

Experiment 3

Built-In Functions and Transaction Control Language

- AIM - To work with Built-In Functions and Transaction Control Language in SQL
- Built-In Functions

Functions accept zero or more arguments and both return one or more results. They can be classified as :

- Single Row Functions
- Group Functions

Single Row Functions can be broadly classified as:

- Date Function
- Numeric Function
- Character Function
- Conversion Function
- Miscellaneous Function

- Date Function

- Add_month

Returns a date after adding a specific date with specific number of months.

Syntax :

Add_month (d, n)

d - date , n - number of months

- Select add_months (sysdate, 2) from dual;

Output - Date Functions

```
SQL> Select add_months(sysdate,2) from dual;  
  
ADD_MONTH  
-----  
04-APR-22  
  
SQL> select sysdate from dual;  
  
SYSDATE  
-----  
04-FEB-22  
  
SQL> Select add_months(sysdate,2) from dual;  
  
ADD_MONTH  
-----  
04-APR-22  
  
SQL> Select last_day ('1-feb-2022') from dual;  
  
LAST_DAY(  
-----  
28-FEB-22  
  
SQL> Select months_between('1-jun-2009','1-aug-2009') from dual;  
  
MONTHS_BETWEEN('1-JUN-2009','1-AUG-2009')  
-----  
-2  
  
SQL> Select months_between('11-jun-2009','1-aug-2009') from dual;  
  
MONTHS_BETWEEN('11-JUN-2009','1-AUG-2009')  
-----  
-1.6774194  
  
SQL> Select next_day (sysdate,'wednesday') from dual;  
  
NEXT_DAY(
```

→ last_day

Displays last date of that month.

Syntax :

last_day (d)

where d - date

> Select last_day ('1-feb-2022') from dual;

→ Months between

Gives difference in number of months between d1 & d2

Syntax :

month_between (d1, d2)

where d1 & d2 are dates

> Select month_between ('1-jun-2009', '1-aug-2009')
from dual;

→ next_day

Returns a day followed by the specified date.

Syntax :

next_day (d, day)

> Select next_day (sysdate, 'wednesday') from dual;

Date Functions

```
SQL> Select next_day (sysdate,'wednesday') from dual;  
  
NEXT_DAY(  
-----  
09-FEB-22  
  
SQL> Select round (to_date('1-jun-2009','dd-mm-yy'),'year') from dual;  
ERROR:  
ORA-01756: quoted string not properly terminated  
  
SQL> Select round ('1-jun-2009','year') from dual;  
ERROR:  
ORA-01756: quoted string not properly terminated  
  
SQL> Select round ('1-jun-2009','year') from dual;  
Select round ('1-jun-2009','year') from dual  
*  
ERROR at line 1:  
ORA-01722: invalid number  
  
SQL> Select round (to_date('1-jun-2009','dd-mm-yy'),'year') from dual;  
  
ROUND(TO_  
-----  
01-JAN-09  
  
SQL> Select round (to_date('31-dec-2009','dd-mm-yy'),'year') from dual;  
  
ROUND(TO_  
-----  
01-JAN-10  
  
SQL> select * from dual;
```

→ round

Returns date rounded to the unit specified by the format model.

Syntax :

round (d, [fmt]);

where d - date, [fmt] - format (optional)

- > Select round (to_date ('1-jun-2009', 'dd-mm-yy'), 'year') from dual;
- > Select round ('1-jun-2009', 'year') from dual;

o Numerical Functions

Command	Query	Output
Abs (n)	Select abs (-25) from dual;	25
Ceil (n)	Select ceil (55.67) from dual;	56
Exp (n)	Select exp(4) from dual;	54.59
Floor(n)	Select floor(100.2) from dual;	100
Powr(m,n)	Select powr('4',2) from dual;	16
Mod (m,n)	Select mod (10,3) from dual;	1
Round (m,n)	Select round (100.256,2) from dual;	100.26
Trunc (m,n)	Select trunc (100.256,2) from dual;	100.23
Sqrt (m,n)	Select sqrt (16) from dual;	4

