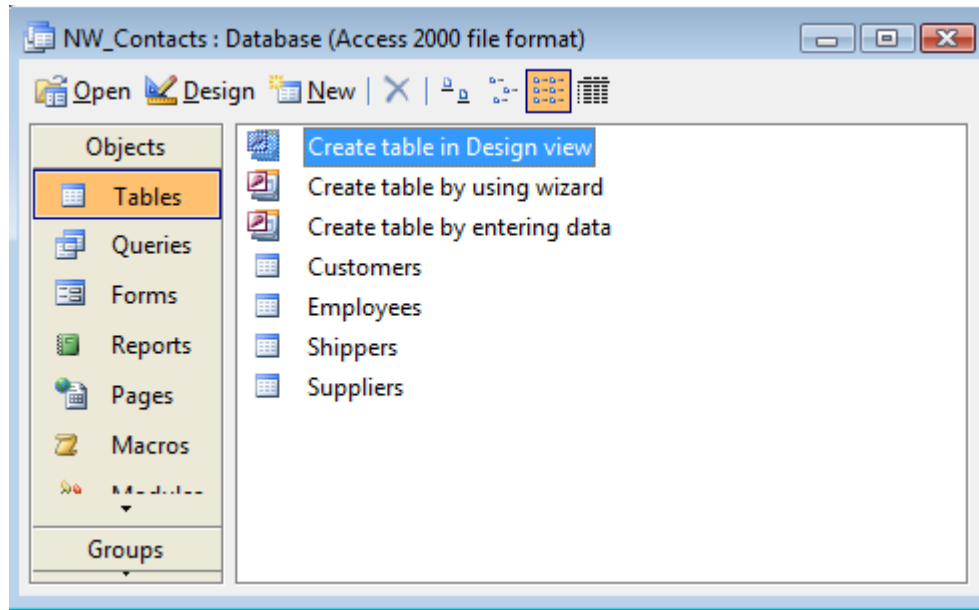


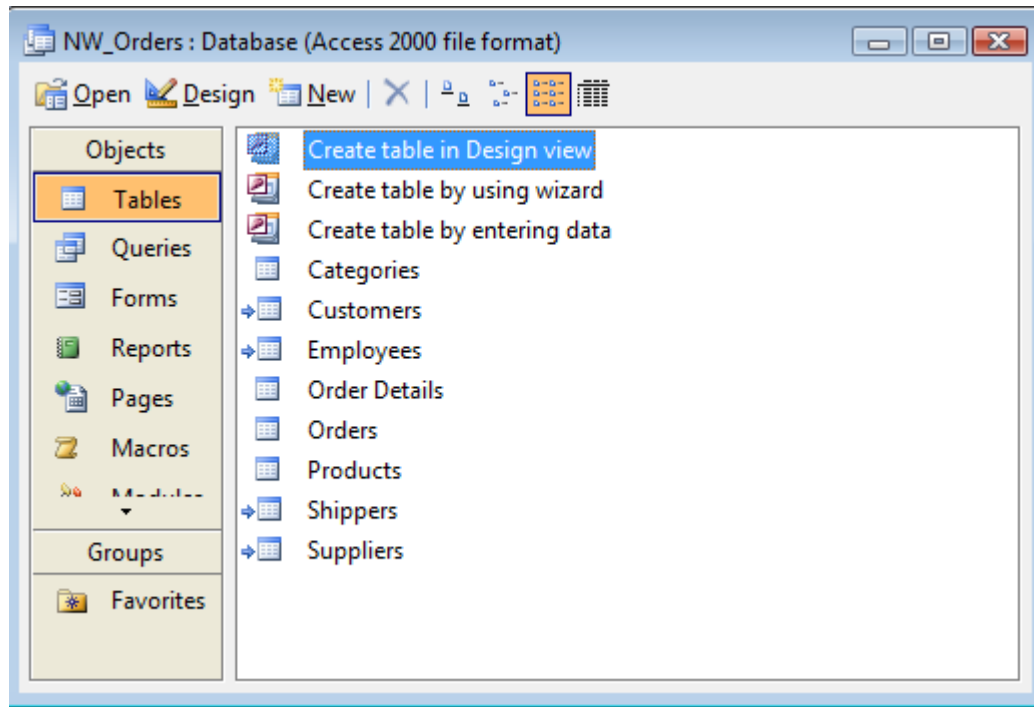


# Upsizing using schemas

<http://www.vb123.com/up>



NW\_Contacts, is a database in the company which holds information on all our employees, customers and suppliers. The database is owned and managed by the accounts team.



