**Single Row Functions**

**Opening Note:**

**SELECT command and as well as any keyword with SELECT (Such as FROM , WHERE, IN ,BETWEEN) are not Case sensitive… Also Table Names and Column names are not case sensitive. But Data itself in the database is case sensitive.**

seleCT \* FROM employees

where first\_name = 'Donald' -- yes it works because data is stored as Initial Capital and rest is lowercase

seleCT \* FROM employees

where first\_name = 'DONALD' -- it does not work

seleCT \* FROM employees

where UPPER(first\_name) = 'DOnAlD' ---yes it works

seleCT \* FROM employees

where UPPER(first\_name) = UPPER('DOnAlD') -- yes it works

seleCT \* FROM employees

where first\_name = 'Donald' -- yes it works

seleCT \* FROM employees

where first\_name = 'donald' -- it does not works

seleCT \* FROM employees

where LOWER (first\_name) = 'donald' -- yes it works

SELECT initcap('CENTENNIAL college') FROM DUAL;

SELECT LASTNAME,FIRSTNAME FROM CUSTOMERS WHERE lower(LASTNAME)='smith';

SELECT LASTNAME,FIRSTNAME ,address FROM CUSTOMERS WHERE

STATE=UPPER('&STATE');

SELECT \* FROM CUSTOMERS;

SELECT ('HEY TODAY is WEdnesday and Sunny day') FROM DUAL

SELECT LOWER('HEY TODAY is WEdnesday and Sunny day') FROM DUAL

SELECT INITCAP('HEY TODAY is WEdnesday and Sunny day') FROM DUAL

SELECT last\_name, UPPER(last\_name) FROM EMPLOYEES

SELECT last\_name, LENGTH (last\_name) FROM EMPLOYEES

SELECT last\_name, UPPER(last\_name) FROM EMPLOYEES

select first\_name, lower(first\_name) as "Lower Case", last\_name, upper(last\_name)

from employees;

select concat(concat(first\_name, ', '), last\_name)

from employees;

select first\_name || ', ' || last\_name

from employees;

select 'Contact Name:' || concat(concat(first\_name, ' '),last\_name)

from contacts;

select first\_name,

substr(lower(first\_name),1,1) || lower(last\_name) || '@email.com' as "Email"

from contacts;

select product\_name, length(product\_name) as "Length of Name"

from products

order by length(product\_name);

select first\_name, length(first\_name) as "First Name Length"

from contacts

where length(first\_name) > 6

order by length(first\_name);

select first\_name, length(first\_name) as "First Name Length"

from contacts

where length(first\_name) > 3 and length(first\_name) < 6

order by "First Name Length";

select email, initcap(email)

from employees;

CONCAT Function… (Use || instead of CONCAT)

CONCAT function : to combine two character values. first name and last together

it only accept 2 different values .. Not much flexible

SELECT first\_name, last\_name FROM employees

SELECT CONCAT ( first\_name , last\_name ) FROM employees -- only two fields

SELECT CONCAT ( first\_name , last\_name, phone\_number) FROM employees -- this will not work as it provide 3 fields

We suggest to use || ( pipe pipe) to combine as many as character values

SELECT first\_name || last\_name||phone\_number|| salary FROM employees ---good but not good enough

SELECT first\_name || ' ' || last\_name || ' ' || phone\_number|| ' ' || salary FROM employees

SELECT 'Hi how are you' || ' today is wednesday' FROM DUAL

SELECT order#,CONCAT(shipdate - orderdate,'DAY(S)') "DTIME" FROM

ORDERS ;

SELECT CONCAT(CONCAT(CONCAT(SHIPDATE, ' SHIPPED '), ORDERDATE), SHIPDATE - ORDERDATE) FROM ORDERS

INNER JOIN orderitems USING (ORDER#);

SELECT CONCAT(CONCAT(CONCAT(CONCAT('ORDER #', ORDER#), ' TOOK '),(SHIPDATE-ORDERDATE)), ' DAYS TO SHIP.')

AS DTIME

FROM ORDERS

WHERE SHIPDATE IS NOT NULL;

SELECT CONCAT(CONCAT(FIRSTNAME,' ORDERED '),TITLE) AS "CUSTOMER ORDER" FROM CUSTOMERS

JOIN ORDERS USING(CUSTOMER#)

JOIN ORDERITEMS USING (ORDER#)

JOIN BOOKS USING(ISBN);

SELECT CONCAT(CONCAT(CONCAT(CONCAT(c.lastname,'-'),C.FIRSTNAME),' ORDER THE BOOK '), b.title)

FROM customers c

JOIN orders o USING(customer#)

JOIN orderitems io USING(order#)

JOIN books b USING(isbn);

--CONCAT

SELECT CONCAT(CONCAT(LASTNAME, '''s CITY IS '), CITY)

FROM CUSTOMERS;

SELECT CONCAT(CONCAT(firstname, ' ID: '),customer#) FROM customers;

SELECT \* FROM BOOKS;

SELECT LASTNAME, RTRIM(LASTNAME, 'S') FROM CUSTOMERS;

-- CHALLENGE 1

-- JOIN CUSTOMERS,ORDERS,ORDERITEM ,BOOKS . WHICH CUSTOMER

--HAS ORDERED WHAT BOOK.

