Data Base Management System Lab

Single-Row Functions

Employee Table

```
CREATE TABLE EMP
      (EMPNO NUMBER(4) NOT NULL,
       ENAME VARCHAR2(10),
       JOB VARCHAR2(9),
       MGR NUMBER(4),
       HIREDATE DATE,
       SAL NUMBER(7, 2),
       COMM NUMBER(7, 2),
       DEPTNO NUMBER(2));

INSERT INTO EMP VALUES
      (7369, 'SMITH',  'CLERK',     7902,
      TO_DATE('17-DEC-1980', 'DD-MON-YYYY'),  800, NULL, 20);
INSERT INTO EMP VALUES
      (7499, 'ALLEN',  'SALESMAN',  7698,
      TO_DATE('20-FEB-1981', 'DD-MON-YYYY'), 1600,  300, 30);
INSERT INTO EMP VALUES
      (7521, 'WARD',   'SALESMAN',  7698,
      TO_DATE('22-FEB-1981', 'DD-MON-YYYY'), 1250,  500, 30);
```

Employee Table

```
INSERT INTO EMP VALUES
      (7566, 'JONES',  'MANAGER',   7839,
      TO_DATE('2-APR-1981', 'DD-MON-YYYY'),  2975, NULL, 20);
INSERT INTO EMP VALUES
      (7654, 'MARTIN', 'SALESMAN',  7698,
      TO_DATE('28-SEP-1981', 'DD-MON-YYYY'), 1250, 1400, 30);
INSERT INTO EMP VALUES
      (7698, 'BLAKE',  'MANAGER',   7839,
      TO_DATE('1-MAY-1981', 'DD-MON-YYYY'),  2850, NULL, 30);
INSERT INTO EMP VALUES
      (7782, 'CLARK',  'MANAGER',   7839,
      TO_DATE('9-JUN-1981', 'DD-MON-YYYY'),  2450, NULL, 10);
INSERT INTO EMP VALUES
      (7788, 'SCOTT',  'ANALYST',   7566,
      TO_DATE('09-DEC-1982', 'DD-MON-YYYY'), 3000, NULL, 20);
INSERT INTO EMP VALUES
      (7839, 'KING',   'PRESIDENT', NULL,
      TO_DATE('17-NOV-1981', 'DD-MON-YYYY'), 5000, NULL, 10);
```

Employee Table

```
INSERT INTO EMP VALUES
      (7844, 'TURNER', 'SALESMAN',  7698,
      TO_DATE('8-SEP-1981', 'DD-MON-YYYY'),  1500, NULL, 30);
INSERT INTO EMP VALUES
      (7876, 'ADAMS',  'CLERK',     7788,
      TO_DATE('12-JAN-1983', 'DD-MON-YYYY'), 1100, NULL, 20);
INSERT INTO EMP VALUES
      (7900, 'JAMES',  'CLERK',     7698,
      TO_DATE('3-DEC-1981', 'DD-MON-YYYY'),   950, NULL, 30);
INSERT INTO EMP VALUES
      (7902, 'FORD',   'ANALYST',   7566,
      TO_DATE('3-DEC-1981', 'DD-MON-YYYY'),  3000, NULL, 20);
INSERT INTO EMP VALUES
      (7934, 'MILLER', 'CLERK',     7782,
      TO_DATE('23-JAN-1982', 'DD-MON-YYYY'), 1300, NULL, 10);
```

Single-Row Functions

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

Single-Row Functions

1. Character functions
    1. Case-manipulation
        1. LOWER
        2. UPPER
        3. INITCAP
    2. Character-manipulation
        1. CONCAT
        2. SUBSTR
        3. LENGTH
        4. INSTR
        5. LPAD | RPAD
        6. TRIM
        7. REPLACE
2. Number Functions
3. Date
4. Data type conversion
    1. Implicit data type conversion
    2. Explicit data type conversion
5. General Functions

Case-manipulation

LOWER('SQL Course') -> sql course
UPPER('SQL Course') -> SQL COURSE
INITCAP('SQL Course') -> Sql Course

```
SELECT EMPNO, ENAME, DEPTNO FROM EMP WHERE LOWER(ENAME) = 'allen';
```

