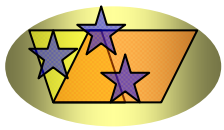
Database software Level 3 Notes for City & Guilds 7574 ITQ Unit 319

Written for Access 2007®
for Windows™ XP

Tench Computing Ltd
Pines
Glendale Road
Burgess Hill
West Sussex
RH15 0EJ

Web address: www.tenchcomputing.co.uk
Email address: jtench@globalnet.co.uk



About the author: *Jackie Tench MSc, ACIB, Cert Ed*

Jackie started her working career in branch banking with the Midland Bank (now HSBC) and was transferred to their Computing Department after achieving 100% in their ability test for programmers. She then worked for more than a decade in this department and was one of the first women to achieve a junior management grade at the age of 21. She attended a significant number of IBM programming training courses during her time there.

Jackie was the first woman to pass the ACIB (Associate Chartered Institute of Bankers) examinations in the Midland Bank (HSBC) and the youngest person at 21 years of age.

Jackie then left to raise a family but still found time to teach part-time at a college in Sheffield and to obtain a MSc in Computing and a Cert Ed in teaching.

When her children were old enough Jackie returned to work full-time and was a Senior Lecturer in Software Engineering and Computer Studies at a college in Brighton for nearly 10 years teaching all levels up to and including HND.

Therefore, Jackie has considerable business knowledge and qualifications plus wide experience in practical computing and training – covering areas such as structured design, analysis, coding, testing and implementing software applications plus training students to fulfil an important role in the computer industry.

Jackie has worked as a consultant for several blue chip companies and examination boards using her software engineering and educational training skills and is now one of the foremost experts in computing with an extensive knowledge of programming languages and applications.

Contents Page

Testing

After a software application has been created it needs testing. The testing should be done in a structured way and not haphazardly. If testing is done in a structured way then any errors in the software should be found. Also if the testing is structured then when an error is found it will be known what caused the error because the testing procedures have been documented.

Test plan

A test plan should be completed to show what testing is to be done. The test plan identifies the test number, purpose of the test, the input and the expected result. If the input and the expected result are small they can actually be written on the test plan. If the input and test data are too large to include on the test plan then a number to cross reference them should be inserted on the test plan. A test plan once prepared can be used throughout testing even for repeated tests.

Tests should be done to test the software under normal operating conditions, under exceptional conditions and boundaries should be tested for input data.

Exceptional conditions are tested to make sure that if a user does something wrong the software will not crash or lose data or the data does not become corrupted. An example of an exceptional condition is if a user tries to delete a record e.g. a customer record, which has another record associated with it e.g. a booking record. The database should not allow this to happen because the booking record will be left with no customer record associated with it. If enforced referential integrity has been applied to the link between the CUSTOMER and BOOKING tables this will not be allowed.

Boundaries around input data must be tested. If a field can only have values between 100 and 1000 (inclusive) as input values then validation should be applied to the field. Testing should be done to ensure that only these values are allowed. Testing the boundaries means that the values 99, 100, 101, 999, 1000 and 1001 should be entered. These values will test the boundaries of the input data. It is the boundary input that is most likely to cause an error. The values 100, 101, 999 and 1000 should be accepted and the values 99 and 1001 should be rejected with an error message.

Expected results are the results that are expected as output given the stated input. The expected results have to be worked out manually. So, if the input is an action such as a click on a button to produce a report the result for the report should be worked out and would be the expected results. The expected results would then be compared with the actual results to confirm that the software works correctly.

An example of a blank test plan page is shown on the next page. For a large software application a test plan can run to many pages.

The purpose of testing is to prove that the software works under normal and exceptional conditions and that it complies with the design specification.

TEST PLAN

DATABASE NAME:		**VERSION NO:** **TESTER NAME:**	**PAGE NO:**
Test No	**Purpose/Type Of Test**	**Input**	**Expected Result**

Test data

Test data must be prepared that will fulfil the tests shown in the test plan.

Enough test data must be used to test the database but never enter too much test data because if a test results in incorrect output the data may have to be entered again or amended.

Test log

A test log is used to track the testing as it is being done. A test log should include the test number, the date, the actual results and comments on any discrepancies between the expected results and the actual results.

If any discrepancies are found then the cause of the discrepancy must be found and corrected and the test must be done again to check the results. Sometimes a test may have to be done several times before the problem is resolved.

A new test log is used for each batch of testing. A test log provides a complete record of all the testing done on a system.

