

## Part 12. Using MS Access 2010 as a frontend to Oracle Database 11gR2 backend via ODBC

Last Updated on Thursday, 14 March 2013 00:44

Published on Wednesday, 13 March 2013 23:40

Written by Paweł Wawrzyniak

Hits: 12808

 SHARE   

We've already finished all the basic installation and configuration tasks on both: server side, where we have a working Oracle Database 11gR2 on Solaris 11.1; and client side, where we installed MS Windows 8 and Oracle Database 11gR2 Client software. This means we're ready to install – any – Microsoft Office package, as long as it is equipped with MS Access. In my case I installed MS Office 2010 Professional, which could be downloaded for trial purposes at the time I was preparing materials for this tutorial. So, let's go with the ODBC (Open DataBase Connectivity) part.

First of all, small assumption – MS Office 2010 Professional installation is very easy and we will skip this process. Therefore, you should install MS Office on your own here. Moreover, one can use any modern version of MS Office – the differences between MS Access versions available in MS Office 2007, 2010 and (even) 2013 are not big enough. You'll easily get the idea behind, so this tutorial is rather quite universal.



Secondly, we've installed Oracle Database 11gR2 Client on MS Windows 8 (see – [part 9](#)). You can use any other MS Windows version here, as long as Oracle Database 11gR2 Client is installed and properly configured to connect with the Oracle Database server. So, I can recommend to at least refer to [part 10](#) and [part 11](#) of this article.

Especially, one should refer to the [part 11](#) to be more than sure that his tnsnames.ora file is correctly composed – which enables Oracle Database 11gR2 Client to use TNS interface for connections with the Oracle Database server. This is crucial, because available ODBC drivers for Oracle Database require TNS interface.

Thirdly, there are two available ODBC drivers for Oracle Database. One is provided with MS Windows and was developed by Microsoft. Personally, I don't have too much experience with this driver. I only remember that once upon a time I had some stability and performance issues with Microsoft-made ODBC driver for Oracle. These were gone when I decided to go with the original, Oracle-made ODBC driver. So, I don't want to argue which driver is better (and for what), but my opinion is simple: as long as Oracle provides their own ODBC drivers for their Oracle Database product, I'm not interested in using any – third-party in this case – drivers. Also, I have read on [OTN](#) that MS ODBC driver for Oracle uses an old Oracle 7 version of the OCI (Oracle Call Interface) API, which has been obsolete for about 15 years. This can easily explain the stability and performance issues we can have with newer versions of the Oracle client (or the database).

Fourthly, MS Access 2010 (and in fact all versions since MS Access 97) is not "database" – in the common understanding of this term. MS Access is – so called – RDBMS (Relational DataBase Management System), which helps not only in storing data, but also in database development, frontend applications development (forms to control the data, reports, sophisticated queries), data analysis and – more generally – widely understood data manipulation. This tool can be divided into two parts:

- Microsoft ACE database engine (formerly known as MS JET database engine) – this is database engine, which is very good for smaller databases. However, it is not aimed to be competition for other database engines, like innodb or MyISAM from Oracle MySQL, Oracle Database engine, PostgreSQL, MS SQL Server etc. MS ACE is compact and flexible, but has some significant limitations. However, we can easily migrate data stored by MS Access in its own ACE database engine to – practically – any other database, as long as appropriate ODBC drivers are available. The obvious choice seems to be MS SQL Server, which in fact is very commonly used as a backend database for MS Access applications (frontends), however, we should value MS Access for its connectivity features. This is really very open and quite scalable technology – we will see an example here, because our backend database for MS Access is Oracle Database 11gR2 (in fact – previously installed on Solaris 11.1 server).
- Microsoft Access database development and data manipulation tools – here we have a possibility to create forms for data manipulation, reports, queries (SQL statements), macros (for automation purposes) and VBA (Visual Basic for Applications) code (stored in modules – this gives MS Access a real power in hands of experienced developer). We can also use additional technologies like ActiveX components, which give interesting options when we have to extend the basic features of our applications (please note, we can write our own ActiveX components in – for example – C++ or use thousands of components provided by third-parties). Also, DLL (Dynamically Linked Libraries) can be used from the VBA level – this gives even more power, as we can write our own libraries in almost any other language available for MS Windows platform, to enhance the basic MS Access application functionality. What else? Since MS Access 2007 we can integrate our applications with MS Sharepoint (previously, in MS Access 2003 there were – so called – Access Data Pages, ADPs, which allowed us to create very simple web applications). Additionally, MS Access gives us lots of options when it comes to data export or import between many data sources (not only database tables). All these features can be used with or without help of – so called – wizards (prepared for non-professionals).

Finally, I don't have to cover the fact, that I like MS Access very much. However, I was skeptical about this tool – as many other IT guys – some years ago. Therefore, please don't get me wrong. I appreciate all the strengths of MS Access, but I know its weaknesses, too (let's leave them for dedicated article). If someone still cannot understand what the MS Access is, maybe we can use a kind of generalized comparison: "in the world of Microsoft technologies, MS Access is like Oracle Forms, integrated with Oracle Reports, oriented for smaller database projects, simplified in some areas (VBA, Sharepoint integration, many wizards etc.), enhanced with its own database engine (JET or ACE) and an option to connect with any ODBC-ready database backend (including Oracle Database and Oracle MySQL), together with very easy data import/export capabilities between many data sources".

Yes, I know! That's said like in advertisement, but I hope you get my idea.

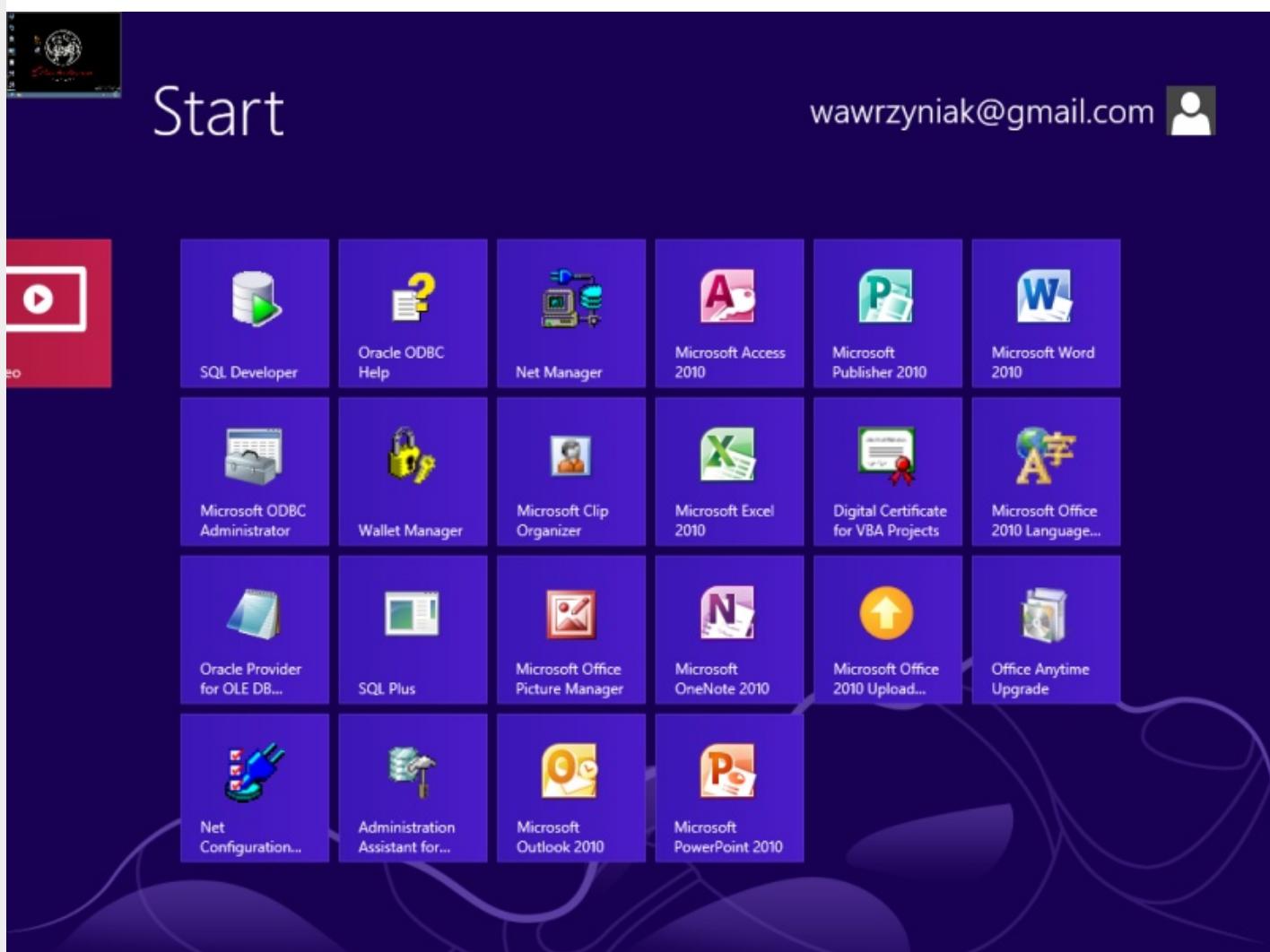
Since we clarified all the initial assumptions and doubts, we can start with the short experiment – we will create ODBC connection to the Oracle Database 11gR2 and use it from MS Access to:

- Link MS Access 2010 (frontend) with the Oracle backend database – we will use hr schema for this case. This scenario is quite common when we have to work with the data from external database or we want to quickly develop frontend application (for example – for the prototype purposes).
- Import all hr schema tables from Oracle to the MS Access – to process them locally with the usage of MS ACE engine. This scenario is quite popular among data analysts, who sometimes have to import a part of data from external database for local processing.

