Database Systems 2

Lecture 7

Dates and Functions

Functions

- Date Formats
- Date and time functions
- Number Formats

- Arithmetic functions
- Character functions
- Conversion functions

Date Formats

The SQL2003 standard specifies two data types:

- DATE and TIME
- One for **time** and one for **dates**

Oracle does things **slightly differently**.

It **just** has the **DATE** data type.

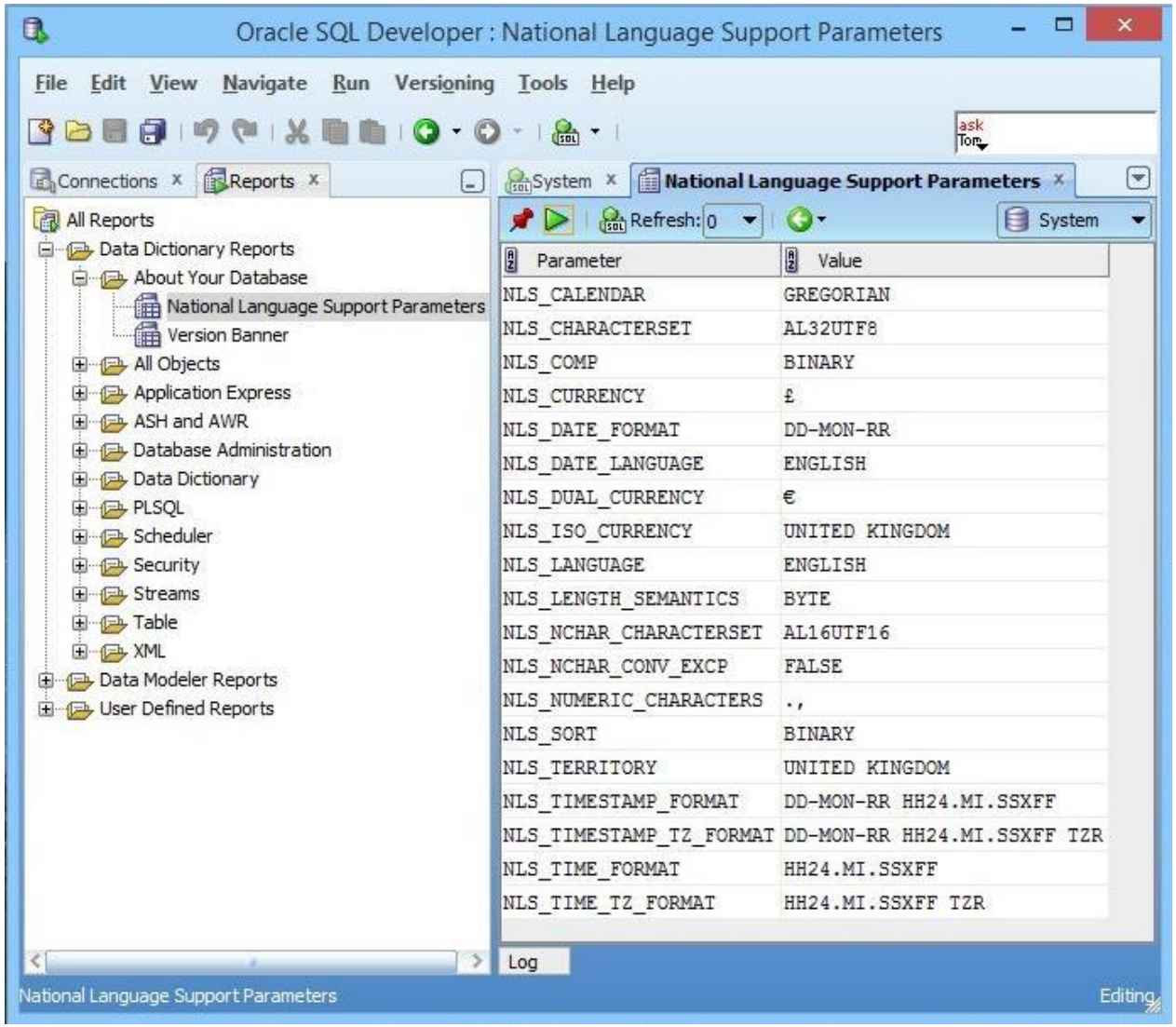
Date Formats

DATE stores its value in a special internal format that includes:

`Year - Month - Day - Hour - Minute - Second`

If you want to insert a date, or a time, you must supply the date in the default format that Oracle expects dates to be in:

Default Date Format



Common Format Codes

MM	**Numeric month**
MON	**Abbreviated month name**
MONTH	**Full month name**
DD	**Numeric day**
DY	**Abbreviated name of day**
YYYY	**4 digit year**
YY	**2 digit year**
AM	**Shows AM or PM according to time**
HH	**Hour of day (1-12)**
HH24	**Hour of day (0-23)**
MI	**Minute (0-59)**
SS	**Second (0-59)**