Numerical Functions

```
SQL> select abs(-25) from dual;  
  
ABS(-25)  
-----  
      25  
  
SQL> select ceil(55.67) from dual  
2 ;  
  
CEIL(55.67)  
-----  
      56
```

```
SQL> select mod(10,3) from dual;  
  
MOD(10,3)  
-----  
      1  
  
SQL> select sqrt(16) from dual;  
  
SQRT(16)  
-----  
      4
```

Numerical Functions

```
SQL> select floor(100.2) from dual;  
  
FLOOR(100.2)  
-----  
      100  
  
SQL> select floor(100.9) from dual;  
  
FLOOR(100.9)  
-----  
      100  
  
SQL> select exp(4) from dual;  
  
EXP(4)  
-----  
    54.59815  
  
SQL> select exp(2) from dual;  
  
EXP(2)  
-----  
   7.3890561  
  
SQL> select power(5,3) from dual;  
  
POWER(5,3)  
-----  
     125  
  
SQL> select round(100.256,2) from dual;  
  
ROUND(100.256,2)  
-----  
    100.26  
  
SQL> select round(100.256,0) from dual;  
  
ROUND(100.256,0)  
-----  
     100  
  
SQL> select round(100.999,0) from dual;  
  
ROUND(100.999,0)  
-----  
     101
```

Character Functions

Command	Query	Output
initcap (char);	select initcap ('hello') from dual;	Hello
lower (char);	select lower ('HELLO') from dual;	hello
upper (char);	select upper ('hello') from dual;	HELLO
ltrim (char, (st));	select ltrim ('creat ', 'cr') from dual;	at
rtrim (char, (st));	select rtrim ('xit ', 'it') from dual;	xe
replace (char, search string, replace string)	select replace ('jack and jie', 'j', 'bl') from dual;	black and blue
substr (char, m,n);	select substr ('information', 3, 4) from dual;	form

Conversion Functions

→ to_char()

Syntax :

to_char (d, [format]);

>Select to_char (sysdate, 'dd-mm-yy') from dual;

→ to_date()

Syntax :

to_date (d, [format]);

>Select to_date ('aug 15 2009', 'mm-dd-yy') from dual;

Character Functions

```
SQL> select initcap('hello') from dual;  
  
INITC  
-----  
Hello  
  
SQL> select lower('HELLo') from dual;  
  
LOWER  
-----  
hello  
  
SQL> select upper('helloworld') from dual;  
  
UPPER(  
-----  
HELLOO  
  
SQL> select ltrim('cseswe','cse') from dual;  
  
LT  
--  
we  
  
SQL> select ltrim('csexwe','cse') from dual;  
  
LTR  
---  
xwe  
  
SQL> select rtrim('csext','xt') from dual;  
  
RTR  
---  
cse
```

Character Functions

```
SQL> select replace('jack and jue','j','bl') from dual;

REPLACE('JACKA
-----
black and blue

SQL> select rtrim('cseewe','we') from dual;

RT
--
cs

SQL> select rtrim('cesewe','we') from dual;

RTR
---
ces

SQL> select rtrim('cesswe','we') from dual;

RTI
----
cess

SQL> select rtrim('csesewwe','we') from dual;

RTI
---
cses

SQL> select substr('information',3,4) from dual;

SUBS
-----
form

SQL> select substr('information',5,4) from dual;

SUBS
-----
rmat
```

Conversion Functions

```
SQL> select to_char (sysdate, 'dd-mm-yy') from dual;
TO_CHAR(
-----
04-02-22

SQL> select to_char (sysdate, 'mm-dd-yy') from dual;
TO_CHAR(
-----
02-04-22

SQL> select to_date('aug 15 2009', 'mm-dd-yy') from dual;
ERROR:
ORA-01756: quoted string not properly terminated

SQL> select to_date('aug 15 2009','mm-dd-yy') from dual;
TO_DATE(
-----
15-AUG-09

SQL> elect uid from dual;
SP2-0734: unknown command beginning "elect uid ..." - rest of line ignored.
SQL> select uid from dual;

      UID
-----
      90

SQL> select user from dual;
USER
-----
RA1911033010035

SQL> select to_date('aug 15 2009','mm#dd#yy') from dual;
TO_DATE(
-----
15-AUG-09

SQL> desc employee
   Name          Null?    Type
   ----          -----    -----
EMP_ID           NOT NULL NUMBER(38)
NAME            VARCHAR2(30)
ADDRESS          VARCHAR2(50)
```

- Miscellaneous Functions

→ uid

Returns the integer value (id) corresponding to the user currently logged in.

