12: Advanced Data Manipulation Using SQL

selected SQL-92 built-in functions

SQL-92 Function	Oracle Implementation	MySQL Implementation	Sections Containing Useful Examples in Chapter 12
CASE	DECODE	Function not available	12.4
CHAR_LENGTH	LENGTH	LENGTH**	12.2.2
Concatenation	CONCAT	CONCAT**	12.2.1
CURRENT_DATE	CURRENT_DATE	CURRENT_DATE or CURDATE()	12.3
CURRENT_TIME	CURRENT_DATE (With Mask)	CURRENT_TIME or CURTIME()	12.3
CURRENT_USER	USER	CURRENT_USER	12.1.2.4
Date/Time Conversions	TO_CHAR, TO_DATE	See below*	12.3
EXTRACT	EXTRACT	EXTRACT	
LOWER and UPPER	LOWER and UPPER	LOWER and UPPER	12.1.2.4
POSITION	INSTR	INSTR** or LOCATE**	12.2.5
SUBSTRING	SUBSTR	SUBSTRING**	12.2.1
TRANSLATE	TRANSLATE	Function not available	12.2.4
TRIM	LTRIM and RTRIM	LTRIM and RTRIM**	12.2.3

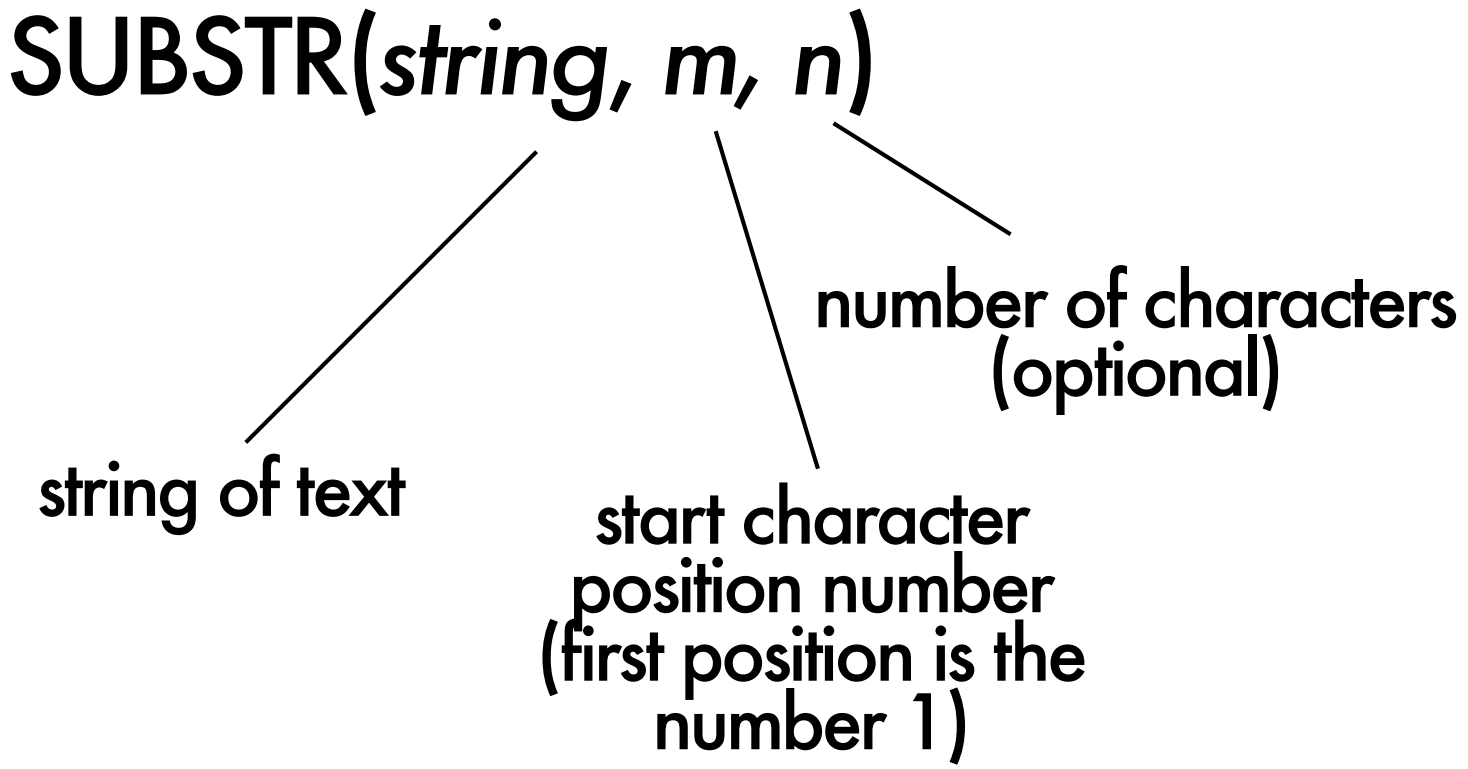
*MySQL supports a series of functions to extract date and time elements.
**Both the Oracle and MySQL implementation of these functions are covered in this text to illustrate the similarities and differences in how they are implemented across products. For additional information on these as well as on other SQL-92 functions not covered in this book, the reader is encouraged to refer to the considerable product-specific documentation available online and in books.