NW\_Orders, is a database in the company which is used for sales order processing. The database is owned and managed by the sales team.

We have decided to bring both the databases into a single SQL Server database.

But we want to keep the contacts parts of the system distinct from the sales part of the system.

We want to do this because of security, and to make it simpler to manage future design changes for the two teams.

It is decided to have a sales schema for the sales team, and an accounts schema for the accounts team.

We will now show how MUST (all versions) supports the migration of the two Access databases into a single SQL Server database.

Using the schema menu, we first create our two new schemas.

Migrate Structure And Data | Full Database Auditing | Browse Data Dictionary | Migrate Queries | Web Conversion | Commissioning Data | Reverse

schemas server version and field data types table exclusions preferences about display status reset system help exit system

Status

**define SQL Server 2005 schemas, schemas allow a large application to be more easily managed**

Data Sources

- ..... Access data files : 0
- ..... Access applications : 0
- ..... SQL Server : 0

Analysis

- ..... Databases analysed : 0
- ..... DB's Tables linked : 0
- ..... DB's Queries replaced : 0
- ..... DB's Forms - views : 0
- ..... DB's Reports - views : 0

Schema	
	dbo
	Reporting
	Accounts
▶	Sales
*	

Then we follow a sequence of steps to upsize the databases.  
First we create our SQL Server database.

The screenshot shows the Microsoft Access software interface during the process of creating a new SQL Server database. The ribbon at the top contains various tools for migration, including 'select SQL Server', 'Access databases', 'analyse Access data', 'review exclusions', 'migrate structure', 'migrate data', 'verify row counts', 'relink application', 'data and structure script files', 'error resolution', 'changes to design', and 'reporting migration results'. The status bar indicates 'select an existing database or create a new SQL server database'. The left-hand navigation pane shows the 'Data Sources' section with 'Access data files : 0', 'Access applications : 0', and 'SQL Server : 0'. The 'Analysis' section shows 'Databases analysed : 0', 'DB's Tables linked : 0', 'DB's Queries replaced : 0', 'DB's Forms - views : 0', and 'DB's Reports - views : 0'. The 'Data Dictionary' section shows 'Schemas : 4', 'Tables : 0', 'Tables excluded : 0', 'Relationships : 0', 'Relationships excluded : 0', 'Indexes : 0', 'Indexes excluded : 0', 'Table Rules : 0', 'Field Rules : 0', 'Design changes : 0', 'Outstanding errors : 0', 'Resolved errors : 0', and 'Yes/No defaults added : 0'. The main window displays the 'select an existing database or create a new SQL server database' dialog. The 'SQL Server Name' is set to 'VISTAULTRA64', 'Use Trusted Security' is checked, and the 'SQL Login Name' is 'sa'. The 'SQL Server Version' is 'Microsoft SQL Server 2005 - 9.00.3068.00 (X64) Feb 26 2008 23:02:54 Copyright (c) 1988-2005 Microsoft'. The '1. Test Your Login' button is highlighted. Below this, the '2. Create The Database' button is also highlighted, with the database name 'SalesAndAccountsDB' entered in the text box. A modal dialog box titled 'Database Has Been Created' is overlaid on the main window, displaying an information icon and the message 'Database Created SalesAndAccountsDB' with an 'OK' button.

Next we select the two databases, and specify the schemas which will be used.

Migrate Structure And Data | Full Database Auditing | Browse Data Dictionary | Migrate Queries | Web Conversion | Commissioning Dat

select SQL Server | Access databases | analyse Access data | review exclusions | migrate structure | migrate data | verify counts | row relink Access application | data and structure script files | error resolution | changes to design | reporting migration results | disp stat

**select Access databases containing data tables to upsize**

Name (a name by which you recognise this database)	Target Database Location (enter the full path, database name and file extension) eg. c:\test\test.mdb	DB Type	Assign To Schema (see notes below) 2005 Only	Do not upsize
NW_Contacts	C:\UsingSchemas\NW_Contacts.mdb	Application & Data	Accounts	<input type="checkbox"/>
NW_Orders	C:\UsingSchemas\NW_Orders.mdb	Application & Data	Sales	<input type="checkbox"/>
*		Application & Data		<input type="checkbox"/>

Next we analyse the databases identifying and correcting any design or data problems.

