File data loading

- ❑Excel data import by for update
- ❑Load data by import sql developer
- ❑Load data by using loader as ctl

Note: table should be present in db.

DATE FUNCTIONS

- ❑ Sysdate
- ❑ Current_date
- ❑ Current_timestamp
- ❑ Systimestamp
- ❑ Localtimestamp
- ❑ Dbtimezone
- ❑ Sessiontimezone
- ❑ To_char
- ❑ To_date
- ❑ Add_months
- ❑ Months_between
- ❑ Next_day
- ❑ Last_day
- ❑ Extract

- ❑Greatest
- ❑Least
- ❑Round
- ❑Trunc
- ❑New_time
- ❑Coalesce

DATE FUNCTIONS

Oracle default date format is DD-MON-YY.

We can change the default format to our desired format by using the following command.

SQL> alter session set nls_date_format = ‘DD-MONTH-YYYY’;

But this will expire once the session was closed.

DATE FUNCTIONS

SYSDATE

This will give the current date and time.

Ex:

```
SQL> select sysdate from dual;
```

CURRENT_DATE

This will returns the current date in the session’s timezone.

Ex:

```
SQL> select current_date from dual;
```

DATE FUNCTIONS

CURRENT_TIMESTAMP

This will returns the current timestamp with the active time zone information.

Ex:

```
SQL> select current_timestamp from dual;
```

SYSTIMESTAMP

This will returns the system date, including fractional seconds and time zone of the database.

Ex:

```
SQL> select systimestamp from dual;
```

DATE FUNCTIONS

TO_CHAR

This will be used to extract various date formats.

The available date formats as follows.

Syntax: to_char (date, format)

DATE FUNCTIONS

DATE FORMATS

D -- No of days in week
DD -- No of days in month
DDD -- No of days in year
MM -- No of month
MON -- Three letter abbreviation of month
MONTH -- Fully spelled out month
RM -- Roman numeral month
DY -- Three letter abbreviated day
DAY -- Fully spelled out day
Y -- Last one digit of the year
YY -- Last two digits of the year
IW -- No of weeks in year from ISO standard
HH -- Hours
MI -- Minutes
YYY -- Last three digits of the year
YYYY -- Full four digit year
SYYYY -- Signed year
I -- One digit year from ISO standard
IY -- Two digit year from ISO standard
IYY -- Three digit year from ISO standard
IYYY -- Four digit year from ISO standard
Y, YYY -- Year with comma
YEAR -- Fully spelled out year
CC -- Century
Q -- No of quarters
W -- No of weeks in month
WW -- No of weeks in year

DATE FUNCTIONS

Ex:

```
SQL> select to_char(sysdate,'dd month yyyy hh:mi:ss am dy') from dual;
```

DATE FUNCTIONS

```
SQL> select to_char(sysdate,'dd month yyyy hh:mi:ss am dy') from dual;

TO_CHAR(SYSDATE,'DD MONTH YYYYHH:MI
```

```
SQL> select to_char(sysdate,'dd month year') from dual;

TO_CHAR(SYSDATE,'DDMONTHYEAR’)
```

```
SQL> select to_char(sysdate,'dd fmmonth year') from dual;

TO_CHAR(SYSDATE,'DD FMMONTH YEAR’)
```

```
SQL> select to_char(sysdate,'ddth DDTH') from dual;
```

DATE FUNCTIONS

```
SQL> select to_char(sysdate,'ddspth DDSPTH') from dual;

TO_CHAR(SYSDATE,'DDSPTHDDSPTH


SQL> select to_char(sysdate,'ddsp Ddsp DDSP ') from dual;

TO_CHAR(SYSDATE,'DDSPDDSPDDSP’)
```

DATE FUNCTIONS

TO_DATE

This will be used to convert the string into data format.

Syntax: to_date (date)

Ex:

```
SQL> select to_char(to_date('24/dec/2006','dd/mon/yyyy'), 'dd * month * day') from
dual;
```

Note-- If you are not using to_char oracle will display output in default date format.

DATE FUNCTIONS

ADD_MONTHS

This will add the specified months to the given date.

Syntax: add_months (date, no_of_months)

Ex:

```
SQL> select add_months(to_date('11-jan-1990','dd-mon-yyyy'), 5) from dual;
```

DATE FUNCTIONS

ADD_MONTHS

This will add the specified months to the given date.

Syntax: add_months (date, no_of_months)

Ex:

```
SQL> select add_months(to_date('11-jan-1990','dd-mon-yyyy'), 5) from dual;
```

DATE FUNCTIONS

```
SQL> select add_months(to_date('11-jan-1990','dd-mon-yyyy'), -5) from dual;
```

ADD_MONTH

If no_of_months is zero then it will display the same date.

If no_of_months is null then it will display nothing.

DATE FUNCTIONS

MONTHS_BETWEEN

This will give difference of months between two dates.

Syntax: months_between (date1, date2)

Ex:

```
SQL> select months_between(to_date('11-aug-1990','dd-mon-yyyy'), to_date('11-jan-1990','dd-mon-yyyy')) from dual;
```

DATE FUNCTIONS

NEXT_DAY

This will produce next day of the given day from the specified date.

