

# **A Visual Guide to Microsoft Access Front-Ends with MySQL**

A MySQL<sup>®</sup> White Paper

**September 2009**

## Table of Contents

<b>Executive Summary</b> .....	<b>3</b>
<b>Why MySQL on Microsoft Windows?</b> .....	<b>3</b>
<b>What is Microsoft Office Access?</b> .....	<b>3</b>
<b>Why Access Front-Ends with MySQL?</b> .....	<b>3</b>
<b>Step 1: Install and Configure MySQL</b> .....	<b>4</b>
<b>Step 2: Configure the ODBC driver</b> .....	<b>5</b>
<b>Step 3: Export Northwind Tables and Data into MySQL</b> .....	<b>6</b>
<b>Step 4: Create Table Relationships</b> .....	<b>11</b>
<b>Step 5: Create a New Access Database</b> .....	<b>13</b>
<b>Step 6: Link the MySQL Tables to Access</b> .....	<b>14</b>
<b>Step 7: Import Queries, Forms, Reports, Macros and Modules</b> .....	<b>16</b>
<b>Step 8: Simple Application Tests</b> .....	<b>18</b>
<b>Step 9: Advanced Configuration</b> .....	<b>19</b>
<b>MySQL on Windows Case Studies</b> .....	<b>21</b>
<b>Conclusion</b> .....	<b>21</b>
<b>Resources</b> .....	<b>21</b>
<b>About MySQL</b> .....	<b>22</b>

## Executive Summary

For many years, Microsoft Windows has been the most popular development platform and second most popular production platform for organizations delivering MySQL applications. In early 2009 we conducted our annual survey and what we found is that 66% percent of those surveyed used Windows for development and 48% ultimately deployed into production on Windows. Given that so many users deploy MySQL on Windows for production, it makes sense to explore the possibility of leveraging Windows specific technologies in conjunction with MySQL. Many MySQL customers are migrating from Microsoft Access because they have concluded that the combination of enhanced scalability (both in terms of concurrent user load and overall data volume), cost-savings, platform freedom, and feature set of MySQL make for a compelling business case to offload some or all their Access applications to MySQL. In this paper we explore how to get started with leveraging Microsoft Access front-ends with MySQL.

## Why MySQL on Microsoft Windows?

MySQL continues to be an excellent choice on the Windows platform due to MySQL's:

- Lower TCO
- Ease of use
- Reliability
- Performance
- Fully featured database with no functional limitations

Windows related downloads at mysql.com continue to be strong for the MySQL server, tools and connectors averaging an astonishing 45,000 downloads per day during the first half of 2009.

## What is Microsoft Office Access?

**Microsoft Office Access**, previously known as **Microsoft Access**, is a relational database management system from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software development tools. It is a member of the Microsoft Office suite of applications and is included in the Professional and higher versions for Windows and also sold separately.<sup>1</sup>

## Why Access Front-Ends with MySQL?

Our 2009 customer and user survey showed that roughly 20% organizations make use of both MySQL and Access. However, we also found that these same organizations when asked if they are going to increase their usage, roughly 75% said they do not plan on increasing their deployments of Access while over 60% said that were planning on increasing their deployment of MySQL. One may surmise that within organizations familiar with both products, MySQL presents enough advantages to consider deploying more aggressively than Access.

Although not labeled a 'lite' database, Microsoft Access is generally targeted for low-end applications and as such, carries with it a number of restrictions and limitations that MySQL does not suffer from.

---

<sup>1</sup> [http://en.wikipedia.org/wiki/Microsoft\\_Access](http://en.wikipedia.org/wiki/Microsoft_Access)

The following is a list of a few of the more important weaknesses of Microsoft Access to consider:

- Access can at most support a 2GB database, MySQL can scale up to multi-terabytes.
- Access is not able to handle many concurrent users, MySQL can handle thousands of concurrent users.
- File databases such as Access do not take advantage of modern hardware with many CPU's or cores; MySQL makes use of the advances in today's hardware to deliver a high performance database server.
- In terms of data protection, if an Access database is open and/or users are accessing it, it cannot be backed up. MySQL offers many forms of backup options while users are connected to and using the database.

For more information concerning Access 2007 limitations, visit:

<http://blogs.msdn.com/access/archive/2006/06/05/access-2007-limits.aspx>

While Access is typically deployed in a simple desktop environment, oftentimes the database and/or application will grow and when it does, it is likely it will hit the limits mentioned above. Rather than start out with Access and then be forced to switch to another DBMS as time goes by, it is often times smarter to begin with a database like MySQL that future-proof's your application's needs.

## ***Scope and Purpose of this Guide***

In this guide we will cover the fundamentals on how to migrate the sample Northwind application included by default in Access 2007 to MySQL. As all migrations are different and present their own unique challenges, we encourage you explore these additional resources:

### **A Guide to Migrating From Microsoft Access to MySQL**

[http://www.mysql.com/why-mysql/white-papers/mysql\\_wp\\_migrate-from-access.php](http://www.mysql.com/why-mysql/white-papers/mysql_wp_migrate-from-access.php)

### **MySQL Forum: Migration from Microsoft Access**

<http://forums.mysql.com/list.php?65>

We should also consider the fact that in general Access will query MySQL tables as if they are Access tables, so there will special considerations to observe when writing new queries, or troubleshooting migrated ones.

## **Step 1: Install and Configure MySQL**

### ***Install MySQL 5.1***

First, we'll need to download a copy of the current version of MySQL, which can be obtained at:

<http://dev.mysql.com/downloads/mysql/5.1.html>

For details on how to install MySQL on Windows, go to:

### **A Visual Guide to Installing MySQL on Windows**

[http://www.mysql.com/why-mysql/white-papers/visual\\_guide\\_to\\_installing\\_mysql\\_windows.php](http://www.mysql.com/why-mysql/white-papers/visual_guide_to_installing_mysql_windows.php)

## Create a blank database

Using your favorite graphical MySQL administration tool or the MySQL command line client, create an empty database named “northwind”.

```
mysql> create database northwind;
```

## Step 2: Configure the ODBC driver

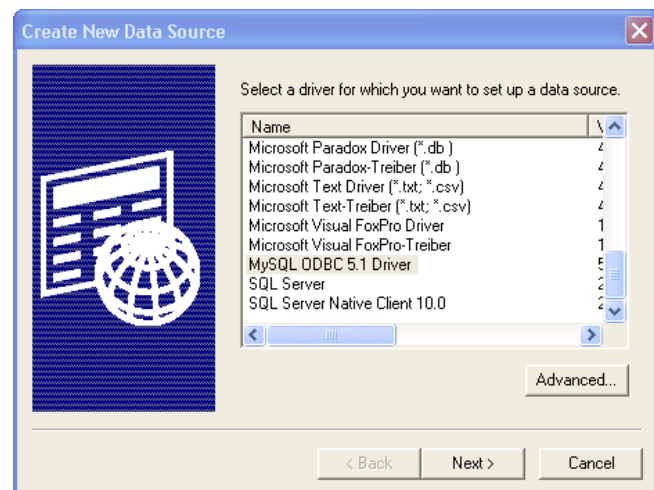
The MySQL ODBC connector provides connectivity for Access to MySQL. The current ODBC connector can be downloaded from:

<http://dev.mysql.com/downloads/connector/odbc/5.1.html#windows>

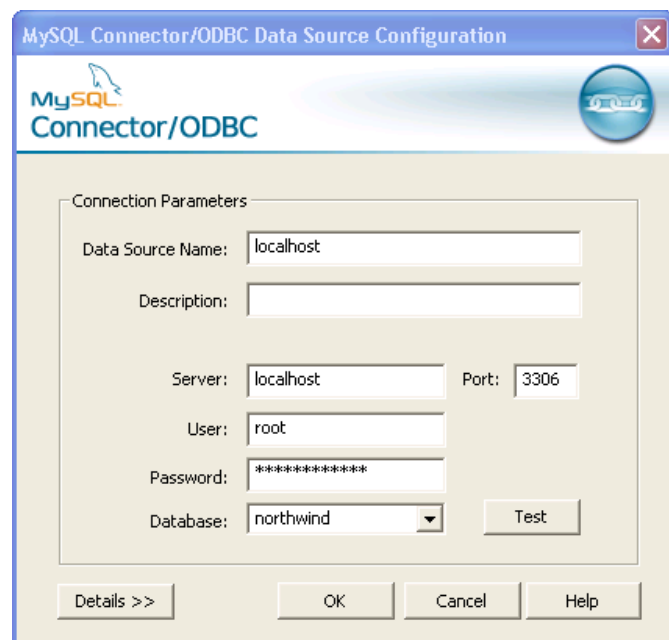
Run the installer. (For the purposes of this guide, a **typical** installation should suffice.)

Next, configure the **Microsoft ODBC Data Source Administrator**. Select the **File DSN** tab and click **Add**.

Scroll down and locate the **MySQL ODBC 5.1 Driver** and type a name for the .dsn file. In this case we used **localhost**.



Finally, configure the Connector/ODBC connection. Here we specify **localhost** as the server, supply login credentials and choose **northwind** as the database.

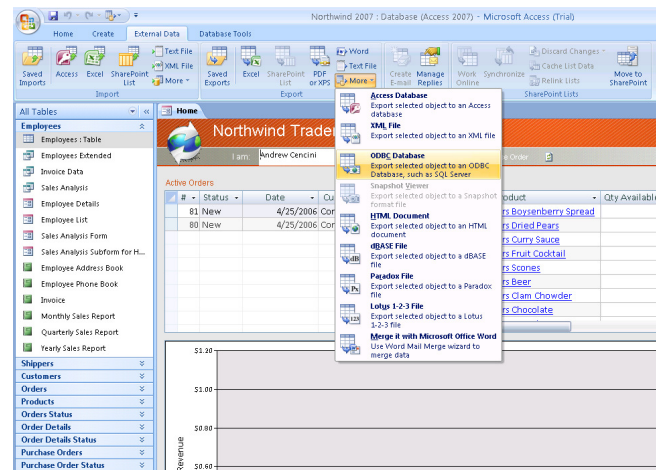


## Step 3: Export Northwind Tables and Data into MySQL

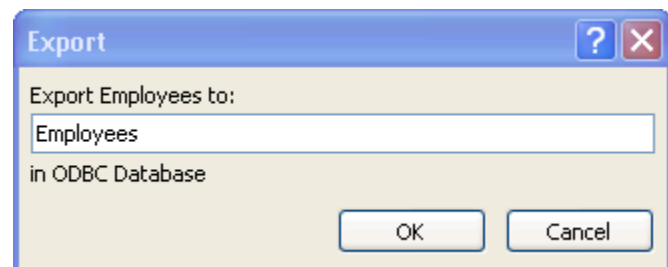
### Using Access for Single Table Migrations

For Access databases with a small number of tables, you can use the built-in object exporter inside of Access 2007. This wizard only allows for the export of tables one at a time, so it may not be practical for Access databases with many tables.

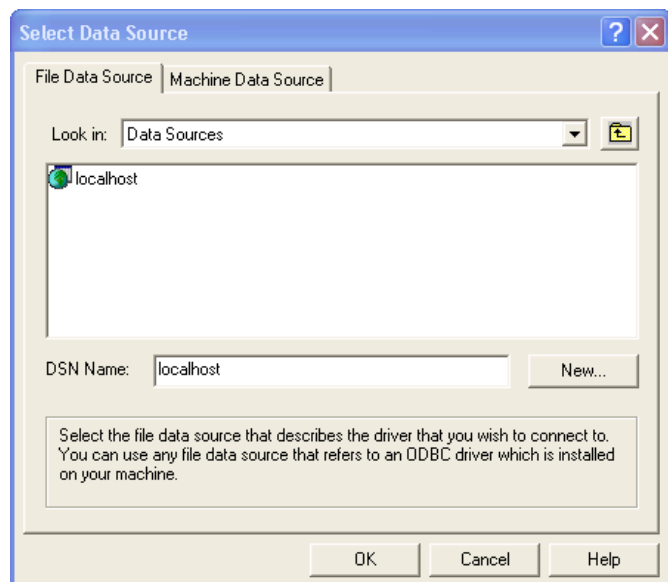
Select a table to export, in this example we chose the **employees** table. Next, click on the **External Data** tab and then select **More** in the **Export** group and select **ODBC Database**



Name the table we will be creating on the MySQL database, in this case we leave it the same, **employees**

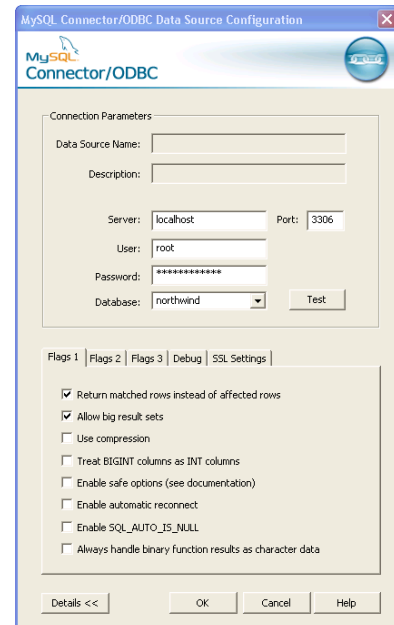


Select the ODBC connection to use for MySQL. In this case we use the previously configured **localhost**.