An example of a blank test log page is shown on the next page. For a large software application a test log can run to many pages.

Test plan and test log forms

These forms can be created in a word processor using tables and filled in using the word processor. The rows can be made as large or as small as required.

TEST LOG

DATABASE NAME:			**VERSION NO:** **TESTER NAME:**	**PAGE NO:**
Test No	**Date**	**Actual Output**	**Comments on discrepancies**	

Example test plan

The test plan should include testing for all the extras that have been added to the basic database. Forms, queries and reports must be tested for correct operation. Any buttons that have been added must be tested to ensure that they operate correctly. The more features that are added to the basic database the more testing that will need to be done.

The basic in-built functions of the database do not need to be tested e.g. deletion of a record. But, deletion of a record from a table should be tested if the links between tables has enforced referential integrity added, as a test must be made to ensure that the referential integrity works.

The skill of testing is to do enough tests to ensure that the database works correctly and complies with the specification but to try not to test the same thing more than once. So, the test plan and the test data need to be carefully thought out.

The following pages show the test plan prepared for the testing of the traveldb database. The references to test data 1-5 in the test plan are to the test data already used previously to set up records for tables to enable some output from the reports.

TEST PLAN

DATABASE NAME: *traveldb*		**VERSION NO:** *001* **TESTER NAME:** *your name*	**PAGE NO:** *1 of 2*
Test No	**Purpose/Type Of Test**	**Input**	**Expected Result**
1	Menu button for HOLIDAY input form	Click on HOLIDAY button	The HOLIDAY input form is displayed on screen
2	Menu button for BOOKING input form	Click on BOOKING button	The BOOKING input form is displayed on screen
3	Menu button for CUSTOMER input form	Click on CUSTOMER button	The CUSTOMER input form is displayed on screen
4	Menu button for TICKET input form	Click on TICKET button	The TICKET input form is displayed on screen
5	Menu button for Quit Travel DB	Click on Quit Travel DB button	The database is closed
6	HOLIDAY records input via input form	Test data 1	HOLIDAY table contains entered records
7	CUSTOMER records input via input form	Test data 2	CUSTOMER table contains entered records
8	BOOKING records input via input form	Test data 3	BOOKING table contains entered records
9	Record rejected if Holiday-Id does not exist in HOLIDAY table	Test data 4	Entered record rejected as invalid
10	TICKET records input via input form	Test data 5	TICKET table contains entered records

DATABASE NAME: *traveldb*		**VERSION NO:** *001* **TESTER NAME:** *your name*	**PAGE NO:** *2 of 2*
Test No	**Purpose/Type Of Test**	**Input**	**Expected Result**
11	Record rejected if Booking-Id does not exist in BOOKING table	Booking-Id 10, Ticket-Id 20	Entered record rejected as invalid
12	Validation works on Holiday-Id	Holiday-Id 0	Entered record rejected and validation error message appears
13	Validation works on Customer-Id	Customer-Id 0	Entered record rejected and validation error message appears
14	Validation works on Booking-Id	Booking-Id 0	Entered record rejected and validation error message appears
15	Validation works on Ticket-Id	Ticket-Id 0	Entered record rejected and validation error message appears
16	Referential integrity for HOLIDAY table	Delete record with Holiday-Id 2	Error message appears and delete not allowed
17	Referential integrity for CUSTOMER table	Delete record with Customer-Id 3	Error message appears and delete not allowed
18	Referential integrity for BOOKING table	Delete record with Booking-Id 5	Error message appears and delete not allowed
19	Menu button for print preview of Accounts report	Click on PREVIEW ACCOUNTS button	The Accounts report is displayed on screen with the correct records as per test output 1
20	Menu button for print preview of Tickets report	Click on PREVIEW TICKETS button	The Tickets report is displayed on screen with the correct records as per test output 2
21	Menu button for print of Accounts report	Click on PRINT ACCOUNTS button	The Accounts report is printed with the correct records as per test output 1
22	Menu button for print of Tickets report	Click on PRINT TICKETS button	The Tickets report is printed with the correct records as per test output 2

Any test data required is normally prepared after the test plan has been created. At this stage any expected results, specified in the test plan, must be created.

The **expected results** specified in the above test plan are test output 1 and 2. Test output 1 is the Accounts report and test output 2 is the Tickets report. The expected results are worked out using the input data that has been specified in the test plan up to this point.