Inputting Dates and Times

See Oracle documentation for a complete list of the format codes.

http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements004.htm

```
INSERT INTO employee (birthday) VALUES ('02-MAR-13');

INSERT INTO employee (starttime) VALUES ('13.05.30');
```

Inputting Dates and Times

If you don't want to supply the date in the default format, you have to tell Oracle what format the date is in, so that it can convert the date into the default format, using TO_DATE.

```
INSERT INTO employee (birthday) VALUES (
        TO_DATE('Apr 27, 2007', [redacted]));

INSERT INTO employee (starttime) VALUES (
        TO_DATE('8:00 PM', 'hh:mi am'));
```

Displaying Dates and Times

When a date value is displayed, Oracle converts it from the internal format to a printable string.

Again, it uses Oracle's default date format, **"DD-MON-YY"**

So if you typed: **`SELECT birthday FROM employee;`**

You would get something like this:

```
birthday
---------
02-JUN-08
```

Displaying Dates and Times

You can override the default by calling TO_CHAR explicitly with your own format.

```
SELECT TO_CHAR(birthday,                    ) AS datebirth
FROM employee;
```

will return:

```
datebirth
----------
2008/06/02
```

Note:

Capitalisation in a format code will be reflected in the displayed data:

DAY	will produce	TUESDAY
Day	will produce	Tuesday
day	will produce	tuesday

You can use **hyphens, slashes, commas, full stops and colons** as well as any quoted characters in your format code.

To test your format codes

```
SELECT To_Char(SYSDATE, 'Day HH:MI:SS am') AS time
FROM dual;
```

dual is a dummy table name which is used when the command doesn't really require any data to work on.

You can also use SYSTIMESTAMP.

SELECT SYSTIMESTAMP FROM DUAL;

SYSTIMESTAMP

28-MAR-00 12.38.55.538741 PM -08:00

Default Date Format

You can change the default DATE format of Oracle by typing something like:

```
ALTER SESSION SET
NLS_DATE_FORMAT='YY/MM/DD';
```

'24/04/20'

But note that the change is only valid for the current SQL*Plus session - i.e. until you logout of Oracle.

DateTime Functions

ADD_MONTHS(date, n)

Returns the specified datetime plus n months

CURRENT_DATE

Returns the current datetime - same as SYSDATE but takes into account the timezone

EXTRACT (datepart, date)

Returns specified part of the date

LAST_DAY(date)

Returns the date of the last day of the month that contains date

MONTHS_BETWEEN(date1, date2)

Returns number of months between the specified dates

DateTime Functions 2

NEXT_DAY(date, dayname)

Returns the datetime of the first day with the specified name following date

NUMTODSINTERVAL(n, interval_unit)

Can specify a number of Days, Hours, Minutes or Seconds

NUMTOYMINTERVAL(n, interval_unit)

Can specify a number of Years or Months

ROUND (date, unit)

Returns a date rounded up to the specified unit

TRUNC (date, unit)

Returns date truncated to the specified unit

Number Formats

Number Formats

In the same way, you can specify a format for the display or input of numbers, using:

```
TO_CHAR( number, format)     or
TO_NUMBER(string, format)
```

For example

```
SELECT TO_CHAR(-10000,'L99G999D99MI') AS Amount
FROM DUAL;
```

Will show

```
Amount
--------------
$10,000.00-
```

Number Formats

Again, refer to Oracle documentation for more on number formats,

http://docs.oracle.com/cd/B19306_01/server.102/b14200/sql_elements004.htm

Arithmetic Functions

Arithmetic Functions

Abs(n) — Returns the **absolute value of** *n*. i.e. it changes all the negative numbers to positive and leaves the positive alone.

Abs(1)=1; Abs(-1)=1

Ceil(n) — Returns the **smallest integer greater than or equal to** *n*

Ceil(2.5) = 3; Ceil(2.9)=3; Ceil(2.1)=3

Floor(n) — Returns the **largest integer equal to or less** than *n*

Floor(2.5) = 2; Floor(2.9)=2; Floor(2.1)=2

Mod(m,n) — Returns the **remainder** of m divided by *n*

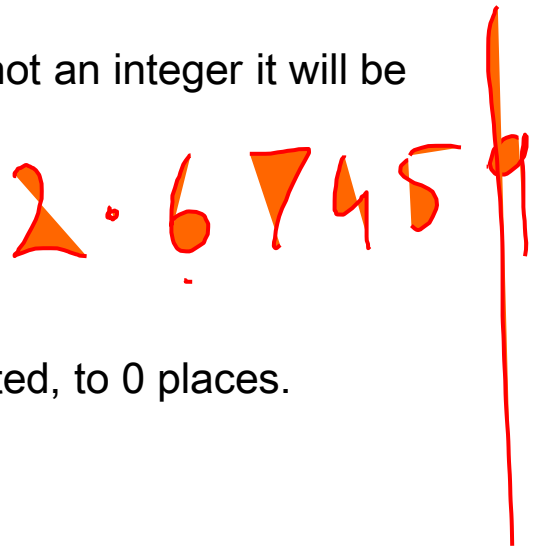
7/2 = 3, Remainder = 1; Therefore Mod(7,2) = 1