> Select uid from dual;

→ user

This function returns the login user's name

> Select user from dual;

→ nvl

Used in case where we want to consider null values as zero.

Syntax :

nvl (exp1, exp2)

If exp1 is null, return exp2. If exp1 is not null,

> return exp1

> Select emp_id, name, nvl(dept, 'default') from employee;

→ vsize

Returns the number of bytes in expression

> Select vsize ('tech') from dual;

Miscellaneous Functions

```
SQL> elect uid from dual;
SP2-0734: unknown command beginning "elect uid ..." - rest of line ignored.
SQL> select uid from dual;

      UID
-----
      90

SQL> select user from dual;

USER
-----
RA1911033010035
```

```
SQL> select * from employee;

      EMP_ID NAME
----- ADDRESS          SALARY
DEPT
----- 1 Harsh           8000.56
SRM
CSE
      2 Alexis           6690.11
Chennai
CSE
      EMP_ID NAME
----- ADDRESS          SALARY
```

Miscellaneous Functions

2	Alexis	
Chennai		6690.11
CSE		
EMP_ID NAME		

ADDRESS		SALARY

DEPT		

3	Anna	
Delhi		6623.9
ECE		
4	Jean	
Mumbai		24479.11
EMP_ID NAME		

ADDRESS		SALARY

DEPT		

CIVIL		
5	Mala	
Pune		24479.11
CIVIL		
6	Malax	
EMP_ID NAME		

ADDRESS		SALARY

DEPT		

Goa		234.71
CSE		
108	hi	
dfg b		4567
7 rows selected.		

Miscellaneous Functions

```
SQL> select emp_id,name,nvl(dept,'default') from employee;

EMP_ID NAME          NVL(DEPT, 'DEFAULT')
----- -----
 1 Harsh           CSE
 2 Alexis          CSE
 3 Anna            ECE
 4 Jean            CIVIL
 5 Mala            CIVIL
 6 Malax           CSE
108 hi             default

7 rows selected.

SQL> select vsize('tech') from dual;
ERROR:
ORA-01756: quoted string not properly terminated

SQL> select vsize('tech') from dual;

VSIZE('TECH')
-----
 4

SQL> select vsize ('hello') from dual;

VSIZE('HELLO')
-----
 5

SQL> select vsize (34567890) from dual;

VSIZE(34567890)
-----
 5

SQL> select vsize (345) from dual;

VSIZE(345)
-----
```

• Group Functions

Returns result based on group of rows.

→ avg

Syntax :

avg (n)

→ Select avg (salary) from employee;

→ max

Syntax :

max (n)

→ Select max (salary) from employee;

→ min

Syntax :

min (n)

→ Select min (salary) from employee;

→ sum

Syntax :

sum (n)

→ Select sum (salary) from employee;

Group Functions

```
SQL> select avg (salary) from employee;  
  
AVG(SALARY)  
-----  
10724.9286  
  
SQL> select max (salary) from employee;  
  
MAX(SALARY)  
-----  
24479.11  
  
SQL> select min(salary) from employee;  
  
MIN(SALARY)  
-----  
234.71  
  
SQL> select sum(salary) from employee;  
  
SUM(SALARY)  
-----  
75074.5
```

- Count Function.

Used to count the number of rows

→ Count (*)

Counts all inclusive of duplicates and nulls

> Select count (*) from employee;

→ Count (col_name)

Avoids null value

> Select count (dept) from employee;

→ Count (distinct col_name)

Avoids repeated and null values

> Select count (distinct emp_id) from employee;

> Select count (distinct dept) from employee;

- Group by clause

Allows to use simultaneous column names and group functions

> Select max (salary), emp_id from employee
group by emp_id;

- Having clause.