single row character functions

SUBSTR
LENGTH
LTRIM
RTRIM
LPAD/RPAD
INSTR

substr

extract a portion of a string of text



substr

extract a portion of a string of text

```
SELECT SUBSTR('ABCDEFG',3,4)
   "Substring" FROM DUAL;
```

CDEF

dummy table

```
SELECT SUBSTR('ABCDEFG',3)
   "Substring" FROM DUAL;
```

CDEFG

substr

extract a portion of a string of text

count from last character

```
SELECT SUBSTR('ABCDEFG',-5,4)
    "Substring" FROM DUAL;
```

CDEF

concatenation - page 608

combining strings of text

||

"pipes" (not the number 11)

```
SELECT 'hot ' || 'dogs' FROM DUAL;
```

hot dogs

concatenation - page 608

combining strings of text

display the name and phone number of all professors with phone numbers that end with two digits ranging between 45 and 65

```
SELECT PROFESSOR.PR_NAME, '('||
SUBSTR(PROFESSOR.PR_PHONE,1,3)||')'||
SUBSTR(PROFESSOR.PR_PHONE,4,3)||'-'||
SUBSTR(PROFESSOR.PR_PHONE,7,4) "PHONE" FROM PROFESSOR
WHERE SUBSTR(PROFESSOR.PR_PHONE, 9,2) BETWEEN 45 AND 65;
```

length

returns the length of a string

LENGTH(*string*)

```
SELECT LENGTH ('Jones, John') FROM DUAL;
```

11

with char data type, will return the length including trailing blanks

length

returns the length of a string

```
SELECT TEXTBOOK.TX_TITLE,
  TEXTBOOK.TX_PUBLISHER,
LENGTH(TEXTBOOK.TX_TITLE)
    FROM TEXTBOOK;
```

varchar

Tx_title	Tx_publisher	LENGTH(TEXTBOOK.Tx_title)
Database Management	Thomson	19
Linear Programming	Prentice-Hall	18
Simulation Modeling	Springer	19
Systems Analysis	Thomson	16
Principles of IS	Prentice-Hall	16
Economics For Managers		22
Programming in C++	Thomson	18
Fundamentals of SQL		19
Data Modeling		13

length

returns the length of a string

```
SELECT TEXTBOOK.TX_TITLE,
  TEXTBOOK.TX_PUBLISHER,
LENGTH(TEXTBOOK.TX_PUBLISHER)
    FROM TEXTBOOK;
```

char

Tx_title	Tx_publisher	LENGTH(TEXTBOOK.Tx_publisher)
Database Management	Thomson	13
Linear Programming	Prentice-Hall	13
Simulation Modeling	Springer	13
Systems Analysis	Thomson	13
Principles of IS	Prentice-Hall	13
Economics For Managers		
Programming in C++	Thomson	13
Fundamentals of SQL		
Data Modeling		13

ltrim - page 611

remove unwanted characters from the left side of a string

LTRIM(*string, set of characters*)

optional, defaults to removing leading blanks

```
SELECT LTRIM('      LAST WORD') FROM DUAL;
```