Arithmetic Functions

Power(m,n)

Returns *m* **raised to the** *nth* **power**. If *n* is not an integer it will be truncated.

2^4=16; **`Power(2,4)=16`**



Round(n[,m])

Rounded to *m* decimal places; if m is omitted, to 0 places.

`Round(2.8472985345,1)=2.8`

Sqrt(n)

Returns the **square root of *n.*** If n is less than 0, NULL is returned.

`Scrt(16)=4`

Trunc(n[,m])

n truncated to *m* decimal places. If *m* is omitted to 0 decimal places.

`Trunc(2.8472985345,2)=2.84`

But we can also multiply etc.

```
SELECT sal, sal*1.13 AS newsal
FROM emp;

SAL    NEWSAL
----   --------
800     904
1600   1808
1250   1412.5
2975   3361.75
1250   1412.5
2850   3220.5
2450   2768.5
```

More than one function in query

```
SELECT sal, ROUND(sal*1.13 ) AS "Round newsal",
                              TRUNC(sal*1.13) AS "Trunc newsal"
FROM emp;
```

SAL	ROUND NEWSAL	TRUNC NEWSAL
800	904	904
1600	1808	1808
1250	1413	1412
2975	3362	3361
1250	1413	1412
2850	3221	3220
2450	2769	2768

Character Functions

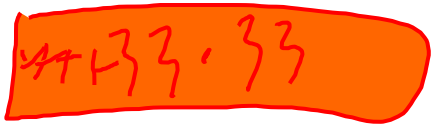
Character Functions

Initcap(char)	The first letter of each word is capitalised **`Initcap("example") = "Example"`**
Length(char)	Returns the length of the *char* **`Length("example") = 7`**
Lower(char)	The contents of the field are altered to lower case **`Lower("BIG") = "big"`**

Character Functions

Lpad(char, n, [,char2]) Left padded to length n with the characters in char2; or if char2 is omitted, with blanks

```
Lpad("example",10,".") = "...example"
```



Ltrim(char,[set]) Removes characters from the left of *char,* with initial characters removed up to the first *set; set* defaults to ' ', a single blank

```
Ltrim("example","exam") = "ple"
```

Character Functions

Replace(char, set [,set2])	Replaces the initial character *set* found within *char* with the second character *set2.* If the second character *set2* is omitted, it defaults to nothing.

```
SELECT ename, REPLACE(ename, 'AR', '**') AS replacement
FROM emp;
```

ENAME	REPLACEMENT
SMITH	SMITH
ALLEN	ALLEN
WARD	W**D
JONES	JONES
MARTIN	M**TIN
BLAKE	BLAKE
CLARK	CL**K
SCOTT	SCOTT
KING	KING
TURNER	TURNER

Character Functions

Rpad(char, n, [,char2]) Right-padded to length *n* with the characters in *char2;* if *char2* is omitted, then with blanks

```
Right("sample string",15, 'x')
                        =  "sample  stringxx"
```

Rtrim(char,[set]) Removes characters specified in *set* from the right of *char. set* defaults to ' ', a single blank.

```
Rtrim("example    ")  =  "example";

Rtrim("sample stringxx", 'x')
                          =  "ample string"
```

Character Functions

Substr(char,m[,n]) Returns the portion *of char,* beginning at character m, *n* characters long.

Substr("example",2,4)="xamp"

Translate(char,from,to)

"In Oracle/PLSQL, the translate function **replaces** a **sequence of characters** in a string with **another set of characters**. However, it replaces a single **character at a Time**.

http://www.techonthenet.com/oracle/functions/translate.php

```
Translate('1tech23', '123', '456');  would return       '4tech56'

Translate('222tech', '2ec', '3it');  would return        '333tith'
```

Character Functions

Upper(char)

All the letters are forced to upper case

Upper("example") = "EXAMPLE";

Conversion Functions

Conversion Functions

To_char(n[,fmt])	
To_char(d[,fmt])	**Converts a number** (n) **or a date** (d) **to a character value in the format specified** by fmt.
To_date(char[,fmt])	**Converts a char value to a date**; if the **format** is omitted the char must have the format 'DD-MON-YY.
To_number(char)	**Converts char value to number**