Used to specify conditions on rows retrieved by using group by clause.

> Select max (salary), emp_id from employee group by emp_id having count(*) >= 1;

Count Functions

sqlplus.exe - Shortcut

```
SQL> select count(*) from employee;  
COUNT(*)  
-----  
8  
  
SQL> select count(dept) from employee;  
COUNT(DEPT)  
-----  
6  
  
SQL> select count(distinct emp_id) from employee;  
COUNT(DISTINCTEMP_ID)  
-----  
8  
  
SQL> select count(distinct dept) from employee;  
COUNT(DISTINCTDEPT)  
-----  
3  
  
SQL> select count(dept) from employee;  
COUNT(DEPT)  
-----  
6
```

Group By and Having Clause

sqlplus.exe - Shortcut

```
SQL> select max(salary),emp_id from employee group by emp_id;

MAX(SALARY)      EMP_ID
-----
 8000.56          1
 6690.11          2
 6623.9           3
 24479.11         4
 24479.11         5
 234.71           6
 4567             108
 4567             109

8 rows selected.
```

```
SQL> select max(salary),emp_id from employee group by emp_id having count(*)>=1;

MAX(SALARY)      EMP_ID
-----
 8000.56          1
 6690.11          2
 6623.9           3
 24479.11         4
 24479.11         5
 234.71           6
 4567             108
 4567             109

8 rows selected.
```

Transaction Control Language

TCL Statements :

- Commit
- Rollback
- Savepoint

State of Data before Commit or Rollback :

- The previous state of the data can be recovered.
- Current user can review the results of DML operations using Select Command.
- Other users cannot view results of DML statements.
- The affected rows are locked.

State of Data After Commit :

- Data changes are permanent in the database.
- Previous state is permanently lost.
- All users can view results.
- All savepoints are erased.

- Q Change Address= 'Finland' for donorID= 106 in donor table and commit the transaction.
- > Update donor set address= 'Finland' where donorID = 106 ;
- > Commit ;

TCL – COMMIT

Command Prompt - mysql -u root -p

```
mysql> select * from donor;
+-----+-----+-----+-----+-----+-----+-----+-----+
| donorID | firstName | lastname | address | phone | dob | bloodType | quantity |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 101 | Alex | Paul | Germany | 4556789098 | 2000-03-01 | O+ | 230.12 |
| 102 | Bell | Decroix | Dutch | 3333339098 | 2003-05-06 | B+ | 111.12 |
| 103 | Cass | James | US | 6556569098 | 2001-06-11 | AB+ | 356.89 |
| 104 | Jess | Day | UK | 678977098 | 2002-08-04 | O- | 456.77 |
| 106 | Jerry | k | NULL | NULL | NULL | B+ | 211.78 |
| 107 | Terry | P | NULL | NULL | NULL | AB+ | 211.78 |
| 108 | Mia | M | US | 56789907 | 2000-03-07 | B+ | 431.18 |
| 109 | Anna | May | UK | 223489907 | 2004-05-07 | B- | 543.18 |
+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.02 sec)

mysql> update table donor set address='Finland' where donorID=106;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'table donor' at line 1
mysql> update donor set address='Finland' where donorID=106;
Query OK, 1 row affected (0.17 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from donor;
+-----+-----+-----+-----+-----+-----+-----+-----+
| donorID | firstName | lastname | address | phone | dob | bloodType | quantity |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 101 | Alex | Paul | Germany | 4556789098 | 2000-03-01 | O+ | 230.12 |
| 102 | Bell | Decroix | Dutch | 3333339098 | 2003-05-06 | B+ | 111.12 |
| 103 | Cass | James | US | 6556569098 | 2001-06-11 | AB+ | 356.89 |
| 104 | Jess | Day | UK | 678977098 | 2002-08-04 | O- | 456.77 |
| 106 | Jerry | k | Finland | NULL | NULL | B+ | 211.78 |
| 107 | Terry | P | NULL | NULL | NULL | AB+ | 211.78 |
| 108 | Mia | M | US | 56789907 | 2000-03-07 | B+ | 431.18 |
| 109 | Anna | May | UK | 223489907 | 2004-05-07 | B- | 543.18 |
+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.03 sec)

mysql>
```

- Rollback

States of data after Rollback :

- Discard all pending changes by using the Rollback statement.
- Data changes are undone.
- Previous state of the data is restored.
- Locks on the affected rows are released.