The screenshot shows a software interface with a menu bar containing 'Migrate Structure And Data', 'Full Database Auditing', 'Browse Data Dictionary', and 'Migrate Quer'. Below the menu is a toolbar with icons and labels: 'select SQL Server', 'Access databases', 'analyse Access data', 'review exclusions', 'migrate structure', 'migrate data', 'verify row counts', 'relink application', and 'data and str script file migration'. A descriptive text below the toolbar reads: 'reads your Access databases, checks for errors and allows auto-correction of co'. The main interface area includes a section 'Select A Database, Or Use Batch Mode To Process ALL Databases' with a dropdown menu showing 'NW\_Contacts' and a text field containing 'C:\UsingSchemas\NW\_Contacts.mdb'. To the right, there is a 'BATCH MODI (Process All Databases) c drop-down to make a selec'. Below this is a 'Perform Analysis' section with a button 'Analyse Access Database Tables' and a checked checkbox. To the right of this section is a 'change your analysi' section with text: 'At this point you can review l switching on/off table auditin Analysis'. A dialog box titled 'Access Analysed' is overlaid on the interface, containing an information icon and the text 'Access databases analysed, no errors', with an 'OK' button at the bottom.

We can also look in the data dictionary and see which schemas the tables have been placed in and make any required changes.

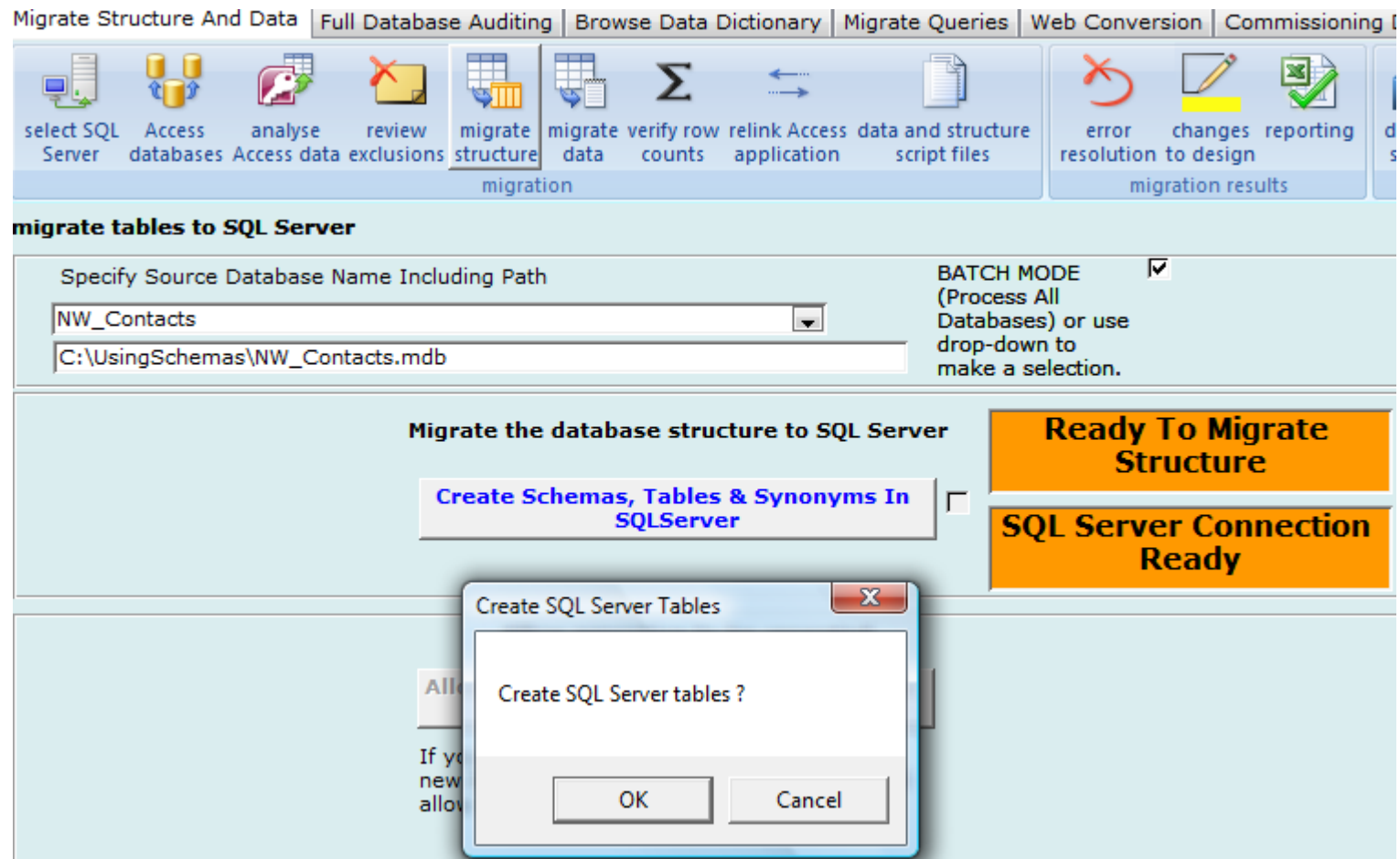
MUST also creates synonyms which make future translation of SQL very simple when working with schemas.