--O/P BONITAS ORDERED THE BOOK BODYBUILD IN 1O

SELECT c.firstname || ' ' || c.lastname AS name, b.title

FROM customers c

JOIN orders o USING(customer#)

JOIN orderitems io USING(order#)

JOIN books b USING(isbn);

----Character data related functions

**SUBSTR**

SELECT SUBSTR('HEY TODAY is WEdnesday and Sunny day', 1,7) FROM DUAL

SELECT last\_name from employees

SUBSTR (list first 3 digits of each employees' last name)

SUBSTR ( column or 'string info ' , starting point, how many digit to extract )

SELECT last\_name, SUBSTR ( last\_name , 1, 3 )

FROM employees

**-- SUBSTRING**

**Find all the customers whose having P.O. in their address info**

**Opt1: LIKE**

**select \* from customers where address LIKE 'P.O.%';**

**Opt2:SUBSTR**

**select \* from customers where substr(address,1,4)='P.O.';**

SELECT ZIP,SUBSTR(ZIP,2,4) FROM CUSTOMERS;

SELECT ZIP,SUBSTR(ZIP,4) FROM CUSTOMERS;

Find all the customers who provided P.O. as address?

With LIKE

SELECT \* FROM CUSTOMERS WHERE address like 'P.O.%'

With SUBSTR

SELECT \* FROM CUSTOMERS WHERE SUBSTR (address, 1,4) ='P.O.'

SELECT FIRSTNAME,LENGTH(FIRSTNAME) FROM CUSTOMERS ORDER BY

LENGTH(FIRSTNAME) DESC;

-- CHECK YOUR NUMBER OF CHAR IN YOUR FIRSTNAME AND LASTNAME

SELECT LENGTH('VIJI') FROM DUAL;

SELECT \* FROM CUSTOMERS;

-- LTRIM AND RTRIM IN CUSTOMERS TABLE

SELECT ADDRESS,LTRIM(ADDRESS,'P.O. BOX') AS PBNUMBER FROM CUSTOMERS;

SELECT LTRIM(ADDRESS) || ' ' || RTRIM(CITY,'-') FROM CUSTOMERS;

INSTR

INSTR(string, substring [, start [, occurrence]])

INSTR (search and finding special character or letter in column or string set)

SELECT INSTR ('ersan.cam@senecacollege.ca', '.') FROM dual --- Finding first occurance of period.

SELECT INSTR ('ersan.cam@senecacollege.ca', '.',7) FROM dual -- Finding second occurance position of Period

SELECT INSTR ('ersan.cam@senecacollege.ca', '.',1,2) FROM dual

select product\_name, instr(product\_name, 't')

from products;

select product\_name, instr(product\_name, ' ')

from products;

select first\_name, instr(first\_name, 'w') as "With w"

from contacts

where instr(first\_name, 'w') > 1

order by "With w";

select first\_name, instr(first\_name, ' ') as "With space"

from contacts

where instr(first\_name, ' ') > 1

order by "With space";

select first\_name, lpad(first\_name, 20, '\*')

from contacts;

select first\_name, rpad(first\_name, 20, '\*')

from contacts;

select first\_name, '$' || trim('S' from first\_name)

from contacts

where first\_name like 'S%';

LENGTH

=====

REPLACE

SELECT ('Hi, how are you! today is wednesday' ) FROM DUAL

SELECT REPLACE ('Hi, how are you! today is wednesday' , '!' ,'?') FROM DUAL

SELECT \* FROM employees

Question.

Select last\_name, first\_name, salary and phone number of employee but also add one more column as new phone

number , in there show phone number as 650-507-9833 instead of '.'

