

Database Programming with SQL 19-1: Testing

Practice Activities

# Objectives

* Develop and apply a strategy for testing that a database functions as designed.

# Try It / Solve It

1. Design and carry out tests to check the following:
   1. The business rule that requires that employees have a job\_id

"JOB\_ID" VARCHAR2(10) CONSTRAINT "EMP\_JOB\_NN" NOT NULL ENABLE,

**Test No.**

 101

**Date**

11/15/2016

**Test Description**

Confirm NOT NULL constraint on job\_id

**Input**

INSERT INTO employees(employee\_id,first\_name,last\_name,email,phone\_number,hire\_date,job\_id,salary,commission\_pct,manager\_id,department\_id)

VALUES(300,'Hemant','Kumar','HKUMAR',NULL,TO\_DATE('2016-11-16','yyyy-mm-dd'),NULL,17000,null,100,90 );

**Expected Output**

Should not be able to insert NULL

**Result/Discrepancy**

  
**Action**

None

* 1. The business rule that requires that the end date of an employment is after a start date in the job history table.

CONSTRAINT "JHIST\_DATE\_INTERVAL" CHECK (end\_date > start\_date) ENABLE,

**Test No.**

 102

**Date**

11/15/2016

**Test Description**

End date must be greater than start date

**Input**

INSERT INTO job\_history(employee\_id,start\_date,end\_date,job\_id,department\_id)

VALUES(102,TO\_DATE('11-15-2016','mm-dd-yyyy'),TO\_DATE('06-17-1993','mm-dd-yyyy'),'AD\_ASST',90 );

**Expected Output**

Should not be able to insert since start\_date is greater than end\_date

**Result/Discrepancy**

  
**Action**

None

* 1. The business rule that states that departments can be closed down with employees in that department (resulting in the department\_id becoming unknown).

ALTER TABLE  "EMPLOYEES" ADD CONSTRAINT "EMP\_DEPT\_FK" FOREIGN KEY ("DEPARTMENT\_ID")

              REFERENCES  "DEPARTMENTS" ("DEPARTMENT\_ID") ENABLE

**Test No.**

 103

**Date**

11/15/2016

**Test Description**

When department is deleted from departments table, department\_id in employees table is set to NULL for the corresponding rows.

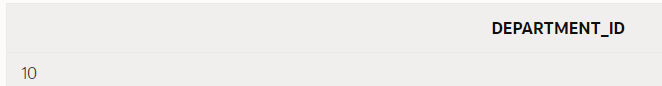
**Input**

DELETE FROM departments WHERE department\_id = 10;

*(Why I chose 10:*

*SELECT CONSTRAINT\_NAME , table\_name from user\_constraints where r\_constraint\_name = (SELECT r\_constraint\_name from user\_constraints where CONSTRAINT\_NAME= 'EMP\_DEPT\_FK');*

select department\_id from employees WHERE department\_id not in ( select nvl(department\_id , 0) from job\_history);



* 1. The minimum salary of an employee is 1000.

CONSTRAINT "EMP\_SALARY\_MIN" CHECK (salary > 0) ENABLE,

**Test No.**

 104

**Date**

11/15/2016

**Test Description**

The minimum salary of an employee is 1000. Should not be able to edit salary to NULL or less than 1000.

**Input**

UPDATE employees

SET salary = NULL

WHERE employee\_id = 144;

UPDATE employees

SET salary = 999

WHERE employee\_id = 144;

**Expected Output**

Both of the update statements should fail.

**Result/Discrepancy**



1. If one of the above tests fails, write out the SQL statement(s) that would be needed to correct the test. With the permission of your teacher, implement the change and then rerun the test with the same input and confirm that it works.

Problem 1 c Action:

**ALTER TABLE employees**

**DROP CONSTRAINT emp\_dept\_fk;**

**ALTER TABLE  employees**

**ADD CONSTRAINT emp\_dept\_fk   FOREIGN KEY (department\_id)**

**REFERENCES  departments (department\_id) ON DELETE SET NULL ;**

Now execute test id 103 again:

DELETE FROM departments WHERE department\_id = 10;



SELECT department\_id FROM employees WHERE employee\_id = 200;



Problem 1 d action:

**ALTER TABLE employees**

**MODIFY salary CONSTRAINT emp\_salary\_nn NOT NULL;**



**ALTER TABLE employees**

**DROP CONSTRAINT emp\_salary\_min;**



**UPDATE employees**

**SET salary = 1000 WHERE NVL(salary,0) < 1000;**



**ALTER TABLE employees**

**ADD CONSTRAINT emp\_salary\_min CHECK (salary >= 1000);**

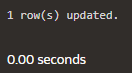


**Execute test 104 again:**

UPDATE employees

SET salary = NULL

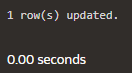
WHERE employee\_id = 144;



UPDATE employees

SET salary = 999

WHERE employee\_id = 144;



Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.