Migrate Structure And Data | Full Database Auditing | Browse Data Dictionary | Migrate Queries | Web Conversion | Commissioning Data | Reverse E

**view tables, resolve and conflicts resulting in changes to table names, review primary keys and assign tables to schemas**

	System Name	SourceTable Name	Target Table Name	Schema (NOT FOR SQL 2000!)	Synonym To Create	Row Count	TimeStamp Added
▶	NW_Contacts	Customers	Customers	Accounts	Customers	91	✓
	NW_Contacts	Employees	Employees	Accounts	Employees	9	✓
	NW_Contacts	Shippers	Shippers	Accounts	Shippers	3	✓
	NW_Contacts	Suppliers	Suppliers	Accounts	Suppliers	29	✓
	NW_Orders	Categories	Categories	Sales	Categories	8	✓
	NW_Orders	Order Details	Order Details	Sales	Order Details	2155	✓
	NW_Orders	Orders	Orders	Sales	Orders	830	✓
	NW_Orders	Products	Products	Sales	Products	77	✓

Next we return to the upsizing sequence and upsize the database structures.



Following this we can migrate the data, and add the indexes, checks and relationships.

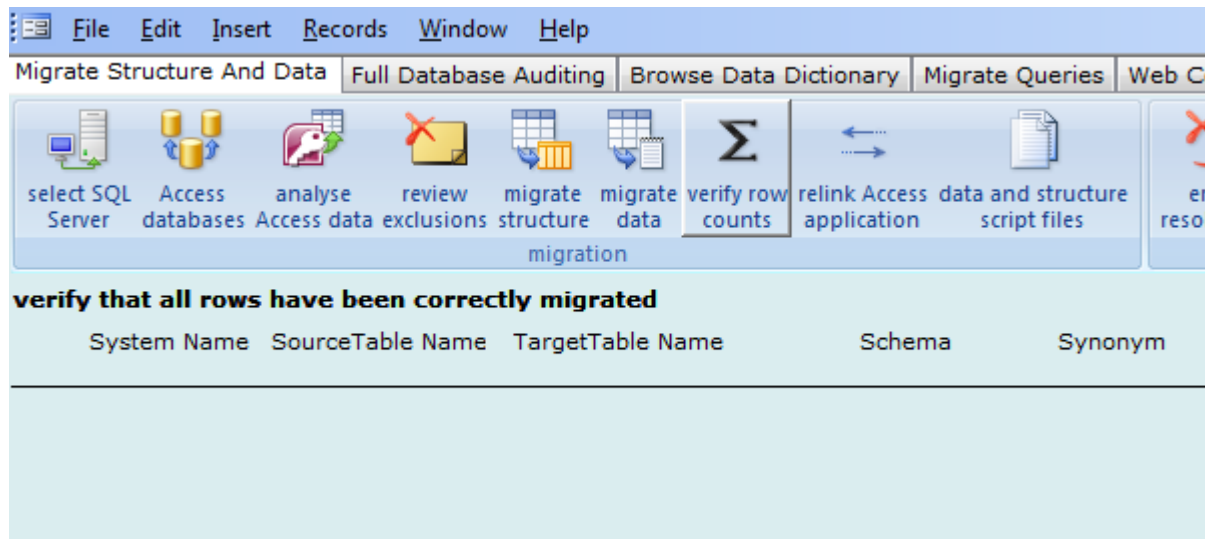
The screenshot displays the Microsoft Access Migration Wizard interface. The top ribbon includes tabs for 'Migrate Structure And Data', 'Full Database Auditing', 'Browse Data Dictionary', 'Migrate Queries', 'Web Conversion', and 'Commissioning'. The 'Migrate Structure And Data' ribbon is active, showing icons for 'select SQL Server', 'Access databases', 'analyse Access data', 'review exclusions', 'migrate structure', 'migrate data', 'verify row counts', 'relink Access application', 'data and structure script files', 'error resolution', 'changes to design', and 'reporting migration results'. The 'migrate data' icon is highlighted.

