

# Relational Database Management System



## Education, Training and Assessment

We enable you to leverage knowledge anytime, anywhere!

**Infosys**<sup>®</sup> | Building Tomorrow's Enterprise

# Copyright Guideline

© 2014 Infosys Limited, Bangalore, India. All Rights Reserved.

Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.

# Confidential Information

- This Document is confidential to Infosys Limited. This document contains information and data that Infosys considers confidential and proprietary (“Confidential Information”).
- Confidential Information includes, but is not limited to, the following:
  - ❑ Corporate and Infrastructure information about Infosys
  - ❑ Infosys’ project management and quality processes
  - ❑ Project experiences provided included as illustrative case studies
- Any disclosure of Confidential Information to, or use of it by a third party, will be damaging to Infosys.
- Ownership of all Infosys Confidential Information, no matter in what media it resides, remains with Infosys.
- Confidential information in this document shall not be disclosed, duplicated or used – in whole or in part – for any purpose other than reading without specific written permission of an authorized representative of Infosys.
- This document also contains third party confidential and proprietary information. Such third party information has been included by Infosys after receiving due written permissions and authorizations from the party/ies. Such third party confidential and proprietary information shall not be disclosed, duplicated or used – in whole or in part – for any purpose other than reading without specific written permission of an authorized representative of Infosys.

# Course Information

- Course Code: CCFP4.0-RDBMS
- Course Name: Relational Database Management System
- Document Number: RDBMS-05
- Version Number: 4.0

# SQL Best Practices



# Best practices for writing SQL queries

- **Coding SQL statements to avoid unnecessary processing**

By keeping your SQL statements simple, you can limit the amount of processing that they require.

- **Using predicates efficiently**

You can improve how DB processes SQL statements by following certain practices when writing predicates.

- **Writing efficient subqueries**

By understanding how DB processes subqueries, you can estimate the best method to use when writing a given query when several methods can achieve the same result.

Guided Activity : CCFP4.0-RDBMSAssignments – Assignment on SQL Best Practices - 1 (Estimated Time: 30 mins.)

VIEW





# Views

Views are database objects or virtual tables which are formed by a SQL query. Views are mainly used for secured access of a data.

2002	Jason	jason@adgm.com	01-10-1999
2003	Smith	smith@xyz.corp	12-05-2001
2004	Mona	Mona@xyz.com	30-Nov-2004
2005	Sam	Sam@adgm.com	22-Oct-2006

customerid	customername
2003	Smith
2004	Mona
2005	Sam

Base Table  
Customer

To be displayed as a view



# Views

## Syntax:

```
CREATE [OR REPLACE] VIEW viewname
    [(alias[, alias]...)]
AS subquery
[WITH CHECK OPTION]
[WITH READ ONLY]

-- To read the data through view
SQL> SELECT * FROM viewname;
```

Demo : 321-001-Assignments\_DBMS, Views, Assignment 1

Guided Activity : CCFP4.0-RDBMSAssignments – Assignment on Views 1, 2  
(Estimated Time: 40 mins.)

## Restrictions on views

- A view can be updated if the query that defines the view meets all of these restrictions:
  - DISTINCT must not be specified; that is, duplicate rows must not be eliminated from the query results
  - The FROM clause must specify only one updateable table; the view must have a single underlying source table
  - The SELECT list cannot contain expressions, calculated columns, or column functions
  - The WHERE clause must not include a sub query; only simple row-by-row search conditions may appear