- Q Add a new row to employee table then rollback the transaction
- > Insert into Employee values (110, 'hi', 'dfg b', 4567, 'hiko');

> Rollback transaction

o Rollback to a Marker

→ Create a marker within a current transaction using:
<Savepoint savepoint-name>

→ Rollback to that marker using:
<Rollback to savepoint-name>

TCL – ROLLBACK

```
SQL> insert into employee values (110,'hi','dfg b',4567,'hiko');
```

```
1 row created.
```

```
SQL> select * from employee;
```

EMP_ID	NAME	ADDRESS	DEPT	SALARY
1	Harsh	SRm		8000.56
		CSE		
2	Alexis	Chennai		6690.11
		CSE		

EMP_ID	NAME	ADDRESS	DEPT	SALARY
3	Anna	Delhi		6623.9
		ECE		
4	Jean	Mumbai		24479.11

TCL – ROLLBACK

ADDRESS	SALARY
DEPT	
CIVIL	
5 Mala	
Pune	24479.11
CIVIL	
6 Malax	
EMP_ID NAME	
ADDRESS	SALARY
DEPT	
Goa	234.71
CSE	
108 hi	
dfg b	4567
EMP_ID NAME	
ADDRESS	SALARY
DEPT	
109 hi	
dfg b	4567
110 hi	
dfg b	4567
hiko	
EMP_ID NAME	
ADDRESS	SALARY
DEPT	

```
SQL> rollback;
Rollback complete.

SQL> select * from employee;

EMP_ID NAME
-----
ADDRESS                               SALARY
DEPT

          1 Harsh           8000.56
SRm
CSE

          2 Alexis           6690.11
Chennai
CSE

EMP_ID NAME
-----
ADDRESS                               SALARY
DEPT

          3 Anna             6623.9
Delhi
ECE

          4 Jean             24479.11
Mumbai

EMP_ID NAME
-----
ADDRESS                               SALARY
DEPT

CIVIL

          5 Mala             24479.11
Pune
CIVIL

          6 Malax
```

Q. Start a transaction and make a savepoint.

- > Start Transaction;
- > Savepoint a;

Q Change the address of user Mia To 'UK' from table personnel.

- > Update Personnel set address = 'UK' where firstName = 'Mia';

Q Rollback to the savepoint created above.

- > Rollback to savepoint a;

o Result:

Built - In functions and Transaction Control language was were understood and implemented in SQL.

TCL – SAVEPOINT AND ROLLBACK

```
ERROR 1505 (42000): SAVEPOINT s1 does not exist
```

```
mysql> START TRANSACTION;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from personnel;
```

firstName	lastname	address	email	phone	dob
Alex	LAX	Australia	xyz@gmail.com	4556789098	2000-03-01
Sia	I	UK	SIA@gmail.com	34567890	2001-07-01
Kia	K	USA	KIA@gmail.com	111227890	2002-07-02
Mia	M	US	MIA@gmail.com	111227890	2002-07-02

```
4 rows in set (0.00 sec)
```

```
mysql> savepoint a;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> Update Personnel set address='UK' where firstName='Mia';
```

```
Query OK, 1 row affected (0.02 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> rollback to savepoint a;
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from personnel;
```

firstName	lastname	address	email	phone	dob
Alex	LAX	Australia	xyz@gmail.com	4556789098	2000-03-01
Sia	I	UK	SIA@gmail.com	34567890	2001-07-01
Kia	K	USA	KIA@gmail.com	111227890	2002-07-02
Mia	M	US	MIA@gmail.com	111227890	2002-07-02

```
4 rows in set (0.04 sec)
```