

# **Partnering Academics and Community Engagement (PACE) program Database Project**

## **Database Integrity Constraints Test Plan**

Date: March 27, 2016

**Prepared by:** Group 1

## Table of Contents

<b>Introduction .....</b>	<b>3</b>
1.1    OBJECTIVES .....	3
1.2    TEAM MEMBERS.....	3
<b>2    Scope .....</b>	<b>3</b>
<b>3    Assumptions / Risks.....</b>	<b>4</b>
3.1    ASSUMPTIONS .....	4
3.2    RISKS.....	4
<b>4    Test Approach.....</b>	<b>4</b>
4.1    TEST AUTOMATION.....	5
<b>5    Test Environment .....</b>	<b>5</b>
<b>6    Milestones / Deliverables.....</b>	<b>5</b>
6.1    TEST SCHEDULE .....	5
6.2    DELIVERABLES.....	5

# Introduction

This test plan has been created to address the testing of specific Oracle database constraints, namely Referential Integrity Constraints. Included in this document are the specific objectives, scope, schedule, risks and approach to the testing. This document will also clearly identify what the test deliverables are and what is deemed in and out of scope.

## 1.1 Objectives

This test plan describes the constraint testing procedures used for the PACE database. These testing procedures will help ensure that data integrity is maintained throughout the project.

This document includes the following objectives:

- List of the constraints to be tested.
- Describe the testing scripts and procedures to be employed.
- Provide an estimate of the testing efforts.
- Describe the post-test deliverable.

## 1.2 Team Members (Group 1)

Resource Name	Role
Sarina Bell	Developer/Tester
Henry Jeter	Developer/Tester
Larry Uchime	Developer/Tester
Jessica Wilson	Developer/Tester

The four members of Group- 1 will act as both developers and testers. As the projects evolves, additional testing needs may be identified and a second phase of testing may be required.

# 2 Scope

There are many types of tests that can be performed on Oracle databases. This test plan will describe a short set of testing procedures that will help ensure data integrity of the PACE database.

This test plan describes testing procedures in the following areas:

- Ensure that duplicate primary keys cannot be inserted into any table.

- Ensure that foreign keys prevent inconsistent data from being entered in any table.
- Ensure that constraints prevent invalid data (such as null values) from being entered in any table.
- Ensure that minimum cordiality is enforced (such as preventing the deletion of a parent record that has a child record).

### 3 Assumptions / Risks

#### 3.1 Assumptions

It is assumed that all functions and features of the Web application – as they relate to the PACE database – have been identified at this time.

#### 3.2 Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project.

#	Risk	Impact	Trigger	Mitigation Plan
1	Scope Creep – as the project progresses, additional functionality may be required, which may cause database schema changes.	High	Delays in the project delivery date	Set and adhere to the PACE project priorities. Discuss priorities with Prof. Booth and the class. Both time and functionality are drivers of the project so the delivery date is not very flexible (can't push the date out too much).
2	Changes to the functionality may negate the tests already written and we may lose test cases already written	High – to schedule and quality	Loss of test cases	Rewrite only specific test procedures and retest.
3	Scheduled test phase completion may not be possible because of class schedules and work load.	Medium	Test phase (e.g. Installation testing) was not delivered on schedule	Recruit external testers when possible (e.g. friends, family, members from other groups within the class if available).

### 4 Test Approach

The PACE project database constraint test is intended as a one-time test. There will be no iterative testing as the project progresses. The test will be repeated only if modifications are to the database schema – in which case test procedures may be added, modified, or deleted based on the extent of the modifications.

## 4.1 Test Automation

No automated test procedures are incorporated in this project or described in this document.

## 5 Test Environment

A refined version of the PACE database that was created as a project in the Fall 2015 ITDB 4201 course will be used as a base for the test procedures.