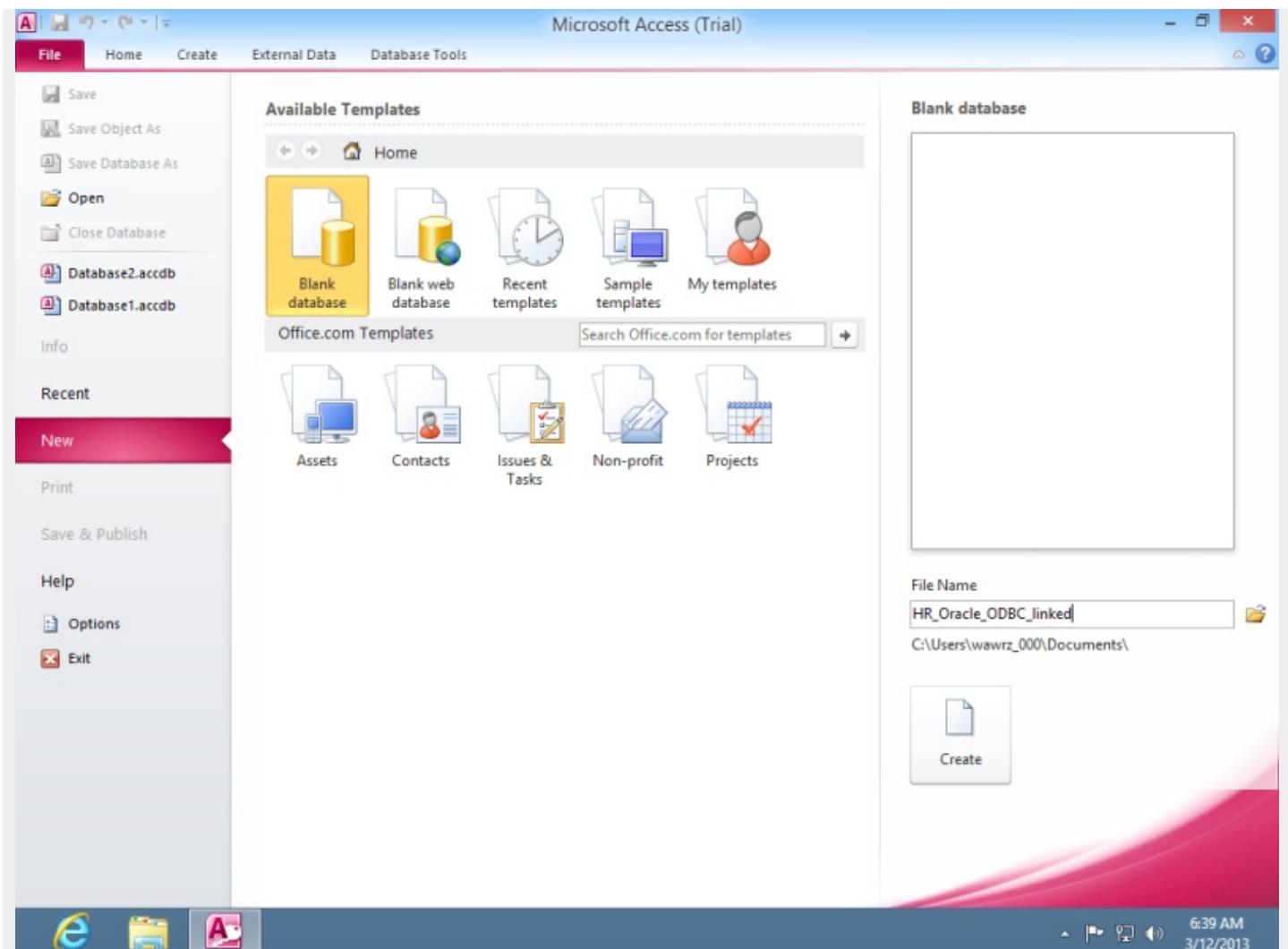
So, here it goes.

### Scenario 1. Link MS Access 2010 (frontend) with the Oracle backend database

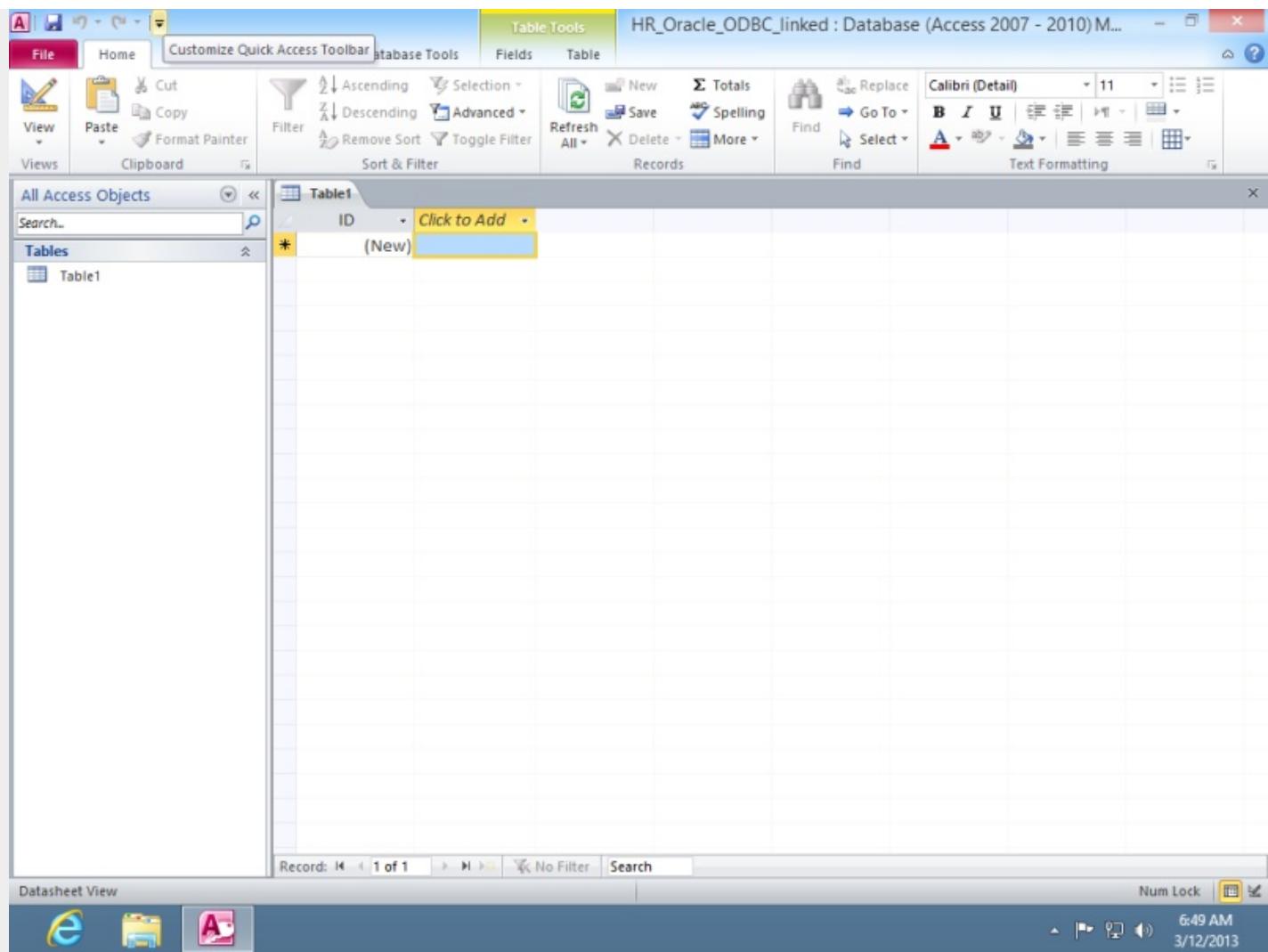
Let's start MS Access – in my case I will use this highly-sophisticated MS Windows 8 metro-style Start menu (smartphone-like):



Once MS Access is started, we should select "Blank database", provide a filename (I propose: "HR\_Oracle\_ODBC\_linked") for our database and click "Create" button:

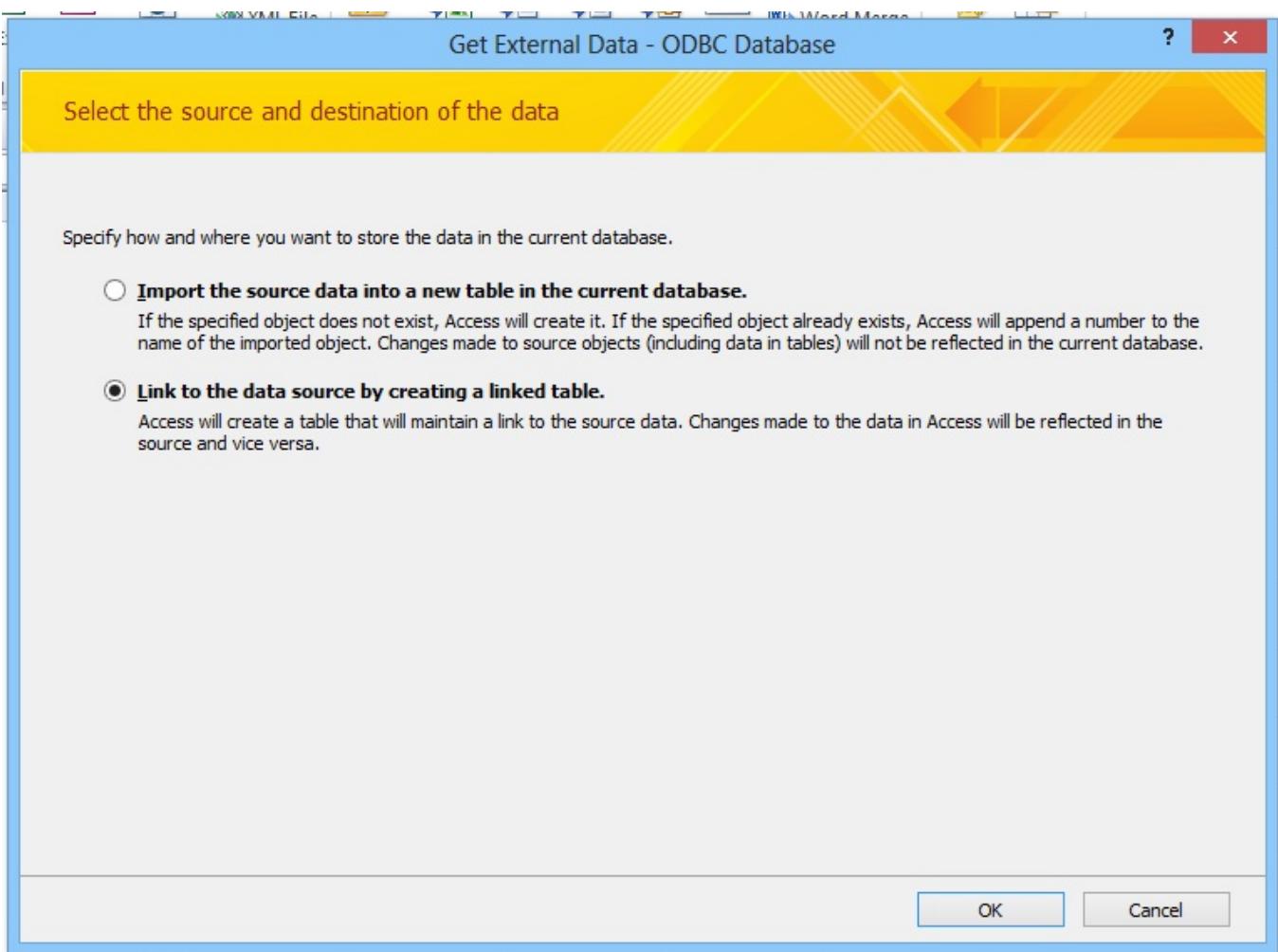


A new database will be created (we can see there is one local table in edit mode):

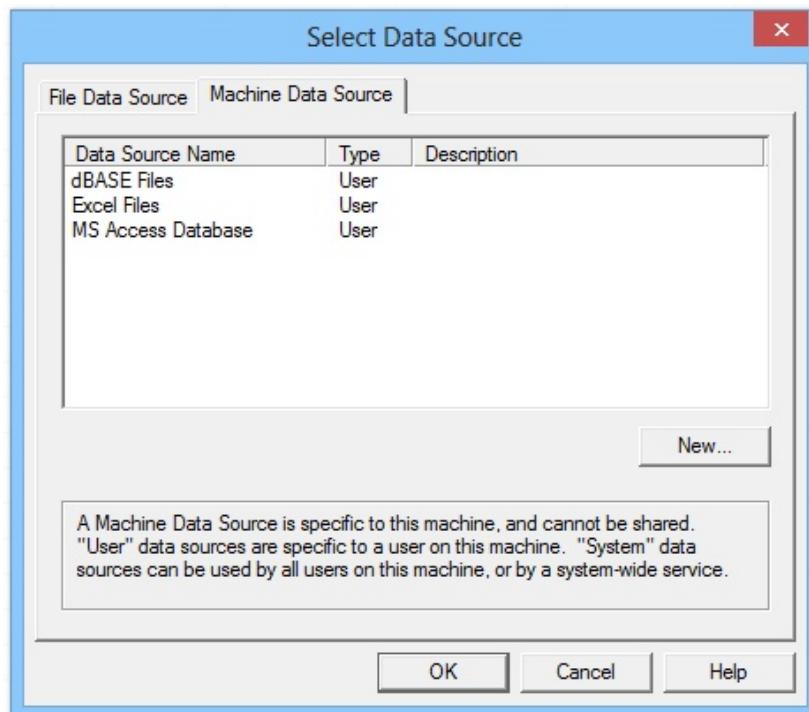


Then, we should go to the "External data" tab on the ribbon menu and select "ODBC database" button – "Get External Data – ODBC Database" window will be displayed with the two available options (we should select the second one):

- Import the source data into a new table in the current database – this will be used in second scenario.
- Link to the data source by creating a linked table – this is what we want to do. Select this option.



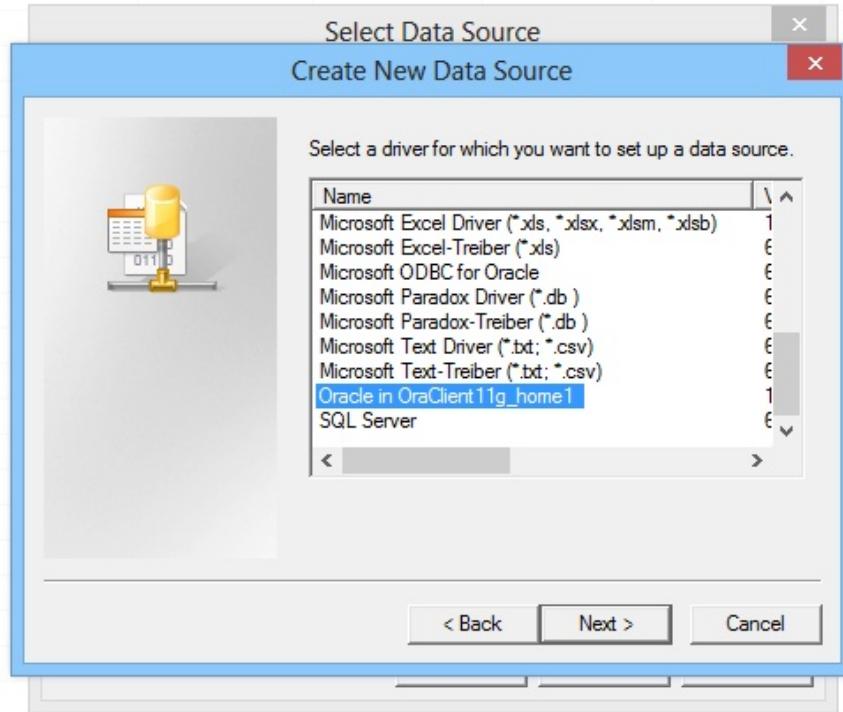
Then click "OK" button – there will be "Select data source" window displayed – in my case it is empty, as there are no data sources defined in the system. We should change the tab from "File data source" to "Machine data source" – like this:



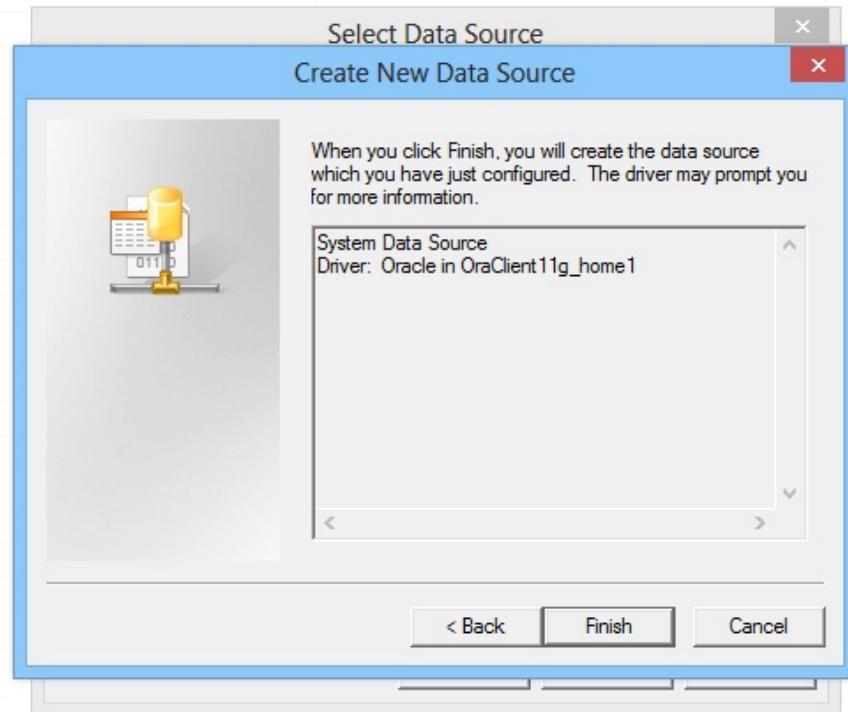
Click "New" button to create new data source for our Oracle database – a new windows will be displayed:



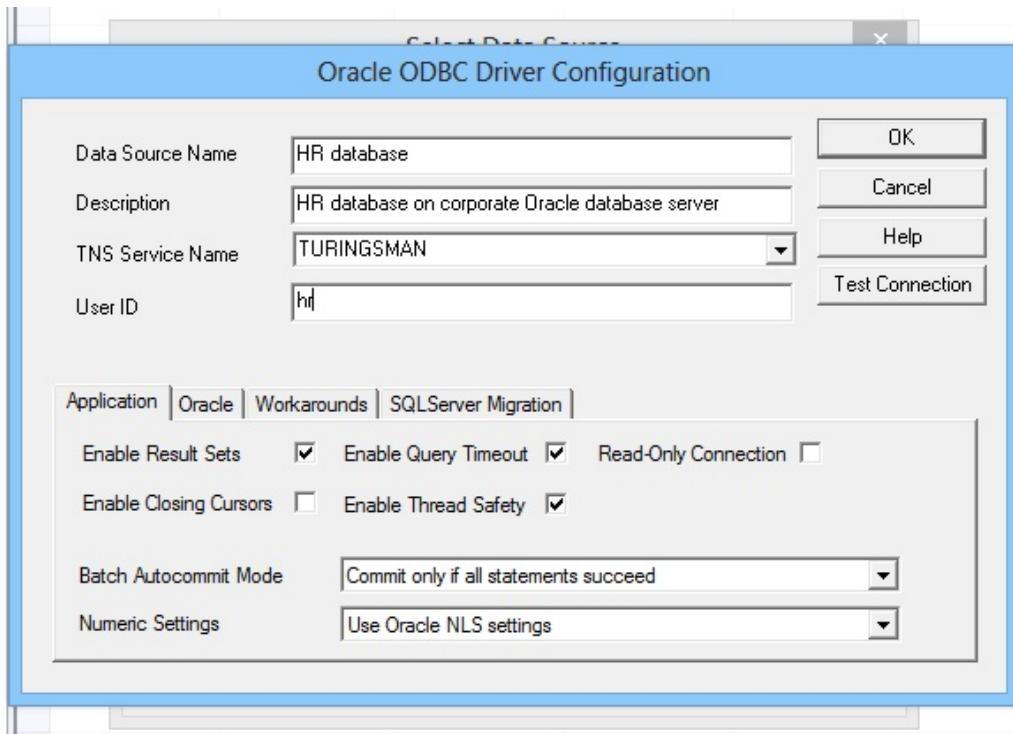
Select "System Data Source (Applies to this machine only)" instead of "User Data Source (Applies to this machine only)" which is selected by default. Then click "Next" button and select Oracle in "OraClient11g\_home1" – or similar name in your case – to choose Oracle-made ODBC driver (be aware – don't use Microsoft-made ODBC driver here):