Test output 1

Accounts report for accounts not paid. One account should be printed per page.

ACCOUNT

Name	Mr J Williams	**Booking -Id**	1
Address	129 Church Avenue	**Customer-Id**	1
	Wilmington	**Holiday-Id**	1
	Sussex		
Post Code	WA3 5RT		
Destination	Barcelona		
Cost per person	£250.99		
Departure date	12/04/2009		
Return date	26/04/2009		
No of people	2		

Total Cost: £501.98

17 April 2008 **Page 1 of 3**

ACCOUNT

Name	Mr K Ingham	**Booking -Id**	4
Address	78 Waldren Rise	**Customer-Id**	4
	Walthington	**Holiday-Id**	4
	Sussex		
Post Code	WA14 7UH		
Destination	Amsterdam		
Cost per person	£410.50		
Departure date	24/07/2009		
Return date	07/08/2009		
No of people	6		

Total Cost: £2,463.00

ACCOUNT

Name	Mrs W Adams	**Booking -Id**	5
Address	69 Fleet Road	**Customer-Id**	2
	Burnham	**Holiday-Id**	5
	Surrey		
Post Code	RE2 6TY		
Destination	Madrid		
Cost per person	£300.00		
Departure date	25/09/2009		
Return date	04/10/2009		
No of people	4		

Total Cost: £1,200.00

Test output 2

Tickets report for tickets not issued. Three tickets should be printed per page on A4 paper.

Ticket No **1**

Booking-Id	1
Name	Mr J Williams
Address	129 Church Avenue
	Wilmingston
	Sussex
Post Code	WA3 5RT
Destination	Barcelona
Departure date	12/04/2009
Return date	26/04/2009

Ticket No **2**

Booking-Id	1
Name	Mr J Williams
Address	129 Church Avenue
	Wilmingston
	Sussex
Post Code	WA3 5RT
Destination	Barcelona
Departure date	12/04/2009
Return date	26/04/2009

Ticket No 3

Booking-Id	2
Name	Mr R Bodkin
Address	25 Western Road
Brighton	Brighton
	Sussex
Post Code	BN4 6YH
Destination	Madrid
Departure date	11/06/2009
Return date	18/06/2009

Ticket No 4

Booking-Id	2
Name	Mr R Bodkin
Address	25 Western Road
Brighton	Brighton
	Sussex
Post Code	BN4 6YH
Destination	Madrid
Departure date	11/06/2009
Return date	18/06/2009

Ticket No 5

Booking-Id	2
Name	Mr R Bodkin
Address	25 Western Road
Brighton	Brighton
	Sussex
Post Code	BN4 6YH
Destination	Madrid
Departure date	11/06/2009
Return date	18/06/2009

Ticket No 6

Booking-Id	2
Name	Mr R Bodkin
Address	25 Western Road
Brighton	Brighton
	Sussex
Post Code	BN4 6YH
Destination	Madrid
Departure date	11/06/2009
Return date	18/06/2009

Ticket No 16

Booking-Id	5
Name	Mrs W Adams
Address	69 Fleet Road
Brighton	Burnham
	Surrey
Post Code	RE2 6TY
Destination	Madrid
Departure date	25/09/2009
Return date	04/10/2009

Ticket No 17

Booking-Id	5
Name	Mrs W Adams
Address	69 Fleet Road
Brighton	Burnham
	Surrey
Post Code	RE2 6TY
Destination	Madrid
Departure date	25/09/2009
Return date	04/10/2009

Ticket No **18**

Booking-Id	5
Name	Mrs W Adams
Address	69 Fleet Road
Brighton	Burnham
	Surrey
Post Code	RE2 6TY
Destination	Madrid
Departure date	25/09/2009
Return date	04/10/2009

Ticket No **19**

Booking-Id	5
Name	Mrs W Adams
Address	69 Fleet Road
Brighton	Burnham
	Surrey
Post Code	RE2 6TY
Destination	Madrid
Departure date	25/09/2009
Return date	04/10/2009

The more input data that is entered as test data the more difficult and time consuming it is to create the expected results for output such as reports. That is why you should use as little test data as possible but still be able to test the database.

Now that the test plan, test data and expected results have been created the actual testing can be done.

As each test is done the test log should be filled in.