# Transaction

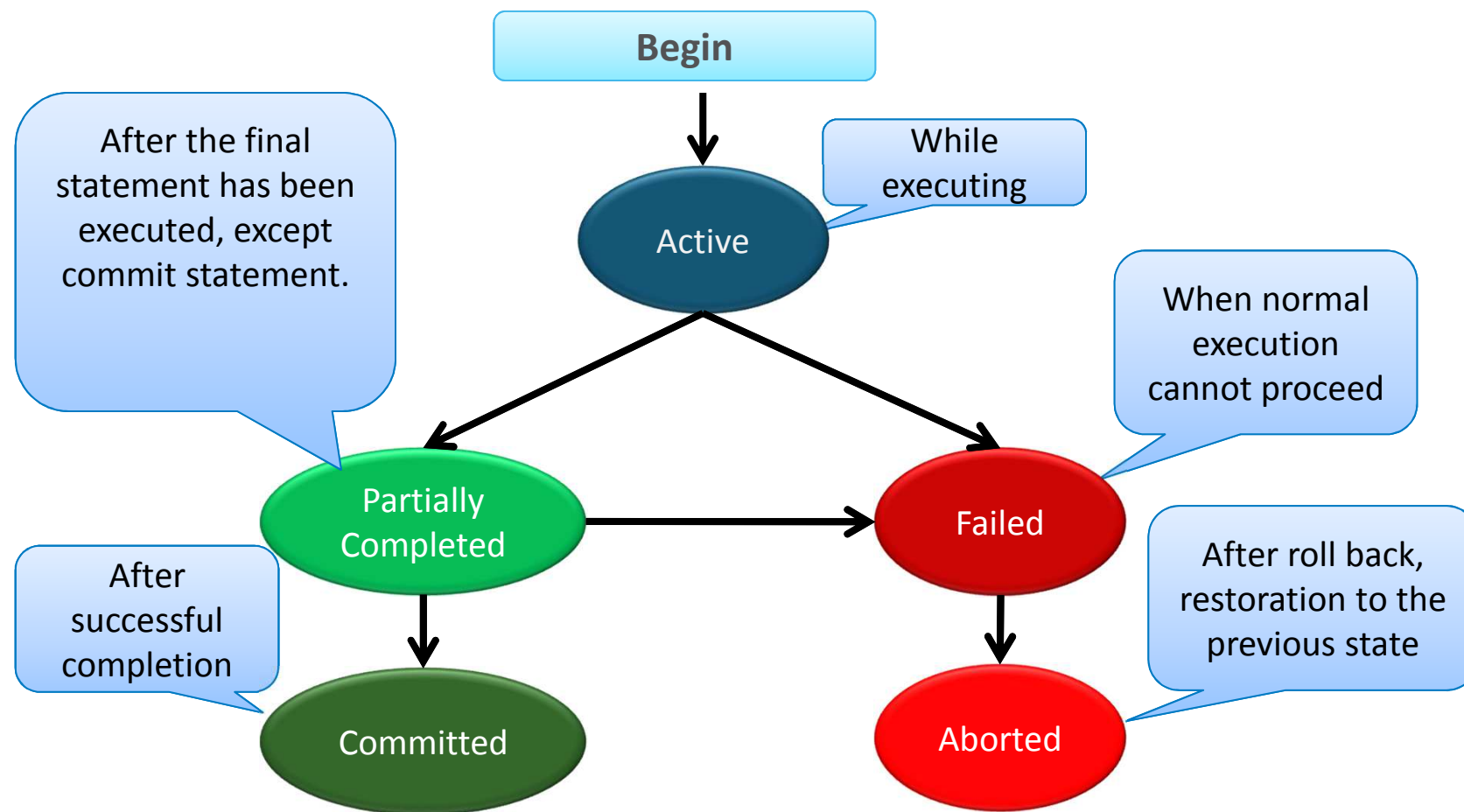


# Transaction

- Consider any of the following scenarios
  1. Purchasing an item in EasyShop application
  2. Money transfer from one account to another
  3. Online reservation of railway tickets or air tickets
  4. Online Shopping
- How many discrete steps are involved in each of the above scenario?
- In case of money transfer scenario what could happen if the activity is interrupted after money is debited from one account and before being credited to the other?
- How do we ensure that the data remains in a consistent state in spite of failures?

**A transaction is a logical unit of work that takes a database from one consistent state to another consistent state**

# State diagram of a transaction



# Transaction properties and types

- **Properties of a transaction**
  - Atomicity
  - Consistency
  - Isolation
  - Durability
- **Types of transaction processing**
  - Batch transaction Processing
  - Real-time transaction Processing
  - Online transaction processing (OLTP)

# Requirements for OLTP

- **Integrity**

- All the data entering into the system must be validated for its correctness and adherence to the organization's business rules
- Data integrity constraints
  - Required data (NOT NULL constraint)
  - Domain integrity (CHECK constraint)
  - Entity integrity (PRIMARY KEY constraint)
  - Referential integrity (FOREIGN KEY constraint)

- **Concurrency**

- Allowing two or more users access a database concurrently
- Problems associated with concurrent execution are:
  - Lost update , Dirty read and Incorrect summary

- **Security, Backup and Recovery**



# Issues in concurrency - Lost update

A store keeper is adding 25 units of item I1001 in retail outlet R1001. Simultaneously 5 units of item I1001 is being purchased from retail outlet R1001 by a customer. Sequence of the transaction is shown below:

What is your observation ?