The main window is titled 'migrate the data to SQL Server'. It features a 'Specify Source Database Name Including Path' field with a dropdown menu showing 'NW\_Orders' and a text box containing 'C:\UsingSchemas\NW\_Orders.mdb'. A 'BATCH MODE' checkbox is checked. Below this, the 'Migrate Data/Indexes/Relationships' section contains three steps, each with a checked checkbox: '1. Migrate Data To SQL Server', '2. Create Indexing In SQL Server', and '3. Create Relationships/Checks'. To the right of these steps are two status boxes: a green box labeled 'Migration Completed' and an orange box labeled 'SQL Server Connection Ready'.

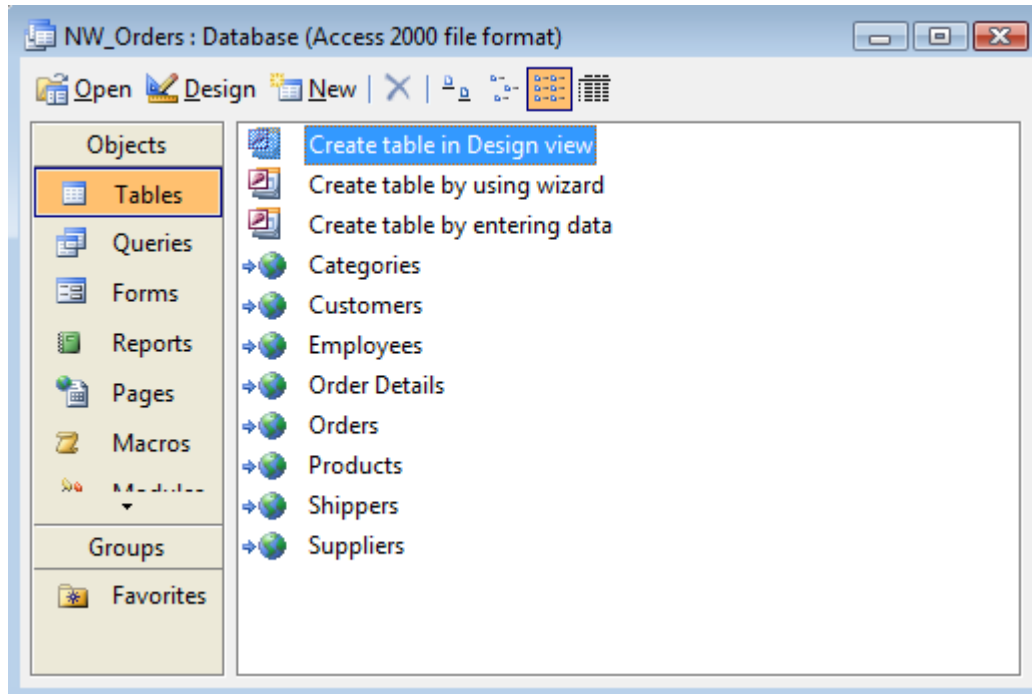
A dialog box titled 'SQL Server Relationships Created (BATCH MO...)' is overlaid on the screen. It contains an information icon and the text 'Relationships created', with an 'OK' button at the bottom right.

At the bottom of the wizard, there is a 'Remove' section with buttons for 'Drop Relationships', 'Drop', and 'Clear'.

