Kabeer Ahmed
SE-19028

Database Management System
(DBMS – 204)

Experiment # 03

Single Row Functions

Student Name: Kabeer Ahmed

Roll Number: SE-19028

Maximum Marks	Performance = 05	Viva = 05	Total = 10
Marks Obtained			
Remarks (if any)			

Experiment evaluated by

Instructor Name: Engr. Adiba Jafar

Signature and Date:

Kabeer Ahmed
SE-19028

OUTCOMES

After completing this lesson, you should be able to do the following:
• Describe various types of functions available in SQL.
• Use character, number, and date functions in SELECT statement.
• Describe the use of conversion functions.

SQL Functions

1. **Input Output Function**
   Function performs action arg 1 arg 2 Result value arg n

Note:
Most of the functions described in this lesson are specific to Oracle Corporation's version of SQL.

Two Types of SQL Functions

1. **Single-row functions**

   - Manipulate data items
   - Accept arguments and return one value
   - Act on each row returned
   - Return one result per row
   - May modify the data type
   - Can be nested
   - Accept arguments which can be a column or an expression
     function_name[(arg1, arg2,...)]

2. **Character Functions**
   - Case-manipulation functions

   Character-manipulation functions
   - CONCAT
   - UPPER
   - SUBSTR
   - INITCAP
   - LENGTH
   - INSTR
   - LPAD | RPAD
   - TRIM
   - REPLACE

Note: The functions discussed in this lesson are only some of the available functions.

Case Manipulation Functions

These functions convert case for character strings.
Function Result
LOWER('SQL Course') sql course
UPPER('SQL Course') SQL COURSE
INITCAP('SQL Course') Sql Course

QUERY

```
SELECT empno, ename, deptno FROM   emp WHERE  ename = 'smith';
no rows selected
```

Kabeer Ahmed
SE-19028

SELECT empno, ename, deptno FROM emp WHERE LOWER(ename) = 'smith`';

Character-Manipulation Functions

These functions manipulate character strings:

Function	RESULT
CONCAT('Hello', 'World')	HelloWorld
SUBSTR('HelloWorld',1,5)	Hello
LENGTH('HelloWorld')	10
INSTR('HelloWorld', 'W')	6
LPAD(sal,10,'*')	*****24000
RPAD(sal, 10, '*')	24000*****
TRIM('H' FROM 'HelloWorld')	elloWorld

QUERY

SELECT empno, CONCAT(ename, job) NAME, sal,
LENGTH (ename), INSTR(ename, 'A') "Contains 'a'?"
FROM emp WHERE SUBSTR(job, 1,3) = 'MAN';

Number Functions

- ROUND : Rounds value to specified decimal
  ROUND(45.926, 2) 45.93
- TRUNC : Truncates value to specified decimal
  TRUNC(45.926, 2) 45.92
- MOD : Returns remainder of division
  MOD(1600, 300) 100

QUERY (ROUND)

SELECT ROUND(45.923,2), ROUND(45.923,0),ROUND(45.923,-1)
FROM DUAL;

DUAL is a dummy table you can use to view results
from functions and calculations.

ROUND(45.923,2)	ROUND(45.923,0)	ROUND(45.923,-1)
45.92	46	50

QUERY (TRUNC)

SELECT TRUNC(45.923,2), TRUNC(45.923), TRUNC(45.923,-2) FROM DUAL;

TRUNC(45.923,2)	TRUNC(45.923)	TRUNC(45.923, 2)
45.92	45	0

QUERY (MOD FUNCTION)

Calculate the remainder of a salary after it is divided by 5000 for all employees whose job title is sales representative.

SELECT ename, sal, MOD(sal, 5000) FROM emp WHERE job = 'MANAGER';

Kabeer Ahmed
SE-19028

LAST_NAME	HIRE_DATE
Gietz	07-JUN-94
Grant	24-MAY-99

Working with Dates

- Oracle database stores dates in an internal numeric format: century, year, month, day, hours,minutes, seconds.
- The default date display format is DD-MM-YY.
  - ❖ Allows you to store 21st century dates in the 20th century by specifying only the last two digits of the year.
  - ❖ Allows you to store 20th century dates in the 21st century in the same way.

