PL/SQL Tables & Exercises

Table of Contents

[EXERCISES 2](#_Toc381862231)

[Exercise 1 2](#_Toc381862232)

[Exercise 2 2](#_Toc381862233)

[Exercise 3 2](#_Toc381862234)

[Exercise 4 2](#_Toc381862235)

[Exercise 5 2](#_Toc381862236)

[Exercise 6 2](#_Toc381862237)

[Exercise 7 3](#_Toc381862238)

[Exercise 8 3](#_Toc381862239)

[Exercise 9 3](#_Toc381862240)

[Exercise 10 3](#_Toc381862241)

[Exercise 11 3](#_Toc381862242)

[Exercise 12 3](#_Toc381862243)

[Exercise 13 3](#_Toc381862244)

[Exercise 14 3](#_Toc381862245)

[Exercise 15 4](#_Toc381862246)

[Exercise 16 4](#_Toc381862247)

[PL/SQL Tables 5](#_Toc381862248)

[The Supplier Table 5](#_Toc381862249)

[The Part Table 5](#_Toc381862250)

[The Project Table 6](#_Toc381862251)

[The Supplier Part Project Table 6](#_Toc381862252)

[The Supplier Part Table 7](#_Toc381862253)

[The Scripts 8](#_Toc381862254)

[The Supplier Table 8](#_Toc381862255)

[The Part Table 10](#_Toc381862285)

[The Project Table 12](#_Toc381862325)

[The Supplier\_Part\_Project Table 14](#_Toc381862356)

[The Supplier\_Part\_Tab Table 18](#_Toc381862448)

## Exercises

Please note that these exercises are interconnected, and depending on performance issues, the procedures and functions created should be reused.

## Exercise 1

Write a procedure to print out error messages. Ideally, the procedure should have a set number of error messages within it, and these should be called as and when needed.

CREATE OR REPLACE PROCEDURE Print\_Error (err\_ IN NUMBER)

IS

BEGIN

CASE err\_

WHEN 1 THEN

Dbms\_Output.Put\_Line(’Error: Row not found’ );

WHEN 2 THEN

Dbms\_Output.Put\_Line(’Error: Row has changed (row\_version has changed)’);

WHEN 3 THEN

Dbms\_Output.Put\_Line(’Error: Row is already present.’);

WHEN 4 THEN

Dbms\_Output.Put\_Line(’Error: Not null columns are being assigned to nulls.’);

END CASE;

END print\_error;

### Exercise 2

Create a function that checks the existence of a record using its rowid value from the Supplier\_Part\_Project\_Tab table. Decide on the return data type.

CREATE OR REPLACE PROCEDURE If\_Exists (id\_ IN VARCHAR2)

RETURN BOOLEAN

IS

CURSOR Retrieve\_row IS

SELECT s.ROWID

FROM s.Supplier\_Part\_Project\_Tab;

BEGIN

FOR Index\_ IN Retrieve\_row

IF Index\_. ROWID = id\_

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END LOOP;

END If\_Exists;

### Exercise 3

Create a function that retrieves the row for a given row id from the Supplier\_Part\_Project\_Tab table.

Remember to raise an error if the:

1) Row has been deleted (row id not present).

CREATE OR REPLACE PROCEDURE Retrieve\_Record\_by\_rowid (id\_ IN VARCHAR2)

RETURN Supplier\_Part\_Project\_Tab%ROWTYPE

IS

Row Supplier\_Part\_Project\_Tab%ROWTYPE;

CURSOR Retrieve\_row IS

SELECT s.\*

FROM s.Supplier\_Part\_Project\_Tab

WHERE s.ROWID=id\_;

BEGIN

OPEN Retrieve\_row;

FETCH Retrieve\_row INTO Row;

IF (Retrieve\_row %NOTFOUND) THEN

Print\_Error (1);

END IF;

CLOSE Retrieve\_row;

RETURN Row;

END Retrieve\_Record\_by\_rowid;

### Exercise 4

Create a function that retrieves the row for a given primary key value from the Supplier\_Part\_Project\_Tab table.

