do Microsoft dBase (*.dbf)
Driver do Microsoft Excel(*.xls)
Driver do Microsoft Paradox (*.db)
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx)
Microsoft Access Driver (*.mdb)

System Data Sources:

Name	Platform	Driver
QT_Flight32	32-bit	Microsoft Access Driver
UFTflightDS	32-bit	Microsoft Access Driver

STEP2: Make connection with database by using DSN and get the recordset

Navigate to UFT | File | New | “DSN_ODBC_Test” | Write following questions | Run

```
Set oConn = CreateObject("ADODB.Connection")
oConn.Open("DSN=UFTflightDS")
sql = "Select *  From orders"
Set oRes = CreateObject("ADODB.Recordset")
Set oRes = oConn.Execute(sql)
print oRes.GetString
oConn.Close
Set oConn = Nothing
```

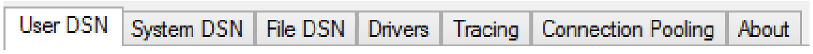
Concept of Exercise – Get data from database without programming

In STEP1, We are creating the DSN i.e. data source name. This DSN will have all the information to connect with database. Microsoft ODBC administration wizard provides you opportunity to create DSN for various kind of data sources including, excel, files, csv and many more formats.

Here we are creating DSN for MS Access database.

In STEP2, we are opening ADODB connection with only DSN name. Point to note that we have not gave any other parameter. After opening the parameter, the process to get the data remains same but this time we are not giving sql query in recordset object yet we are using .Execute method to run the query. The execution of query returns the recordset. Recordset object has many methods and properties. GetString returns the whole recordset data as a string.

A data source name (DSN) contains the information about a specific datasource an Open Database Connectivity driver needs in order to connect to it. There are three type of DSN.



	DSN TYPES	Description
1	User DSN	It allows database access for a single user on a single computer.It store DSN information in the registry.
2	System DSN	It allows database access for all user on system.It store DSN information in the registry.
3	File DSN	It save information in text file with a .DSN file extension. It can be shared by users of different computers

Key Learning of exercise: Data sources can be access by creating the DSN. DSN contains all the relevant information to make the connection with different data sources including database, excel, csv etc. We need to install the proper driver before creating DSN. Microsoft already provide driver for many applications by default e.g. MS Access, Excel, SQL server, Oracle etc.

Exercise – Get data from database without programming – END