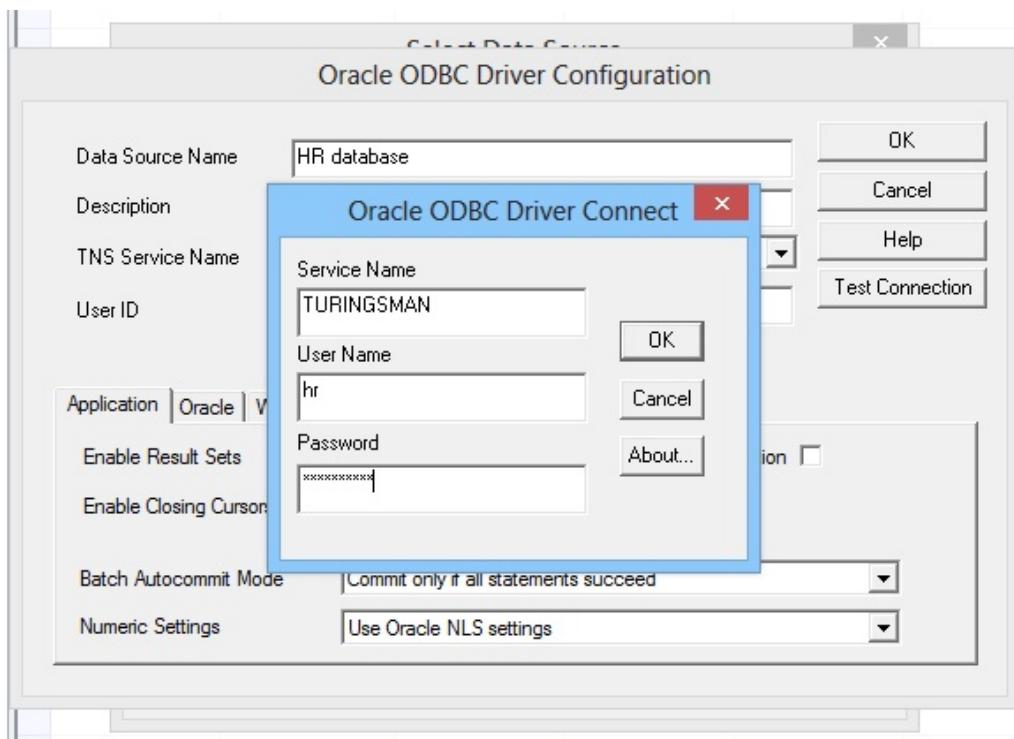
Once selected and double-checked it is in fact Oracle ODBC driver (which was installed during [part 9](#) of this series together with the rest of Oracle Database 11gR2 Client software), we can proceed with "Next" button – a summary windows will be shown:



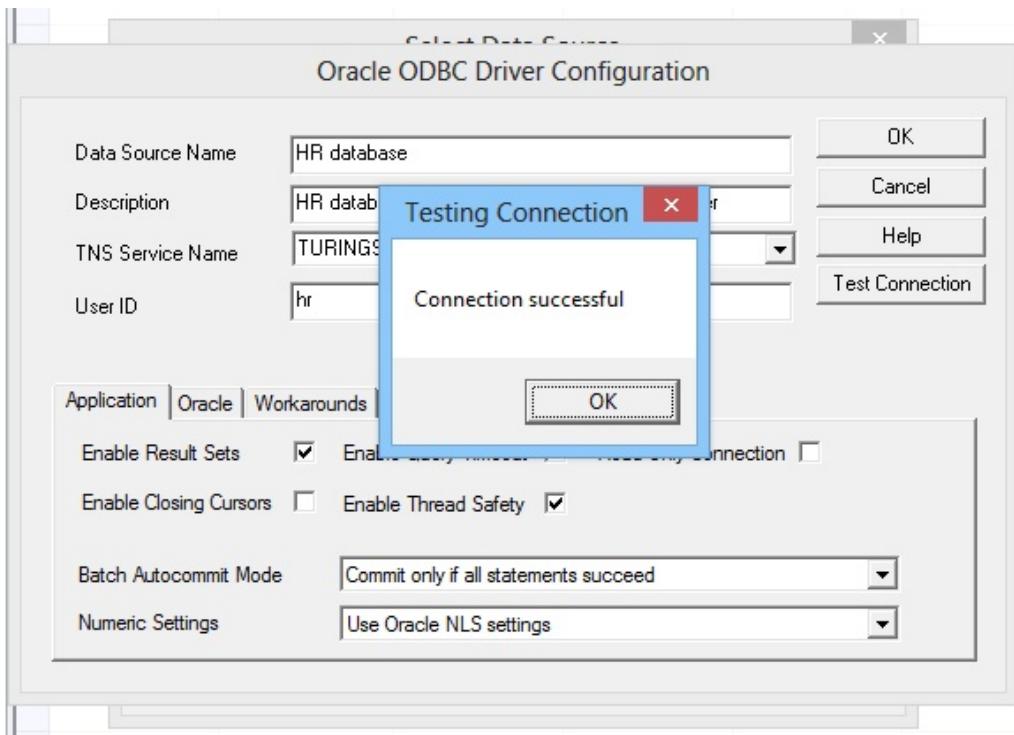
Click "Finish" – you'll see "Oracle ODBC Driver Configuration" windows in which we have to provide correct information to establish a connection with the hr schema in our Oracle database server:



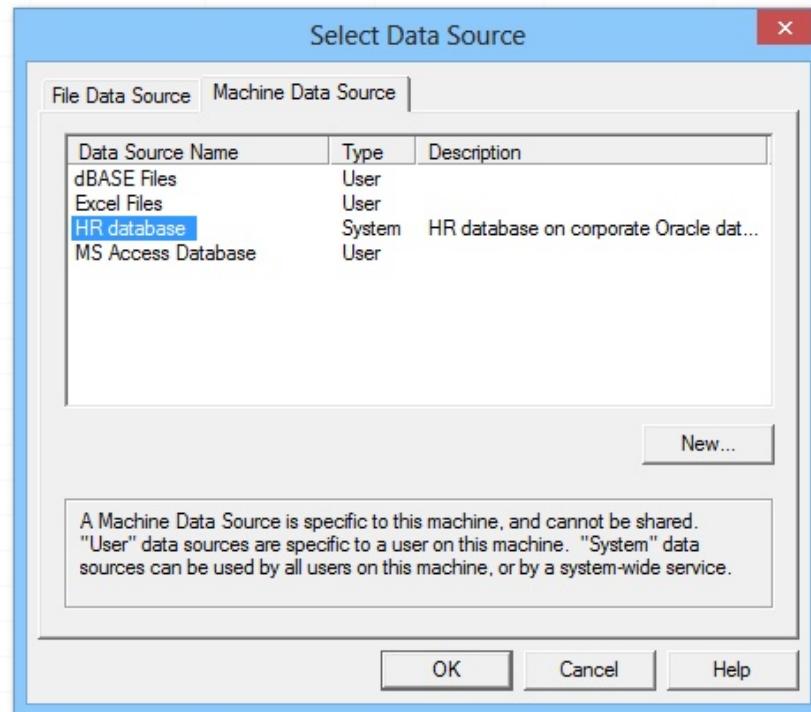
Please note, we have to select "TNS Service Name" here – fortunately, we managed to configure TNS interface on the client side in [part 11](#) and we can use TURINGSMAN service name to establish the connection. So, once we filled all the fields with connection information, we can use "Test Connection" button to check if really everything is correct – let's do it ("Oracle ODBC Driver Connect" windows will be displayed):



On "Oracle ODBC Driver Connect" windows we have to provide a password for our user (hr user was unlocked and the password "turing\$man" was set in [part 11](#)) – then, click "OK":



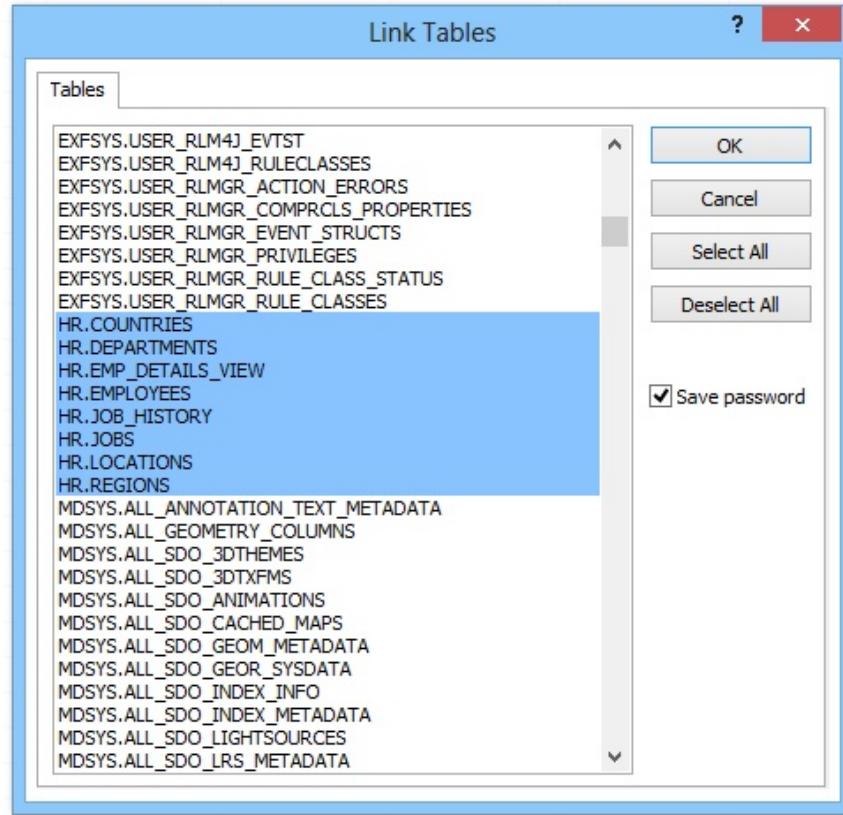
If the test connection was successful, please click "Ok" on the "Testing Connection" window. Then click "Ok" on the "Oracle ODBC Driver Configuration" – we will have "HR database" data source on the list in "Select Data Source" window:



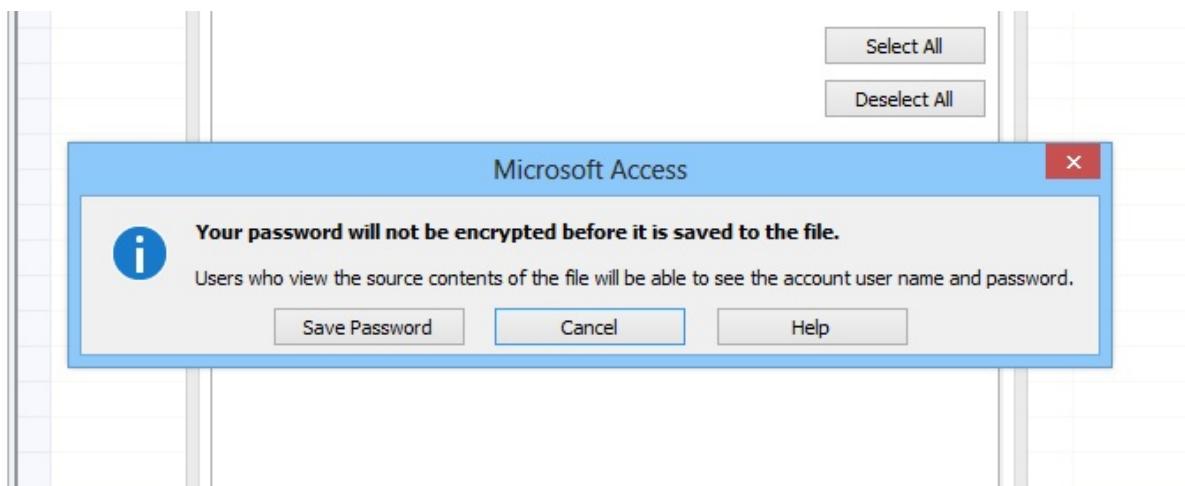
We should select his data source and click "Ok". When "Oracle ODBC Driver Connect" window will appear, provide the password ("turing\$man") for hr user and click "Ok":



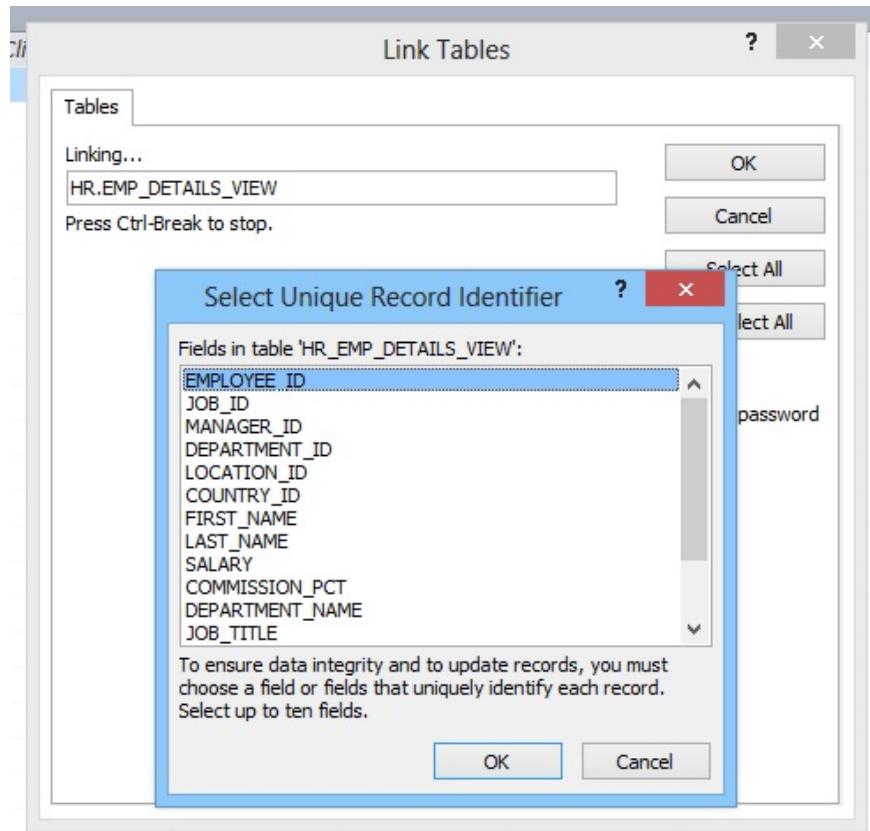
After a while, we will see "Link Tables" window, which presents all the tables in TURINGSMAN database. We don't want to use them all, so please select only the tables that belong to hr user (for the sake of convenience, check "Save password" checkbox):



Then click "Ok". We can ignore security messages (this is only development environment – we want to focus on other things) by pressing "Save Password" button each time:



When "Select Unique Record Identifier" window appeared, please select the column which will be treated as an indexed unique identifier (this is not required, but recommended from the performance point of view):



After a while we will have all hr schema tables linked from MS Access to the Oracle Database via ODBC. We can see them all on the left "Tables" pane – double-click one of the linked tables (please note, there are 8 linked tables, which are stored in Oracle Database and 1 local table: Table1 – which is stored in MS Access/ACE database engine):

All data in tables, which are stored on the Oracle database server (in our case Oracle Database 11gR2 on Solaris 11.1), is now available from our MS Access 2010. We can now take the advantage of all MS Access features to create forms, reports, SQL queries etc. – and even the whole forms based frontend application with some smart VBA coding. Fortunately, this task is quite a different story – we will come back to the application development with MS Access one day (but now, let's come back to our exercises with ODBC):

# Turing's Man Blog

<http://turingsman.net>



Last name	OConnell	Country	United States of America
First name	Donald	City	South San Francisco
Position	Shipping Clerk		
Salary	2600		
Hire date	6/21/2007		



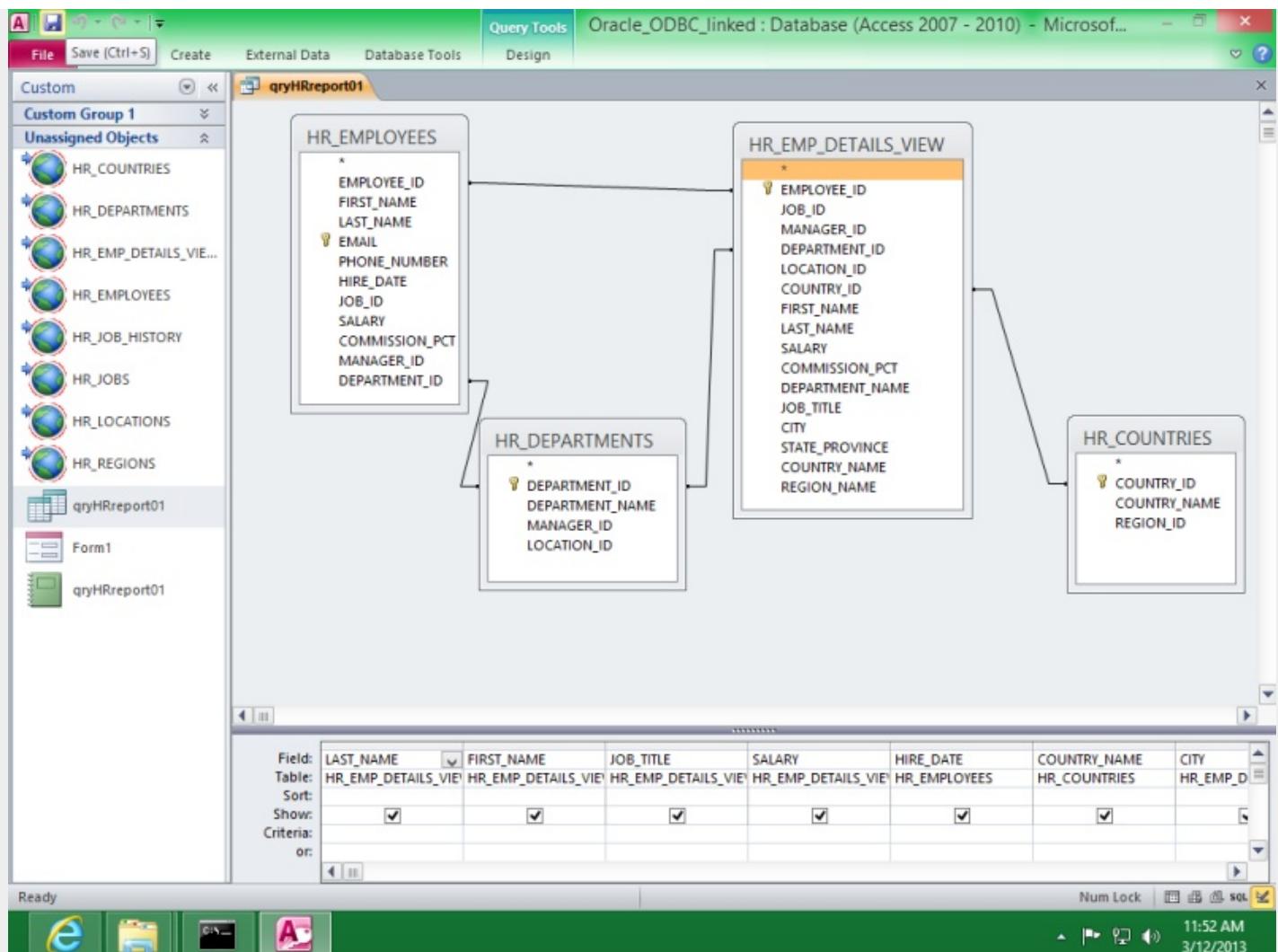
Exit

# Turing's Man

<http://turingsman.net>

Sample report from HR  
database stored in Oracle  
11gR2 on Solaris 11.1 and  
linked to MS Access 2010 (on  
MS Windows 8) via ODBC