Remember to raise an error if the:

1) Row has been deleted.

CREATE OR REPLACE PROCEDURE Retrieve\_Record (id\_ IN VARCHAR2)

RETURN Supplier\_Part\_Project\_Tab%ROWTYPE

IS

Row Supplier\_Part\_Project\_Tab%ROWTYPE;

CURSOR Retrieve\_row IS

SELECT s.\*

FROM s.Supplier\_Part\_Project\_Tab

WHERE s.SUPPLIER\_ID=id\_;

BEGIN

OPEN Retrieve\_row;

FETCH Retrieve\_row INTO Row;

IF (Retrieve\_row %NOTFOUND) THEN

Print\_Error (1);

END IF;

CLOSE Retrieve\_row;

RETURN Row;

END Retrieve\_Record;

### Exercise 5

Create a function that locks and returns a uniquely identified row for the Supplier\_Part\_Project\_Tab table. Use the primary keys to identify the row uniquely.

Remember to raise an error if the:

1) Row has been deleted.

CREATE OR REPLACE PROCEDURE Retrieve\_Record\_Lock (id\_ IN VARCHAR2)

RETURN Supplier\_Part\_Project\_Tab%ROWTYPE

IS

Row Supplier\_Part\_Project\_Tab%ROWTYPE;

CURSOR Retrieve\_row IS

SELECT s.\*

FROM s.Supplier\_Part\_Project\_Tab

WHERE s.SUPPLIER\_ID=id\_;

FOR UPDATE NOWAIT;

BEGIN

OPEN Retrieve\_row;

FETCH Retrieve\_row INTO Row;

IF (Retrieve\_row %NOTFOUND) THEN

Print\_Error (1);

END IF;

CLOSE Retrieve\_row;

RETURN Row;

END Retrieve\_row\_Lock;

### Exercise 6

Create a function that locks and returns a uniquely identified row for the Supplier\_Part\_Project\_Tab table. Use the rowid and the row version to identify the record uniquely.

Remember to raise an error if the:

1) Row has changed (row\_version has changed).

2) Row has been deleted.

CREATE OR REPLACE PROCEDURE Retrieve\_Record\_Lock (id\_ IN VARCHAR2, row\_version\_ IN VARCHAR2)

RETURN Supplier\_Part\_Project\_Tab%ROWTYPE

IS

Row Supplier\_Part\_Project\_Tab%ROWTYPE;

CURSOR Retrieve\_row IS

SELECT s.\*

FROM s.Supplier\_Part\_Project\_Tab

WHERE s.ROW\_ID=id\_ AND s.ROW\_VERSION= row\_version\_;

FOR UPDATE NOWAIT;

CURSOR Retrieve\_row\_no\_version IS

SELECT s.\*

FROM s.Supplier\_Part\_Project\_Tab

WHERE s.ROW\_ID=id;

BEGIN

OPEN Retrieve\_row;

FETCH Retrieve\_row INTO Row;

IF (Retrieve\_row %NOTFOUND) THEN

OPEN Retrieve\_row\_no\_version;

FETCH Retrieve\_row\_no\_version INTO Row\_no\_version;

IF (Retrieve\_row \_no\_version %NOTFOUND) THEN

Print\_Error (1);

ELSE

Print\_Error (2);

CLOSE Retrieve\_row;

END IF;

CLOSE Retrieve\_row;

RETURN Row;

END Retrieve\_row\_Lock;

### Exercise 7

Create a procedure that retrieves the row id and the row version for a given primary key value from the Supplier\_Part\_Project\_Tab table.

CREATE OR REPLACE PROCEDURE Retrieve\_Record\_Ver (id\_ IN VARCHAR2)

RETURN row\_id\_ver\_type

IS

TYPE row\_id\_ver\_type IS RECORD

(row\_id NUMBER,

Row\_version DATE);

Row row\_id\_ver\_type;

CURSOR Retrieve\_row IS

SELECT s.ROW\_ID, s.ROW\_VERSION

FROM s.Supplier\_Part\_Project\_Tab

WHERE s.SUPPLIER\_ID=id\_;

BEGIN

OPEN Retrieve\_row;

FETCH Retrieve\_row INTO Row;

IF (Retrieve\_row %NOTFOUND) THEN

Print\_Error (1);

END IF;

CLOSE Retrieve\_row;

RETURN Row;

END Retrieve\_row\_Ver;

### Exercise 8

Create a procedure that inserts a record into the Supplier\_Part\_Project\_Tab table.

