

Introduction to Access Development

Balter Chapter 1

Access as a Development Platform

- Personal applications
 - Database templates and Wizards make things easy... up to a point
 - novices reach a dead end when need to customize
- Small-business applications
 - excellent
 - rapid development
 - VBA code to make forms/reports come to life
- Departmental applications
 - perfect
 - shared data
 - adequate performance w/o need for client-server technology

Access as a Development Platform

- Corporation-wide applications
 - lots of factors affect performance
 - Balter's rule of thumb: 10-15 concurrent users, 100,000 records/table
 - bottleneck: each user's workstation does all the processing
- As a front-end in a client/server application
 - database server (egs: Oracle, SQL Server, MySQL) stores data and performs processing
 - Access presents data (forms/reports) and does some local processing
- Intranet/Internet Applications
 - can export tables/forms/reports as HTML documents for web
 - can use other technologies to build web pages that harvest db content to present via browser
 - Data Access Pages (DAP) are HTML forms that use IE and Access to expose data over web
 - when published to web server users can view/change data via browser
 - deprecated in Access 2007

Database Objects

- Tables (data storage)
- Relationships (data integrity)
- Queries (processing)
- Forms (view/edit/insert/delete)
- Reports (output)
- Macros (processing)
 - to automate a series of actions
 - enhanced in Access 2007
 - embedded macros, error handling, variables
- Modules (processing)
 - more powerful and flexible than macros
 - a major focus for CIS217
 - contain **procedures**
 - a series of VBA code statements that perform an operation or calculate a value, frequently in response to an event

Object Naming Conventions

- Helps make joint development & maintenance easier
- [prefixes]tagBaseName[Suffixes]
- Examples
 - tblCustomer
 - frmContacts
 - fdlgReports
 - frmReportsDialog
 - cmdClose
 - cboClientID
 - strMessage
 - mstrMessage
- Callahan doesn't implement consistently
 - we will in our other assignments

Development Framework

- Task Analysis
 - a database models a real-world organization and supports its operation
 - understand each user's needs, flow of information, business procedures
- Data Analysis & Design
 - identify each data element, its data type, required size, validation rules
 - determine whether the value is entered or can be calculated
 - Normalization
 - a process yields a final set of correctly designed tables
 - avoid data redundancy
 - each type of entity is stored in a separate table
 - each record has a unique primary key identifier
 - desirable if pk is also minimal & static
 - fields are attributes that describe an entity
 - each field is atomic

Development Framework

- Prototyping
 - build tables/relationships/queries/forms/reports/modules
 - Wizards are a big productivity boost
- Testing
 - test each object individually and the entire system to ensure they provide specified functionality
 - actual end-users
- Implementation
 - distribute/install
 - collect user feedback
- Maintenance
 - bug fixes
 - specification changes
 - enhancements /new features
 - optimization
 - backup

Database Engine

File name:	Database1.accdb
Save as type:	Microsoft Office Access 2007 Databases (*.accdb)
	Microsoft Office Access Databases (2002-2003 Format)(*.mdb)
	Microsoft Office Access Databases (2000 format)(*.mdb)
	Microsoft Office Access 2007 Databases (*.accdb)
	Microsoft Office Access Projects (*.adp)

- The underlying software a database management system uses to create, store, retrieve, process, update, and delete relational data and associated objects
 - Jet (Joint Engine Technology)
 - database engine in versions prior to Acc07
 - .MDB file format
 - ACE (Microsoft Access Engine)
 - the new database engine used in Acc07
 - uses .ACCDB file format
 - improved security and encryption, multivalued fields, the Attachment data type, and history tracking in Memo fields
 - backward-compatible with previous versions of Jet, reads/writes . mdb
 - SQL Server Express
 - free, easy-to-use, lightweight version of SQL Server (previously MSDE)
 - included with Access 2007, Microsoft Visual Studio
 - used by . adp file format



Access ≠ Database Engine

- Access itself manages forms, reports, and macros
- Access relies on a database engine to manage table objects, inter-table relationships, and query execution
- Access works cooperatively with Visual Basic to execute procedures stored in modules
- Access is a graphical user interface which translates user interactions with data into requests that are passed to ACE/Jet/database server which actually performs data storage and query processing behind the scenes and passes results to Access to display

Access Architecture Options

- **Single-User Approach**
 - user's workstation has a local .ACCDB/.MDB file which stores all objects
 - tables, queries, forms, reports, macros, modules
 - all processing done on user's workstation, where Access & ACE/Jet run
 - Access = front-end and back-end
- **File Server Approach**
 - multiple concurrent users
 - data is stored in .ACCDB/.MDB file in a shared network folder
 - all processing done on user's workstation, where Access & ACE/Jet run
 - eg: open a report whose record source is a query, the query definition, report definition, and entire table of data come down network to workstation where ACE/Jet runs the query, applies criteria, passes results to Access to format as specified by report's design
 - lots of network traffic, especially when lots of users
 - each user must have Access/ACE/Jet installed on their workstation
 - [Access runtime version](#)

Access Architecture Options

■ Client/Server Approach

- **Back-end** (egs: SQL Server, Oracle, MySQL)
 - database server software runs on a server
 - stores table data, performs processing, sends results to requesting workstation
 - provides security, referential integrity, data validation
- Access .ACCDB/.MDB **front-end** uses ADO to connect to Oracle back end
 - Access sends **data request** (SQL statement) to Oracle
 - Oracle fetches only the data requested, performs processing
 - sends **results** back to Access to present via a form/report
- Access Data Project (.ADP) **front-end** connects to SQL Server back-end
 - Access sends **data request** to SQL Server, which fetches the data requested, performs processing, sends **results** back to Access to present via form/report
- **Advantages**
 - reduces network traffic (improves performance)
 - can handle more concurrent users
 - greater control over data integrity
 - increased security