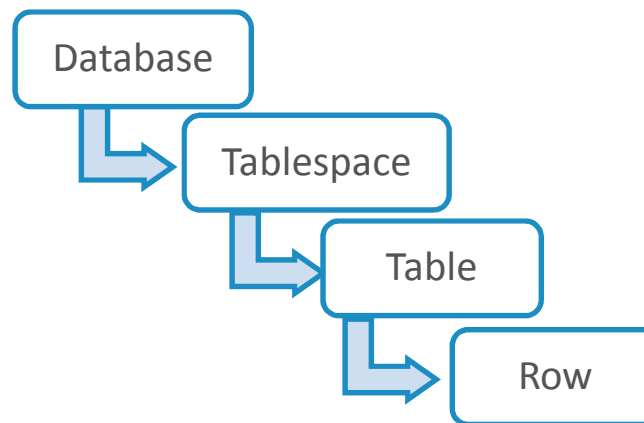
Time	Store Keeper's Transaction	Quantity of I1001	Customer's Transaction
10:22	Read qtyavailable of item 'I1001'	25	
10:23	qtyavailable = 25+25		
10:24			read qtyavailable (25)
10:25	Write new qtyavailable	50	
10:26	Commit		
10:27			qtyavailable = 25 - 5
10:28		20	Write new qtyavailable
10:29			Commit

How many items are lost during these transactions ? How can this be resolved ?

25 Items has been lost. Solution to issues in concurrency is serialization through locking.

# Transaction serialization through database locks

- Transaction serialization can be achieved through locking
- A lock is a variable associated with each data item in a database
- The DBMS locks the data item when it is updated by a transaction
- Lock could be **Shared (S)** or **Exclusive (X)**
- Locks are released by applying COMMIT / ROLLBACK
- Locks can be applied at various granular levels



Guided Activity : CCFP4.0-RDBMSAssignments – Assignment on Transactions and locks - 2  
(Estimated Time: 15 mins.)

## Deadlock – Guided activity

User-1

```
1. UPDATE emp SET sal=7000  
   WHERE empno=7900;
```

```
3. UPDATE emp SET sal=7000  
   WHERE empno=7902;
```

User-2

```
2. UPDATE emp SET sal=6000  
   WHERE empno=7902;
```

```
4. UPDATE emp SET sal=6000  
   WHERE empno=7900;
```

ORA-00060: deadlock detected while waiting for resource

Note: This situation is deadlock. Oracle is smart enough to detect it and it rolls back the statement present in step 4.

# Procedural Language SQL



## EasyShop case study

A customer purchases an item from EasyShop. Customer is eligible for a discount based on his/her total bill amount. Based on the discount, net bill has to be computed. After successful bill generation, these details have to be recorded in the database



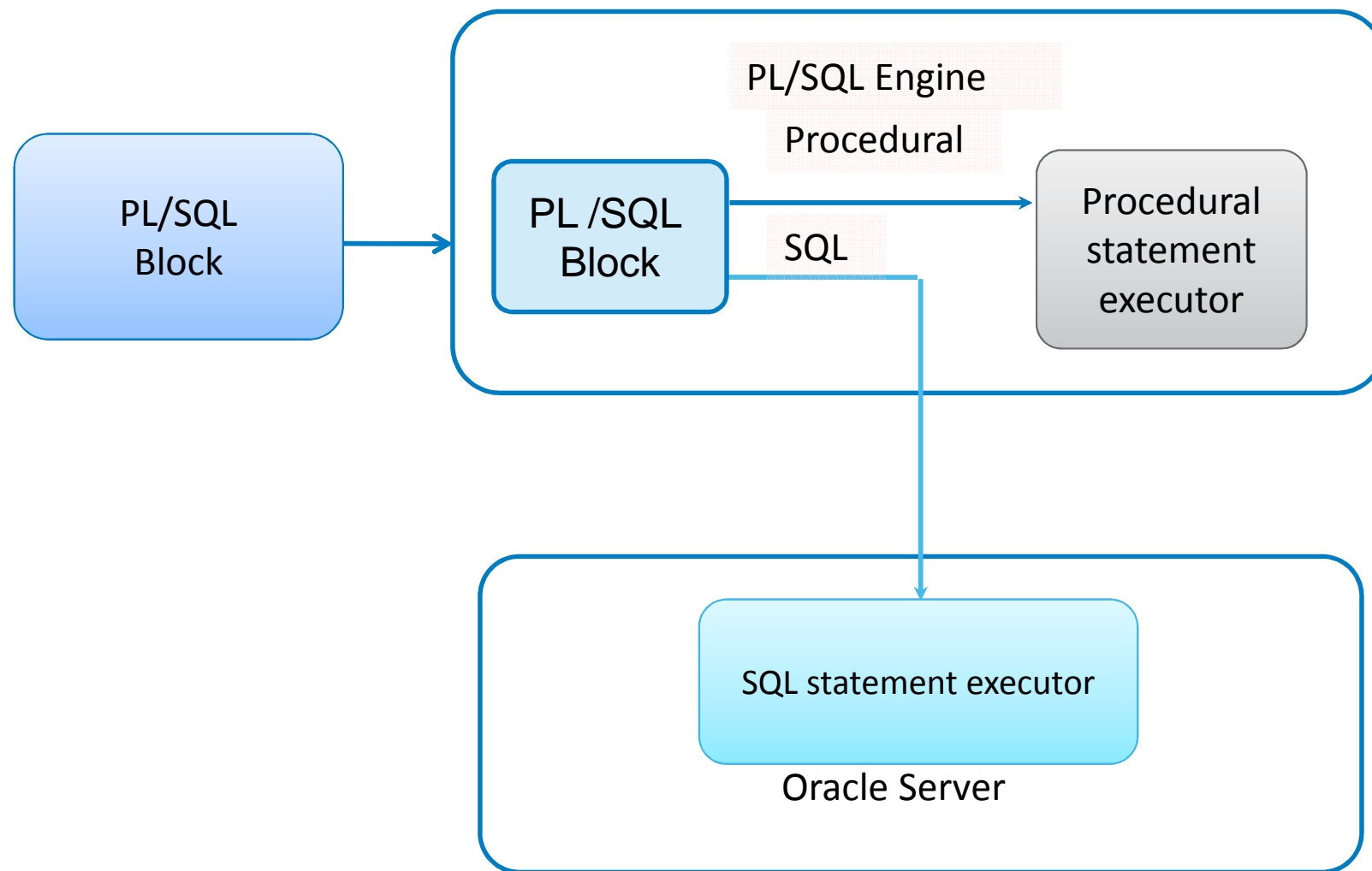
Can this requirement be implemented using SQL ?