Option 1: SELECT REPLACE

select last\_name, first\_name, salary, phone\_number, replace(phone\_number, '.','-') from employees;

Option 2: UPDATE EMPLOYEES SET phone = REPLACE ()

Option 3: PL/SQL programmming Open cursor, loop thru , check if there is . then replace with -

===

Show me all books under 'cooKIng' category

SELECT \* FROM books

WHERE LOWER(category)='cooking'

SELECT \* FROM books

WHERE LOWER(category)= lower('cooKING')

SELECT \* FROM books

WHERE UPPER(category)= 'COOKING'

SELECT title, INITCAP(title) from books

SELECT title, SUBSTR (title,1,8) from books

SELECT title, SUBSTR (title,5) from books WHERE category='COMPUTER'

SELECT SUBSTR ('We will setup make up class for saturdays',14) FROM DUAL

SELECT SUBSTR ('We will setup make up class for saturdays',-5) FROM DUAL

LENGTH

Finding lenght of character set

SELECT LENGTH (title) from books

CONCAT Combines two characters set (Only two)

SELECT title, category FROM books

combine two columns into one column

SELECT CONCAT (title, category,isbn) as titleandcat

FROM books

SELECT CONCAT ('we will have ' ,'make up class') FROM DUAL

SELECT 'we will have ' ,'make up class' FROM DUAL

Use || instead of CONCAT to have unlimited option

SELECT 'name of the book: ' || title || ' and the category is: ' || category || ' and code ' || isbn as titleandcat

FROM books

====

Books Database

select first\_name, replace(first\_name, 's', '$')

from contacts

where first\_name like '%s%';

select first\_name, replace(lower(first\_name), 's', '$')

from contacts

where lower(first\_name) like '%s%';

SELECT lastname,firstname, REPLACE (address, 'P.O.', 'Do not sent mail') FROM CUSTOMERS

SELECT last\_name, first\_name, salary, phone\_number

FROM employees

SELECT last\_name, first\_name, salary, phone\_number,REPLACE (phone\_number, '.','-')

FROM employees

SELECT last\_name, first\_name, salary, phone\_number

FROM employees

these REPLACE activities or any other functions will not actually update the data in the tables.

data still remain the same

Question: What change actually the data in the table?

It is not ALTER...

ALTER TABLE change table structure. like adding column , changung column data type from NUMBER TO CHAR

ALTER ADD CONSTRAINT FK

UPDATE is the right answer....

UPDATE employees

SET phone\_number = REPLACE (phone\_number, '.','-')

---now after this all the records phone number got changed

SELECT last\_name, first\_name, salary, phone\_number

FROM employees

Email Tokenizing Example – Game1

create table contacts (email varchar2(50) );

INSERT INTO contacts (email ) VALUES ('ersan.cam@my.senecacollege.ca');

INSERT INTO contacts (email ) VALUES ('james.curry@gmail.com');

INSERT INTO contacts (email ) VALUES ('sandra.jenson@yahoo.com');

INSERT INTO contacts (email ) VALUES ('samantha.jenson@yahoo.ca');

Commit;

SELECT \* FROM contacts

option1 (Easy solution) separate below information from email column

|  |  |  |
| --- | --- | --- |
| **firstname** | **lastname** | **servprovider** |
| Ersan | Cam | my.senecacollege.ca |

Option2: (Challenged version.) Separate below info from email column

|  |  |  |  |
| --- | --- | --- | --- |
| **firstname** | **lastname** | **servprovider** | **domain** |
| Ersan | Cam | my. senecacollege | ca |
| james | curry | gmail.com | com |

Email Tokenizing Example – Game2

create table email (mailaddress varchar2(50) );

INSERT INTO email (mailaddress) VALUES ('ersan.cam@my.centennialcollege.ca');

INSERT INTO email (mailaddress) VALUES ('james.curry@gmail.com');

INSERT INTO email (mailaddress) VALUES ('sandra.jenson@yahoo.com');

INSERT INTO email (mailaddress) VALUES ('samantha.jenson@yahoo.ca');

select \* from email

In this table

ecam@my.centennialcollege.ca

sandra.jenson@yahoo.com

james.curry@gmail.com

Find if there is period in their email? If no it is zero 0, if yes what is the position of period?

INSTR (column or 'string' , searching character) -- -this will find first occurance of that character

SELECT mailaddress, INSTR (mailaddress, '.') FROM email

Find if there is @ character in their email? If no it is null, if yes what is the position of @?

INSTR (column or 'string' , searching character) -- -this will find first occurance of that character

LENGHT is to find lenght of character (either column or string)

SELECT LENGTH ('HEY TODAY is WEdnesday and Sunny day') FROM DUAL

find out how many characters in each person email address in email table

SELECT mailaddress, LENGTH (mailaddress)

FROM email

===

What if the period location that we are looking for is not the first one? Let's say there is two period and you are

looking for 2nd period's position

create table email2 (mailaddress varchar2(50) )

INSERT INTO email2 (mailaddress) VALUES ('ersan.cam@my.centennialcollege.ca')

INSERT INTO email2 (mailaddress) VALUES ('james.curry@gmail.com')

INSERT INTO email2 (mailaddress) VALUES ('sandra.jenson@yahoo.com')

INSERT INTO email2 (mailaddress) VALUES ('sandra.jenson@yahoo.com')