QUERY

SYSDATE is a function that returns:
•Date
•Time
Example Display the current date using the DUAL table.

```
SELECT SYSDATE FROM   DUAL;
```

SYSDATE
08-MAR-01

Arithmetic with Dates

- Add or subtract a number to or from a date for a resultant date value.
  - ❖ date + number Date Adds a number of days to a date
  - ❖ date - number Date Subtracts a number of days from a date

- Subtract two dates to find the number of days between those dates.
  - ❖ date - date Number of days Subtracts one date from another

- Add hours to a date by dividing the number of hours by 24.
  - ❖ date + number/24 Date Adds a number of hours to a date

Using Arithmetic Operators with Dates

```
SELECT ename AS last_name , (SYSDATE-hiredate)/7 AS WEEKS
FROM   emp
WHERE  deptno= 20;
```

LAST_NAME	WEEKS
King	716.227563
Kochhar	598.084706
De Haan	425.227563

Note:SYSDATE is a SQL function that returns the current date and time. Your results may differ from the example. If a more current date is subtracted from an older date, the difference is a negative number.

Kabeer Ahmed
SE-19028

Using Date Functions

- MONTHS_BETWEEN ('01-SEP-95','11-JAN-94')
  19.6774194
- ADD_MONTHS ('11-JAN-94',6) '11-JUL-94'
- NEXT_DAY ('01-SEP-95','FRIDAY') '08-SEP-95'
- LAST_DAY('01-FEB-95') '28-FEB-95'

Assume SYSDATE = '25-JUL-95':

- ROUND(SYSDATE,'MONTH') 01-AUG-95
- ROUND(SYSDATE ,'YEAR') 01-JAN-96
- TRUNC(SYSDATE ,'MONTH') 01-JUL-95
- TRUNC(SYSDATE ,'YEAR') 01-JAN-95

Example

Compare the hire dates for all employees who started in 1981. Display the employee number, hire date, and month started using the ROUND and TRUNC functions.

SELECT empno, hiredate, ROUND(hiredate, 'MONTH'), TRUNC(hiredate, 'MONTH') FROM emp WHERE hiredate LIKE '%81';
Practice 3, Part 1
This practice is designed to give you a variety of exercises using different functions available for character,number, and date data types.
Complete questions 1 through 5 of Practice 3, found at the end of this lesson.

Conversion Functions

Data-type conversion
Implicit data-type conversion

Explicit data-type conversion
Note: Although implicit data-type conversion is available, it is recommended that you do explicit data type conversion to ensure the reliability of your SQL statements.

Implicit Data-Type Conversion
For assignments, the Oracle server can automatically convert the following:

From	To
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

Note: CHAR to NUMBER conversions succeed only if the character string represents a valid

number.

Explicit Data-Type Conversion

1. TO_NUMBER
2. TO_DATE
3. NUMBER
4. DATE
5. CHARACTER
6. TO_CHAR

Function	**Purpose**
TO_CHAR(*number* | *date* ,[*fmt*], VARCHAR2 *[nlsparams*])	Converts a number or date value to a character string with format model *fmt*.
TO_NUMBER(*char,[fmt],* *[nlsparams]*)	Converts a character string containing digits to a number in the format specified by the optional format model *Fmt* . The nlsparams parameter has the same purpose in this function as in the TO_CHAR function for number conversion.
TO_DATE(*char* ,[*fmt*],[*nlsparams*])	Converts a character string representing a date to a date value according to the *fmt* specified. If *fmt* is omitted, the format is DD-MON-YY.The nlsparams parameter has the same purpose in this function as in the TO_CHAR function for date conversion.

Note: The list of functions mentioned in this lesson includes only some of the available conversion functions.

