

DMSTEX Business Solutions

DDGS Presentation 2

Generated API

Sponsored by DMSTEX Business Solutions

- Founded 2010
- 25 Years Oracle Experience
- Database Design and Generation Service
 - Oracle Database Schema
 - PL/SQL Application Programming Interfaces
 - Application Specific Test Data
 - APEX Data Maintenance Forms

Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates.

Database Design and Generation Service (DDGS) Presentations

A Series of 5 Presentations

1. DDGS Deliverables
2. **DDGS Generated API**
3. DDGS Generated Data Handling
4. DDGS Generated Subtypes
5. DDGS Generated History and Audit

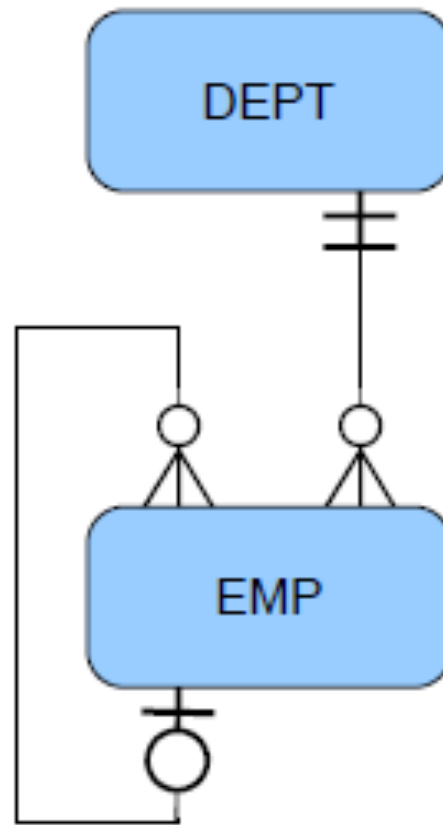
Presentations do not cover all functionality

2. DDGS Generated API

- Documentation
 - E-R Diagram
 - Configuration Addendum
- Insert API
- Update API
- Delete API
- Query and Clear

Documentation

E-R Diagram



Documentation

Configuration Addendum

Column Name	NK	REQ	FLD	WSR	Default
-----	---	---	---	---	-----
id		X			emp_SEQ.nextval
ename	1		U		
job		X			
-- Domain: P1,JOB					
mgr_emp_id					
-- Foreign Key: mgr_ID_PATH, mgr_NK_PATH, mgr_emp_nk1					
hiredate		X			
-- DATE Truncate every 1 Days of the Month					
sal		X			
comm					
dept_id		X			
-- Foreign Key: dept_nk1					

CONSTRAINTS:

10) Only SALESMAN can be on commission

2. DDGS Generated API

- Documentation
- Insert API
 - Table SQL (2 slides)
 - View SQL (2 slides)
 - Column Based
 - Column Based with Return Values (2)
 - Record Based (3 slides)
 - Dimension Load
- Update API
- Delete API
- Query and Clear

Insert API - Table SQL

Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER (38)
ENAME	NOT NULL	VARCHAR2 (16)
JOB	NOT NULL	VARCHAR2 (9)
MGR_EMP_ID		NUMBER (38)
HIREDATE	NOT NULL	DATE
SAL	NOT NULL	NUMBER (7, 2)
COMM		NUMBER (7, 2)
DEPT_ID	NOT NULL	NUMBER (38)

Insert API - Table SQL

```
SQL> insert into emp (ename, job, mgr_emp_id,  
2      hiredate, sal, dept_id)  
3          values ('Duane', 'CLERK', 5,  
4      '01-JUL-10', 100, 2);
```

1 row created.

```
SQL> select id, ename, job, mgr_emp_id, dept_id  
2      from emp where ename = 'DUANE';
```

ID	ENAME	JOB	MGR_EMP_ID	DEPT_ID
15	DUANE	CLERK	5	2

Insert API - View SQL

Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER (38)
ENAME	NOT NULL	VARCHAR2 (16)
JOB	NOT NULL	VARCHAR2 (9)
MGR_EMP_ID		NUMBER (38)
MGR_ID_PATH		VARCHAR2 (4000)
MGR_NK_PATH		CLOB
MGR_EMP_NK1		VARCHAR2 (16)
HIREDATE	NOT NULL	DATE
SAL	NOT NULL	NUMBER (7, 2)
COMM		NUMBER (7, 2)
DEPT_ID	NOT NULL	NUMBER (38)
DEPT_NK1	NOT NULL	VARCHAR2 (14)