SELECT \* FROM email2

SELECT INSTR or LENGHT or SUBSTR FROM email2

ersan

james

sandra

sandra

SELECT

ersan cam

james curry

--- extract first name

================

**NUMBER functions**

SELECT 34.56 FROM DUAL

--- If we want to see xxxx.x digit with rounding

SELECT ROUND(34.56,1) FROM DUAL --> 34.6

SELECT ROUND(34.42,1) FROM DUAL --> 34.4

SELECT ROUND(34.42,2) FROM DUAL --> 34.42

SELECT TRUNC (34.42,1) FROM DUAL --> 34.4

SELECT TRUNC (34.42,0) FROM DUAL --> 34

===========================================

**DATE related functions**

Special Function for today' date:

CURRENT\_DATE and SYSDATE are functions to bring us today's date

**SELECT sysdate from DUAL**

**SELECT CURRENT\_DATE FROM DUAL**

Date + Number == > Date

Date – Number 🡺 Date

Date – Date 🡺 number of days

find health benefit enrollment day for each employee.

Please note that every person' benefit enrollment start 30 days after they got hired

SELECT first\_name, hire\_date, hire\_date+30 FROM employees

SELECT first\_name, hire\_date, hire\_date+900 FROM employees

Find Security checking date for each employee. Note than ideally each empl's sec. check is done 7 days before they start

SELECT first\_name, hire\_date, hire\_date-7 FROM employees

====

Q: Find out how many month pasted since we hired each employee?

SELECT last\_name, hire\_date FROM employees

MONTHS\_BETWEEN (date1, date2) it will the different in between date1-date2

SELECT last\_name, hire\_date, MONTHS\_BETWEEN (SYSDATE, hire\_date) as hmmonths

FROM employees

**Q: find out how many months past between today and publish date of each book?**

SELECT pubdate, MONTHS\_BETWEEN (SYSDATE, pubdate)

FROM books

SELECT \* FROM employees

ADD\_MONTHS: it add x number of months to given date. ADD\_MONTHS (sysdate, 3)

In this company everybody is suppose to get free medical and dental coverage after 3 months since they hired.

Find out for each employee when they are/were eligible to get free coverage

last\_name, first\_name, hire\_date, as eligbledate

SELECT last\_name, first\_name, hire\_date, ADD\_MONTHS (hire\_date,3) as eligbledate

FROM employees

SELECT ADD\_MONTHS(sysdate,3) FROM DUAL

SELECT SYSDATE FROM DUAL;

If you substract two dates from each other it return number of days in between

find how many days each employee have been working with this company

SELECT last\_name, sysdate-hire\_date FROM EMPLOYEES

find how many weeks each employee have been working with this company

SELECT last\_name, (sysdate-hire\_date)/7 FROM EMPLOYEES

find how many months each employee have been working with this company

SELECT last\_name, (sysdate-hire\_date)/30 FROM EMPLOYEES

Add 4 hours to today's date and tell me which time is gonna be

SELECT to\_char(SYSDATE, ' dd-mm-yyyy hh:mm am'), to\_char(SYSDATE + 4/24 , ' dd-mm-yyyy hh:mm am') FROM DUAL

1)how to find how many days past from the day we hire each employees?

SELECT employee\_id, last\_name, first\_name, SYSDATE - hire\_date AS numofday

FROM EMPLOYEES

2) How to find how many weeks past from the day we hire each employee?

SELECT employee\_id, last\_name, first\_name, (SYSDATE - hire\_date)/7 AS numofweeks

FROM EMPLOYEES

round this this number to closest 2 digit

SELECT employee\_id, last\_name, first\_name, ROUND ( (SYSDATE - hire\_date)/7,2 ) AS numofweeks

FROM EMPLOYEES

3) How to find how many months past from the day we hire each employee?

SELECT employee\_id, last\_name, first\_name, ROUND ( (SYSDATE - hire\_date)/7,2 ) AS numofweeks

FROM EMPLOYEES

this above one is not scientific

Scientific one is the one using MONTHS\_BETWEEN

SELECT employee\_id, last\_name, first\_name, MONTHS\_BETWEEN (SYSDATE , hire\_date) AS numofmonths

FROM EMPLOYEES

SELECT employee\_id, last\_name, first\_name, TRUNC (MONTHS\_BETWEEN (SYSDATE , hire\_date),0) AS numofmonths

FROM EMPLOYEES

--Date Type Data

select employee\_id, hire\_date from employees;

-- sysdate

select sysdate, current\_date + 1

from dual;

select employee\_id, (sysdate - hire\_date) / 7 as "Weeks"

from employees;

select employee\_id, trunc((sysdate - hire\_date) / 7,0) as "Weeks"

from employees

order by "Weeks";

-- ADD\_MONTHS

select hire\_date, add\_months(hire\_date, 1)

from employees;