- No !
- The data retrieved using SQL need to be further processed
- This can be implemented using a sequence of SQL statements as a unit
- Different DBMS have different ways to implement this requirement
- PL/SQL is one such technology provided by Oracle to implement procedural logic.

# Implementing transaction with PL/SQL

- PL/SQL block structure allows us to group a set of DML statements which does a logical unit of work
- Changes made by these series of DML statements can be committed (permanently saved) to the database or rolled back (undone) immediately
- If the program fails in the middle of the transaction due to some reasons such as system crash or memory crash, then the database is automatically restored to its earlier state

# PL/SQL Architecture





## Sample PL/SQL

DECLARE ←

```
v_totalbillamt NUMBER:=2500;  -- declare and assign  
v_discount NUMBER:=10;  
v_netbillamt NUMBER;  
v_billid NUMBER := 5466;  
v_retailoutletid VARCHAR2(10):= 'R1001';  
v_customerid NUMBER := 2001;  
v_discountamt NUMBER;  
v_qtypurchased NUMBER := 25;  
v_itemid VARCHAR2(10) := 'I1005';
```

BEGIN ←

```
v_discountamt := (v_totalbillamt * v_discount)/100;  
v_netbillamt:= (v_totalbillamt - v_discountamt);
```

```
DBMS_OUTPUT.PUT_LINE('Net bill amount ' || v_netbillamt);
```

```
COMMIT;  
UPDATE item SET quantityonhand = quantityonhand - v_qtypurchased WHERE itemcode =  
        v_itemid;  
INSERT INTO purchasebill (billid,retailoutletid,customerid,billamount,billdate)  
        VALUES(v_billid,v_retailoutletid,v_customerid, v_netbillamt,SYSDATE);  
COMMIT;
```

END; ←

## Using SQL SELECT in PL/SQL - guided activity

```
-- To select an item and display
SET SERVEROUTPUT ON
DECLARE
    v_quantityonhand NUMBER;
BEGIN
    SELECT quantityonhand INTO v_quantityonhand FROM item
    WHERE itemcode='I1001';

    DBMS_OUTPUT.PUT_LINE('Quantity On Hand : ' ||
        v_quantityonhand);
END;
```

## Using SQL INSERT in PL/SQL - guided activity

- INSERT can be used in PL/SQL block as it used is in SQL
- The following example is a direct insertion of values to the respective columns

```
--Inserting values to customer table directly by providing  
--values  
  
BEGIN  
    INSERT INTO customer (customerid, customername, mobileno)  
        VALUES (2011, 'JAMES', 0018695641);  
  
END;
```

## Using SQL UPDATE in PL/SQL - guided activity

- UPDATE can be used in PL/SQL block as it is used in SQL
- The following example is a direct update of value of the respective column

```
--To update the customer contact number where customerid is  
--2001  
  
BEGIN  
    UPDATE customer SET mobileno = 4567892341  
    WHERE customerid = 2001;  
  
END;
```

## Using SQL DELETE in PL/SQL - guided activity

- DELETE can be used in PL/SQL block as it is used in SQL
- The below PL/SQL is an example of deleting a record for a given customer id

```
--To delete a customer record where customerid is equal  
--to 2001  
  
BEGIN  
    DELETE FROM customer WHERE customerid = 2001;  
  
END;
```

Guided Activity : CCFP4.0-RDBMSAssignments – Assignments on PL/SQL - 1 , 2  
(Estimated Time: 30 mins.)