Insert API - View SQL

```
SQL> insert into emp_act (id, ename, job, mgr_emp_nk1,  
2      hiredate, sal, dept_nk1)  
3          values (-1, 'LISA', 'ANALYST', 'JONES',  
4      '02-JUL-10', 1000, 'RESEARCH');
```

1 row created.

```
SQL> select id, ename, job, mgr_emp_id, dept_id  
2      from emp_act where ename = 'LISA';
```

ID	ENAME	JOB	MGR_EMP_ID	DEPT_ID
-1	LISA	ANALYST	2	2

Insert API - Column Based

```
SQL> BEGIN
  2     EMP_DML.ins(ename_in      => 'TOM'
  3                   ,job_in      => 'CLERK'
  4                   ,mgr_emp_nk1_in => 'MARTIN'
  5                   ,hiredate_in  => '02-JUL-10'
  6                   ,sal_in       => 200
  7                   ,dept_nk1_in  => 'SALES');
  8 END;
  9 /
```

PL/SQL procedure successfully completed.

Insert API

Column Based with return values

```
SQL> describe DEPT
```

Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER(38)
DNAME	NOT NULL	VARCHAR2(14)
LOC	NOT NULL	VARCHAR2(13)

```
SQL> describe DEPT_ACT
```

Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER(38)
DNAME	NOT NULL	VARCHAR2(14)
LOC	NOT NULL	VARCHAR2(13)

Insert API

Column Based with return values

```
SQL> set serveroutput on
```

```
SQL> DECLARE
```

```
2     dept_id          number;
```

```
3     dept_name        varchar2(14) := 'HR';
```

```
4     dept_location    varchar2(13) := 'CHICAGO';
```

```
5 BEGIN
```

```
6     DEPT_DML.ins2(id_io      => dept_id
```

```
7                    ,dname_io => dept_name
```

```
8                    ,loc_io    => dept_location);
```

```
9     dbms_output.put_line('dept_id = ' || dept_id);
```

```
10 END;
```

```
11 /
```

```
dept_id = 5
```

Insert API - Record Based

PROCEDURE INS2

Argument Name	Type	In/Out	Default?
-----	-----	-----	-----
ID_IO	NUMBER	IN/OUT	
ENAME_IO	VARCHAR2	IN/OUT	
JOB_IO	VARCHAR2	IN/OUT	
MGR_EMP_ID_IO	NUMBER	IN/OUT	
MGR_ID_PATH_IO	VARCHAR2	IN/OUT	
MGR_NK_PATH_IO	VARCHAR2	IN/OUT	
MGR_EMP_NK1_IO	VARCHAR2	IN/OUT	
HIREDATE_IO	DATE	IN/OUT	
SAL_IO	NUMBER	IN/OUT	
COMM_IO	NUMBER	IN/OUT	
DEPT_ID_IO	NUMBER	IN/OUT	
DEPT_NK1_IO	VARCHAR2	IN/OUT	

Insert API - Record Based

PROCEDURE INS2

Argument Name	Type	In/Out	Default?
-----	-----	-----	-----
ID_IO	NUMBER	IN/OUT	
ENAME_IO	VARCHAR2	IN/OUT	
JOB_IO	VARCHAR2	IN/OUT	
MGR_EMP_ID_IO	NUMBER	IN/OUT	
HIREDATE_IO	DATE	IN/OUT	
SAL_IO	NUMBER	IN/OUT	
COMM_IO	NUMBER	IN/OUT	
DEPT_ID_IO	NUMBER	IN/OUT	

Insert API - Record Based

```
SQL> set serveroutput on
```

```
SQL> DECLARE
```

```
2     dept_rec    dept%ROWTYPE;
```

```
3 BEGIN
```

```
4     dept_rec.dname    := 'SECURITY';
```

```
5     dept_rec.loc      := 'BOSTON';
```

```
6     DEPT_DML.ins(dept_rec);
```

```
7     dbms_output.put_line('dept_rec.id = ' ||
```

```
8                             dept_rec.id);
```

```
9 END;
```

```
10 /
```

```
dept_rec.id = 6
```