TEST LOG

DATABASE NAME: *traveldb*			**VERSION NO:** 001 **TESTER NAME:** *your name*	**PAGE NO:** *1 of 2*
Test No	**Date**	**Actual Output**	**Comments on discrepancies**	
1	02/04/2008	HOLIDAY form is displayed on screen	OK	
2	02/04/2008	BOOKING form is displayed on screen	OK	
3	02/04/2008	BOOKING form is displayed on screen	The wrong form is displayed on screen. It should be the CUSTOMER form. The wrong form was attached to the button. This has now been corrected.	
4	02/04/2008	TICKET form is displayed on screen	OK	
5	02/04/2008	Database is closed	OK	
6	02/04/2008	HOLIDAY table contains records as per test data 1	OK	
7	02/04/2008	CUSTOMER table contains records as per test data 2	OK	
8	02/04/2008	BOOKING table contains records as per test data 3	OK	
9	02/04/2008	Record rejected for HOLIDAY table as per test data 4	OK	
10	02/04/2008	TICKET table contains records as per test data 5	OK	
11	02/04/2008	Record rejected for BOOKING table	OK	
12	02/04/2008	Record rejected for HOLIDAY table and validation error message appears	OK	
13	02/04/2008	Record rejected for CUSTOMER table and validation error message appears	OK	

DATABASE NAME: *Traveldb*			**VERSION NO:** 001 **TESTER NAME:** *your name*	**PAGE NO:** *2 of 2*
Test No	**Date**	**Actual Output**	**Comments on discrepancies**	
14	02/04/2008	Record rejected for BOOKING table and validation error message appears	OK	
15	02/04/2008	Record rejected for TICKET table and validation error message appears	OK	
16	02/04/2008	Record not deleted from HOLIDAY table and error message appears	OK	
17	02/04/2008	Record not deleted from CUSTOMER table and error message appears	OK	
18	02/04/2008	Record not deleted from BOOKING table and error message appears	OK	
19	02/04/2008	Accounts report is displayed on screen	OK	
20	02/04/2008	Tickets report is displayed on screen	OK	
21	02/04/2008	Accounts report is printed	Not printing each account on a separate page. Page adjusted.	
22	02/04/2008	Tickets report is printed	OK	
3b	03/04/2008	CUSTOMER form is displayed on screen	OK	
21b	03/04/2008	Accounts report is printed	OK	

The actual results should be compared with the expected results and any discrepancies found noted in the test log. The cause of the discrepancies should be located. It may be the expected results that are incorrect or it may be the results produced by the database are incorrect.

Note that tests number 3 and 21 show discrepancies with the actual and expected outcome to illustrate the type of comments that appear in the test log. Also the tests must be redone (3b, 21b) so that the amendments made to correct the problem are tested.

Printed output should be annotated with the test number and date so that if there are any queries about the test the output can be traced back to the actual test number. Test numbers 21, 22 and 21b produce printed output so the Accounts reports should be annotated with the numbers 21 and 21b and the date and the Tickets report should be annotated with the number 22 and the date.

Technical documentation

Technical documentation must be created to provide information for future maintenance of the database. Most of the documentation will already be present because it has been created during the design, creation and testing of the database.

The technical documentation should include the following:

- Entity Relationship diagram (ERD)
- Explanation of normalisation (1NF, 2NF, 3NF)
- Table structures
- Data dictionary
- Screen/form layouts
- Report layouts
- Query designs
- Illustration and description of the menu interface
- Test plan
- Test log
- Testing output

A list of the contents with page numbers in the order in which they appear in the document should be included at the front. An alphabetical list of the contents with page numbers should be included as an index at the back.

Print the structure of a table

You cannot print a table structure directly from Access you must copy and paste it into another application.

- Open ACCESS if not already using it.
- Open the required database.
- Right click on the required table and select **Design View** on the pop up menu
- Hold down the **ALT** key and press the **PRT Scrn** key. This will copy the structure onto the clipboard.
- Select **All Programs|Microsoft Office|Microsoft Office Word 2007** from the Windows **Start** menu.
- Select the **Home** command tab and click the top half of the **Paste**  icon on the **Clipboard** command set ***OR*** press the **CTRL+V** keys.
- Click the **Office Button** and select **Print** on the dropdown list.

The Print dialog appears.

- Click OK.
- Select the **Office Button** and click the **Exit Word** icon at the bottom of the dropdown list to close the **Word** program and save the file with a suitable filename.
- If Access does not appear click on the minimised Access button at the bottom of the screen to return to Access.
- Right click on the table tab and select **Close** on the pop up menu.

[This page is intentionally blank]