

# Relational Database Management System



## Education, Training and Assessment

We enable you to leverage knowledge anytime, anywhere!

**Infosys**® | 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

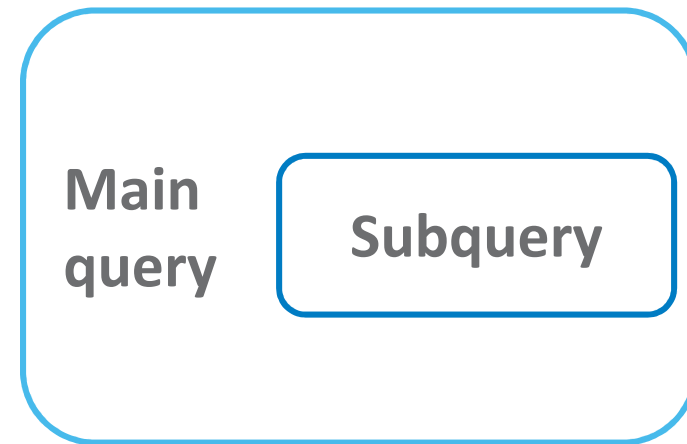
- Course Code: CCFP4.0-RDBMS
- Course Name: Relational Database Management System
- Document Number: RDBMS-04
- Version Number: 4.0

# Subquery



# Subquery

- A query within a query
- Inner query caters to one part of the requirement
- Inner query is enclosed in parentheses
- The result of inner query would be given as input to outer query
- Sub-queries can be used in SELECT , FROM, WHERE and HAVING clauses



## Independent subquery : Single row

- Requirement:

The super manager of the EasyShop would like to know the details of item/s having maximum price

Write a SQL query for this requirement

- Step1:** Get the maximum price of item/s

```
SELECT MAX(price)
FROM item;
```

- Step2:** Retrieve the details of the item/s with maximum price

```
SELECT itemcode, description FROM item WHERE
price =(SELECT MAX(price) from item);
```

Outer query

Inner query

## Independent subquery: Multiple row

If the inner query does not have any reference of outer query tables then it is independent subquery

- Requirement

The manager of EasyShop would like to retrieve the supplierid(s) and the name(s) of those suppliers whose quotations have been accepted

### Solution

```
SELECT supplierid, suppliername FROM supplier WHERE supplierid  
IN  
(SELECT supplierid FROM quotation WHERE quotationstatus = 'Accepted');
```



# Independent query and joins

- Requirement

Retrieve the details of customers who have bill amount of more than 5000

- **Solution with independent subquery:**

```
SELECT customerid, customername FROM customer c WHERE customerid  
IN  
(SELECT customerid FROM purchasebill WHERE billamount > 5000);
```

- **Solution with join:**

```
SELECT DISTINCT customerid, customername FROM customer c  
INNER JOIN purchasebill p  
ON c.customerid = p.customerid WHERE billamount > 5000
```

## Guided activity

- The management of the retail chain wants to identify the retail outlet generating the maximum revenue. Write a query to display all details of such retail outlet(s).

```
SELECT retailoutletid FROM purchasebill
GROUP BY retailoutletid
HAVING SUM(billamount) =
(SELECT MAX(SUM(billamount))
FROM purchasebill
GROUP BY retailoutletid);
```

## Independent subquery

**Requirement:** The payroll department of EasyShop requires the details of those employees who have the highest salary in each designation.

Write a query to retrieve the details of all these employees. (Hint: This requires multiple column subquery)

Guided Activity: CCFP4.0-RDBMSAssignments – Assignments on Sub queries -1 - 3  
(Estimated Time: 90 mins.)