Insert API - Dimension Load

```
SQL> set serveroutput on
```

```
SQL> DECLARE
```

```
2     dept_rec    dept%ROWTYPE;
```

```
3 BEGIN
```

```
4     dept_rec.dname    := 'SECURITY';
```

```
5     dept_rec.loc      := 'OAKLAND';
```

```
6     DEPT_DML.load_dim(dept_rec);
```

```
7     dbms_output.put_line('dept_rec.id = ' ||
```

```
8                               dept_rec.id);
```

```
9 END;
```

```
10 /
```

```
dept_rec.id = 6
```

```
SQL> select * from dept where dname = 'SECURITY';
```

```
      ID DNAME          LOC
```

```
-----
```

```
      6 SECURITY          OAKLAND
```

2. DDGS Generated API

- Documentation
- Insert API
- Update API
 - Table/View SQL
 - Column Based
 - Record Based
- Delete API
- Query and Clear

Update API - Table/View SQL

```
SQL> update dept set loc = 'BOSTON'  
2 where dname = 'SECURITY';
```

```
SQL> update emp_act set mgr_emp_nk1 = 'JONES'  
2 where ename = 'DUANE';
```

```
SQL> select id, ename, mgr_emp_id, mgr_emp_nk1  
2 from emp_act where ename = 'DUANE'  
3 or ename = 'JONES';
```

ID	ENAME	MGR_EMP_ID	MGR_EMP_NK1
15	DUANE	2	JONES
2	JONES	1	KING

Update API - Column Based

```
SQL> set serveroutput on
```

```
SQL> DECLARE
```

```
2     dept_id          number;
```

```
3     dept_name        varchar2(14) := 'HR';
```

```
4     dept_location    varchar2(13) := 'SEATTLE';
```

```
5 BEGIN
```

```
6     DEPT_DML.upd2(id_io      => dept_id
```

```
7                    ,dname_io => dept_name
```

```
8                    ,loc_io   => dept_location);
```

```
9     dbms_output.put_line('dept_id = ' || dept_id);
```

```
10 END;
```

```
11 /
```

```
dept_id = 5
```

Update API - Record Based

```
SQL> DECLARE
  2      emp_rec      emp_act%ROWTYPE;
  3  BEGIN
  4      select * into emp_rec from emp_act
  5          where ename = 'LISA';
  6      emp_rec.ename := 'Leeza';
  7      emp_dml.upd(emp_rec);
  8      dbms_output.put_line('emp_rec.ename = ' ||
  9                          emp_rec.ename);
 10  END;
 11  /
emp_rec.ename = LEEZA
```

2. DDGS Generated API

- Documentation
- Insert API
- Update API
- Delete API
- Query and Clear

Delete API

```
SQL> delete from emp_act where mgr_emp_nk1 = 'MARTIN';  
1 row deleted.
```

```
SQL> execute emp_dml.del(-1);
```

```
PL/SQL procedure successfully completed.
```

```
SQL> declare
```

```
2     emp_id  emp_act.id%TYPE;
```

```
3     ename   emp_act.ename%TYPE := 'DUANE';
```

```
4 begin
```

```
5     emp_dml.del2(emp_id, ename);
```

```
6     dbms_output.put_line('emp_id = ' || emp_id);
```

```
7 end;
```

```
8 /
```

```
emp_id = 15
```

```
PL/SQL procedure successfully completed.
```


2. DDGS Generated API

- Insert API
- Update API
- Delete API
- Query and Clear

Query and Clear

```
SQL> declare
  2     emp_rec  emp_act%ROWTYPE;
  3  begin
  4     emp_rec.id := emp_dml.get_id('KING');
  5     dbms_output.put_line('emp_rec.id = ' || emp_rec.
id);
  6     emp_dml.get_rec_by_id(emp_rec);
  7     dbms_output.put_line('emp_rec.job = ' || emp_rec.
job);
  8     emp_dml.clear(emp_rec);
  9     dbms_output.put_line('emp_rec.id = ' || emp_rec.
id);
 10  end;
 11  /
```

```
emp_rec.id = 1
```

```
emp_rec.job = PRESIDENT
```

```
emp_rec.id =
```

2. DDGS Generated API

- Documentation
- Insert API
- Update API
- Delete API
- Query and Clear

<http://www.dmstex.com>