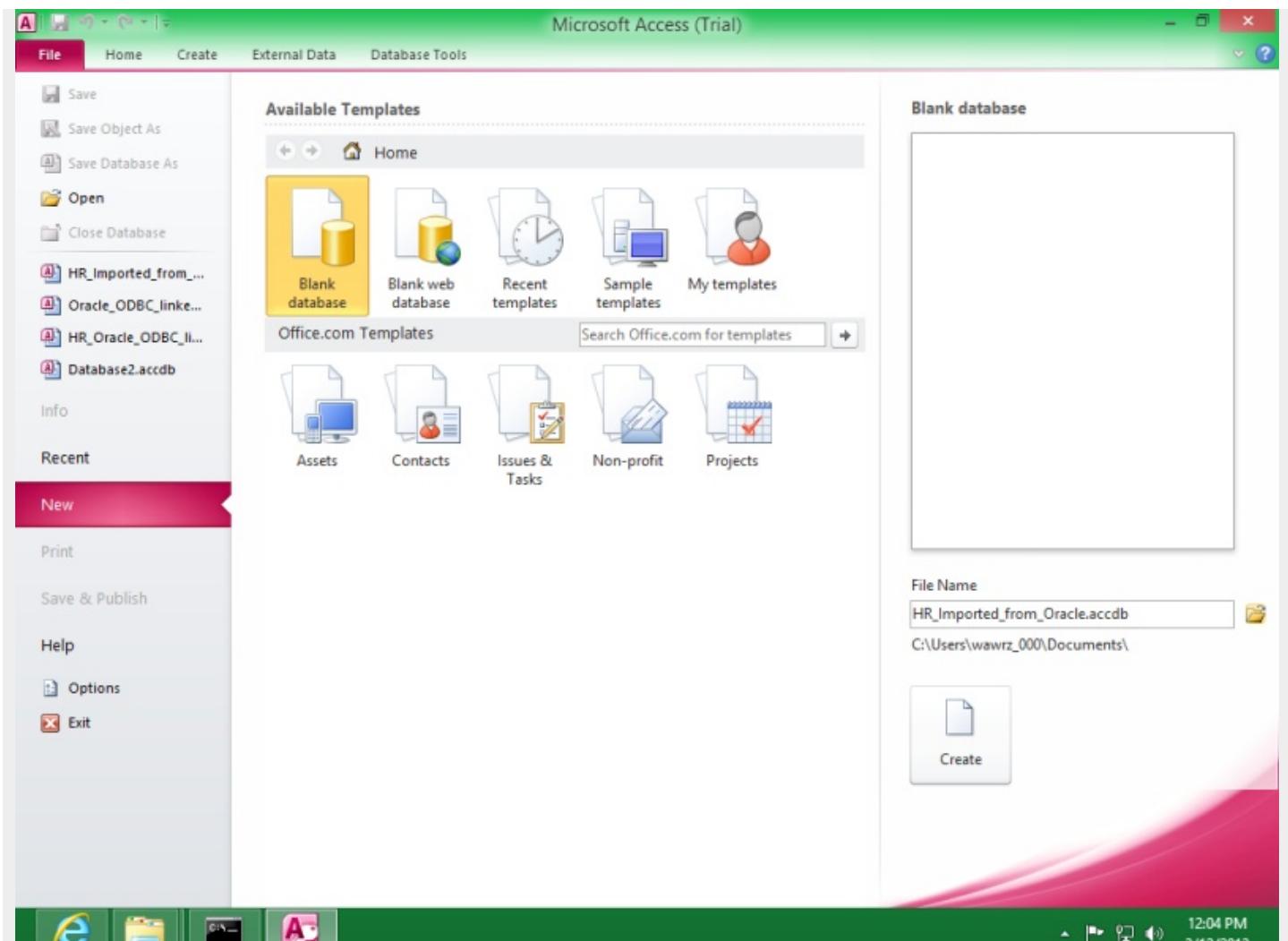
	Last name	Salary	First name	Job title	Hire date	Country	City
	Abel	11000	Ellen	Sales Representative	5/11/2004	United Kingdom	Oxford
Last name	Ande	6400	Sundar	Sales Representative	3/24/2008	United Kingdom	Oxford
First name	Atkinson	2800	Mozhe	Stock Clerk	10/30/2005	United States of America	South San F
Position	Austin	4800	David	Programmer	6/25/2005	United States of America	Southlake
Salary	Baer	10000	Hermann	Public Relations Rep	6/7/2002	Germany	Munich
Hire date	Baida	2900	Shelli	Purchasing Clerk	12/24/2005	United States of America	Seattle
	Banda	6200	Amit	Sales Representative	4/21/2008	United Kingdom	Oxford
	Bates	7300	Elizabeth	Sales Representative	3/24/2007	United Kingdom	Oxford
	Bell	4000	Sarah	Shipping Clerk	2/4/2004	United States of America	South San F
	Bernstein	9500	David	Sales Representative	3/24/2005	United Kingdom	Oxford
	Bissot	3300	Laura	Stock Clerk	8/20/2005	United States of America	South San F
	Bloom	10000	Harrison	Sales Representative	3/23/2006	United Kingdom	Oxford
	Bull	4100	Alexis	Shipping Clerk	2/20/2005	United States of America	South San F
	Cabrio	3000	Anthony	Shipping Clerk	2/7/2007	United States of America	South San F
	Cambrault	7500	Nanette	Sales Representative	12/9/2006	United Kingdom	Oxford
		11000	Gerald	Sales Manager	10/15/2007	United Kingdom	Oxford
	Chen	8200	John	Accountant	9/28/2005	United States of America	Seattle
	Chung	3800	Kelly	Shipping Clerk	6/14/2005	United States of America	South San F
	Colmenares	2500	Karen	Purchasing Clerk	8/10/2007	United States of America	Seattle



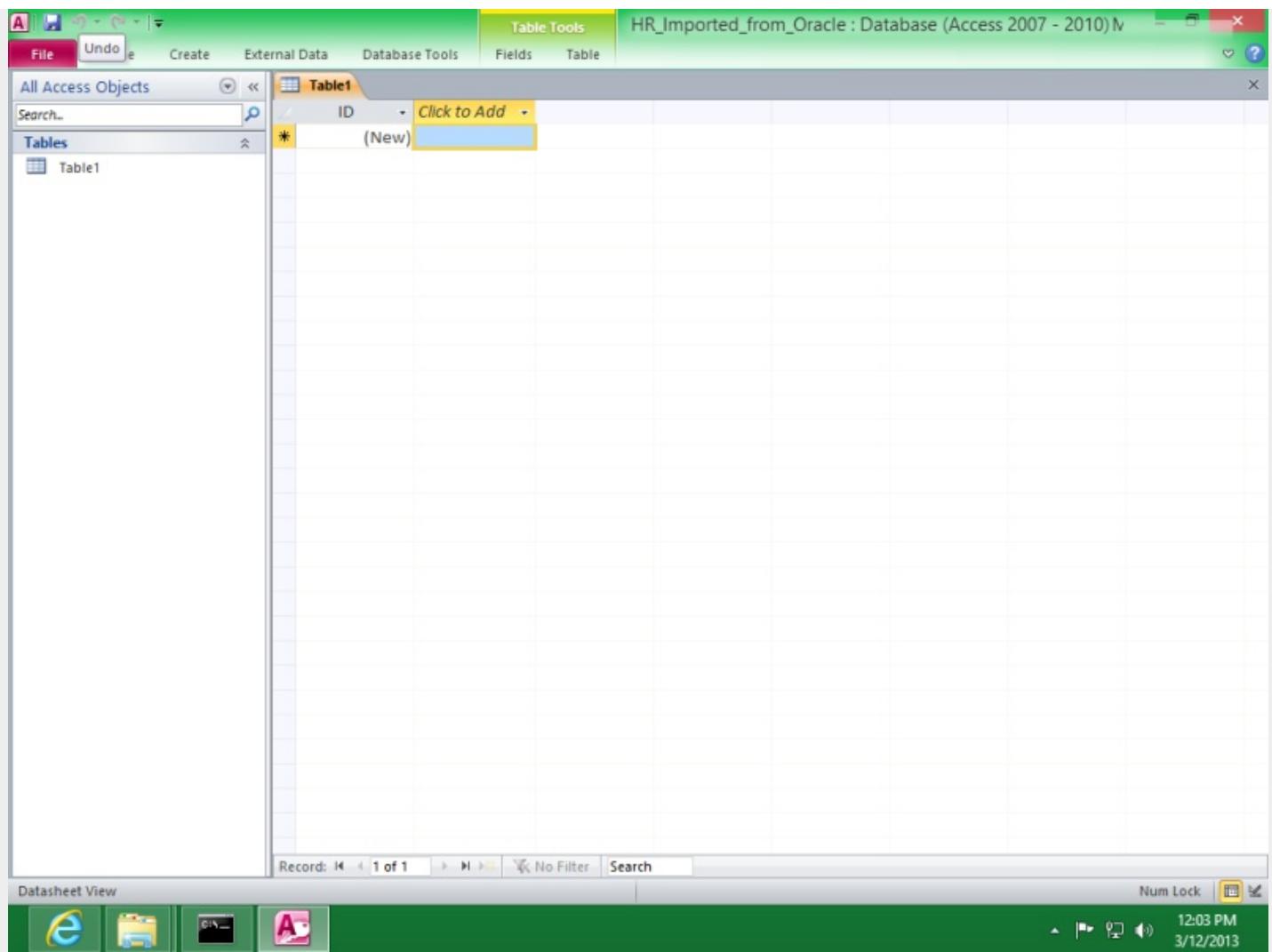
## Scenario 2. Import all hr schema tables from Oracle to the MS Access

We will reuse "HR database" connection created before to import all tables from Oracle Database 11gR2 to our MS Access 2010. Let's imagine we have to work with these data locally, for example – during business trip. MS Access will help us a lot – maybe we will not have an access to the most recent data, but we can assume this is not an issue. Or, let's imagine we want to copy all data from Oracle database do MS Access and then, after some tweaking, we will export all tables to another database, like MS SQL Server? There are many possible options and justifications for this scenario. Let's begin.