Depending on how you have configured your ODBC driver, you may also be able to create a connection via the Machine Data Source tab

Supply MySQL connectivity credentials, select the **northwind** database and select the appropriate flags after clicking on Details. For the purposes of our example we chose **Return matched rows instead of affected rows** and **Allow big result sets**. You may then choose to save your Export settings if desired



Verify that the table was created using the **SHOW TABLES** command

```
C:\> C:\WINDOWS\system32\cmd.exe - mysql -uroot -p
mysql> create database northwind;
Query OK, 1 row affected (0.00 sec)

mysql> use northwind;
Database changed
mysql> show tables;
+-----+
| Tables_in_northwind |
+-----+
| employees            |
+-----+
1 row in set (0.00 sec)
```

Verify that the data was successfully exported using the **SELECT COUNT(\*) FROM employees** command and comparing to the row count in Access

```
C:\> C:\WINDOWS\system32\cmd.exe - mysql -uroot -p
mysql> use northwind;
Database changed
mysql> select count(*) from employees;
+-----+
| count(*) |
+-----+
|          9 |
+-----+
1 row in set (0.00 sec)

mysql>
```

## MySQL Partner Solutions for Access Migrations

There are a variety of commercial and free tools available for more sophisticated migrations of Access databases to MySQL beyond the simple table by table process that Access natively supports. MySQL offers a free graphical tool called “Migration Toolkit” which is available for download at:

<http://dev.mysql.com/downloads/gui-tools/5.0.html>

Another good place to start in locating an appropriate solution, is MySQL’s Partner Solution pages located at:

<http://solutions.mysql.com/solutions/>

A partial list includes:

- DBConvert for MS Access and MySQL
- DBForms from MS Access to PHP + MySQL
- DBSync for Access and MySQL

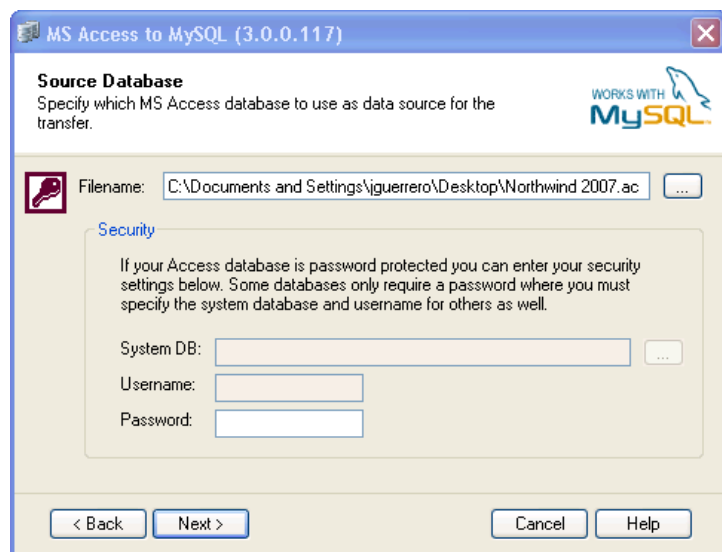
## Migrating Multiple Tables with Access to MySQL

For the purposes of this paper we have chosen to use the program Access to MySQL (Freeware) from Bullzip available for download at:

<http://www.bullzip.com/download.php>

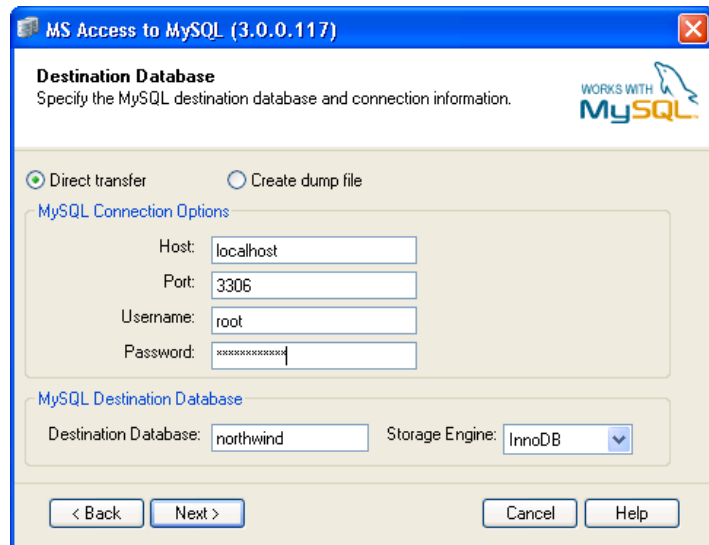
Although the workflow and features may differ depending on the tool you choose to use, of the several we tested, they all followed a very similar process.

Browse and **select** the **northwind.accdb** file (or .mdb if using an older version of Access)



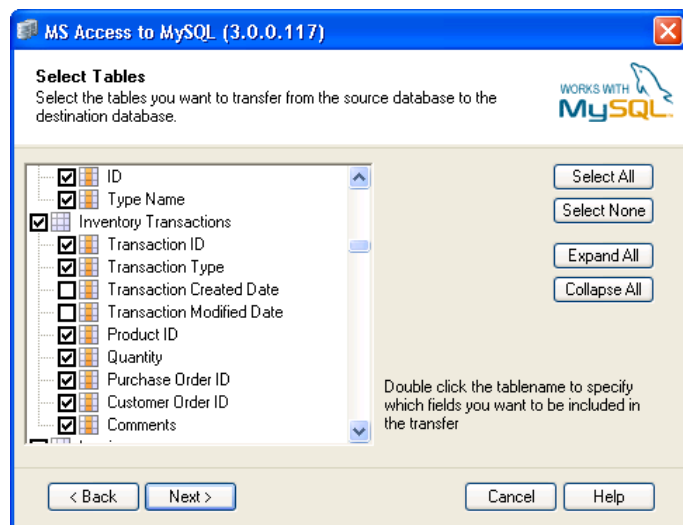


Select the target MySQL database, in this case **northwind**, provide connectivity credentials and choose either a **direct transfer** or the creation of a **dump file**



Next **select all tables but exclude the following columns** which we will create after the migration to resolve issues with default values<sup>2</sup>

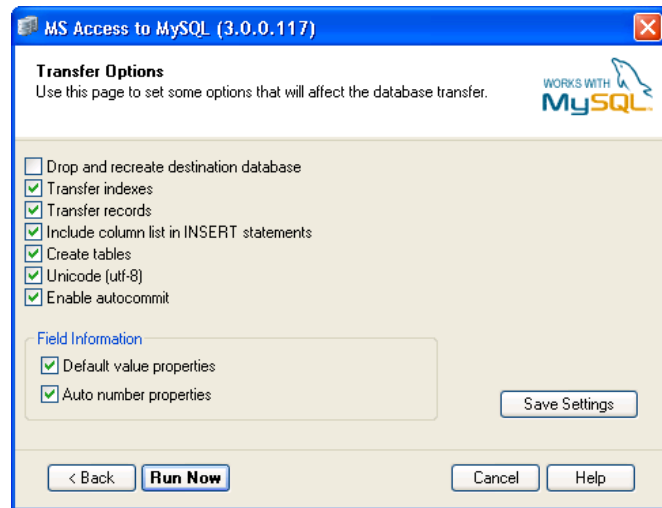
- **Inventory Transactions**  
*Transaction Created Date*  
*Transaction Modified Date*
- **Invoices**  
*Invoice Date*
- **Orders**  
*Order Date*
- **Purchase Orders**  
*Creation Date*