The Verify Row Counts shows that no data has been missed, as this screen will only show data if some data has not been upsized.



Each Access application can then be re-linked to the SQL Server database.



We now create a security role for each department which allows user to only edit the data belonging to their department.

Migrate Structure And Data | Full Database Auditing | Browse Data Dictionary | Migrate Queries | Web Conversion | Commissioning Da

create database roles | manage users | update primary keys | commission update primary keys | review built in function library | display status | help | exit system

security | backflush primary keys | reference data | status

**using schemas and database roles greatly simplify security, first create a role and then assign permissions on the schema:**

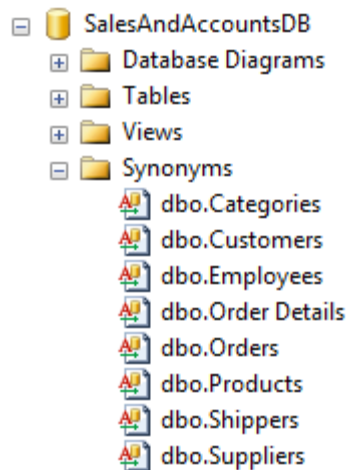
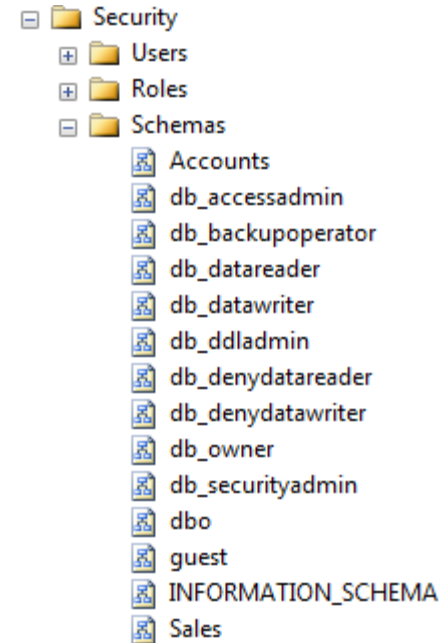
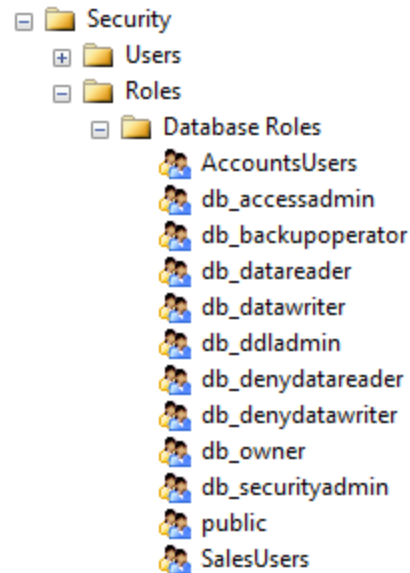
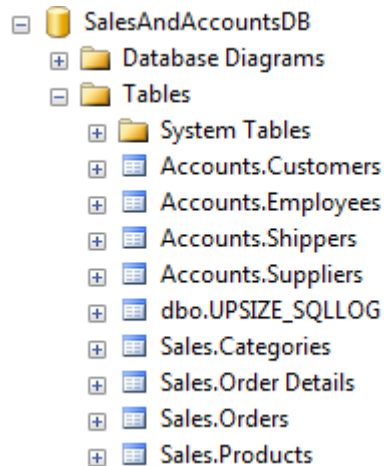
Database Role:  [Create ALL Database Roles And Assign Schema Permissions](#)

Schema	Can SELECT/VIEW Data	Can INSERT Data	Can DELETE Data	Can UPDATE Data	Can Execute Procedures
▶ Accounts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sales	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
*	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

**using schemas and database roles greatly simplify security, first create a role and then assign permissions on the schema:**

Database Role:  [Create ALL Database Roles And Assign Schema Permissions](#)

Schema	Can SELECT/VIEW Data	Can INSERT Data	Can DELETE Data	Can UPDATE Data	Can Execute Procedures
▶ Accounts	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Sales	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
*	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>



Schemas have allowed us to bring the two databases together, establish a solid security model and provide a clear naming convention for managing design work between the two departments.

We could also have created a third schema for shared objects. MUST would then simply have allowed selected tables to be placed in the shared schema.