Remember to raise an error if the:

1) Row is already present.

2) Not null columns are being assigned to nulls.

**Note:** Use Oracle errors for checking the not null constraint and row duplicates.

CREATE OR REPLACE PROCEDURE Insert\_Sup\_Part

(Supplier\_Id IN NUMBER,

Part\_Id IN VARCHAR2,

Project \_Id IN VARCHAR2,

Quantity IN NUMBER

)

IS

Duplicate\_row EXCEPTION;

Null\_insertion EXCEPTION;

PRAGMA EXCEPTION\_INIT(Duplicate\_row, -00001);

PRAGMA EXCEPTION\_INIT(Null\_insertion, -01400);

BEGIN

INSERT INTO Supplier\_Part\_Project\_Tab

VALUES(Supplier\_Id, Part\_Id, Project \_Id , Quantity, SYSDATE);

EXCEPTION

WHEN Duplicate\_row THEN

Print\_Error (3);

WHEN Null\_insertion THEN

Print\_Error (4);

END Insert\_Sup\_Part;

### Exercise 9

Create a procedure that updates a record in the Supplier\_Part\_Project\_Tab table.

Remember to raise an error if the:

1) Row is not found.

2) Not null columns are being assigned nulls.

CREATE OR REPLACE PROCEDURE Update\_Sup\_Part

(Sup\_Id IN NUMBER,

P\_Id IN VARCHAR2,

Pr \_Id IN VARCHAR2,

Qty IN NUMBER

)

IS

No\_row EXCEPTION;

Null\_insertion EXCEPTION;

PRAGMA EXCEPTION\_INIT(No\_row, NO\_DATA\_FOUND);

PRAGMA EXCEPTION\_INIT(Null\_insertion, -01400);

BEGIN

INSERT INTO Supplier\_Part\_Project\_Tab

SET QUANTITY = Qty, ROW\_VERSION=SYSDATE

WHERE Supplier\_Id= Sup\_Id AND Part\_Id=P\_Id AND Project \_Id=Pr\_Id ;

EXCEPTION

WHEN No\_row THEN

Print\_Error (1);

WHEN Null\_insertion THEN

Print\_Error (4);

END Update\_Sup\_Part;

### Exercise 10

Create a procedure that deletes a record in the Supplier\_Part\_Project\_Tab table.

CREATE OR REPLACE PROCEDURE Delete\_Sup\_Part

(Sup\_Id IN NUMBER,

P\_Id IN VARCHAR2,

Pr \_Id IN VARCHAR2

)

IS

No\_row EXCEPTION;

PRAGMA EXCEPTION\_INIT(No\_row, NO\_DATA\_FOUND);

BEGIN

DELETE FROM Supplier\_Part\_Project\_Tab

WHERE Supplier\_Id= Sup\_Id AND Part\_Id=P\_Id AND Project \_Id=Pr\_Id;

EXCEPTION

WHEN No\_row THEN

Print\_Error (1);

END Delete\_Sup\_Part;

**Note:** The previous exercises where meant to form the implementation layer for a package that is to be created in the last exercise. The next three exercises will form private methods that utilize the implementation methods.

### Exercise 11

Create a procedure that creates a new record in the Supplier\_Part\_Project\_Tab table.

CREATE OR REPLACE PROCEDURE Insert\_New\_Supplier\_Part\_Project

(Supplier\_Id IN NUMBER,

Part\_Id IN VARCHAR2,

Project \_Id IN VARCHAR2,

Quantity IN NUMBER

)

IS

BEGIN

Insert\_Sup\_Part (Supplier\_Id, Part\_Id, Project \_Id , Quantity);

END Insert\_New\_Supplier\_Part\_Project;

### Exercise 12

Create a procedure that modifies a record in the Supplier\_Part\_Project\_Tab table.