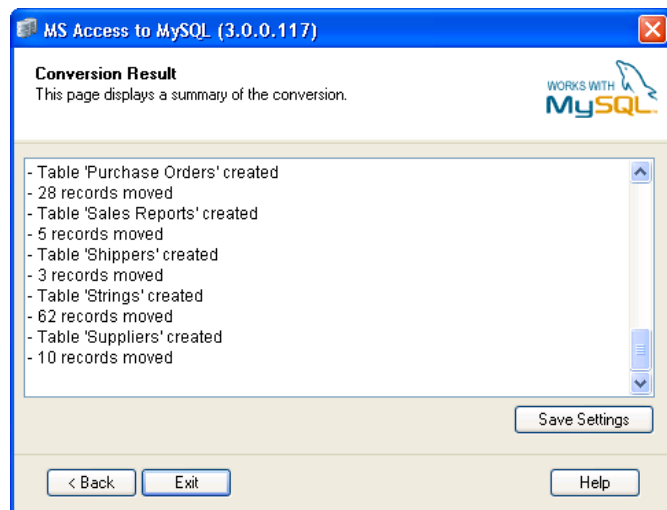
<sup>2</sup> The Access 2007 version of the sample Northwind database introduces an incompatibility concerning default values. Specifically, the DATETIME MySQL datatype does not accept the NOW() function as a default value. For more information concerning this issue please consult:

<http://dev.mysql.com/doc/refman/5.0/en/data-type-defaults.html>  
<http://bugs.mysql.com/bug.php?id=27645>

Select the appropriate migration options ensuring that indexes, relationships data and default values are selected



Examine the conversion results and exit



### **Add Missing Columns with Appropriate Default Values**

Execute the following statements to create the columns we previously excluded or create them using your favorite MySQL Administration GUI:

```
ALTER TABLE `northwind`.`inventory transactions`
ADD COLUMN `Transaction Created Date` TIMESTAMP DEFAULT '0000-00-00 00:00:00',
ADD COLUMN `Transaction Modified Date` TIMESTAMP DEFAULT NOW() ON UPDATE NOW();
```

```
ALTER TABLE `northwind`.`invoices`
ADD COLUMN `Invoice Date` TIMESTAMP DEFAULT NOW() NULL ;
```

```
ALTER TABLE `northwind`.`orders`
```

```
ADD COLUMN `Order Date` TIMESTAMP DEFAULT NOW() NULL ;

ALTER TABLE `northwind`.`purchase orders`
ADD COLUMN `Creation Date` TIMESTAMP DEFAULT NOW() NULL ;
```

At this point you may choose to verify that the table, row counts, indexes are correct using your favorite MySQL administration tool.

## Step 4: Create Table Relationships

The following relationships in the original Access Northwind database need to be created in MySQL's version of the Northwind database:

Foreign Key Table.Source Column	Primary Key Table.Target Column
employee privileges.employee ID	employees.ID
inventory transactions.Transaction Type	inventory transaction types.ID
order details.status ID	order details status.Status ID
inventory transactions.Customer Order ID	orders.Order ID
orders.Employee ID	employees.ID
order details.Order ID	orders.Order ID
orders.Customer ID	customers.ID
invoices.Order ID	orders.Order ID
orders.Status ID	orders status.Status ID
orders.Tax Status	orders tax status.ID
employee privileges.Privilege ID	privileges.Privilege ID
order details.Product ID	products.ID
inventory transactions.Product ID	products.ID
purchase order details.Product ID	products.ID
purchase order details.Inventory ID	inventory transactions.Transaction ID
purchase orders.Created By	employees.ID
inventory transactions.Purchase Order ID	purchase orders.Purchase Order ID
purchase order details.Purchase Order ID	purchase orders.Purchase Order ID
purchase orders.Status ID	purchase order status.Status ID
purchase orders.employees	employees.ID
orders.Shipper ID	shippers.ID
purchase orders.Supplier ID	suppliers.ID

You can use your preferred MySQL GUI tool or simply execute the DDL statements below to reestablish the relationships lost in the initial migration.

```
ALTER TABLE `northwind`.`employee privileges` ADD CONSTRAINT `FK_employee privileges`
FOREIGN KEY (`Employee ID`) REFERENCES `employees` (`ID`);

ALTER TABLE `northwind`.`inventory transactions` ADD CONSTRAINT `FK_inventory
transactions` FOREIGN KEY (`Transaction Type`) REFERENCES `inventory transaction
types` (`ID`);

ALTER TABLE `northwind`.`order details` ADD CONSTRAINT `FK_order details` FOREIGN KEY
(`Status ID`) REFERENCES `order details status` (`Status ID`);

ALTER TABLE `northwind`.`inventory transactions` ADD CONSTRAINT `FK_inventory
transactions_ord` FOREIGN KEY (`Customer Order ID`) REFERENCES `orders` (`Order ID`);

ALTER TABLE `northwind`.`orders` ADD CONSTRAINT `FK_orders_employees` FOREIGN KEY
(`Employee ID`) REFERENCES `employees` (`ID`);

ALTER TABLE `northwind`.`order details` ADD CONSTRAINT `FK_order details_orders`
FOREIGN KEY (`Order ID`) REFERENCES `orders` (`Order ID`) ON DELETE CASCADE ;

ALTER TABLE `northwind`.`orders` ADD CONSTRAINT `FK_orders_customers` FOREIGN KEY
(`Customer ID`) REFERENCES `customers` (`ID`);

ALTER TABLE `northwind`.`invoices` ADD CONSTRAINT `FK_invoices_orders` FOREIGN KEY
(`Order ID`) REFERENCES `orders` (`Order ID`) ON DELETE CASCADE ;

ALTER TABLE `northwind`.`orders` ADD CONSTRAINT `FK_orders_orders_status` FOREIGN KEY
(`Status ID`) REFERENCES `orders status` (`Status ID`);

ALTER TABLE `northwind`.`orders` ADD CONSTRAINT `FK_orders` FOREIGN KEY (`Tax Status`)
REFERENCES `orders tax status` (`ID`);

ALTER TABLE `northwind`.`employee privileges` ADD CONSTRAINT `FK_employee
privileges_privileges` FOREIGN KEY (`Privilege ID`) REFERENCES `privileges`
(`Privilege ID`);

ALTER TABLE `northwind`.`order details` ADD CONSTRAINT `FK_order details_products`
FOREIGN KEY (`Product ID`) REFERENCES `products` (`ID`);

ALTER TABLE `northwind`.`inventory transactions` ADD CONSTRAINT `FK_inventory
transactions_products` FOREIGN KEY (`Product ID`) REFERENCES `products` (`ID`);

ALTER TABLE `northwind`.`purchase order details` ADD CONSTRAINT `FK_purchase order
details_products` FOREIGN KEY (`Product ID`) REFERENCES `products` (`ID`);

ALTER TABLE `northwind`.`purchase order details` ADD CONSTRAINT `FK_purchase order
details_inventory_transactions` FOREIGN KEY (`Inventory ID`) REFERENCES `inventory
transactions` (`Transaction ID`);

ALTER TABLE `northwind`.`purchase orders` ADD CONSTRAINT `FK_purchase
orders_employees` FOREIGN KEY (`Created By`) REFERENCES `employees` (`ID`);

ALTER TABLE `northwind`.`inventory transactions` ADD CONSTRAINT `FK_inventory
transactions_purchase_orders` FOREIGN KEY (`Purchase Order ID`) REFERENCES `purchase
orders` (`Purchase Order ID`);

ALTER TABLE `northwind`.`purchase order details` ADD CONSTRAINT `FK_purchase order
details_purchase_orders` FOREIGN KEY (`Purchase Order ID`) REFERENCES `purchase
orders` (`Purchase Order ID`) ON DELETE CASCADE ;
```

```
ALTER TABLE `northwind`.`purchase orders` ADD CONSTRAINT `FK_purchase
orders_purchase_orders_status` FOREIGN KEY (`Status ID`) REFERENCES `purchase order
status` (`Status ID`);
```