-- CURRENT\_DATE

select add\_months(current\_date,2)

from dual;

-- current\_timestamp

select current\_timestamp

from dual;

-- EXTRACT

select extract (year from current\_date)

from dual;

select extract (month from current\_date)

from dual;

select extract (day from current\_date)

from dual;

-- LAST\_DAY

select last\_day(current\_date)

from dual;

-- MONTHS\_BETWEEN

select trunc(months\_between(current\_date , hire\_date),0)

from employees;

-- NEXT\_DAY

select next\_day(current\_date, 'Monday')

from dual;

-- TO\_CHAR

-- TO\_DATE

-- ROUND

select round(sysdate, 'MONTH')

from dual;

select round(sysdate, 'YEAR')

from dual;

-- TRUC

select trunc(sysdate, 'MONTH')

from dual;

select trunc(sysdate, 'MONTH')

from dual;

--

SELECT employee\_id,

hire\_date,

MONTHS\_BETWEEN (SYSDATE, hire\_date) "Seniority",

ADD\_MONTHS (hire\_date, 6) "Review Date",

NEXT\_DAY (hire\_date, 'Friday'),

LAST\_DAY (hire\_date)

FROM employees

WHERE MONTHS\_BETWEEN (SYSDATE, hire\_date) > 70;

SELECT employee\_id, last\_name, first\_name, ROUND (TRUNC (MONTHS\_BETWEEN (SYSDATE , hire\_date),0)/12,1) AS numofyears

FROM EMPLOYEES

Date to Char conversion

SELECT employee\_id, first\_name, hire\_date ,to\_char(hire\_date, 'dd -Mon-yyyy hh:mi am')

from employees

SELECT employee\_id, first\_name, hire\_date ,to\_char(hire\_date, 'day/month/year')

from employees

SELECT employee\_id, first\_name, hire\_date ,to\_char(hire\_date, 'dd-mm-yyyy hh:mi am')

from employees

SELECT employee\_id, first\_name, hire\_date ,to\_char(hire\_date, 'ddth -mm-yyyy hh:mi am')

from employees

select product\_id, product\_name, list\_price,

list\_price \* 1.01 as "New Price",

round(list\_price \* 1.01, 2) as "Rounded"

from products;

select round(2.552, 0)

from dual;

select trunc(2.352, 0)

from dual;

select mod(4,3)

from dual;

select employee\_id, mod(employee\_id, 2)

from employees;

select city, nvl(state, 'unkmown') as "State"

from locations;

select city, state, nvl2(state, 'state is known', 'state is unknown')

from locations;

select city, state, coalesce(state,city,'unknow')

from locations;

select \* from locations;

-- use round, trunc, Mod function

--

select last\_name, hire\_date,

TO\_CHAR (hire\_date, 'YYYY-Month-DD')

from employees;

SELECT EMPLOYEE\_ID,TO\_CHAR (HIRE\_DATE, 'MM/YY') Month\_Hired

FROM EMPLOYEES

WHERE LAST\_NAME like 'H%';

--

SELECT EMPLOYEE\_ID,TO\_CHAR (HIRE\_DATE, 'fm MM/DD/YY') HireDate

FROM EMPLOYEES

WHERE LAST\_NAME like 'H%';

SELECT EMPLOYEE\_ID,TO\_CHAR (HIRE\_DATE, 'MM/DD/YY') HireDate

FROM EMPLOYEES

WHERE LAST\_NAME like 'H%';

-- MONTH

select to\_char(sysdate, 'MONTH')

from dual;

select to\_char(hire\_date, 'MONTH')

from employees;

select to\_char(sysdate + 90, 'MON')

from dual;

select to\_char(sysdate, 'DAY')

from dual;

select to\_char(sysdate, 'WW') || 'th week of the year'

from dual;

select to\_char(sysdate, 'W') || 'th week of the month'

from dual;

select to\_char(sysdate, 'DD "of" MONTH')

from dual;

select to\_char(sysdate + 1, 'ddspth "of" MONTH')

from dual;

select to\_char(sysdate + 20, 'ddspth "of" MONTH')

from dual;

select sysdate,to\_char(sysdate + 10, 'DD-MONTH-YYYY')

from dual;

select to\_char(sysdate + 10, 'FMDD-MONTH-YYYY')

from dual;

SELECT last\_name,TO\_CHAR(sysdate, 'Ddspth "of" Month YYYY HH:MI')

FROM employees;

SELECT product\_name, TO\_CHAR(list\_price, '$99,999.00') as price

FROM products;

select to\_number('1234') - 2

from dual;

-- TO\_DATE

select current\_date, to\_date('09-01-2020', 'DD-MM-YYYY')

from dual;

select employee\_id, hire\_date

from employees

where hire\_date > to\_date('09-SEP-16', 'DD-MON-YY');