CREATE OR REPLACE PROCEDURE Update\_Supplier\_Part\_Project

(Supplier\_Id IN NUMBER,

Part\_Id IN VARCHAR2,

Project \_Id IN VARCHAR2,

Quantity IN NUMBER

)

IS

BEGIN

Update\_Sup\_Part (Supplier\_Id, Part\_Id, Project \_Id , Quantity);

END Update\_Supplier\_Part\_Project;

### Exercise 13

Create a procedure that removes a record in the Supplier\_Part\_Project\_Tab table.

CREATE OR REPLACE PROCEDURE Delete\_Supplier\_Part\_Project

(Supplier\_Id IN NUMBER,

Part\_Id IN VARCHAR2,

Project \_Id IN VARCHAR2,

)

IS

BEGIN

Delete\_Sup\_Part(Supplier\_Id, Part\_Id, Project \_Id);

END Delete\_Supplier\_Part\_Project;

**Note:** The next two methods will from public methods for the package.

### Exercise 14

Create a method that retrieves the quantity supplied of a part in a given project for a given supplier from the Supplier\_Part\_Project\_Tab table.

CREATE OR REPLACE PROCEDURE Retrieve\_Quantity

(Supplier\_Id IN NUMBER,

Part\_Id IN VARCHAR2,

Project \_Id IN VARCHAR2,

)

RETURN NUMBER

IS

Qty NUMBER

CURSOR Get\_qty

SELECT s.QUANTITY

FROM s.Supplier\_Part\_Project\_Tab

WHERE Supplier\_Id= Sup\_Id AND Part\_Id=P\_Id AND Project \_Id=Pr\_Id;

BEGIN

OPEN Get\_qty;

FETCH Get\_qty INTO Qty;

IF (Retrieve\_row %NOTFOUND) THEN

Print\_Error (1);

END IF;

CLOSE Get\_qty;

RETURN Row;

END Retrieve\_row\_Ver;

### Exercise 15

Provide a method for increasing the quantity supplied of a given part for a given project. Check with the available quantities for each supplier. Remove the stocks from the suppliers, and added them to the Supplier\_Part\_Project\_Tab table. If a supplier does not have enough parts to supply, add the functionality to add the quantity from a second supplier. If all the suppliers are exhausted, then print a message stating the part number with the quantity required.

CREATE OR REPLACE Stock\_change (

P\_id IN NUMBER,

Pr\_id IN VARCHAR2,

)

IS

BEGIN

END Stock\_change;

### Exercise 16

Create two packages. The first package should contain exercise 1. The second package should contain the exercises 2-15. Make the necessary changes to call the procedure made for exercise 1 from within the other procedures and function in the second package.

CREATE OR REPLACE PACKAGE EX\_ONE

IS

END EX\_ONE;

## PL/SQL Tables

### The Supplier Table

|  |  |  |
| --- | --- | --- |
| **Supplier Table** | | |
| **Field Name** | **Field Type** | **Field Length** |
| Supplier\_Id | NUMBER |  |
| Suplier\_Name | STRING | 25 |
| City | STRING | 25 |
| Row\_Version | DATE |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Supplier Table** | | | | |
| **Supplier\_Id** | **Supplier\_Name** | **City** | **Row\_Version** |
| 1001 | Smithson | Hamburg | SYSDATE |
| 1002 | Robinson | Stockholm | SYSDATE |
| 1003 | Robert | Linden | SYSDATE |
| 1004 | Richardson | Dublin | SYSDATE |
| 1005 | Peterson | Hamburg | SYSDATE |