EMPLOYEE_ID	MONTH
205	06/94

Using the TO_CHAR Function with Dates

TO_CHAR(date, ' format_model ')
The format model:

- Must be enclosed in single quotation marks and is case Sensitive.
- Can include any valid date format element.
- Has an fm element to remove padded blanks or suppress leading zeros.
- Is Separated from the date value by a comma.

```
SELECT empno, TO_CHAR(hiredate, 'MM/YYYY') Hiredate
FROM   emp
WHERE  ename = 'SMITH';
```

Elements of the Date Format Model

YYYY	Full year in numbers
YEAR	Year spelled out
MM	Two-digit value for month
MONTH	Full name of the month
MON	Three-letter abbreviation of the
Month	Three-letter abbreviation of the

Kabeer Ahmed
SE-19028

DY	day of the week
DAY	Full name of the day of the week
DD	Numeric day of the month

Elements of the Date Format Model

- Time elements format the time portion of the date.
  HH24:MI:SS AM 15:45:32 PM
- Add character strings by enclosing them in double quotation marks.
  DD "of" MONTH 12 of OCTOBER
- Number suffixes spell out numbers. ddspth fourteenth

Using the TO_CHAR Function with Dates

SELECT ename, TO_CHAR(hiredate, 'fmDD Month YYYY') HIREDATE FROM emp;
SELECT ename, TO_CHAR(hiredate, 'fmDdspth "of" Month YYYY fmHH:MI:SS AM') HIREDATE FROM emp;
Notice that the month follows the format model specified: in other words, the first letter is capitalized and the rest are lowercase.

Using the TO_CHAR Function with Numbers

TO_CHAR(number, ' format_model ')
These are some of the format elements you can use with the TO_CHAR function to display a

a. number value as a character:
   9 Represents a number 0
b. Forces a zero to be displayed $ Places a floating dollar sign
   L Uses the floating local currency symbol. Prints a decimal point, Prints a thousand indicator
c. Element Description Example Result

9	Numeric positions (number of 9s determine display width)	999999 1234
0	Display leading zeros	099999 001234
$	Floating dollar sign	$999999 $1234
L	Floating local currency symbol	L999999 FF1234
.	Decimal point in position specified	999999.99 1234.00
,	Comma in position specified	999,999 1,234
MI	Minus signs to right (negative values)	999999MI 1234-
PR	Parenthesize negative numbers	999999PR <1234>
EEEE	Scientific notation (format must specify four Es)	99.999EEEE 1.234E+03
V	Multiply by 10 *n* times (*n* = number of 9s after V)	9999V99 123400
B	Display zero values as blank, not 0	B9999.99 1234.00

SALARY
$3,000.00

Using the TO_CHAR Function with Numbers

SELECT TO_CHAR(sal, '$99,999.00') SAL FROM emp;

Using the TO_NUMBER And TO_DATE Functions

- Convert a character string to a number format
  using the TO_NUMBER function:

TO_NUMBER(char [,'format_model'])

- Convert a character string to a date format using the TO_DATE function:
  TO_DATE(char [, 'format_model'])
- These functions have an fx modifier. This modifier specifies the exact matching for the character argument and date format model of a TO_DATE function.

Example

Display the names and hiredates of all the employees who joined on May 01, 1981. Because the fx modifier is used, an exact match is required and the spaces after the word "May" are not recognized.

```
SELECT ename, hiredate FROM   emp
WHERE  hiredate = TO_DATE('May 01, 1981', 'fxMonth DD, YYYY')
```

Example of RR Date Format

To find employees hired prior to 1982, use the RR format, which produces the same results whether the command is run in 1982 or now:

```
SELECT ename, TO_CHAR(hiredate, 'DD-Mon-YYYY')
FROM   emp
WHERE  hiredate < TO_DATE('01-Jan-82', 'DD-Mon-RR');
SELECT ename, TO_CHAR(hiredate, 'DD -Mon-YYYY')
FROM   emp
WHERE  TO_DATE(hiredate, 'DD-Mon-YY') < '01-Jan-1982';
no rows selected
```

Nesting Functions

• Single-row functions can be nested to any level.
• Nested functions are evaluated from deepest level to the least deep level.

```
F3(F2(F1(col,arg1),arg2),arg3)
Step 1 = Result 1
Step 2 = Result 2
Step 3 = Result 3
```

Nesting Functions

Single-row functions can be nested to any depth. Nested functions are evaluated from the innermost level to the outermost level. Some examples follow to show you the flexibility of these functions.