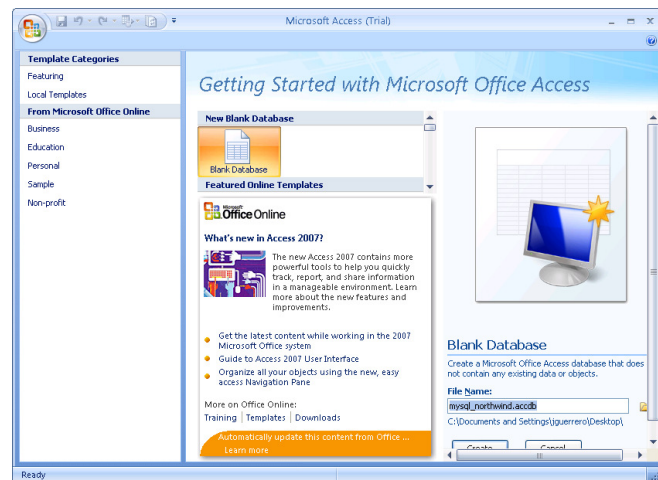
```
ALTER TABLE `northwind`.`purchase orders` ADD CONSTRAINT `FK_purchase
orders_employees_ID` FOREIGN KEY (`Created By`) REFERENCES `employees` (`ID`);
```

```
ALTER TABLE `northwind`.`orders` ADD CONSTRAINT `FK_orders_shippers` FOREIGN KEY
(`Shipper ID`) REFERENCES `shippers` (`ID`);
```

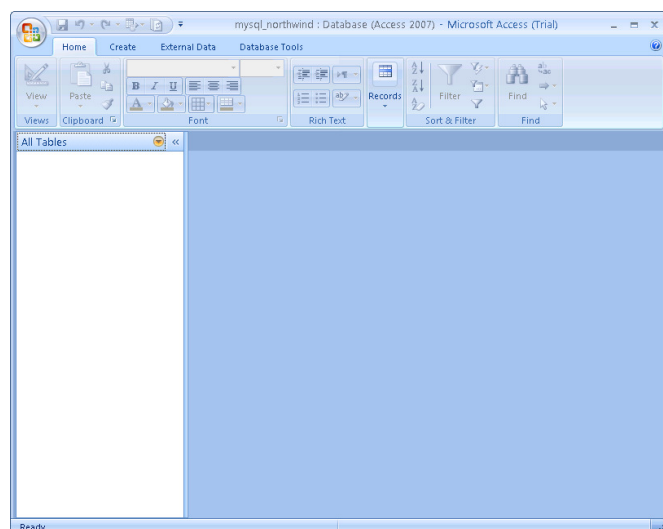
```
ALTER TABLE `northwind`.`purchase orders` ADD CONSTRAINT `FK_purchase
orders_suppliers` FOREIGN KEY (`Supplier ID`) REFERENCES `suppliers` (`ID`);
```

## Step 5: Create a New Access Database

Within Access create a new database. In this example we have named it **mysql\_northwind**

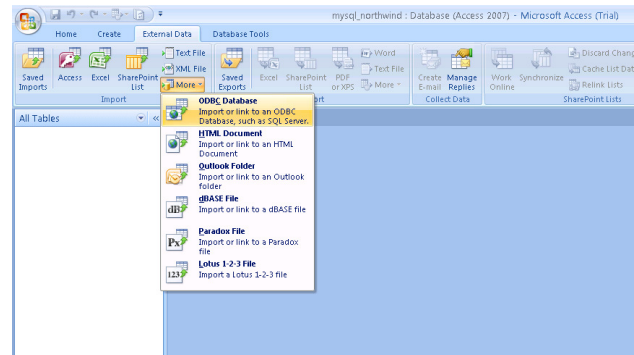


Remove any default objects so that the database is blank

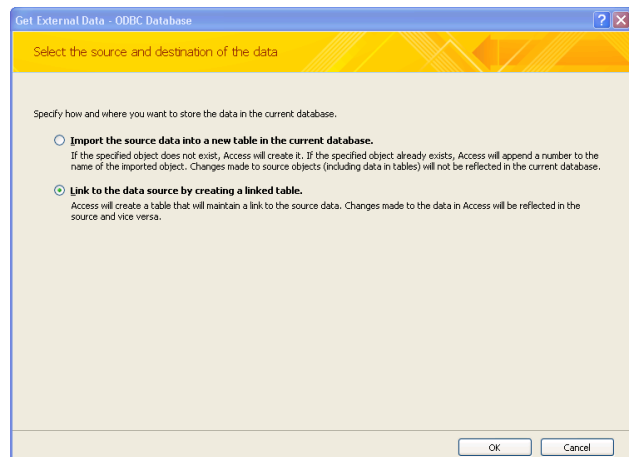


## Step 6: Link the MySQL Tables to Access

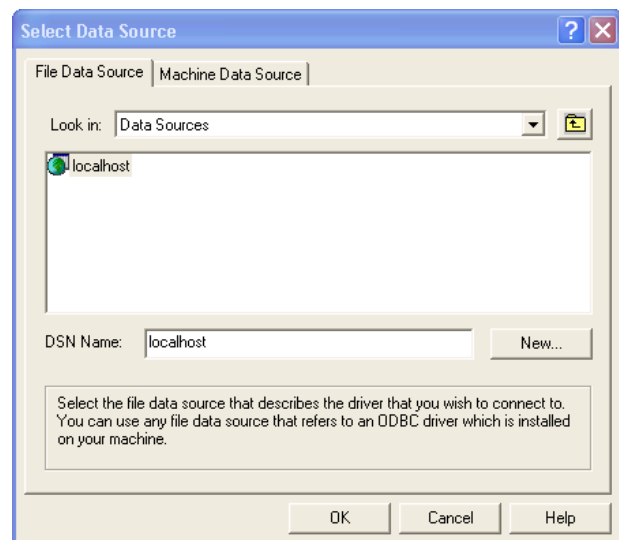
From the **External Data** tab select **Import** and **ODBC Database**