## Correlated subquery

- Requirement:  
The Manager of payroll department wants to know details of all employees whose salary is greater than or equal to average salary of the employees in each designation.

employee

empid	empame	designation	salary
E1001	John	Manager	6500
E1002	Sam	Sales Person	3000
E1003	Allen	Manager	5000
E1004	Henry	Sales Person	2900
E1005	Chris	Sales Person	2800
E1006	Peter	Cashier	3200
E1007	Donald	Sales Person	2900

- How to meet above requirement ?
- Solution requires iterative comparison of employee's salary with computed average salary for that designation.
- This is achieved using correlated subquery

## Correlated subquery

- Requirement:

The Manager of payroll department wants to know details of all employees whose salary is greater than or equal to average salary of the employees in each designation.

employee

empid	empname	designation	salary
E1001	John	Manager	6500
E1002	Sam	Sales Person	3000
E1003	Allen	Manager	5000
E1004	Henry	Sales Person	2900
E1005	Chris	Sales Person	2800
E1006	Peter	Cashier	3200
E1007	Donald	Sales Person	2900

```
SELECT empid, empname, designation, salary
FROM employee e1
WHERE salary >= (SELECT AVG(salary)
                  FROM employee e2
                  WHERE e1.designation = e2.designation )
```

inner query is referring a column of the table used in outer query

## Correlated subquery

- Correlated subquery is one in which inner query refers to a table in the FROM clause of outer query
- The inner query is executed iteratively for each row of the outer query

## Working of correlated subquery (1 of 4)

e1								e2					
empid	empname	designation	salary	<b>&gt;=</b>	Avg	5750		empid	empname	designation	salary	Avg	5750
E1001	John	Manager	6500					E1001	John	Manager	6500		
E1002	Sam	Sales Person	3000					E1002	Sam	Sales Person	3000		
E1003	Allen	Manager	5000					E1003	Allen	Manager	5000		
E1004	Henry	Sales Person	2900					E1004	Henry	Sales Person	2900		
E1005	Chris	Sales Person	2800					E1005	Chris	Sales Person	2800		
E1006	Peter	Cashier	3200					E1006	Peter	Cashier	3200		
E1007	Donald	Sales Person	2900					E1007	Donald	Sales Person	2900		

empid	empname	designation	salary
E1001	John	Manager	6500

```
SELECT empid, empname, designation, salary FROM employee e1
WHERE salary >=
(SELECT avg(salary) FROM employee e2
WHERE e1.designation = e2.designation );
```

## Working of correlated subquery (2 of 4)

e1

empid	empname	designation	salary
E1001	John	Manager	6500
E1002	Sam	Sales Person	3000
E1003	Allen	Manager	5000
E1004	Henry	Sales Person	2900
E1005	Chris	Sales Person	2800
E1006	Peter	Cashier	3200
E1007	Donald	Sales Person	2900

&gt;=

Avg

2900

e2

empid	empname	designation	salary
E1001	John	Manager	6500
E1002	Sam	Sales Person	3000
E1003	Allen	Manager	5000
E1004	Henry	Sales Person	2900
E1005	Chris	Sales Person	2800
E1006	Peter	Cashier	3200
E1007	Donald	Sales Person	2900

Avg

2900

empid	empname	designation	salary
E1001	John	Manager	6500
E1002	Sam	Sales Person	3000

```
SELECT empid, empname, designation, salary FROM employee e1
WHERE salary >=
(SELECT avg(salary) FROM employee e2
WHERE e1.designation = e2.designation );
```



## Working of correlated subquery (3 of 4)

e1

empid	empname	designation	salary
E1001	John	Manager	6500
E1002	Sam	Sales Person	3000
E1003	Allen	Manager	5000
E1004	Henry	Sales Person	2900
E1005	Chris	Sales Person	2800
E1006	Peter	Cashier	3200
E1007	Donald	Sales Person	2900

e2

empid	empname	designation	salary
E1001	John	Manager	6500
E1002	Sam	Sales Person	3000
E1003	Allen	Manager	5000
E1004	Henry	Sales Person	2900
E1005	Chris	Sales Person	2800
E1006	Peter	Cashier	3200
E1007	Donald	Sales Person	2900

>=

Avg

5750

Avg

5750

```
SELECT empid, empname, designation, salary FROM employee e1
WHERE salary >=
(SELECT avg(salary) FROM employee e2
WHERE e1.designation = e2.designation );
```