Example

Display the date of the next Friday that is six months from the hire date. The resulting date should appear as Friday, August 13th, 1999. Order the results by hire date.

```
SELECT   hiredate,TO_CHAR(NEXT_DAY(ADD_MONTHS(hiredate, 6), 'FRIDAY'),
'fmDay, Month DDth, YYYY')
"Next 6 Month Review"
FROM      emp
ORDER BY  hiredate;
```

General Functions

These functions work with any data type and pertain to using null value.

• NVL (expr1, expr2)

Kabeer Ahmed
SE-19028

• NVL2 (expr1, expr2, expr3)
• NULLIF (expr1, expr2)
•COALESCE (expr1, expr2, ..., exprn)

Function	Description
NVL	Converts a null value to an actual value
NVL2	If expr1 is not null, NVL2 returns expr2.
NULLIF	If expr1 is null, NVL2 returns. The argument can have any data type. expr3 expr1
COALESCE	Compares two expressions and returns null if they are equal, or the first expression if they are not equal Returns the first non-null expression in the expression list

NVL Function

- Converts a null to an actual value
- Data types that can be used are date, character,and number.
- Data types must match:
  - NVL(commission,0)
  - NVL(hiredate,'01-JAN-97')
  - NVL(job,'No Job Yet')

NVL Conversions for Various Data Types

Data Type	Conversion Example
NUMBER	NVL(*number_column* ,9)
DATE	NVL(*date_column*, '01-JAN-95')
CHAR or VARCHAR2	NVL(*character_column* , 'Unavailable')

Using the NVL Function

```
SELECT ename, sal, NVL(comm, 0),
(sal*12) + (sal*12*NVL(comm, 0)) AN_SAL FROM emp;
```

The NVL Function

To calculate the annual compensation of all employees, you need to multiply the monthly salary by 12 and then add the commission percentage to it.

```
SELECT ename, sal, comm, (sal*12) + (sal*12*comm) AN_SAL FROM emp;
```

Using the NVL2 Function

```
SELECT ename,  sal, comm,
NVL2(comm,'SAL+COMM', 'SAL') income
FROM   emp WHERE deptno IN (10, 20);
```

Using the NULLIF Function

```
SELECT ename, LENGTH(ename) "expr1",
ename,  LENGTH(ename)  "expr2",
NULLIF(LENGTH(ename), LENGTH(ename)) result
FROM   emp;

SELECT ename, LENGTH(ename) "expr1",
JOB,  LENGTH(JOB)  "expr2",
```

Kabeer Ahmed
SE-19028

NULLIF(LENGTH(ename), LENGTH(JOB)) result
FROM emp;
Note:
The NULLIF function is logically equivalent to the following CASE expression. The CASE expression is discussed in a subsequent page:
CASE WHEN expr1 = expr 2 THEN NULL ELSE expr1 END

Kabeer Ahmed
SE-19028

LAB # 03

Single Row Functions

Practice 2, Part 1:

Write a query that displays the employee's names with the first letter capitalized and all other letters lowercase and the length of the names, for all employees whose name starts with *J* , *A* , or *M* . Give each column an appropriate label. Sort the results by the employees' names.

ANS.

```
SQL> SELECT INITCAP(ename) "Name", LENGTH(ename) "Length"
  2  FROM emp
  3  WHERE ename LIKE 'J%' OR ename LIKE 'M%' OR ename LIKE 'A%'
  4  ORDER BY ename;

Name                      Length
-------------------- ----------
Adams                          5
Allen                          5
James                          5
Jones                          5
Martin                         6
Miller                         6

6 rows selected.
```

Kabeer Ahmed
SE-19028

Practice 2, Part 2

For each employee, display the employee's name, and calculate the number of months between today and the date the employee was hired. Label the column MONTHS_WORKED . Order your results by the number of months employed. Round the number of months up to the closest whole number. Note: Your results will differ.