LAST WORD

```
SELECT LTRIM('xxxXxxLAST WORD', 'x') FROM
DUAL;
```

XxxLAST WORD

ltrim - page 611

remove unwanted characters from the left side of a string

```
SELECT TEXTBOOK.TX_TITLE,
LTRIM(TEXTBOOK.TX_TITLE, 'Systems')
"Trimmed Title" FROM TEXTBOOK;
```

Tx_title	Trimmed Title
Database Management	Database Management
Linear Programming	Linear Programming
Simulation Modeling	Simulation Modeling
Systems Analysis	Analysis
Principles of IS	Principles of IS
Economics For Managers	Economics For Managers
Programming in C++	Programming in C++
Fundamentals of SQL	Fundamentals of SQL
Data Modeling	Data Modeling

rtrim

remove unwanted characters from the right side of a string

RTRIM(*string, set of characters*)

optional, defaults to removing leading blanks

```
SELECT RTRIM ('STINSONxxXxx','x') "Right
      Trim Example" FROM DUAL;
```

STINSONxxX

rtrim and length

bringing it together

```
SELECT TEXTBOOK.TX_TITLE,
  TEXTBOOK.TX_PUBLISHER,
LENGTH(TEXTBOOK.TX_PUBLISHER) “Length Pub",
RTRIM(TEXTBOOK.TX_PUBLISHER) “Trimmed Pub",
    LENGTH(RTRIM(TEXTBOOK.TX_PUBLISHER))
                "Trimmed Length"
  FROM TEXTBOOK;
```

rtrim and length

bringing it together

Tx_title	Tx_publisher	Length Pub	Trimmed Pub	Trimmed Length
Database Management	Thomson	13	Thomson	7
Linear Programming	Prentice-Hall	13	Prentice-Hall	13
Simulation Modeling	Springer	13	Springer	8
Systems Analysis	Thomson	13	Thomson	7
Principles of IS	Prentice-Hall	13	Prentice-Hall	13
Economics For Managers				
Programming in C++	Thomson	13	Thomson	7
Fundamentals of SQL				
Data Modeling		13		

rpad and lpad

not discussed in textbook

LPAD(*string*, *length*, *set*)

optional, defaults to adding leading blanks

RPAD(*string*, *length*, *set*)

optional, defaults to adding trailing blanks

rpad and lpad

not discussed in textbook

```
SELECT LPAD('Page 1', 14, '*.') "LPAD
       Example" FROM DUAL;
```

..*.*.Page 1

```
SELECT LPAD('Page 1', 14) "LPAD Example"
       FROM DUAL;
```

Page 1

there are
leading blanks here

rpad and lpad

not discussed in textbook

```
SELECT RPAD ('Page 1', 14, '*.') "RPAD
Example" FROM DUAL;
```

Page 1*.*.*.*.

```
SELECT RPAD(‘Page 1', 14) "RPAD Example"
FROM DUAL;
```

Page 1

there are
trailing blanks here

instr

return numeric value of the location of a character string

INSTR(*string to search in, string to find, start position, number occurrence*)

optional, defaults to one

optional, defaults to first

instr

return numeric value of the location of a character string

```
SELECT INSTR ('MISSISSIPPI','S',5,2) "In
       String Example" FROM DUAL;
```

7

```
SELECT INSTR ('MISSISSIPPI','S',5,1) "In
       String Example" FROM DUAL;
```

6

instr and substr together

begin with the second word of each textbook title and
display all titles with the character string ‘ing’ in the title

```
SELECT TEXTBOOK.TX_TITLE,
    INSTR(SUBSTR(TEXTBOOK.TX_TITLE,
 INSTR(TEXTBOOK.TX_TITLE, ' ‘)+1),'ing')
          "ing string in word 2”,
  INSTR(SUBSTR(TEXTBOOK.TX_TITLE,1),’ing')
"ing string in overall title” FROM TEXTBOOK
   WHERE INSTR(SUBSTR(TEXTBOOK.TX_TITLE,
 INSTR(TEXTBOOK.TX_TITLE, ' ')+1),'ing') >
                    0;
```

instr and substr together

begin with the second word of each textbook title and
display all titles with the character string 'ing' in the title

