WEDNESDAY, MARCH 3, 2010

[How to do real database testing (10 tips to perform serious database tests)?](http://inderpsingh.blogspot.com/2010/03/how-to-do-real-database-testing-10-tips.html)

Many (but not all) applications under test use one or more databases. The purposes of using a database include *long-term storage* of data in an *accessible* and *organized* form. Many people have only a vague idea about database testing. If you are serious about learning database testing, then read on...

Firstly, we need to understand **what is database testing?** As you would know, a database has two main parts - the **data structures** (the schema) that store the data AND the **data** itself. Let us discuss them one by one.

The data is stored in the database in tables. However, tables may not be the only objects in the database. A database may have other objects like views, stored procedures and functions. These other objects help the users access the data in required forms. The data itself is stored in the tables. Database testing involves finding out the answers to the following questions:

**Questions related to database structure**

1. Is the data organized well logically?

2. Does the database perform well?

3. Do the database objects like views, triggers, stored procedures, functions and jobs work correctly?

4. Does the database implement constraints to allow only correct data to be stored in it?

5. Is the data secure from unauthorized access?

**Questions related to data**

1. Is the data complete?

2. Is all data factually correct i.e. in sync with its source, for example the data entered by a user via the application UI?

3. Is there any unnecessary data present?

Now that we understand database testing, it is important to know about the **5 common challenges** seen before or during database testing:

**1. Large scope of testing**

It is important to identify the test items in database testing. Otherwise, you may not have a clear understanding of what you would test and what you would not test. You could run out of time much before finishing the database test.  
Once you have the list of test items, you should estimate the effort required to design the tests and execute the tests for each test item. Depending on their design and data size, some database tests may take a long time to execute. Look at the test estimates in light of the available time. If you do not have enough time, you should select only the important test items for your database test.

**2. Incorrect/ scaled-down test databases**

You may be given a copy of the development database to test. This database may only have little data (the data required to run the application and some sample data to show in the application UI). Testing the development or test or staging databases may not be sufficient. You should also be testing a copy of the production database.  
  
**3. Changes in database schema and data**

This is a particularly nasty challenge. You may find that after you design a test (or even after you execute a test), the database structure (the schema) has been changed. This means that you should be aware of the changes made to the database during testing. Once the database structure changes, you should analyze the impact of the changes and modify any impacted tests. Further, if your test database is being used by other users, you would not be sure about your test results. Therefore, you should ensure that the test database is used for testing purpose only. You may also see this problem if you run multiple tests at the same time. You should run one test at a time at least for the performance tests. You do not want your database performing multiple tasks and under-reporting performance.

**4. Messy testing**

Database testing may get complex. You do not want to be executing tests partially or repeating tests unnecessarily. You should create a test plan and proceed accordingly while carefully noting your progress.

**5. Lack of skills**

The lack of the required skills may really slow things down. In order to perform database testing effectively, you should be comfortable with SQL queries and the required database management tools.

Next, let us discuss the approach for database testing. You should keep the scope of your test as well as the challenges in mind while designing your particular test design and test execution approach. Note the following **10 tips**:  
  
1. List all database-specific requirements. You should gather the requirements from all sources, particularly technical requirements. It is quite possible that some requirements are at a high level. Break-down those requirements into the small testable requirements.

2. Create test scenarios for each requirement as suggested below.

3. In order to check the logical database design, ensure that each entity in the application e.g. actors, system configuration are represented in the database. An application entity may be represented in one or tables in the database. The database should contain only those tables that are required to represent the application entities and no more.  
  
4. In order to check the database performance, you may focus on its throughput and response times. For example, if the database is supposed to insert 1000 customer records per minute, you may design a query that inserts 1000 customer records and print/ store the time taken to do so. If the database is supposed to execute a stored procedure in under 5 seconds, you may design a query to execute the stored procedure with sample test data multiple times and note each time.

5. If you wish to test the database objects e.g. stored procedures, you should remember that a stored procedure may be thought of as a simple program that (optionally) accepts certain input(s) and produces some output. You should design test data to exercise the stored procedure in interesting ways and predict the output of the stored procedure for every test data set.

6. In order to check database constraints, you should design invalid test data sets and then try to insert/ update them in the database. An example of an invalid data set is an order for a customer that does not exist. Another example is a customer test data set with an invalid ZIP code.

7. In order to check the database security, you should design tests that mimic unauthorized access. For example, log in to the database as a user with restricted access and check if you can view/ modify/ delete restricted database objects or view or view and update restricted data. It is important to backup your database before executing any database security tests. Otherwise, you may render your database unusable. You should also check to see that any confidential data in the database e.g. credit card numbers is either encrypted or obfuscated (masked).

8. In order to test data integrity, you should design valid test data sets for each application entity. Insert/ update a valid test data set (for example, a customer) and check that the data has been stored in the correct table(s) in correct columns. Each data in the test data set should have been inserted/ updated in the database. Further, the test data set should be inserted only once and there should not be any other change in the other data.

9. Since your test design would require creating SQL queries, try to keep your queries as simple as possible to prevent defects in them. It is a good idea for someone other than the author to review the queries. You should also dynamically test each query. One way to test your query is to modify it so that it just shows the resultset and does not perform the actual operation e.g. insert, delete. Another way to test your query is to run it for a couple of iteration s and verify the results.

10. If you are going to have a large number of tests, you should pay special attention to organizing them. You should also consider at least partial automation of frequently run tests.

Now you should know what database testing is all about, the problems that you are likely to face while doing database testing and how to design a good database test approach for the scope decided by you. Why don't you consider professional database testing for your application? Let me know your experiences. You can contact me at *isingh 30 AT gmail DOT com (no spaces)*.

POSTED BY INDER P SINGH AT [5:19 PM](http://inderpsingh.blogspot.com/2010/03/how-to-do-real-database-testing-10-tips.html) [http://www.blogger.com/img/icon18_email.gif](http://www.blogger.com/email-post.g?blogID=3806208351698608392&postID=8589824708374020688)