ANS.

```
SQL> SELECT ename, ROUND(MONTHS_BETWEEN(SYSDATE, hiredate)) MONTHS_WORKED
  2  FROM emp
  3  ORDER BY MONTHS_BETWEEN(SYSDATE,hiredate);

ENAME                MONTHS_WORKED
-------------------- -------------
ADAMS                          451
SCOTT                          452
MILLER                         463
FORD                           464
JAMES                          464
KING                           465
MARTIN                         467
TURNER                         467
CLARK                          470
BLAKE                          472
JONES                          472

ENAME                MONTHS_WORKED
-------------------- -------------
WARD                           474
ALLEN                          474
SMITH                          476

14 rows selected.
```

Kabeer Ahmed
SE-19028

Practice 3, Part 2

1. Write a query that produces the following for each employee: < employee name> earns <salary> monthly but wants <3 times salary >. Label the column Dream Salaries .

ANS.

```
SQL> SELECT ename || ' earns ' || TO_CHAR(sal, 'fm$99,999.00')
  2  || ' monthly but wants ' || TO_CHAR(sal*3, 'fm$99,999.00')
  3  || ',' "Dream Salaries"
  4  FROM emp;

Dream Salaries
--------------------------------------------------------------------
SMITH earns $800.00 monthly but wants $2,400.00,
ALLEN earns $1,600.00 monthly but wants $4,800.00,
WARD earns $1,250.00 monthly but wants $3,750.00,
JONES earns $2,975.00 monthly but wants $8,925.00,
MARTIN earns $1,250.00 monthly but wants $3,750.00,
BLAKE earns $2,850.00 monthly but wants $8,550.00,
CLARK earns $2,450.00 monthly but wants $7,350.00,
SCOTT earns $3,000.00 monthly but wants $9,000.00,
KING earns $5,000.00 monthly but wants $15,000.00,
TURNER earns $1,500.00 monthly but wants $4,500.00,
ADAMS earns $1,100.00 monthly but wants $3,300.00,

Dream Salaries
--------------------------------------------------------------------
JAMES earns $950.00 monthly but wants $2,850.00,
FORD earns $3,000.00 monthly but wants $9,000.00,
MILLER earns $1,300.00 monthly but wants $3,900.00,

14 rows selected.
```

2. Create a query to display the name and salary for all employees. Format the salary to be 15 characters long, left-padded with $. Label the column SALARY.

ANS.

Kabeer Ahmed
SE-19028

```
SQL> SELECT ename, LPAD(sal,15,'$') SALARY
  2  FROM emp;

ENAME
--------------------
SALARY
--------------------------------------------------------------------
SMITH
$$$$$$$$$$$800

ALLEN
$$$$$$$$$$1600

WARD
$$$$$$$$$$1250


ENAME
--------------------
SALARY
--------------------------------------------------------------------
JONES
$$$$$$$$$$2975

MARTIN
$$$$$$$$$$1250

BLAKE
$$$$$$$$$$2850


ENAME
--------------------
SALARY
--------------------------------------------------------------------
CLARK
$$$$$$$$$$2450

SCOTT
$$$$$$$$$$3000

KING
$$$$$$$$$$5000
```

Kabeer Ahmed
SE-19028

3. Display each employee's name, hiredate, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear similar to "Monday, the Thirty-First of July, 2000."