EMPNO	ENAME	DEPTNO
7499	ALLEN	30
7499	ALLEN	30

Character-Manipulation

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

```
SELECT EMPNO, CONCAT(CONCAT(ENAME,'_'), JOB) NAME, LENGTH (ENAME), INSTR(ENAME, 'a') "Contains 'a'?"
FROM EMP WHERE SUBSTR(JOB, 4) = 'LYST';
```

EMPNO	NAME	LENGTH(ENAME)	Contains 'a'?
7788	SCOTT_ANALYST	5	0
7902	FORD_ANALYST	4	0
7788	SCOTT_ANALYST	5	0
7902	FORD_ANALYST	4	0

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

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

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

SELECT ENAME, SAL, MOD(SAL, 2000) FROM EMP WHERE JOB = 'ANALYST';

ENAME	SAL	MOD(SAL,2000)
SCOTT	3000	1000
FORD	3000	1000
SCOTT	3000	1000

Remainder of salary of ANALYST after it is divided by 2000

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 number value as a character

9	->	Represents a number
0	->	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

SELECT TO_CHAR(SAL, '$99,999.00') SALary FROM EMP WHERE ENAME = 'WARD';

SALARY
$1,250.00
$1,250.00

TO_NUMBER and TO_DATE Functions

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

TO_NUMBER (char , 'format_model')

```
SELECT  TO_NUMBER('1210.73', '9999.99')  FROM DUAL;
```

Convert a character string to a date format using the TO_DATE function:

TO_DATE(char, 'format_model ')

```
SELECT TO_DATE('January 15, 1989, 11:00 A.M.',  'Month dd, YYYY, HH:MI A.M.') FROM DUAL;
```

These functions have an fx modifier. This modifier specifies the exact matching for the character argument and date format model

RR Date Format

Current Year	Specified Date	RR Format	YY Format
1995	27-OCT-95	1995	1995
1995	27-OCT-17	2017	1917
2001	27-OCT-17	2017	2017
2001	27-OCT-95	1995	2095

To find employees hired prior to 1990, use the RR format, which produces the same results whether the command is run in 1999 or now:

		If the specified two-digit year is:	
		0–49	50–99
If two digits of the current year are:	0–49	The return date is in the current century	The return date is in the century before the current one
	50–99	The return date is in the century after the current one	The return date is in the current century

```
SELECT  ENAME, TO_CHAR(HIREDATE, 'DD-Mon-YYYY')
FROM EMP WHERE HIREDATE < TO  DATE('01-Jan-81', 'DD-
Mon-RR');
```

ENAME	TO_CHAR(HIREDATE,'DD-MON-YYYY')
SMITH	17-Dec-1980
SMITH	17-Dec-1980

General Functions

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

- NVL (expr1, expr2)
- NVL2 (expr1, expr2, expr3)
- NULLIF (expr1, expr2)
- COALESCE (expr1, expr2, ..., exprn)

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_pct,0)
  - NVL(hire_date,'01-JAN-97')
  - NVL(job_id, 'No Job Yet')

SELECT ENAME, NVL(TO_CHAR(MGR), 'No Manager') FROM EMP WHERE MGR IS NULL;