Tx_title	ing string in word 2	ing string in overall title
Linear Programming	9	16
Simulation Modeling	6	17
Data Modeling	6	11

instr and substr together

display the names of all professors who
have the same first name as at least one other professor

```
SELECT PROFESSOR.PR_NAME FROM PROFESSOR
  WHERE SUBSTR(PROFESSOR.PR_NAME, 1,
INSTR(PROFESSOR.PR_NAME,' ')-1) IN (SELECT
    SUBSTR(PROFESSOR.PR_NAME, 1,
  INSTR(PROFESSOR.PR_NAME,' ')-1)
     FROM PROFESSOR GROUP BY
    SUBSTR(PROFESSOR.PR_NAME, 1,
  INSTR(PROFESSOR.PR_NAME,' ')-1)
      HAVING COUNT(*) > 1);
```

```
Pr_name
---------------
John Smith
John B Smith
John Nicholson
Mike Faraday
Mike Crick
```

dates

each platform handles dates differently

oracle's default display of dates is:

19-NOV-13

but the date is stored in julian calendar format

2456615.5

days since January 1st, 4713 B.C. @ noon

dates

show the age of each professor in department 3 when they were hired

```
SELECT PROFESSOR.PR_NAME,
TRUNC((PROFESSOR.PR_DATEHIRED -
PROFESSOR.PR_BIRTHDATE)/365.25,0)
"Age When Hired" FROM PROFESSOR WHERE
PROFESSOR.PR_DPT_DCODE = 3;
```

Pr_name	Age When Hired
Chelsea Bush	46
Tony Hopkins	47
Alan Brodie	56
Jessica Simpson	40
Laura Jackson	26

dates

show the current age in days and years of each professor in department 3

```
SELECT PROFESSOR.PR_NAME,
CURRENT_DATE - PROFESSOR.PR_BIRTHDATE "Age
in Days", TRUNC((CURRENT_DATE -
PROFESSOR.PR_BIRTHDATE)/365.25,0) "Age in
Years" FROM PROFESSOR
WHERE PROFESSOR.PR_DPT_DCODE = 3;
```

Pr_name	Age in Days	Age in Years
Chelsea Bush	21876.8535	59
Tony Hopkins	20698.8535	56
Alan Brodie	22839.8535	62
Jessica Simpson	18598.8535	50
Laura Jackson	11971.8535	32

dates

display the number of professors born during each decade of the 1900s.

```
SELECT 19||SUBSTR(PROFESSOR.PR_
BIRTHDATE,8,1)||0||'''s' "Decade of", COUNT
(*) "Number Born"FROM PROFESSOR
 WHERE PROFESSOR.PR_BIRTHDATE IS NOT NULL
GROUP BY SUBSTR(PROFESSOR.PR_BIRTHDATE,8,1)
```

Decade of	Number Born
1940's	4
1950's	2
1960's	6
1970's	6

to_char and to_date

convert a date to a string format or a string to a date

TO_CHAR

extract the different parts of a date/time/
number and convert them to a character string

TO_DATE

convert character strings to a valid date

to_char and to_date

page 619

Element	Description	Example
MONTH, Month, or month	Name of the month spelled out—padded with blank spaces to a total width of nine spaces; case follows format.	JULY, July, or july (5 spaces follows each representation of July)
MON, Mon, or mon	Three-letter abbreviation of the name of the month; case follows format.	JUL, Jul, or jul
MM	Two-digit numeric value of the month.	7
D	Numeric value of the day of the week.	Monday = 2
DD	Numeric value of the day of the month.	23
DAY, Day, or day	Name of the day of the week spelled out—padded with blank spaces to a length of nine characters.	MONDAY. Monday, or Monday (3 spaces follows each representation of Monday)
fm	"Fill mode." When this element appears, subsequent elements (such as MONTH) suppress blank padding leaving a variable-length result.	fmMonth, yyyy produces a date such as March, 2007
DY	Three-letter abbreviation of the day of the week.	MON, Mon, or mon
YYYY	The four-digit year.	2007
YY	The last two digits of the year.	07
YEAR, Year, or year	Spells out the year; case follows year.	TWO THOUSAND SEVEN
BC or AD	Indicates B.C. or A.D.	2007 A.D.
AM or PM	Meridian indicator	10:00 AM
J	Julian date. January 1, 4712 B.C. is day 1.	July 27, 2007 is Julian date 2454309
SS	Seconds (value between 0 and 59)	21
MI	Minutes (value between 0 and 59)	32
HH	Hours (value between 1 and 12)	9
HH24	Hours (value between 0 and 23)	13