## 6 Milestones / Deliverables

### 6.1 Test Schedule

The test schedule is as follows:

Task Name	Start	Finish	Effort	Comments
Test Planning	TBD	TBD	TBD	
Review test plan	TBD	TBD	TBD	
Installation (scripts) of the PACE database testing	TBD	TBD	TBD	
Defect resolution testing	N/A	N/A	N/A	
Regression testing	N/A	N/A	N/A	
Total				

### 6.2 Deliverables

Deliverable	For	Date / Milestone
Test Plan	Test Team, Prof. Booth	TDB
Test Results	Test Team	TBD
Test Status report	Test Team	TBD
Metrics	All team members	

## 7 Test Procedures

### 7.1 PACE Database Creation Testing

Verify that the base PACE database can be successfully created.

**Test Objective:** To ensure that the scripts that build and populate the database work as expected.

▪ **Resources:**

- Oracle (installed).
- SQL\*Plus Instant Client.
- Scripts provided by various members of the ITDB 4202 class.

- **Procedure:** Run the scripts in a specified order.
- **Expected Results:** The PACE database will be created with no errors. It will consist of the following 11 tables:
- **Actual Results:** (Passed/Failed)
- **Comment:**

## 7.2 Database Constraint Testing

Verify that the database constraints that are included in the original schema work as expected. Constraint testing will not include enabling and disabling Oracle constraints.

**Test Objective:** Verify that the constraints exist:

- **Procedure:** Run the following script on each table in the database and take note of the list of constraints:  

```
SELECT constraint_name, constraint_type, search_condition
FROM user_constraints
WHERE table_name = 'Table Name';
```
- **Expected Results:** The constraint name, constraint type and search condition should be listed for each table.
- **Actual Results:** (Passed/Failed)
- **Comment:**

**Test Objective:** Verify the *Unique Integrity Constraint* on each primary key in each table in the database.

- **Procedure:** Run the following script on each table in the database.  

```
INSERT INTO <table name>
VALUES (1, <remaining valid values>);
```
- **Expected Results:** SQL\*Plus should display the following error message:  

```
ERROR at line 1:
ORA-00001: unique constraint (<TableOwner>.<TableName>_PK) violated.
(Example: (HJETER.RELATED_COURSES_PK)).
```
- **Actual Results:** (Passed/Failed)
- **Comment:**

**Test Objective:** Verify the *NOT NULL Constraint* on at least one NOT NULL column in each table in the database.

- **Procedure:** Run the following script on each table in the database:  

```
INSERT INTO <table name>
VALUES (<unique primary key>, <Null>, <remaining valid values>);
```

Example:  

```
INSERT INTO Topic_Areas
```

VALUES (24, 7, NULL);

- **Expected Results:** SQL\*Plus should display the following error message:  
ERROR at line 1:  
ORA-01400: cannot insert NULL into  
(<TableOwner>.<TableName>.<ColumnName> )
- **Actual Results:** (Passed/Failed)
- **Comment:**

**Test Objective:** Verify the *Foreign Key Constraint* on each *Child* table in the database (each table that has its primary key in a parent table).

- **Procedure:** Run the following script on each *Child* table in the database :  
INSERT INTO <Table Name>  
VALUES (<Valid Values>);
- **Expected Results:** SQL\*Plus should display the following error message:  
ERROR at line 1:  
ORA-02291: integrity constraint (<Foreign Key Name>) violated - parent key not found
- **Actual Results:** (Passed/Failed)
- **Comment:**

**Test Objective:** Verify the *Foreign Key Constraint* on each *Parent* table in the database (has a foreign key from a child table).

- **Procedure:** Run the following script on each *Parent* table in the database :  
DELETE FROM <Table Name>  
WHERE (<Valid Values>);
- **Expected Results:** SQL\*Plus should display the following error message:  
ERROR at line 1:  
ORA-02291: integrity constraint (<Foreign Key Name>)
- **Actual Results:** (Passed/Failed)
- **Comment:**

[Back](#)