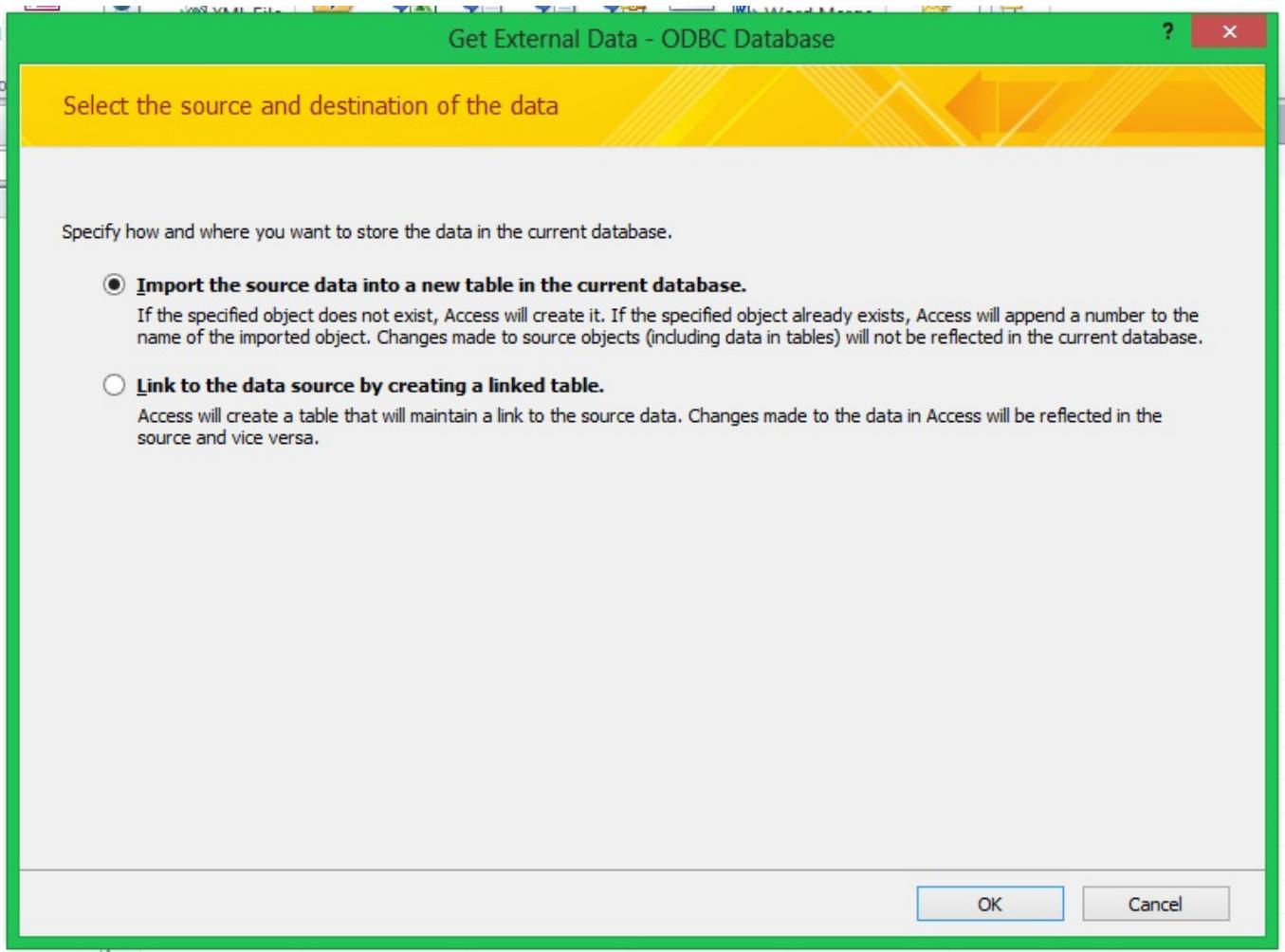
First of all, we have to execute MS Access 2010 – then, we have to choose "Blank database", provide the file name (in my case: "HR\_imported\_from\_Oracle") and click "Create" button:



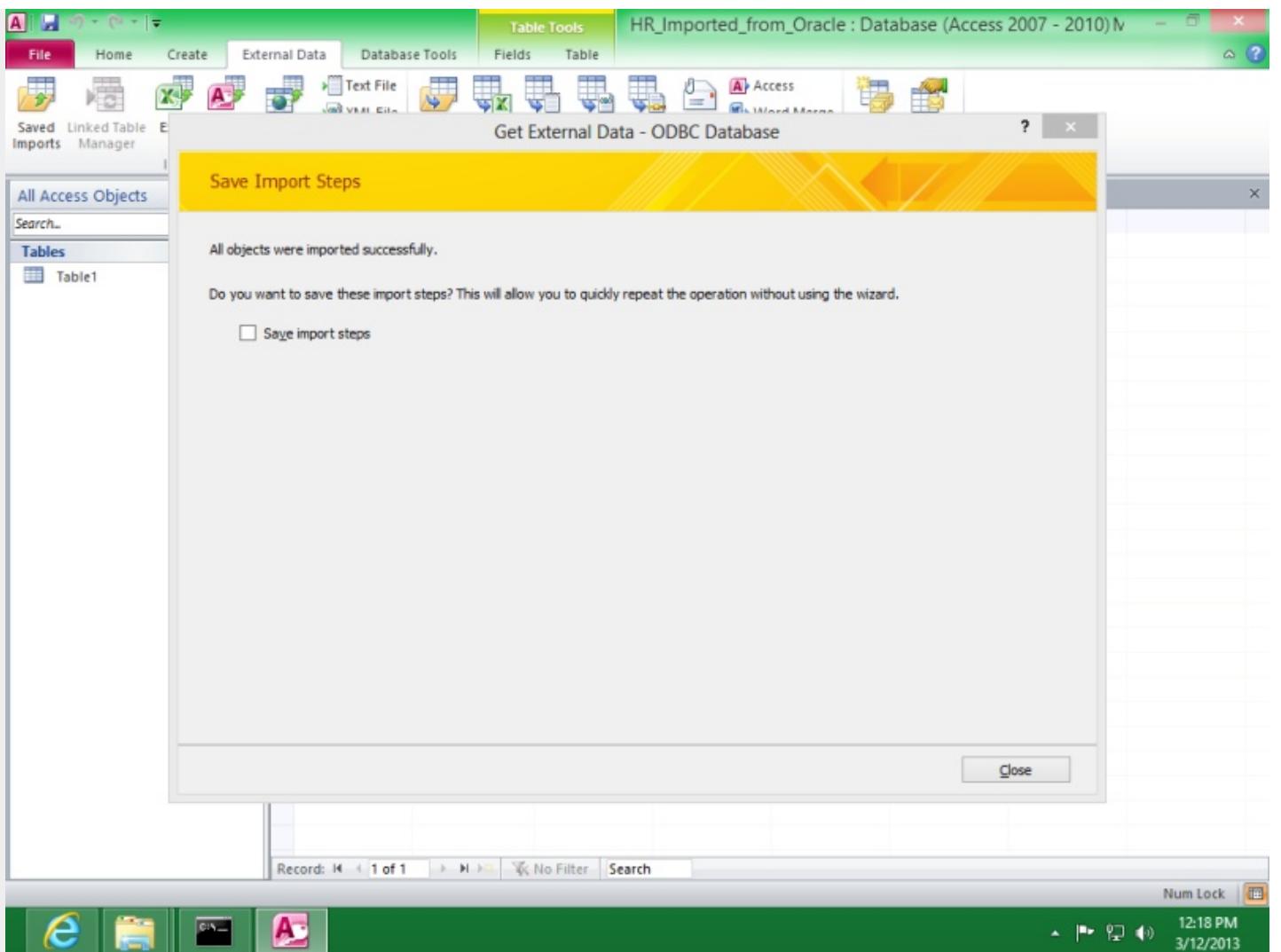
A new database will be created:



Similarly to the previous example, click "External data" tab on the ribbon menu, then click "ODBC Database" button:



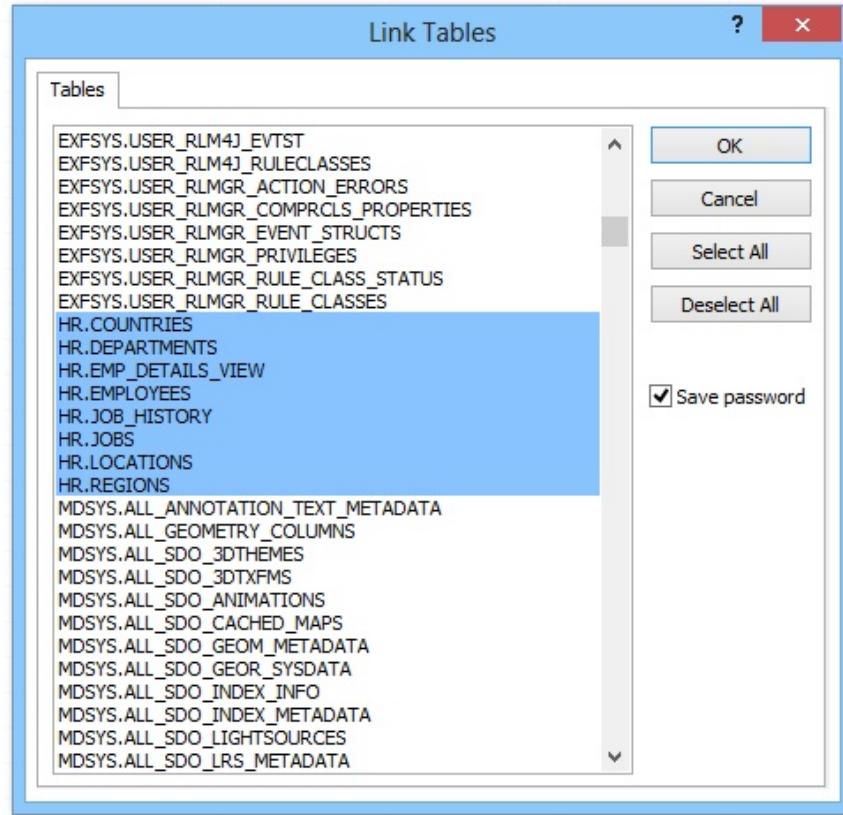
However, this time we will select the first option (should be selected by default) – "Import the source data into a new table in the current database" – then we have to click "Ok". On the "Select Data Source" window we have to click the tab: "Machine Data Source" and select "HR\_imported\_from\_Oracle" from the list). The data source is configured already, so we can click "Ok":



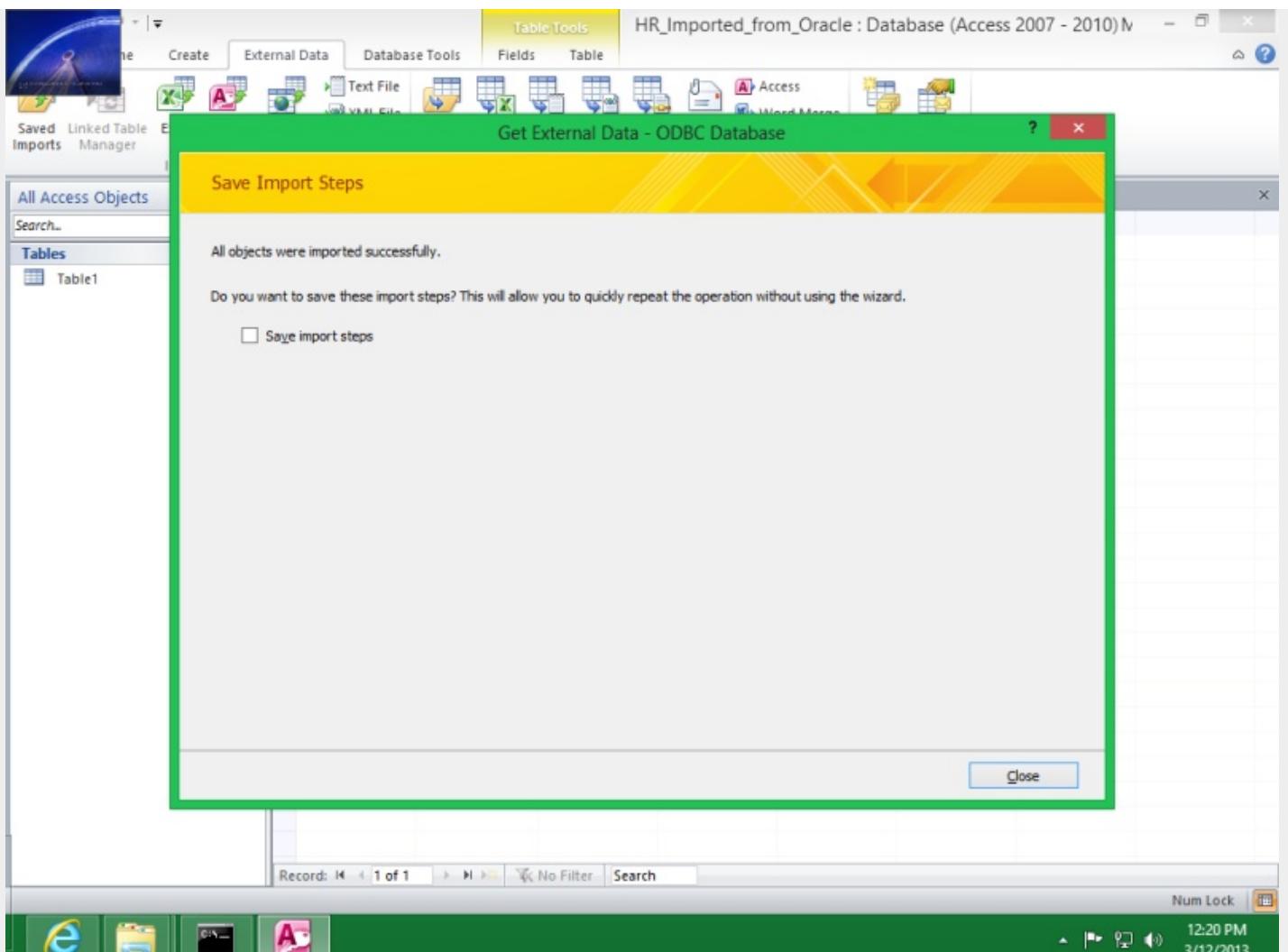
The procedure is the same here like in previous example – remember to provide the correct password ("turing\$man"):



Then select only the tables which belongs to the hr schema – these will be imported to our MS Access 2010 database and handled by MS ACE database engine locally:



After import is ended there will be "Get External Data – ODBC Database" window displayed – we're not interested in saving import template at this moment, so click "Close" button:



Here it is, we have all tables, which were previously stored on Oracle Database 11gR2 on our database server (running Solaris 11.1), imported to the local MS Access 2010 database. Since that moment, we can work with data locally – every database operation will be handled by built-in MS Access 2010 database engine – MS ACE. Please double-click on any available table which we can see in the left pane:

The screenshot shows the Microsoft Access 2010 interface. The ribbon at the top has tabs for File, Save (Ctrl+S), Create, External Data, Database Tools, Fields, and Table. Under Database Tools, there are icons for Text File, XML File, Saved Exports, Excel, XML File, PDF or XPS, E-mail, More, and Export. The Export section includes options for Access, Word Merge, Create E-mail, Manage Replies, and Collect Data. On the left, the navigation pane shows 'All Access Objects' and lists tables: HR\_COUNTRIES, HR\_DEPARTMENTS, HR\_EMP\_DETAILS\_VIEW, HR\_EMPLOYEES (which is selected), HR\_JOB\_HISTORY, HR\_JOBS, HR\_LOCATIONS, and HR\_REGIONS. The main area displays the 'HR\_EMPLOYEES' table in Datasheet View. The table has 8 columns: EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUM, HIRE\_DATE, JOB\_ID, and SALARY. The data shows various employees from different departments and countries. Record 107, Diana Lorentz, is highlighted. At the bottom, there are navigation buttons for Records, a status bar showing '17 of 107', and a toolbar with icons for Microsoft Edge, File Explorer, Task View, and Access.

We can now work with MS Access 2010 and do whatever we want – forms, reports, SQL queries, VBA, Sharepoint integration, data export/import, anything... The only difference is that all data is stored locally and handled by MS ACE, since we're not using linked tables anymore.

[SHARE](#) [F](#) [T](#) [E](#)

## Comments

#8 Lyudmil 2014-03-14 12:38

Thanks so much you made my day.  
Also recomend to check PowerShell

[L](#) [P](#)

0

[Quote](#)

#7 jay 2013-12-27 06:11

Great work will give it a try soon  
I have a odbc compliant database hope this works .., thanks so much

[L](#) [P](#)

0

[Quote](#)

#6 Matti 2013-10-24 16:38

Question: With MS Access as your front end, are you able to import and create new database tables in Oracle. For our datawarehouse, I would like to give users the capability of loading their own data files into the datawarehouse, linking/querying these files with existing DW tables and creating reports. The problem is that our DW is missing many feeds and users must download data from the DW and process mission critical data on their PCs.

[L](#) [P](#)

0

[Quote](#)**#5 Endurance** 2013-10-17 07:39

Thanks for this tutorial. Its very straight forward.



0

[Quote](#)**#4 David Burch** 2013-09-17 20:19

Instead of linking database tables, is it possible to just call PL/SQL queries through packages that reside in the Oracle DB?



0

[Quote](#)**#3 Ken Wais** 2013-07-08 18:26

I am working on a correcting a problem with MS Access 2003 as a front-end to a large Oracle back-end (10g). The data type number has a specification in characters for Oracle as 38 char which Access 2003 auto defaults to text. I am not finding a way to correct this thru the ODBC. I can't change the Orc side. Does MS Access 2010 solve this problem?



0

[Quote](#)**#2 Turingsman** 2013-03-28 19:53

Thank you for kind words 😊 This keeps me motivated. I plan to continue my work on this topic soon and will consider an article on differences, according to your suggestion.



0

[Quote](#)**#1 TheSmileyCoder** 2013-03-25 10:22

+2

Nicely written, and detailed article. I have not yet had the chance to try Oracle as a backend for Access, but have worked with Access for years (and love it).

I hope to see a follow-up article outlining what differences one should be aware of when designing forms against a linked table (ODBC driven backend) versus a standard ACE/JET driven backend.

Keep up the good sharing. Best of luck with your project.

[Quote](#)[Refresh comments list](#)[RSS feed for comments to this post](#)

## Add comment

Name (required)

E-mail (required, but will not display)

Website



1000 symbols left

 Notify me of follow-up comments
[Refresh](#)

**Send**

JComments