to_char and to_date

page 619

Element	Description	Example
9	Series of 9s indicates width of display (with insignificant leading zeros not displayed).	99999
0	Displays insignificant leading zeros.	0009999
$	Displays a floating dollar sign to prefix value.	$99999
.	Indicates number of decimals to display.	999.99
,	Displays a comma in the position indicated.	9,999

to_char

```
SELECT PROFESSOR.PR_NAME,
 TO_CHAR(PROFESSOR.PR_BIRTHDATE,'DD-
MON-YYYY') "Alternate Format”,
TO_CHAR(PROFESSOR.PR_SALARY, '$99,999.00')
"Alternate Format"
FROM PROFESSOR
WHERE PROFESSOR.PR_DPT_DCODE = 3;
```

Pr_name	Alternate Format	Alternate Format
Chelsea Bush	03-SEP-1946	$77,000.00
Tony Hopkins	24-NOV-1949	$77,000.00
Alan Brodie	14-JAN-1944	$76,000.00
Jessica Simpson	25-AUG-1955	$67,000.00
Laura Jackson	16-OCT-1973	$43,000.00

to_char

```
SELECT PATIENT.PAT_NAME,
TO_CHAR(PAT_ADMIT_DT, 'fmMonth dd, yyyy HH:
MI AM') "Date of Admission"
FROM PATIENT;
```

Pat_name	Date of Admission
Davis, Bill	July 7, 2007 12:00 AM
Li, Sue	August 25, 2007 12:00 AM
Grimes, David	July 12, 2007 12:00 AM

dates

```
INSERT INTO PATIENT VALUES ('ZZ', '06912',
        'Zhang, Zhaoping', 'F', 35,
TO_DATE('2007-08-11 10:15 AM', 'YYYY-MM-DD
        HH:MI AM'), NULL, NULL, NULL);
```

dates

```
SELECT PATIENT.PAT_NAME,
TO_CHAR(PAT_ADMIT_DT, 'fmMonth dd, yyyy HH:
MI AM') "Date of Admission"
FROM PATIENT;
```

Pat_name	Date of Admission
Davis, Bill	July 7, 2007 12:00 AM
Li, Sue	August 25, 2007 12:00 AM
Zhang, Zhaoping	August 11, 2007 10:15 AM
Grimes, David	July 12, 2007 12:00 AM

dates

suppose we discharge zhaoping zhang at 3:35 pm on august 12

```
SELECT PATIENT.PAT_NAME, CURRENT_DATE -
  PATIENT.PAT_ADMIT_DT "Length of Stay"
FROM PATIENT WHERE PATIENT.PAT_P#A = 'ZZ'
     AND PATIENT.PAT_P#N = ‘06912';
```

```
Pat_name                  Length of Stay
------------------------- --------------
Zhang, Zhaoping                1.2222338
```

additional date functions

ADD_MONTHS

add months to a date to signal a target date
in the future

MONTHS_BETWEEN

determines the number of months between two dates

assertions and triggers

ASSERTION

enforce a rule on an entire relation

ex: at least 100 orders present in the system

TRIGGER

if an event (insert, delete, update) meets a certain condition, perform an action

ex: insert a row into the PATIENT_AUDIT table whenever a patient is deleted from the clinic

VIEW

a single table derived
from a relational expression
(think of it as a stored query)

```
CREATE VIEW senior_citizen AS SELECT
patient.Pat_name, patient.Pat_age,
patient.Pat_gender FROM patient WHERE
patient.Pat_age > 64;
```

```
select * from senior_citizen;
```