Syntax: next_day (date, day)

Ex:

```
SQL> select next_day(to_date('24-dec-2006','dd-mon-yyyy'),'sun') from dual;
```

DATE FUNCTIONS

LAST_DAY

This will produce last day of the given date.

Syntax: last_day (date)

Ex:

```
SQL> select last_day(to_date('24-dec-2006','dd-mon-yyyy'),'sun') from dual;
```

DATE FUNCTIONS

EXTRACT

This is used to extract a portion of the date value.

Syntax: extract ((year | month | day | hour | minute | second), date)

Ex:

```
SQL> select extract(year from sysdate) from dual;
```

Note-- You can extract only one value at a time.

DATE FUNCTIONS

GREATEST

This will give the greatest date.

Syntax: greatest (date1, date2, date3 … daten)

Ex:

```
SQL> select greatest(to_date('11-jan-90','dd-mon-yy'),to_date('11-mar-90','dd-monyy'),
to_date('11-apr-90','dd-mon-yy')) from dual;
```

DATE FUNCTIONS

LEAST

This will give the least date.

Syntax: least (date1, date2, date3 … daten)

Ex:

```
SQL> select least(to_date('11-jan-90','dd-mon-yy'),to_date('11-mar-90','dd-monyy'),
to_date('11-apr-90','dd-mon-yy')) from dual;
```

DATE FUNCTIONS

ROUND

Round will rounds the date to which it was equal to or greater than the given date.

Syntax: round (date, (day | month | year))

```
SQL> select round(to_date('24-dec-04','dd-mon-yy'),'year'), round(to_date('11-mar-06','dd-mon-yy'),'year') from dual;

SQL> select round(to_date('11-jan-04','dd-mon-yy'),'month'), round(to_date('18-jan-04','dd-mon-yy'),'month') from dual;

SQL> select round(to_date('26-dec-06','dd-mon-yy'),'day'), round(to_date('29-dec-06','dd-mon-yy'),'day') from dual;

select to_char(round(to_date('24-dec-06','dd-mon-yy')), 'dd mon yyyy hh:mi:ss am') from dual;
```

DATE FUNCTIONS

TRUNC

Trunc will chops off the date to which it was equal to or less than the given date.

Syntax: trunc (date, (day | month | year))

```
SQL> select trunc(to_date('24-dec-04','dd-mon-yy'),'year'), trunc(to_date('11-mar-06','dd-mon-yy'),'year') from dual;
```

```
SQL> select trunc(to_date('11-jan-04','dd-mon-yy'),'month'), trunc(to_date('18-jan-04','dd-mon-yy'),'month') from dual;
```

DATE FUNCTIONS

```
SQL> select trunc(to_date('26-dec-06','dd-mon-yy'),'day'), trunc(to_date('29-dec-06','ddmon-yy'),'day') from dual;
```

```
SQL> select to_char(trunc(to_date('24-dec-06','dd-mon-yy')), 'dd mon yyyy hh:mi:ss am') from dual;
```

DATE FUNCTIONS

NEW_TIME

This will give the desired timezone’s date and time.

Syntax: new_time (date, current_timezone, desired_timezone)

```
SQL> select to_char(new_time(sysdate,'gmt','yst'),'dd mon yyyy hh:mi:ss am') from dual;
```

DATE FUNCTIONS

Available timezones are as follows.

TIMEZONES

AST/ADT -- Atlantic standard/day light time

BST/BDT -- Bering standard/day light time

CST/CDT -- Central standard/day light time

EST/EDT -- Eastern standard/day light time

GMT -- Greenwich mean time

HST/HDT -- Alaska-Hawaii standard/day light time

MST/MDT -- Mountain standard/day light time

NST -- Newfoundland standard time

PST/PDT -- Pacific standard/day light time

YST/YDT -- Yukon standard/day light time

DATE FUNCTIONS

COALESCE

This will give the first non-null date.

Syntax: coalesce (date1, date2, date3 … daten)

Ex:

```
SQL> select coalesce('12-jan-90','13-jan-99'), coalesce(null,'12-jan-90','23-mar-98',null)
from dual;
```

MISCELLANEOUS FUNCTIONS

Uid

User

Vsize

Rank

Dense_rank

MISCELLANEOUS FUNCTIONS

UID

This will returns the integer value corresponding to the user currently logged in.

Ex:

```
SQL> select uid from dual;
```

MISCELLANEOUS FUNCTIONS

USER

This will returns the login's user name.

Ex:

```
SQL> select user from dual;
```

MISCELLANEOUS FUNCTIONS

VSIZE

This will returns the number of bytes in the expression.

Ex:

```
SQL> select vsize(123), vsize('computer'), vsize('12-jan-90') from dual;
```

MISCELLANEOUS FUNCTIONS

RANK

This will give the non-sequential ranking.

Ex:

```
SQL> select rownum,sal from (select sal from emp order by sal desc);
```

MISCELLANEOUS FUNCTIONS

DENSE_RANK

This will give the sequential ranking.

Ex:

```
SQL> select dense_rank(2975) within group(order by sal desc) from emp;
```

CONVERSION FUNCTIONS

BIN_TO_NUM

This will convert the binary value to its numerical equivalent.

Syntax: bin_to_num(binary_bits)

Ex:

```
SQL> select bin_to_num(1,1,0) from dual;
```