ENAME	NVL(TO_CHAR(MGR),'NOMANAGER')
KING	No Manager
KING	No Manager

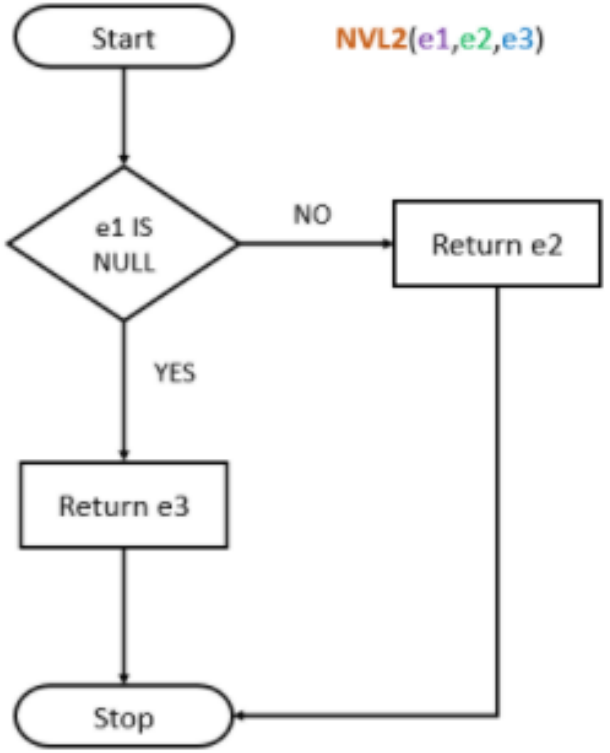
NVL2

NVL2(e1,e2,e3) function accepts three arguments.

e1 is the source value or expression that may contain null
e2 is the value returned if expr1 is not null
e3 is the value returned if expr1 is null

SELECT NVL2(NULL, 1, 2) FROM dual;

NVL2(NULL,1,2)
2



NULLIF Function

The NULLIF() function returns NULL if two expressions are equal, otherwise it returns the first expression.

SELECT ENAME,EMPNO, LENGTH(EMPNO) "expr1", MGR, LENGTH(MGR) "expr2", NULLIF(LENGTH(EMPNO), LENGTH(MGR)) result FROM EMP WHERE MGR IS NULL OR MGR= 7902;

ENAME	EMPNO	expr1	MGR	expr2	RESULT
SMITH	7369	4	7902	4	-
KING	7839	4	-	-	4
SMITH	7369	4	7902	4	-
KING	7839	4	-	-	4

COALESCE Function

- The advantage of the COALESCE function over the NVL function is that the COALESCE function can take multiple alternate values.
- If the first expression is not null, it returns that expression; otherwise, it does a COALESCE of the remaining expressions.

```
SELECT  COALESCE(NULL, NULL, 1, 2) FROM  dual;
```

COALESCE(NULL,NULL,1,2)
1

Conditional Expressions

Provide the use of IF-THEN-ELSE logic within a SQL statement

- Use two methods:
  - CASE expression
  - DECODE function

```
CASE expr WHEN comparison_expr1 THEN return_expr1
          [WHEN comparison_expr2 THEN return_expr2
          WHEN comparison_exprn THEN return_exprn
          ELSE else_expr]
END
```

```
SELECT ENAME, JOB, SAL,
       CASE JOB WHEN 'CLERK' THEN 1.10*SAL
                WHEN 'SALESMAN' THEN 1.15*SAL
                WHEN 'MANAGER' THEN 1.20*SAL
       ELSE SAL END "REVISED_SALARY“
FROM EMP;
```

ENAME	JOB	SAL	REVISED_SALARY
SMITH	CLERK	800	880
ALLEN	SALESMAN	1600	1840
WARD	SALESMAN	1250	1437.5
JONES	MANAGER	2975	3570

The DECODE Function

It also facilitates conditional inquiries

DECODE(col|expression, search1, result1
[, search2, result2,...,]
[, default])

```
SELECT ENAME, JOB, SAL,
        DECODE(JOB, 'CLERK', 1.10*SAL,
                    'SALESMAN', 1.15*SAL,
                    'MANAGER', 1.20*SAL,
        SAL)
REVISED_SALARYFROM EMP;
```

ENAME	JOB	SAL	REVISED_SALARY
SMITH	CLERK	800	880
ALLEN	SALESMAN	1600	1840
WARD	SALESMAN	1250	1437.5
JONES	MANAGER	2975	3570

The DECODE Function

Display the applicable tax rate for each employee in department 80.

```
SELECT ENAME, SAL,
        DECODE (TRUNC(SAL/1000, 0),
                0, 0.00,
                1, 0.09,
                2, 0.20,
                3, 0.30,
                4, 0.40,
                5, 0.42,
                6, 0.44,
                    0.45) TAX_RATE
        FROM EMPWHERE DEPTNO  = 20;
```

ENAME	SAL	TAX_RATE
SMITH	800	0
JONES	2975	.2
SCOTT	3000	.3
ADAMS	1100	.09
FORD	3000	.3