### The Part Table

|  |  |  |
| --- | --- | --- |
| **Part Table** | | |
| **Field Name** | **Field Type** | **Field Length** |
| Part\_Id | NUMBER |  |
| Part\_Name | STRING | 25 |
| Colour | STRING | 10 |
| Weight | NUMBER |  |
| City | STRING | 25 |
| Row\_Version | DATE |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part Table** | | | | | |
| **Part\_Id** | **Part\_Name** | **Colour** | **Weight** | **City** | **Row\_Version** |
| 100 | Nut 5’ | Brown | 5 | Stockholm | SYSDATE |
| 101 | Bolt 5’ | Brown | 10 | Stockholm | SYSDATE |
| 102 | Cam | Red | 25 | Dublin | SYSDATE |
| 103 | Cog | Grey | 15 | Linden | SYSDATE |
| 104 | Nut 2’ | Black | 2 | Hamburg | SYSDATE |

### The Project Table

|  |  |  |
| --- | --- | --- |
| **Project Table** | | |
| **Field Name** | **Field Type** | **Field Length** |
| Project \_Id | NUMBER |  |
| Project\_Name | STRING | 25 |
| City | STRING | 25 |
| Row\_Version | DATE |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Table** | | | |
| **Project \_Id** | **Project\_Name** | **City** | **Row\_Version** |
| 2001 | K-199 | Rome | SYSDATE |
| 2002 | G-891 | Oslo | SYSDATE |
| 2003 | G-341 | New York | SYSDATE |
| 2004 | K-201 | Linden | SYSDATE |
| 2005 | D-878 | Athens | SYSDATE |

### The Supplier Part Project Table

|  |  |  |
| --- | --- | --- |
| **Supplier\_Part\_Project\_Tab Table** | | |
| **Field Name** | **Field Type** | **Field Length** |
| Supplier\_Id | NUMBER |  |
| Part\_Id | STRING | 5 |
| Project \_Id | STRING | 5 |
| Quantity | NUMBER |  |
| Row\_Version | DATE |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Supplier\_Part\_Project\_Tab Table | | | | |
| **Supplier\_Id** | **Part\_Id** | **Project \_Id** | **Quantity** | **Row\_Version** |
| 1002 | 100 | 2001 | 50000 | SYSDATE |
| 1002 | 101 | 2001 | 50000 | SYSDATE |
| 1002 | 100 | 2002 | 50 | SYSDATE |
| 1002 | 100 | 2003 | 15000 | SYSDATE |
| 1002 | 101 | 2003 | 15000 | SYSDATE |
| 1002 | 100 | 2004 | 150 | SYSDATE |
| 1002 | 100 | 2005 | 50 | SYSDATE |
| 1003 | 103 | 2001 | 250 | SYSDATE |
| 1003 | 103 | 2002 | 250 | SYSDATE |
| 1004 | 102 | 2001 | 100 | SYSDATE |
| 1004 | 102 | 2002 | 500 | SYSDATE |
| 1005 | 104 | 2001 | 15000 | SYSDATE |
| 1005 | 104 | 2003 | 500 | SYSDATE |

### The Supplier Part Table

|  |  |  |
| --- | --- | --- |
| **Supplier\_Part\_Tab Table** | | |
| **Field Name** | **Field Type** | **Field Length** |
| Supplier\_Id | NUMBER |  |
| Part\_Id | STRING | 2 |
| Quantity | NUMBER |  |
| Row\_Version | DATE |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Supplier\_Part\_Tab Table | | | |
| **Supplier\_Id** | **Part\_Id** | **Quantity** | **Row\_Version** |
| 1001 | 100 | 50000 | SYSDATE |
| 1001 | 101 | 50000 | SYSDATE |
| 1002 | 100 | 50000 | SYSDATE |
| 1002 | 102 | 15000 | SYSDATE |
| 1003 | 103 | 250000 | SYSDATE |
| 1004 | 101 | 25000 | SYSDATE |
| 1004 | 102 | 24000 | SYSDATE |
| 1005 | 104 | 500000 | SYSDATE |

## The Scripts

### The Supplier Table

**CREATE TABLE Supplier\_Tab**

**(Supplier\_Id NUMBER NOT NULL,**

**Supplier\_Name VARCHAR2(25) NOT NULL,**

**City VARCHAR2(25) NULL,**

**Quantity\_Ordered NUMBER NULL,**

**Row\_Version DATE NOT NULL);**

**ALTER TABLE Supplier\_Tab**

ADD (CONSTRAINT supplier\_tab\_pk PRIMARY KEY (Supplier\_Id));