SELECT last\_name, hire\_date

from employees

where hire\_date = to\_date('May 24, 2016', 'Month DD, YYYY');

END OF DATE Functions

Select last\_name, salary

from employees

where last\_name like '%g'

SELECT results from soccer

WHERE teamname like 'PSG%'

SELECT results from soccer

WHERE teamname like 'Barcel%'

NVL vs NVL2 functions

=====================

Find out all the employees who earn some commission

commission\_pct .3 .4 NOT NULL

SELECT \* FROM EMPLOYEES

WHERE commission\_pct IS NOT NULL

Find out all the employees who do not earn any commission

or

commission\_pct NULL

SELECT \* FROM EMPLOYEES

WHERE commission\_pct IS NULL

SELECT last\_name, salary, commission\_pct FROM employees

Find out how much total salary each employees gets by adding copmmission into calculation

salary + salary \*commission\_pct

SELECT last\_name, salary, salary + salary\*commission\_pct as comsal

FROM employees

SELECT last\_name, salary, commission\_pct, salary + salary\* NVL(commission\_pct, 0) as aftcomcal

FROM employees

NVL2 has two options if it is null

SELECT order#, orderdate, shipdate FROM ORDERS

Using NVL2

SELECT order#, orderdate, shipdate, NVL2 (shipdate, 'Shipped', 'Not shipped')

FROM ORDERS

NULLIF

SELECT last\_name, LENGTH (last\_name) , first\_name, LENGTH (first\_name) FROM employees

SELECT last\_name, LENGTH (last\_name) , first\_name, LENGTH (first\_name) , NULLIF (LENGTH (last\_name),LENGTH (first\_name) )

FROM employees

=========

CASE function : it's job to check different conditions on data for each record and decide accordinly

SELECT last\_name, job\_id, salary,

CASE job\_id WHEN 'IT\_PROG' THEN 1.10\*salary

WHEN 'ST\_CLERK' THEN 1.15\*salary

WHEN 'SA\_REP' THEN 1.20\*salary

ELSE salary END "REVISED\_SALARY"

FROM employees;

Same thing can be done with DECODE as below

SELECT last\_name, job\_id, salary,

DECODE(job\_id, 'IT\_PROG', 1.10\*salary,

'ST\_CLERK', 1.15\*salary,

'SA\_REP', 1.20\*salary,

salary) REVISED\_SALARY

FROM employees;

SELECT \* from employees

list employee\_id , last\_name, first\_name , salary, job\_id, department\_id , MANUFACTURING

list employee\_id , last\_name, first\_name , salary, job\_id, department\_id , SALES FORCE on the field

IF employee is in 60 then it means MANUFACTURING

IF employee works for 80 then it means SALES FORCE

everybody else they in BACK OFFICE

SELECT employee\_id , last\_name, first\_name , salary, job\_id, department\_id ,

**CASE**

WHEN department\_id=50 THEN 'MANUFACTURING'

WHEN department\_id=60 THEN 'FRONT LINE'

WHEN department\_id=80 THEN 'SALES FORCE'

ELSE 'BACK OFFICE'

END AS WORKTYPE

FROM employees

SELECT last\_name, salary FROM EMPLOYEES

select min(salary) FROM EMPLOYEES

select max(salary) FROM EMPLOYEES

**Date related Activities**

-- Code Samples - May 29, 2020

-- lower upper

select product\_id, product\_name, list\_price,

list\_price \* 1.01 as "New Price",

round(list\_price \* 1.01, 2) as "Rounded"

from products;

select round(2.552, 0)

from dual;

select trunc(2.352, 0)

from dual;

select mod(4,3)

from dual;

select employee\_id, mod(employee\_id, 2)

from employees;

select city, nvl(state, 'unkmown') as "State"

from locations;

select city, state, nvl2(state, 'state is known', 'state is unknown')

from locations;

select city, state, coalesce(state,city,'unknow')

from locations;

select \* from locations;

-- use round, trunc, Mod function

--

select last\_name, hire\_date,

TO\_CHAR (hire\_date, 'YYYY-Month-DD')

from employees;

SELECT EMPLOYEE\_ID,TO\_CHAR (HIRE\_DATE, 'MM/YY') Month\_Hired

FROM EMPLOYEES

WHERE LAST\_NAME like 'H%';

--

SELECT EMPLOYEE\_ID,TO\_CHAR (HIRE\_DATE, 'fm MM/DD/YY') HireDate

FROM EMPLOYEES

WHERE LAST\_NAME like 'H%';

SELECT EMPLOYEE\_ID,TO\_CHAR (HIRE\_DATE, 'MM/DD/YY') HireDate

FROM EMPLOYEES

WHERE LAST\_NAME like 'H%';

