Using Single-Row Functions to Customize Outp ut

Character functions (Upper, Lower, Initcap)

```
SELECT EMPLOYEE_ID, FIRST_NAME, upper(FIRST_NAME), lower(FIRST_NAME), initcap(first_name)
FROM EMPLOYEES;
```

```
SELECT EMPLOYEE_ID, FIRST_NAME, upper(FIRST_NAME), lower(FIRST_NAME), initcap(first_name)
FROM EMPLOYEES
WHERE UPPER(FIRST_NAME) = 'PATRICK';
```

```
SELECT EMPLOYEE_ID, FIRST_NAME, upper(FIRST_NAME), lower(FIRST_NAME), initcap(first_name)
FROM EMPLOYEES
WHERE UPPER(FIRST_NAME) = UPPER('patrick')
ORDER BY UPPER(FIRST_NAME);
```

Character functions (concat, substr, length)

```
SLEECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, CONCAT(FIRST_NAME, LAST_NAME)
FROM EMPLOYEES;
```

The concat function only took 2 args, but || is more flexible

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, FIRST_NAME || ' ' ||LAST_NAME||salary
FROM EMPLOYEES;
```

substr(column|expression, m, n)
m is the starting position
n is the length of the characters

```
SELECT EMPLOYEE_ID
FIRST_NAME,
SUBSTR(FIRST_NAME, 1, 3)
SUBSTR(FIRST_NAME, 2, 4)
SUBSTR(FIRST_NAME, 2) -- if you don't specify the n value, then it will be the end of the string
SUBSTRT(FIRST_NAME, -3) -- if m is negative, then the count starts from the end
FROM EMPLOYEES;
```

```
SELECT FIRST_NAME, LENGTH(FIRST_NAME)
```

```
FROM EMPLOYEES; -- it takes char and return nnumber
```

Character functions (instr)

instr(column|expression, m, n)
m is the start position
n is the occurrence
1 is the default for m and n

```
SELECT FIRST_NAME
INSTR(FIRST_NAME, 'e'),
INSTR(FIRST_NAME, 'e', 2),
INSTR(FIRST_NAME, 'e', 5),
INSTR(FIRST_NAME, 'e', 1, 2)
FROM EMPLOYEES
WHERE FIRST_NAME = 'Nanette';
```

Character functions (lpad, rpad, replace, trim)

```
SELECT EMPLOYEE_ID, FIRST_NAME, SALARY, LPAD(SALARY, 10, '#'), RPAD(SALARY, 10, '*')
FROM EMPLOYEES;
```

EMPLOYEE_ID	FIRST_NAME	LPAD(SALARY, 10, '#')	RPAD(SALARY, 10, '*')
100	Steven	#####24000	24000*****

```
SELECT EMPLOYEE_ID, FIRST_NAME, SALARY, REPLACE(SALARY, 'a', '*'), REPLACE(SALARY, 'en', '#')
FROM EMPLOYEES;
```

EMPLOYEE_ID	FIRST_NAME	REPLACE(SALARY, 'a', '*')	REPLACE(SALARY, 'en', '#')
166	Sundar	Sund*r	Sundar

to understand trim in very good way, we will try to do examples using dual table
dual is a public table that you can use to view result from functions and calculation

```
SELECT * FROM DUAL;
```

So it is a table contain one column and one dummy value x

```
SELECT 1+1+3 FROM DUAL;
```

```
SELECT 1+5 FROM EMPLOYEES; -- it will show the results but the number of results equals the
number of records
```

TRIM([[Leading | Trailing | Both] trim_character FROM] string1)

```
SELECT TRIM (' ' FROM ' khaled kudari ') V FROM DUAL;
```

```
SELECT TRIM (LEADING ' ' FROM ' khaled kudari ') V FROM DUAL;
```

```
SELECT TRIM (TRAILING ' ' FROM ' khaled kudari ') V FROM DUAL;
```

```
SELECT TRIM (BOTH ' ' FROM ' khaled kudari ') V FROM DUAL;
```

```
SELECT TRIM ('k' FROM 'khaled kudari') V FROM DUAL;
```

```
SELECT TRIM (LEADING 'k' FROM 'khaled kudari') V FROM DUAL;
```

```
SELECT TRIM (TRAILING 'k' FROM 'khaled kudari') V FROM DUAL;
```

```
SELECT TRIM (' khaled kudari ') V FROM DUAL;
```

Date functions (Sysdate)