# Cursor



# Cursor

- A Cursor is a private SQL work area
- Every SQL statement executed by the Oracle Server has an individual cursor associated with it
- There are two types of cursors
  - Implicit Cursors
    - Declared for all DML and PL/SQL SELECT statements
  - Explicit Cursors
    - Declared and named by the programmer
- The Cursor points to a memory region called the context area that holds the following:
  - Rows returned by the query
  - Number of rows processed by the query
  - A pointer to the parsed query



## Implicit Cursor Attributes

- Using SQL cursor attributes, you can test the outcome of your SQL statements

Implicit Cursor Attribute	Meaning
SQL%ROWCOUNT	Number of records affected by the most recent SQL statement
SQL%FOUND	Evaluates to TRUE if the most recent SQL statement affects one or more rows
SQL%NOTFOUND	Evaluates to TRUE if the most recent SQL statement does not affect any rows
SQL%ISOPEN	Always evaluates to FALSE because PL/SQL closes implicit cursors immediately after they are executed

## Implicit Cursor Attribute - Example

- Update the delivery status of all items as 'Delivered' in the orderstatus table where the order date falls after 10-NOV-2010

```
BEGIN
    UPDATE orderstatus SET status='Delivered' WHERE orderdate > '10-NOV-2010';
    DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT || ' rows updated');
    IF SQL%NOTFOUND THEN
        DBMS_OUTPUT.PUT_LINE('No orders after 10-NOV-2010');
    END IF;
    COMMIT;
END;
```

On Successful update: 5 rows updated

On unsuccessful update:  
0 rows updated  
No orders after 10-NOV-2010

Guided Activity: CCFP4.0-RDBMSAssignments – Assignments on PL/SQL - 3 (Estimated Time: 30 mins.)

# Summary

- SQL best practices
- Views
- Transaction
  - State diagram of a transaction
  - Transaction properties and types
  - OLTP
    - Concurrency control
- Procedural language SQL

# Self-Study

Refer to NPTEL course: <http://nptel.ac.in/courses.php>

Course : Course : NPTEL >> Computer Science and Engineering >> **Database Design**

**Videos:**

Transaction Processing Concepts

Foundation for Concurrency Control

Case Study - Part One Database Design

Case Study - Part 2 Database Design

Refer to:

<https://class.stanford.edu/courses/Home/Databases/Engineering/about>

# References

- [http://en.wikipedia.org/wiki/Database\\_index](http://en.wikipedia.org/wiki/Database_index)
- [http://docs.oracle.com/cd/B28359\\_01/appdev.111/b28424/adfns\\_indexes.htm](http://docs.oracle.com/cd/B28359_01/appdev.111/b28424/adfns_indexes.htm)
- <http://publib.boulder.ibm.com/infocenter/cicsts/v3r2/index.jsp?topic=%2Fcom.ibm.cics.ts.productoverview.doc%2Fconcepts%2Facid.html>
- <http://technet.microsoft.com/en-us/library/hh393525.aspx>
- [http://docs.oracle.com/cd/B19306\\_01/server.102/b14215/ldr\\_concepts.htm](http://docs.oracle.com/cd/B19306_01/server.102/b14215/ldr_concepts.htm)
- <http://technet.microsoft.com/en-us/library/cc917551.aspx>
- [http://en.wikipedia.org/wiki/Data\\_warehouse](http://en.wikipedia.org/wiki/Data_warehouse)
- [http://docs.oracle.com/cd/B10501\\_01/appdev.920/a96624/02\\_funds.htm#10519](http://docs.oracle.com/cd/B10501_01/appdev.920/a96624/02_funds.htm#10519)
- [http://infolab.stanford.edu/~ullman/fcdb/oracle/or-plsql.html#simple\\_programs\\_in pl/sql](http://infolab.stanford.edu/~ullman/fcdb/oracle/or-plsql.html#simple_programs_in_pl/sql)

# Thank You



© 2013 Infosys Limited, Bangalore, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/or any named intellectual property rights holders under this document.

**Infosys®** | **Building Tomorrow's Enterprise**