**COMMIT;**

**INSERT INTO Supplier\_Tab**

**(Supplier\_Id,**

**Supplier\_Name,**

City,

**Quantity\_Ordered,**

**Row\_Version)**

**VALUES**

**(1001,**

**‘Smithson’,**

**‘Hamburg’,**

**5000,**

**SYSDATE);**

**INSERT INTO Supplier\_Tab**

**(Supplier\_Id,**

**Supplier\_Name,**

**City,**

**Quantity\_Ordered,**

**Row\_Version)**

**VALUES**

**(1002,**

**‘Robinson’,**

**‘Stockholm’,**

**20000,**

**SYSDATE);**

**INSERT INTO Supplier\_Tab**

**(Supplier\_Id,**

**Supplier\_Name,**

**City,**

**Quantity\_Ordered,**

**Row\_Version)**

**VALUES**

**(1003,**

**‘Robert’,**

**‘Linden’,**

**50000,**

**SYSDATE);**

**INSERT INTO Supplier\_Tab**

**(Supplier\_Id,**

**Supplier\_Name,**

**City,**

**Quantity\_Ordered,**

**Row\_Version)**

**VALUES**

**(1004,**

**‘Richardson’,**

**‘Dublin’,**

**2500,**

**SYSDATE);**

**INSERT INTO Supplier\_Tab**

**(Supplier\_Id,**

**Supplier\_Name,**

**City,**

**Quantity\_Ordered,**

**Row\_Version)**

**VALUES**

**(1005,**

**‘Peterson’,**

**‘Hamburg’,**

**200000,**

**SYSDATE);**

**COMMIT;**

### The Part Table

**CREATE TABLE Part\_Tab**

**(Part\_Id NUMBER NOT NULL,**

**Part\_Name VARCHAR2(25) NOT NULL,**

**Colour VARCHAR2(10) NULL,**

**Weight NUMBER NULL,**

**City VARCHAR2(25) NULL,**

**Row\_Version DATE NOT NULL);**

**ALTER TABLE Part\_Tab**

**ADD(CONSTRAINT part\_tab\_pk**

**PRIMARY KEY (Part\_Id));**

**COMMIT;**

INSERT INTO Part\_Tab

**(Part\_Id,**

**Part\_Name,**

**Colour,**

**Weight,**

**City,**

**Row\_Version)**

**VALUES**

**(100,**

**‘Nut 5’,**

**‘Brown’,**

**5,**

**‘Stockholm’,**

**SYSDATE);**

**INSERT INTO Part\_Tab**

**(Part\_Id,**

**Part\_Name,**

**Colour,**

**Weight,**

**City,**

**Row\_Version)**

**VALUES**

**(101,**

**‘Bolt 5’,**

**‘Brown’,**

**10,**

**‘Stockholm’,**

**SYSDATE);**

**INSERT INTO Part\_Tab**

**(Part\_Id,**

**Part\_Name,**

**Colour,**

**Weight,**

**City,**

**Row\_Version)**

**VALUES**

**(102,**

**‘Cam’,**

**‘Red’,**

**25,**

**‘Dublin’,**

**SYSDATE);**

**INSERT INTO Part\_Tab**

**(Part\_Id,**

**Part\_Name,**

**Colour,**

**Weight,**

**City,**

**Row\_Version)**

**VALUES**

**(103,**

**‘Cog’,**

**‘Grey’,**

**15,**

**‘Linden’,**

**SYSDATE);**

**INSERT INTO Part\_Tab**

**(Part\_Id,**

**Part\_Name,**

**Colour,**

**Weight,**

**City,**

**Row\_Version)**

**VALUES**

**(104,**

**‘Nut 2’,**

**‘Black’,**

**2,**

**‘Hamburg’,**

**SYSDATE);**

**COMMIT;**

### The Project Table

**CREATE TABLE Project\_Tab**

**(Project\_Id NUMBER NOT NULL,**

**Project\_Name VARCHAR2(25) NOT NULL,**

**City VARCHAR2(25) NULL,**

**Row\_Version DATE NOT NULL);**

**ALTER TABLE Project\_Tab**

**ADD(CONSTRAINT project\_tab\_pk**

**PRIMARY KEY (Project\_Id));**

**COMMIT;**

**INSERT INTO Project\_Tab**

**(Project\_Id,**

**Project\_Name,**

**City,**

**Row\_Version)**

**VALUES**

**(2001,**

**‘K-199’,**

**‘Rome’,**

**SYSDATE);**

**INSERT INTO Project\_Tab**

**(Project\_Id,**

**Project\_Name,**

**City,**

**Row\_Version)**

**VALUES**

**(2002,**

**‘G-891’,**

**‘Oslo’,**

**SYSDATE);**

**INSERT INTO Project\_Tab**

**(Project\_Id,**

**Project\_Name,**

**City,**

**Row\_Version)**

**VALUES**

**(2003,**

**‘G-341’,**

**‘New York’,**

**SYSDATE);**

**INSERT INTO Project\_Tab**

**(Project\_Id,**

**Project\_Name,**

**City,**

**Row\_Version)**

**VALUES**

**(2004,**

**‘K-201’,**

**‘Linden’,**

**SYSDATE);**

**INSERT INTO Project\_Tab**

**(Project\_Id,**

**Project\_Name,**

**City,**

**Row\_Version)**

**VALUES**

**(2005,**

**‘D-878’,**

**‘Athens’,**

**SYSDATE);**

COMMIT;