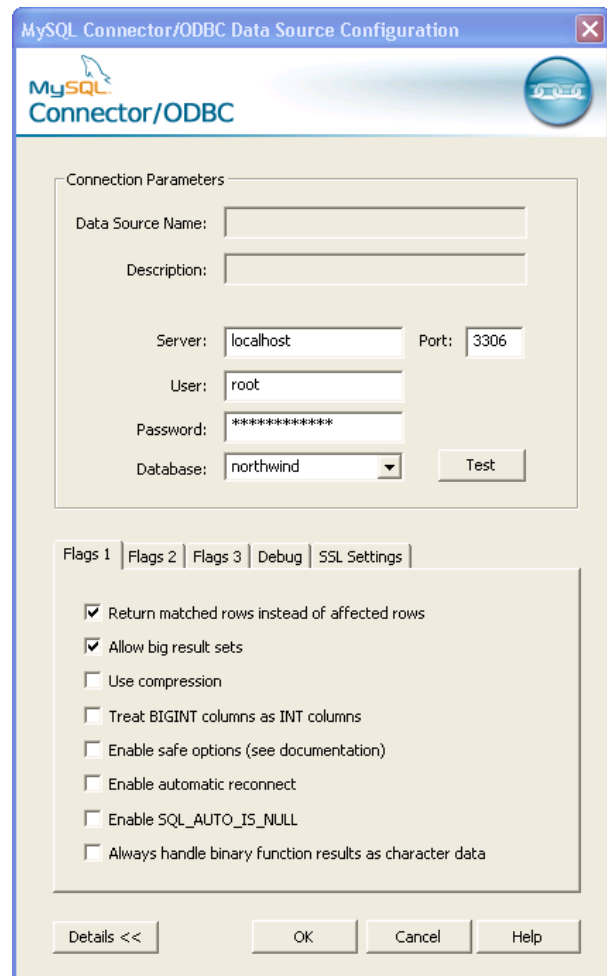
Select the **Link to the data source by creating a linked table** option



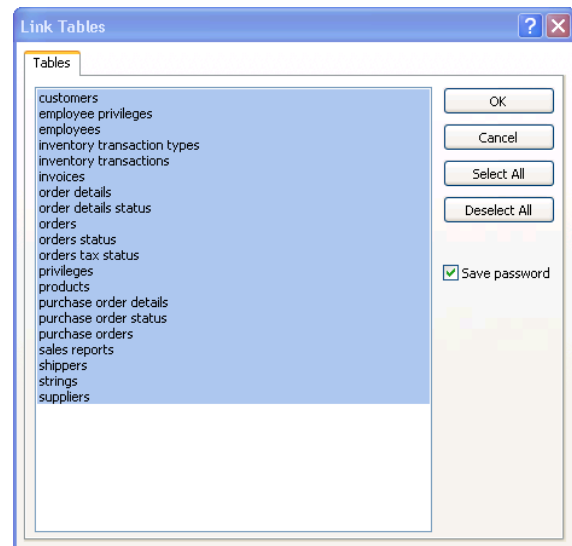
Select the **ODBC connection** to use for MySQL. In this case we use the previously configured **localhost**



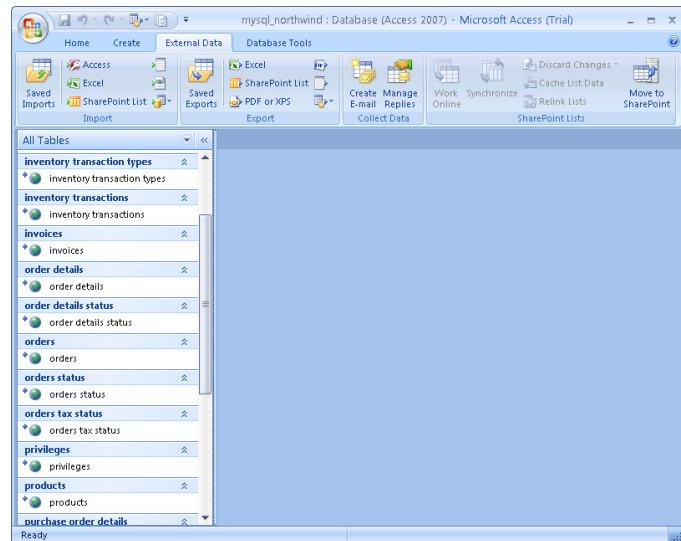
Supply MySQL connectivity credentials select the **northwind** database and select the appropriate flags. For the purposes of our example we chose **Return matched rows instead of affected rows** and **Allow big result sets**



Select all tables and check the **save password** option if desired (although less secure)

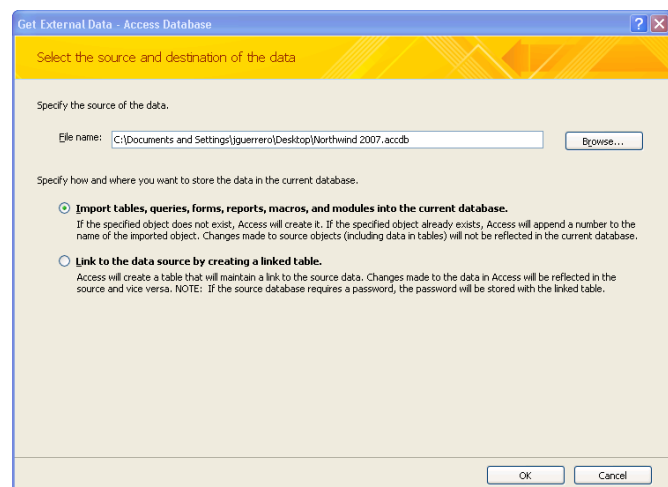


You should now see all the MySQL tables linked inside of Access. At this point you may choose to verify that the table and relationships are correct



