Ex.No.8 30.04.2025	STRING FUNCTONS

AIM:

To perform sql operations by using string functions.

CREATING A TABLES :

SQL> create table student(id number(5), name varchar2(20), address varchar2(20));

Table created.

SQL> create table department(did number(5),name varchar2(20), dname varchar2(20), email varchar(30));

Table created.

INSERTING VALUES INTO TABLE :

SQL> insert into student values(1, 'kabesh', 'namakkal'); 1
row created.

SQL> insert into student values(2,'jegan', 'karur'); 1
row created.

SQL> insert into student values(3, 'jeyasanjay', 'theni');
1 row created.

SQL> insert into student values(4, 'iyyappan', 'coimbatore');
1 row created.

SQL> insert into student values(5, 'naveen', 'erode');
1 row created.

SQL> insert into student values(6, 'job', 'tiruppur'); 1
row created.

SQL> insert into student values(7, 'kamalesh', 'trichy'); 1
row created.

SQL> insert into student values(8, 'jeevan', 'namakkal');

1 row created.
SQL> insert into department values(201, 'kabesh', 'it', 'kabesh.23it@kongu.edu'); 1
row created.

SQL> insert into department values(202, 'jegan', 'eee',
'jegan.23eee@kongu.edu');
1 row created.

SQL> insert into department values(203, 'jeyasanjay', 'cs',
'jeyasanjay.23cs@kongu.edu'); 1
row created.

SQL> insert into department values(204, 'iyyappan', 'cse',
'iyyappan.23cse@kongu.edu'); 1
row created.

SQL> insert into department values(205, 'naveen', 'ml',
'naveen.23ml@kongu.edu'); 1
row created.

SQL> insert into department values(206, 'job', 'aiml', 'job.23aiml@kongu.edu'); 1
row created.

SQL> insert into department values(207, 'kamalesh', 'ft', 'kamalesh.23ft@kongu.edu'); 1
row created.

SQL> insert into department values(208, 'jeevan', 'mts', 'jeevan.23mts@kongu.edu'); 1
row created.

SQL> create view dual as select * from student,department where student.name=department.name;
View created.

ASCII (CHR):

SQL>select ascii('k') as ascii_value from dual;

ASCII_VALUE

75

VALUE(CHR):

SQL>select chr(75) as character from dual;
CHARACTER

---------- K

CONCAT:

SQL>select name || ' lives in ' || address as std_details from student;

STD_DETAILS
---
kabesh lives in namakkal
jegan lives in karur
jeyasanjay lives in theni
iyyappan lives in coimbatore
naveen lives in erode job
lives in tiruppur kamalesh
lives in trichy jeevan lives
in namakkal

8 rows selected.

UPPERCACSE & LOWERCACSE:

SQL>select upper(name) as upper_name, lower(name) as lower_name from student;

UPPER_NAME	LOWER_NAME
KABESH	kabesh
JEGAN	jegan
JEYASANJAY	jeyasanjay
IYYAPPAN	iyyappan
NAVEEN	naveen
JOB	job
KAMALESH	kamalesh
JEEVAN	jeevan

8 rows selected.

LENGTH(STR):

SQL>select name, length(name) as name_length from student;

NAME	NAME_LENGTH
Kabesh	6

Jegan	5
Jeyasanjay	10
Iyyappan	8
Naveen	7
Job	3
Kamalesh	8
Jeevan	6

8 rows selected.

READDRESS(STR):

```
SQL>select readdress(name, 'jeyasanjay', 'ram') as readdress_name from student;

READDRESS_NAME
----------------------------
Kabesh
Jegan
Ram
Iyyappan
Naveen
Job
Kamalesh
Jeevan

8 rows selected.
```

SUBSTR():

```
SQL>select substr('kabesh rocks', 7, 5) as substring from dual;

SUBST
-----------
Rocks
```

RPAD() & LPAD():

```
SQL>select rpad(name, 10, '*') as right_pad, lpad(name, 10, '*') as left_pad from student;

RIGHT_PAD          LEFT_PAD
--------------     --------------
Kabesh**           **Kabesh
Jegan***           ***Jegan
```

Jeyasanjay	Jeyasanjay
Iyyappan**	**Iyyappan
Naveen**	**Naveen
Job***	***Job
Kamalesh**	**Kamalesh
Jeevan**	**Jeevan

8 rows selected.

LTRIM & RTRIM:

```
SQL>select ltrim('  kabesh') as ltrim, rtrim('kabesh  ') as rtrim from dual;

LTRIM          RTRIM


------         ------

KABESH         KABESH
```

REVERSE:

```
SQL>select reverse(name) as reverse_name from student;

REVERSE_NAME


-----------------
    hsebaK
nageJ
yajnasayeJ
nappayyI
neevaN      boJ
hselamaK
    naveeJ

8 rows selected.
```

EXTRACTING THE DOMITN OF EMAIL:

```
SQL>select substr(email, instr(email, '@') + 1) as email_domain from department where email is not
null;

EMAIL_DOMAIN


-------------------------
kongu.edu kongu.edu
kongu.edu kongu.edu
```

kongu.edu kongu.edu
kongu.edu kongu.edu

8 rows selected.

CHANGING THE DOMITN NAME:

SQL>select readdress(email, substr(email, instr(email, '@') + 1), 'gmail.com') as upftted_mail from department;

UPFTTED_MAIL

---

kabesh.23it@gmail.com

jegan.23eee@gmail.com

jeyasanjay.23cs@gmail.com

iyyappan.23cse@gmail.com

Naveen.23ml@gmail.com job.23aiml@gmail.com

kamalesh.23ft@gmail.com

jeevan.23mts@gmail.com

8 rows selected.

SEARCHING OPERATIONS:

LIKE KEYWORD:

SQL>select dname from department where dname like 'd%';

DNAME

EEE
FT

SQL> select dname from department where dname like '_a%';

DNAME

FT

SQL> select dname from department where dname not like 's%';

DNAME

IT

EEE

CS

ML

AIML

FT

MTS

7 rows selected.

NULL:

SQL> select * from department where email is null;

no rows selected

NOT NULL:

SQL> select * from department where email is not null;

DID	DNAME	EMAIL
201	IT	kabesh.23it@kongu.edu
202	EEE	jegan.23eee@kongu.edu
203	CS	jeyasanjay.23cs@kongu.edu
204	CSE	iyyappan.23cse@kongu.edu
205	ML	Naveen.23ml@kongu.edu
206	AIML	job.23aiml@kongu.edu
207	FT	kamalesh.23ft@kongu.edu
208	MTS	jeevan.23mts@kongu.edu

8 rows selected.

CONTENTS	MARKS ALLOTED	MARKS OBTITNED
Aim,Algorithm,SQL,PL/SQL	30	
Execution and Result	20	
Viva	10	
Total	60	

RESULT

Thus various string operations are performed. The operations helped in manipulating, formatting and searching the string function efficiently based on different conditions.