ANS.

```
SQL> SELECT ename,hiredate,TO_CHAR(NEXT_DAY(ADD_MONTHS(hiredate,6),'MONDAY'),'fmDay, "the" Ddspth "of" Month, YYYY')
  2  REVIEW
  3  FROM emp;

ENAME                HIREDATE
-------------------- ---------
REVIEW
--------------------------------------------------------------------------------
SMITH                17-DEC-80
Monday, the Twenty-Second of June, 1981

ALLEN                20-FEB-81
Monday, the Twenty-Fourth of August, 1981

WARD                 22-FEB-81
Monday, the Twenty-Fourth of August, 1981


ENAME                HIREDATE
-------------------- ---------
REVIEW
--------------------------------------------------------------------------------
JONES                02-APR-81
Monday, the Fifth of October, 1981

MARTIN               28-SEP-81
Monday, the Twenty-Ninth of March, 1982

BLAKE                01-MAY-81
Monday, the Second of November, 1981
```

```
ENAME                HIREDATE
-------------------- ---------
REVIEW
--------------------------------------------------------------------------------
CLARK                09-JUN-81
Monday, the Fourteenth of December, 1981

SCOTT                09-DEC-82
Monday, the Thirteenth of June, 1983

KING                 17-NOV-81
Monday, the Twenty-Fourth of May, 1982


ENAME                HIREDATE
-------------------- ---------
REVIEW
--------------------------------------------------------------------------------
TURNER               08-SEP-81
Monday, the Fifteenth of March, 1982

ADAMS                12-JAN-83
Monday, the Eighteenth of July, 1983

JAMES                03-DEC-81
Monday, the Seventh of June, 1982


ENAME                HIREDATE
-------------------- ---------
REVIEW
--------------------------------------------------------------------------------
FORD                 03-DEC-81
Monday, the Seventh of June, 1982

MILLER               23-JAN-82
Monday, the Twenty-Sixth of July, 1982


14 rows selected.
```

Kabeer Ahmed
SE-19028

4. Display the name, hiredate, and day of the week on which the employee started. Label the column DAY . Order the results by the day of the week starting with Monday.

ANS.

```
SQL> SELECT ename,hiredate,TO_CHAR(hiredate,'DAY') DAY
  2  FROM emp
  3  ORDER BY TO_CHAR(hiredate - 1, 'd');

ENAME                HIREDATE  DAY
-------------------- --------- ------------------------------------
MARTIN               28-SEP-81 MONDAY
CLARK                09-JUN-81 TUESDAY
TURNER               08-SEP-81 TUESDAY
KING                 17-NOV-81 TUESDAY
SMITH                17-DEC-80 WEDNESDAY
ADAMS                12-JAN-83 WEDNESDAY
JAMES                03-DEC-81 THURSDAY
JONES                02-APR-81 THURSDAY
FORD                 03-DEC-81 THURSDAY
SCOTT                09-DEC-82 THURSDAY
ALLEN                20-FEB-81 FRIDAY

ENAME                HIREDATE  DAY
-------------------- --------- ------------------------------------
BLAKE                01-MAY-81 FRIDAY
MILLER               23-JAN-82 SATURDAY
WARD                 22-FEB-81 SUNDAY

14 rows selected.
```

5. Create a query that displays the employees' names and commission amounts. If an employee does not earn commission, put "No Commission." Label the column COMM.

ANS.

Kabeer Ahmed
SE-19028

```
SQL> SELECT ename, NVL(TO_CHAR(comm), 'No Commision') comm
  2  FROM emp;

ENAME                COMM
-------------------- ----------------------------------------
SMITH                No Commision
ALLEN                300
WARD                 500
JONES                No Commision
MARTIN               1400
BLAKE                No Commision
CLARK                No Commision
SCOTT                No Commision
KING                 No Commision
TURNER               0
ADAMS                No Commision

ENAME                COMM
-------------------- ----------------------------------------
JAMES                No Commision
FORD                 No Commision
MILLER               No Commision

14 rows selected.
```

6. Create a query that displays the employees' names and indicates the amounts of their annual salaries with asterisks. Each asterisk signifies a thousand dollars. Sort the data in descending order of salary. Label the column EMPLOYEES_AND_THEIR_SALARIES.

ANS.

```
SQL> SELECT rpad(ename,8)||' '||rpad (' ',sal/1000+1,'*') EMPLOYEES_AND_THEIR_SALARIES
  2  FROM emp
  3  ORDER BY sal DESC;

EMPLOYEES_AND_THEIR_SALARIES
--------------------------------------------------------------------------------
KING     *****
FORD     ***
SCOTT    ***
JONES    **
BLAKE    **
CLARK    **
ALLEN    *
TURNER   *
MILLER   *
WARD     *
MARTIN   *

EMPLOYEES_AND_THEIR_SALARIES
--------------------------------------------------------------------------------
ADAMS    *
JAMES
SMITH

14 rows selected.
```