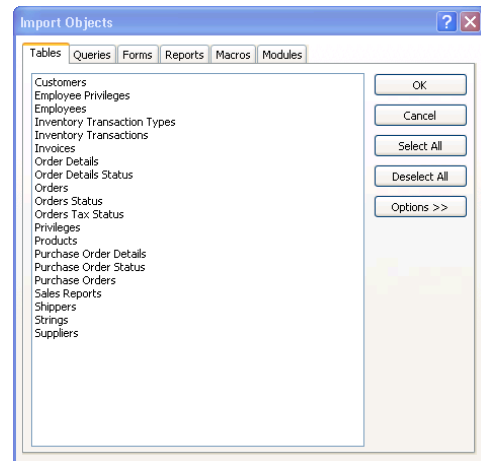
## Step 7: Import Queries, Forms, Reports, Macros and Modules

From the **External Data** tab select **Access** from the **Import** group. Next browse and select the original **Northwind 2007.accdb** file and the **Import tables, queries, forms, reports, macros, and modules into the current database** option

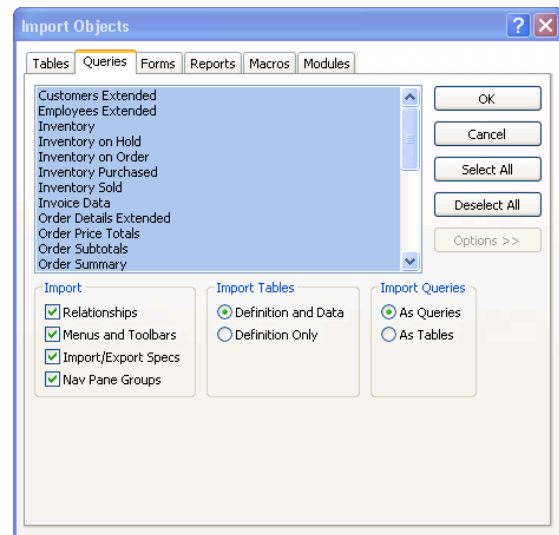




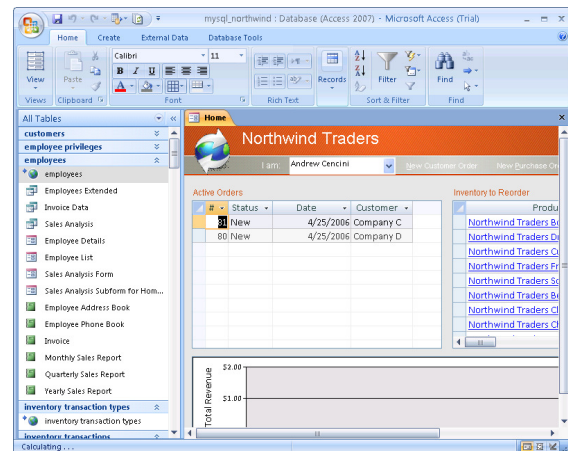
**Deselect all the tables** as we have already linked to them in MySQL in the previous step



On the **Queries** tab select all the queries and select the appropriate **Import** options. For our purposes we have selected all the import options. Similarly select all the **forms, reports, macros** and **modules** in the respective tabs. Save the import options if desired.



You should now see all the imported objects in Access. The grouping of the objects should look identical to the original Northwind database except that all the tables are now linked tables in MySQL



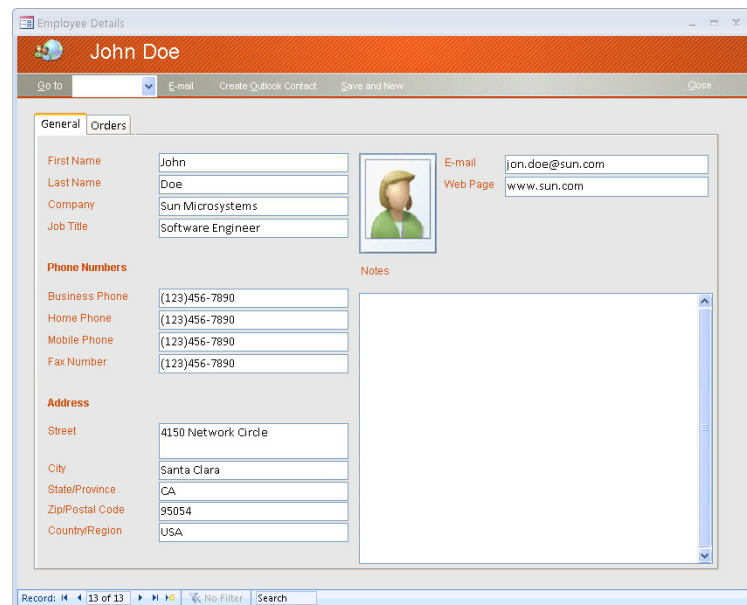
## Step 8: Simple Application Tests

### Testing

At this point you should verify that all your forms, queries and reports return the desired results and accept valid inputs.

### Example: Enter a new employee

As an initial test, **enter a new employee record** using the Employee Details form



Employee Details

John Doe

Go to: [ ] E-mail Create Outlook Contact Save and New Close

General Orders

First Name: John  
 Last Name: Doe  
 Company: Sun Microsystems  
 Job Title: Software Engineer

E-mail: jon.doe@sun.com  
 Web Page: www.sun.com

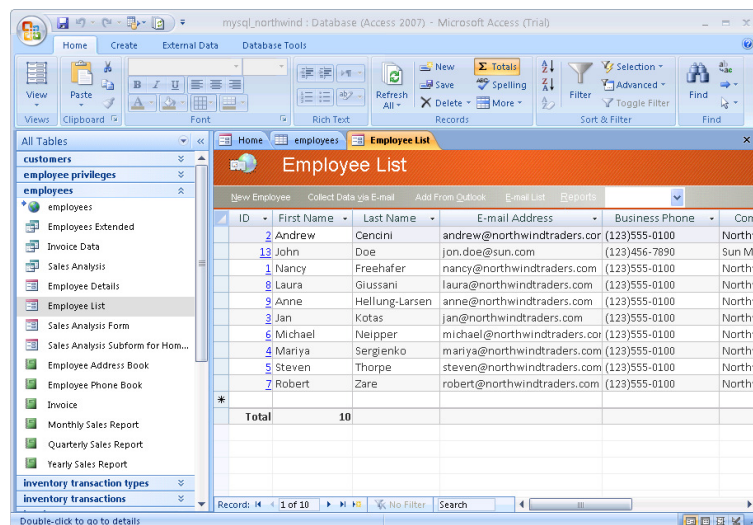
Phone Numbers  
 Business Phone: (123)456-7890  
 Home Phone: (123)456-7890  
 Mobile Phone: (123)456-7890  
 Fax Number: (123)456-7890

Address  
 Street: 4150 Network Circle  
 City: Santa Clara  
 State/Province: CA  
 Zip/Postal Code: 95054  
 Country/Region: USA