-- MONTH

select to\_char(sysdate, 'MONTH')

from dual;

select to\_char(hire\_date, 'MONTH')

from employees;

select to\_char(sysdate + 90, 'MON')

from dual;

select to\_char(sysdate, 'DAY')

from dual;

select to\_char(sysdate, 'WW') || 'th week of the year'

from dual;

select to\_char(sysdate, 'W') || 'th week of the month'

from dual;

select to\_char(sysdate, 'DD "of" MONTH')

from dual;

select to\_char(sysdate + 1, 'ddspth "of" MONTH')

from dual;

select to\_char(sysdate + 20, 'ddspth "of" MONTH')

from dual;

select sysdate,to\_char(sysdate + 10, 'DD-MONTH-YYYY')

from dual;

select to\_char(sysdate + 10, 'FMDD-MONTH-YYYY')

from dual;

SELECT last\_name,TO\_CHAR(sysdate, 'Ddspth "of" Month YYYY HH:MI')

FROM employees;

SELECT product\_name, TO\_CHAR(list\_price, '$99,999.00') as price

FROM products;

select to\_number('1234') - 2

from dual;

-- TO\_DATE

select current\_date, to\_date('09-01-2020', 'DD-MM-YYYY')

from dual;

select employee\_id, hire\_date

from employees

where hire\_date > to\_date('09-SEP-16', 'DD-MON-YY');

SELECT last\_name, hire\_date

from employees

where hire\_date = to\_date('May 24, 2016', 'Month DD, YYYY');

Show me all books under 'cooKIng' category

SELECT \* FROM books

WHERE LOWER(category)='cooking'

SELECT \* FROM books

WHERE LOWER(category)= lower('cooKING')

SELECT \* FROM books

WHERE UPPER(category)= 'COOKING'

SELECT title, INITCAP(title) from books

SELECT title, SUBSTR (title,1,8) from books

SELECT title, SUBSTR (title,5) from books WHERE category='COMPUTER'

SELECT SUBSTR ('We will setup make up class for saturdays',14) FROM DUAL

SELECT SUBSTR ('We will setup make up class for saturdays',-5) FROM DUAL

LENGTH

Finding lenght of character set

SELECT LENGTH (title) from books

CONCAT Combines two characters set (Only two)

SELECT title, category FROM books

combine two columns into one column

SELECT CONCAT (title, category,isbn) as titleandcat

FROM books

SELECT CONCAT ('we will have ' ,'make up class') FROM DUAL

SELECT 'we will have ' ,'make up class' FROM DUAL

Use || instead of CONCAT to have unlimited option

SELECT 'name of the book: ' || title || ' and the category is: ' || category || ' and code ' || isbn as titleandcat

FROM books

select title, REPLACE (title,'BODY','MUSCLE') from books

select \* from books

WHERE isbn=1059831198

UPDATE books

SET title = 'MSCLBUILD in 10 MINUTES A DAY'

WHERE isbn=1059831198

COMMIT

ROLLBACK

SELECT employee\_id, first\_name, hire\_date ,to\_char(hire\_date, 'day/month/year')

from employees

SELECT employee\_id, first\_name, hire\_date ,to\_char(hire\_date, 'dd-mm-yyyy hh:mi am')

from employees

SELECT employee\_id, first\_name, hire\_date ,to\_char(hire\_date, 'ddth -mm-yyyy hh:mi am')

from employees

SELECT employee\_id, first\_name, hire\_date ,to\_char(hire\_date, 'dd -Mon-yyyy hh:mi am')

from employees

SELECT sysdate from dual

Date + Number

Date – Number

Date – Date

find health benefit enrollment day for each employee.

Please note that every person' benefit enrollment start 30 days after they got hired

SELECT first\_name, hire\_date, hire\_date+30 FROM employees

SELECT first\_name, hire\_date, hire\_date+900 FROM employees

Find Security checking date for each employee. Note than ideally each empl's sec. check is done 7 days before they start

SELECT first\_name, hire\_date, hire\_date-7 FROM employees

====

If you substract two dates from each other it return number of days in between

find how many days each employee have been working with this company

SELECT last\_name, sysdate-hire\_date FROM EMPLOYEES

find how many weeks each employee have been working with this company

SELECT last\_name, (sysdate-hire\_date)/7 FROM EMPLOYEES

find how many months each employee have been working with this company

SELECT last\_name, (sysdate-hire\_date)/30 FROM EMPLOYEES

Add 4 hours to today's date and tell me which time is gonna be

SELECT to\_char(SYSDATE, ' dd-mm-yyyy hh:mm am'), to\_char(SYSDATE + 4/24 , ' dd-mm-yyyy hh:mm am') FROM DUAL

**JL Books Database samples**

SELECT SYSDATE FROM DUAL;

SELECT ('CENTENNIAL') FROM DUAL;

-- WHAT IS DUAL.