### The Supplier\_Part\_Project Table

**CREATE TABLE Supplier\_Part\_Project\_Tab**

**(Supplier\_Id NUMBER NOT NULL,**

**Part\_Id NUMBER NOT NULL,**

**Project\_Id NUMBER NOT NULL,**

**Quantity NUMBER NOT NULL,**

**Row\_Version DATE NOT NULL);**

**ALTER TABLE Supplier\_Part\_Project\_Tab**

**ADD(CONSTRAINT supplier\_part\_project\_tab\_pk**

**PRIMARY KEY (Supplier\_Id, Part\_Id, Project\_Id));**

COMMIT;

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1002,** **100,**

**2001,**

**50000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1002,**

**101,**

**2001,**

**50000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1002,**

**100,**

**2002,**

**50,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1002,**

**100,**

**2003,**

**15000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1002,**

**101,**

**2003,**

**15000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1002,**

**100,**

**2004,**

**150,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1002,**

**100,**

**2005,**

**50,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1003,**

**103,**

**2001,**

**250,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1003,**

**103,**

**2002,**

**250,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1004,**

**102,**

**2001,**

**100,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1004,**

**102,**

**2002,**

**500,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1005,**

**104,**

**2001,**

**15000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Project\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Project\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1005,**

**104,**

**2003,**

**500,**

**SYSDATE);**

**COMMIT;**

The Supplier\_Part\_Tab Table

**CREATE TABLE Supplier\_Part\_Tab**

**(Supplier\_Id NUMBER NOT NULL,**

**Part\_Id NUMBER NOT NULL,**

**Quantity NUMBER NOT NULL,**

**Row\_Version DATE NOT NULL);**

**ALTER TABLE Supplier\_Part\_Tab**

**ADD(CONSTRAINT supplier\_part\_tab\_pk**

**PRIMARY KEY (Supplier\_Id, Part\_Id));**

**COMMIT;**

**INSERT INTO Supplier\_Part\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1001,**

**100,**

**50000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1001,**

**101,**

**50000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1002,**

**100,**

**50000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1002,**

**102,**

**15000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1003,**

**103,**

**250000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1004,**

**101,**

**25000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1004,**

**102,**

**24000,**

**SYSDATE);**

**INSERT INTO Supplier\_Part\_Tab**

**(Supplier\_Id,**

**Part\_Id,**

**Quantity,**

**Row\_Version)**

**VALUES**

**(1005,**

**104,**

**500000,**

**SYSDATE);**

**COMMIT;**