Notes

Records: 13 of 13

Double-click on the Employee List form. You should now see a total of 10 records



mysql\_northwind : Database (Access 2007) - Microsoft Access (Trial)

Home Create External Data Database Tools

View Paste Font Rich Text Refresh All Save Delete Spelling More Filter Selection Advanced Find

All Tables customers employee privileges employees Employee List

Employee List

ID	First Name	Last Name	E-mail Address	Business Phone	Com
2	Andrew	Cencini	andrew@northwindtraders.com	(123)555-0100	Northw
13	John	Doe	jon.doe@sun.com	(123)456-7890	Sun Mi
1	Nancy	Freehafer	nancy@northwindtraders.com	(123)555-0100	Northw
8	Laura	Giussani	laura@northwindtraders.com	(123)555-0100	Northw
9	Anne	Hellung-Larsen	anne@northwindtraders.com	(123)555-0100	Northw
3	Jan	Kotas	jan@northwindtraders.com	(123)555-0100	Northw
6	Michael	Neipper	michael@northwindtraders.com	(123)555-0100	Northw
4	Mariya	Sergienko	mariya@northwindtraders.com	(123)555-0100	Northw
5	Steven	Thorpe	steven@northwindtraders.com	(123)555-0100	Northw
7	Robert	Zare	robert@northwindtraders.com	(123)555-0100	Northw
Total		10			

Records: 1 of 10

We can also verify this data with a simple select statement from the MySQL client

```

C:\WINDOWS\system32\cmd.exe - mysql -uroot -p
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\jguerrero>mysql -uroot -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 131
Server version: 5.1.39-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current
statement.

mysql> use northwind;
Database changed
mysql> select count(*) from employees;
+-----+
| count(*) |
+-----+
|         10 |
+-----+
1 row in set (0.00 sec)

mysql>

```

## Step 9: Advanced Configuration

### *Datatype Mappings*

Depending on how you have chosen to migrate your tables and their definitions, there will likely be modifications which have been made to the original Access datatypes. You should verify that the datatype conversions that have been made are suitable for your application. Below are some recommended Access to MySQL datatype mappings.

Access Datatype	MySQL Suggested Datatype
AUTONUMBER	AUTO_INCREMENT
BINARY(SIZE)	BINARY
BYTE	TINYINT
CURRENCY	NUMERIC OR DECIMAL
DATE	DATE OR DATETIME
DECIMAL	DECIMAL
DOUBLE	FLOAT
GUID	IDENTITY COLUMN
INTEGER	SMALLINT
LONGBINARY	VARBINARY
LONG INTEGER	INT OR BIGINT
MEMO	VARCHAR(SIZE) OR TEXT
SINGLE	REAL
TEXT	VARCHAR(SIZE)
YESNO	TINYINT

## Function Mappings

As with the migration of datatypes, you should also verify that any functions that have been migrated are suitable for your application.

Below is a table containing popular Access functions and their suggested MySQL equivalents.

Access Function	MySQL Suggested Function
asc	ascii
ccur	convert(decimal)
cdbl	convert(float)
chr	char
chr\$	char
cint	convert(smallint)
clng	convert(int)
csng	convert(real)
cstr	convert(varchar)
cvdate	convert(datetime/date)
date	convert or cast
day	day
hour	hour
int	floor
lcase	lower
lcase\$	lower
len	length
ltrim\$	ltrim
mid	substr or substring
mid\$	substr or substring
month	month
now()	now()
minute	minute
rtrim\$	rtrim
right\$	right
sgn	sign
second	second
space	space
time()	time
str\$	strcmp
ucase	upper
ucase\$	upper
weekday	dayofweek
year	year

## MySQL on Windows Case Studies

Below are some examples of MySQL customers realizing lower TCO by running MySQL on Windows.

### ***Adobe Relies on MySQL to Make Creative Professionals More Productive***

Adobe Systems is one of the largest software companies and is the leading provider of creative tools for print, web, interactive, mobile, video and film. Adobe embeds MySQL into several Adobe Creative Suite 3 components, including Adobe Acrobat CS3, Adobe® Bridge CS3, and Adobe® Version Cue® CS3 so that workgroups can work more efficiently on complex projects. For more information, please visit:

[http://www.mysql.com/why-mysql/case-studies/MySQL\\_CaseStudy\\_Adobe.pdf](http://www.mysql.com/why-mysql/case-studies/MySQL_CaseStudy_Adobe.pdf)

### ***NetQoS Delivers Distributed Network Management Solution with Embedded MySQL***

NetQoS delivers products and services that enable some of the world's most demanding enterprises to improve network performance. American Express, Barclays, Boeing, Chevron, Cisco, Citrix, DuPont, Sara Lee, and Schlumberger are among the corporations that rely on NetQoS performance management solutions to ensure consistent delivery of business critical applications, monitor application service levels, troubleshoot problems quickly, contain infrastructure costs, and manage user expectations. To find the right embedded database solution to fit its innovative product architecture, NetQoS evaluated everything from flat-files to proprietary databases. NetQoS found that MySQL provided the ideal combination of performance, reliability, and ease of administration on Windows. For more information, please visit:

<http://www.mysql.com/why-mysql/case-studies/mysql-netqos-casestudy.pdf>

For a complete list of case studies and other resources concerning organizations making use of MySQL on Windows, please visit:

<http://www.mysql.com/customers/operatingsystem/?id=109>

## Conclusion

In this paper we presented a guide for getting started with migrating an existing Access application to use MySQL as a back end database. Although every migration presents unique challenges, migrating from Access to MySQL holds many benefits for administrators and users alike. Many parts of the migration process can be made easier by understanding the limitations and functional equivalents of each product. In conclusion, many MySQL customers are migrating from Microsoft Access because they have reached the conclusion that the combination of enhanced scalability (both in terms of concurrent user load and overall data volume), cost-savings, platform freedom, and feature set of MySQL make for a compelling business case to offload some or all their Access applications to the MySQL database server.

## Resources

### ***White Papers***

<http://www.mysql.com/why-mysql/white-papers/>

## ***Case Studies***

<http://www.mysql.com/why-mysql/case-studies/>

## ***Press Releases, News and Events***

<http://www.mysql.com/news-and-events/>

## ***Live Webinars***

<http://www.mysql.com/news-and-events/web-seminars/>

## ***Webinars on Demand***

<http://www.mysql.com/news-and-events/on-demand-webinars/>

## **About MySQL**

MySQL is the most popular open source database software in the world. Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems and packaged software -- including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube and Zappos.com. At <http://www.mysql.com>, Sun provides corporate users with commercial subscriptions and services, and actively supports the large MySQL open source developer community.

To discover how Sun's offerings can help you harness the power of next-generation Web capabilities, please visit <http://www.sun.com/web>.