DESC DUAL;

SELECT initcap('CENTENNIAL college') FROM DUAL;

SELECT LASTNAME,FIRSTNAME FROM CUSTOMERS WHERE lower(LASTNAME)='smith';

SELECT LASTNAME,FIRSTNAME ,address FROM CUSTOMERS WHERE

STATE=UPPER('&STATE');

SELECT \* FROM CUSTOMERS;

-- SUBSTRING

SELECT ADDRESS,SUBSTR(ADDRESS,10) FROM CUSTOMERS WHERE

ADDRESS LIKE 'P.O. BOX%';

SELECT ZIP,SUBSTR(ZIP,2,4) FROM CUSTOMERS;

SELECT ZIP,SUBSTR(ZIP,4) FROM CUSTOMERS;

SELECT FIRSTNAME,LENGTH(FIRSTNAME) FROM CUSTOMERS ORDER BY

LENGTH(FIRSTNAME) DESC;

-- CHECK YOUR NUMBER OF CHAR IN YOUR FIRSTNAME AND LASTNAME

SELECT LENGTH('VIJI') FROM DUAL;

SELECT \* FROM CUSTOMERS;

-- LTRIM AND RTRIM IN CUSTOMERS TABLE

SELECT ADDRESS,LTRIM(ADDRESS,'P.O. BOX') AS PBNUMBER FROM CUSTOMERS;

SELECT LTRIM(ADDRESS) || ' ' || RTRIM(CITY,'-') FROM CUSTOMERS;

--CONCAT

SELECT CONCAT(CONCAT(LASTNAME, '''s CITY IS '), CITY)

FROM CUSTOMERS;

SELECT CONCAT(CONCAT(firstname, ' ID: '),customer#) FROM customers;

SELECT \* FROM BOOKS;

SELECT LASTNAME, RTRIM(LASTNAME, 'S') FROM CUSTOMERS;

-- CHALLENGE 1

-- JOIN CUSTOMERS,ORDERS,ORDERITEM ,BOOKS . WHICH CUSTOMER

--HAS ORDERED WHAT BOOK.

--O/P BONITAS ORDERED THE BOOK BODYBUILD IN 1O

SELECT c.firstname || ' ' || c.lastname AS name, b.title

FROM customers c

JOIN orders o USING(customer#)

JOIN orderitems io USING(order#)

JOIN books b USING(isbn);

SELECT CONCAT(CONCAT(FIRSTNAME,' ORDERED '),TITLE) AS "CUSTOMER ORDER" FROM CUSTOMERS

JOIN ORDERS USING(CUSTOMER#)

JOIN ORDERITEMS USING (ORDER#)

JOIN BOOKS USING(ISBN);

SELECT CONCAT(CONCAT(CONCAT(CONCAT(c.lastname,'-'),C.FIRSTNAME),' ORDER THE BOOK '), b.title)

FROM customers c

JOIN orders o USING(customer#)

JOIN orderitems io USING(order#)

JOIN books b USING(isbn);

SELECT \* FROM ORDERS;

SELECT \* FROM ORDERITEMS;

-- CHALLEGE 2 . HOW MANY DAYS ARE BETWEEN ORDERDATE AND SHIPDATE

SELECT CONCAT(CONCAT(CONCAT(CONCAT(c.lastname,'-'),C.FIRSTNAME),' ORDER THE BOOK '), b.title),

(o.shipdate-o.orderdate) as days,o.SHIPDATE

FROM customers c

JOIN orders o USING(customer#)

JOIN orderitems io USING(order#)

JOIN books b USING(isbn)

where o.shipdate is not null;

SELECT order#,CONCAT(shipdate - orderdate,'DAY(S)') "DTIME" FROM

ORDERS ;

SELECT CONCAT(CONCAT(CONCAT(SHIPDATE, ' SHIPPED '), ORDERDATE), SHIPDATE - ORDERDATE) FROM ORDERS

INNER JOIN orderitems USING (ORDER#);

SELECT CONCAT(CONCAT(CONCAT(CONCAT('ORDER #', ORDER#), ' TOOK '),(SHIPDATE-ORDERDATE)), ' DAYS TO SHIP.')

AS DTIME

FROM ORDERS

WHERE SHIPDATE IS NOT NULL;

SELECT customer#,firstname || '''s order ' || DECODE(shipdate - orderdate, 0, 'shipped immediately!', 'took ' || (shipdate - orderdate) || ' day(s) to ship.') "Order info"

FROM customers

INNER JOIN orders USING (customer#)

WHERE shipdate IS NOT NULL

ORDER BY (shipdate - orderdate);