

Topic Two

In this section you will cover the following topics:

Select data-access layer (DAL)

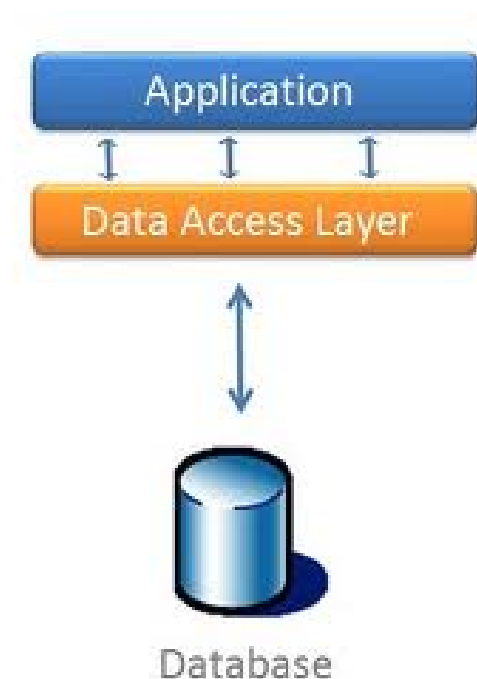
Review function of DAL in multi-layer application model

Determine data-access application programming interface (API) for connecting to various data sources

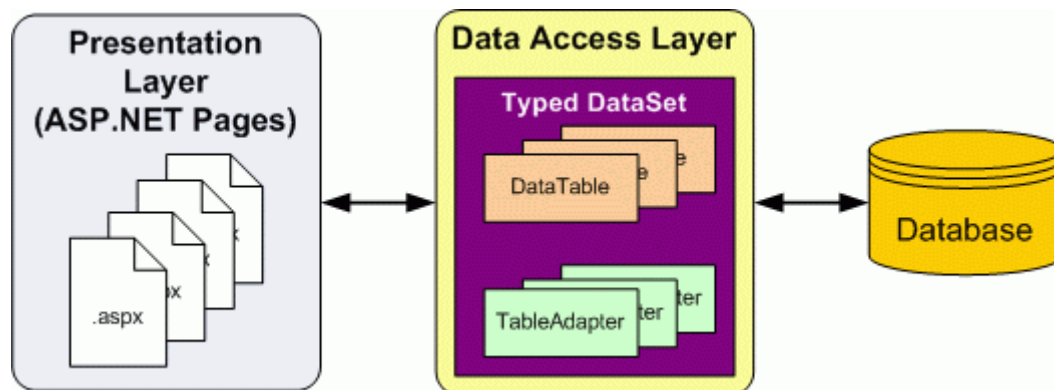
Data Access Layer

As ICT professionals, our lives revolve around working with data. We create databases to store the data, code to retrieve and modify it, and create user interfaces to collect and summarize it. The data access layer (DAL) is the interface between the end user application and the database. Basically, a data access layer is just code we write to interact with the data source; it can be as complex or as simple as we want or need. As an example, we could create a data access class to handle interactions with the MySQL and the “products” database we created in the Hardware Database student activity. Our application could be programmed to perform the following database operations:

- Add new products to the table
- Delete products from the table
- View product information contained in the table



The primary purpose of the DAL is to keep the code you use to interact with the data in your database separate from the business logic and presentation code. This way, if you have to change database, you reduce the need to rewrite all the code as shown in the figure below;



Instead of using commands such as insert, delete, and update to access a specific table in a database, a class and a few stored procedures could be created in the database. The procedures would be called from a method inside the class, which would return an object containing the requested values. Or, the insert, delete and update commands could be executed within simple functions like registeruser or loginuser stored within the data access layer.

Application Programming Interface

In computer programming, an application programming interface (API) is a set of routines, protocols, and tools for building software applications. It is a language and message format used by an application program to communicate with the operating system or some other control program such as a database management system (DBMS) or communications protocol.

There are many different types of APIs for operating systems, applications or for websites. Windows, for example, has many API sets that are used by system hardware and applications — when you copy and paste text from one application to another, it is the API that allows that to work.

To determine the data-access API for connecting to various data sources it is important to read the API reference documentation. This document will include the following:

- a description of all the functions signatures, including:
- a description of all the data structures it depends upon
- any accessibility or visibility constraint.

The following list of Data Access API's represents the most widely used to interact with modern databases. These APIs are the gateways that let SQL Server function as a data source both for traditional, 2-tiered client/server applications and as a data source for Web and n-tiered applications.

1. ADO

ADO is a COM-based object framework. Unlike its predecessors, ADO is an object framework over OLE DB. So, you can use ADO and OLE DB in development tools such as VB, VBScript, Java, JScript, and ASP.

2. OLE DB

Positioned as the successor to ODBC, OLE DB provides access to relational and non-relational data sources. OLE DB is the cornerstone of Microsoft's Universal Data Access strategy, and you can use OLE DB to access any data source that can be represented by a row-and-column format.

3. RDO

The RDO object framework combines the best of DAO and ODBC. Like DAO, RDO is COM-based. Unlike DAO, RDO was developed to work with ODBC. RDO is easy to use and provides good performance. However, although RDO is good at relational database access, that task is all it does.

4. Data Access Object (DAO)

Microsoft developed DAO to provide access to the Microsoft JET database, which Access uses. DAO is a COM-based object framework that Microsoft later expanded to address ODBC connectivity. DAO is much easier to use than its CLI-based predecessors, but its JET orientation made it less than optimal for connecting to ODBC data sources such as SQL Server.

5. ODBC

ODBC provides a vendor-independent database-access API. The ODBC API has been wildly successful as a database-access standard. Virtually all products that require database access support it, and the most recent ODBC driver supports the SQL Server 7.0 enhancements. You use a call-level interface (CLI) to implement the ODBC API.

The Data Definition Language Component

The Data Definition Language (DDL) component is used to create tables and establish relationships among tables, and the Data Manipulation Language (DML) component is used to manage the database by performing such operations as retrieving, updating, deleting, and navigating through data. The Data Control Language (DCL) component is used to provide security to data in a database. Some of the commands within each of these components are as follows:

DDL	DML	DCL
CREATE TABLE DROP TABLE ALTER TABLE CREATE INDEX	INSERT INTO SELECT INTO UPDATE DELETE SELECT UNION TRANSFORM PARAMETER	ALTER DATABASE CREATE GROUP DROP GROUP CREATE USER ALTER USER DROP USER ADD USER GRANT PRIVILEGE REVOKE PRIVILEGE

END of TOPIC TWO