East West University
Department of Computer Science and Engineering

CSE 301: LAB 04 [Assessed Lab]
Course Instructor: Dr. Mohammad Rezwanul Huq

Introducing SQLDeveloper Tool, Oracle Functions and Writing SQL Statements involving nested Subqueries

Lab Objective

Familiarize students with SQL Developer tool to interact with the database and to write intermediate-level queries and nested subqueries.

Lab Outcome

After completing this lab successfully, students will be able to:

1. **Use** *SQL Developer* tool to interact with the database.
2. **Understand** the use of Oracle Built-in functions.
3. **Construct** SQL statements to perform queries involving nested subqueries.

Psychomotor Learning Levels

This lab involves activities that encompass the following learning levels in psychomotor domain.

Level	Category	Meaning	Keywords
P1	Imitation	Copy action of another; observe and replicate.	Relate, Repeat, Choose, Copy, Follow, Show, Identify, Isolate.
P2	Manipulation	Reproduce activity from instruction or memory	Copy, response, trace, Show, Start, Perform, Execute, Recreate.

Instructions

- Download and save banking script from https://goo.gl/q78ANw
- Execute SQLDeveloper tool and follow the instructor during the class.
- A more formal tutorial about SQLDeveloper can be found here: http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/sqldev/r40/sqldev4.0_GS/sqldev4.0_GS.html

Lab Activities (Introducing built-in functions in Oracle)

String Functions

`ASCII( single_character )`	`ASCII('t')` *Result:* `116`
`CHR( number_code )`	`CHR(116)` *Result:* `'t'`
`CONCAT( string1, string2 )`	`CONCAT('Tech on', ' the Net')` *Result:* `'Tech on the Net'`
`string1 || string2 [ || string_n ]`	`'a' || 'b' || 'c' || 'd'` *Result:* `'abcd'`
`INITCAP( string1 )`	`INITCAP('tech on the net');` *Result:* `'Tech On The Net'`

INSTR(string, substring [, start_position [, th_appearance]])	INSTR('Tech on the net', 'e') *Result:* 2 *(the first occurrence of 'e')*
LENGTH(string1)	LENGTH('Tech on the Net') *Result:* 15
LOWER(string1)	LOWER('Tech on the Net'); *Result:* 'tech on the net'
UPPER(string1)	UPPER('Tech on the Net') *Result:* 'TECH ON THE NET'
LPAD(string1, padded_length [, pad_string])	LPAD('tech', 8, '0'); *Result:* '0000tech'
RPAD(string1, padded_length [, pad_string])	RPAD('tech', 8, '0') *Result:* 'tech0000'
LTRIM(string1 [, trim_string])	LTRIM('xyxzyyyTech', 'xyz') *Result:* 'Tech'
RTRIM(string1 [, trim_string])	RTRIM('Techxyxzyyy', 'xyz') *Result:* 'Tech'
REPLACE(string1, string_to_replace [, replacement_string])	REPLACE('222tech', '2', '3'); *Result:* '333tech'
SUBSTR(string, start_position [, length])	SUBSTR('TechOnTheNet', 1, 4) *Result:* 'Tech'

Number Functions

ABS(number)	ABS(-23) *Result:* 23
bitand(expr1, expr2)	BITAND(5,3) *Result:* 1
CEIL(number)	CEIL(32.65) *Result:* 33
FLOOR(number)	FLOOR(5.9) *Result:* 5
GREATEST(expr1 [, expr2, ... expr_n])	GREATEST(2, 5, 12, 3) *Result:* 12
LEAST(expr1 [, expr2, ... expr_n])	LEAST(2, 5, 12, 3) *Result:* 2
LOG(m, n)	LOG(2, 15) *Result:* 3.90689059560852
MEDIAN(expression) [OVER (query partition clause)]	select MEDIAN(salary) from employees where department = 'Marketing';
MOD(m, n)	MOD(11.6, 2) *Result:* 1.6
POWER(m, n)	POWER(3, 2) *Result:* 9
SQRT(n)	SQRT(5.617) *Result:* 2.37002109695251
ROUND(number [, decimal_places])	ROUND(125.315, 2) *Result:* 125.32

TRUNC(number [, decimal_places])	TRUNC(125.815, 2) *Result:* 125.81
	SELECT ROWNUM, customers.* FROM customers WHERE customer_id > 4500;

Date Functions

ADD_MONTHS(date1, number_months)	ADD_MONTHS('21-Aug-03', -3) *Result:* '21-May-03'
EXTRACT ({ YEAR | MONTH | DAY | HOUR | MINUTE | SECOND } | { TIMEZONE_HOUR | TIMEZONE_MINUTE } | { TIMEZONE_REGION | TIMEZONE_ABBR } FROM { date_value | interval_value })	**SELECT** EXTRACT(YEAR FROM DATE '2003-08-22') from dual *Result:* 2003
TO_CHAR(value [, format_mask] [, nls_language])	SELECT TO_CHAR(sysdate, 'yyyy/mm/dd')from dual *Result:* '2003/07/09'
TO_DATE(string1 [, format_mask] [, nls_language])	SELECT TO_DATE('2015/05/15 8:30:25', 'YYYY/MM/DD HH:MI:SS') FROM dual;

Example:

```
Select ASCII('t') from dual;
Select ROUND (125.315, 2) from dual;
```

East WestUniversity
Department of Computer Science and Engineering

CSE 301: LAB 04 (Exercise)
Course Instructor: Dr. Mohammad Rezwanul Huq

Write all SQL statements in notepad first and save them with a.sql extension. Then execute your query in SQLDeveloper.

The following database schema is given.

1) Branch (branch_name, branch_city, assets)
2) Customer (customer_name, customer_street, customer_city)
3) Account (account_number, branch_name, balance)
4) Loan (loan_number, branch_name, amount)
5) Depositor (customer_name, account_number)
6) Borrower (customer_name, loan_number)

- **Write SQL statements for the following queries. Make sure that you have written your query in a notepad document first before executing it in front of instructor.**

1. Find all customer related information who have an account in a branch, located in the same city as they live. (write this query without using subqueries and then using a subquery)
2. Find all customer related information who have a loan in a branch, located in the same city as they live. (write this query without using subqueries and then using a subquery)
3. For each branch city, find the average balance of all the accounts opened in a branch located in that branch city. Do not include any branch city in the result where the total balance of all accounts opened in a branch located in that city is less than 1000. (Write this query with and without using 'having' clause)
4. For each branch city, find the average amount of all the loans opened in a branch located in that branch city. Do not include any branch city in the result where the average amount of all loans opened in a branch located in that city is less than 1500. (write this query with and without using 'having' clause)
5. Find the customer name, customer street, customer city of the account which has the highest balance among all the accounts. (Write this query with and without using all keyword)
6. Find the customer name, customer street, customer city of the loan which has the lowest amount among all the loans. (write this query with and without using all keyword)
7. Find the distinct branches (name and city) that have opened both accounts and loans. (Write this query using in and exists keyword)
8. Find the distinct customers (name and city) who do not have loans but have accounts. (write this query using not in and exists keyword)
9. Find those branch names which have total account balance greater than the average of total balance among all the branches. (write this query with and without using with clause)
10. Find those branch names which have total loan amount less than the average of total loan amount among all the branches. (write this query with and without using with clause)