## Working of correlated subquery (4 of 4)

e1

empid	empname	designation	salary
E1001	John	Manager	6500
E1002	Sam	Sales Person	3000
E1003	Allen	Manager	5000
E1004	Henry	Sales Person	2900
E1005	Chris	Sales Person	2800
E1006	Peter	Cashier	3200
E1007	Donald	Sales Person	2900

e2

empid	empname	designation	salary
E1001	John	Manager	6500
E1002	Sam	Sales Person	3000
E1003	Allen	Manager	5000
E1004	Henry	Sales Person	2900
E1005	Chris	Sales Person	2800
E1006	Peter	Cashier	3200
E1007	Donald	Sales Person	2900

similar execution occurs for rest of the rows and we get the output as

empid	empname	designation	salary
E1001	John	Manager	6500
E1002	Sam	Sales Person	3000
E1004	Henry	Sales Person	2900
E1006	Peter	Cashier	3200
E1007	Donald	Sales Person	2900

Guided Activity : CCFP4.0-RDBMSAssignments – Assignments on Sub queries - 4  
(Estimated Time: 60 mins.)

# Exists and Not Exists



## EXISTS and NOT EXISTS

- The EXISTS/ NOT EXISTS keyword is used to check whether a sub query returns any row(s)
- The main query displays the row if sub query following the EXISTS returns at least one row i.e. EXISTS returns TRUE.
- If sub query does not return any row, the EXISTS returns FALSE

## Exists

- Display the customer id and customer name of those customers who have purchased at least once from any retail outlet.

```
SELECT customerid, customername FROM customer c
WHERE EXISTS (SELECT * FROM purchasebill p
              WHERE c.customerid = p.customerid);
```

Guided Activity :CCFP4.0-RDBMSAssignments – Assignments on Sub queries - 5  
(Estimated Time: 30 mins.)

# Index



## Requirement:

Smith is designing a database for an application. While doing so, he wants more clarity on the topic, “candidate key”. He visits library and refers to a book on database management system.

What is the fastest way to search this topic in the book?

## INDEX (contd.)

- Indexing involves forming a two dimensional matrix completely independent of the table on which index is created.
- Here one column will hold the sorted data of the column which is been indexed
- Another column called the address field identifies the location of the record i.e. Row ID.
- Row Id indicates exactly where the record is stored in the table.
- Unique indexes are created automatically by Oracle when you create a primary key or a unique key constraint in a table
- When the primary key of the table or the unique constraint on column(s) is dropped the index which was built on them is also dropped

Guided Activity: CCFP4.0-RDBMSAssignments – Assignments on Index - 1  
(Estimated Time: 15 mins.)



## INDEX (contd.)

- **Advantages of having an INDEX:**
  - Greatly speeds the execution of SQL statements with search conditions that refer to the indexed column(s)
  - It is most appropriate when retrieval of data from tables are more frequent than inserts and updates
- **Disadvantages of having an INDEX:**
  - It consumes additional disk space
  - Additional Overhead on DML Statements

# Summary

- Subquery
  - Independent subquery
    - Single row
    - Multiple row
    - Multiple column
  - Correlated subquery
- Index

# Self-Study

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

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

**Videos:**

- Structured Query Language
- Structured Query Language II
- Storage Structures
- Indexing Techniques Single Level
- Indexing Techniques Multi-Level

Refer to:

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

# References

- Abraham Silberschatz, Henry Korth and S. Sudarshan, Database System Concepts Jan 27, 2010
- C.J. Date, Database Design and Relational Theory: Normal Forms and All That Jazz (Theory in Practice) Apr 24, 2012
- Kevin Loney, George Koch “Oracle 9i, The Complete reference” Oracle Press
- [http://en.wikipedia.org/wiki/Database\\_normalization](http://en.wikipedia.org/wiki/Database_normalization)
- [http://en.wikipedia.org/wiki/Entity%E2%80%93relationship\\_model](http://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model)
- <http://www.w3schools.com/sql/default.asp>
- [http://docs.oracle.com/cd/E11882\\_01/server.112/e41084/toc.htm](http://docs.oracle.com/cd/E11882_01/server.112/e41084/toc.htm)
- <http://online.stanford.edu/course/intro-to-databases-winter-2014>
- <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-830-database-systems-fall-2010/lecture-notes/>
- <http://www.techopedia.com/definition/1245/structured-query-language-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**