

## DDL の取得

### DBMS\_METADATA.GET\_DDL

GET\_DDL を使うと DDL を取得できる。

### DDL の出力形式を変更する

[https://siguniang.wordpress.com/2013/09/04/oracle-dbms\\_metadata-get\\_ddl-fetch-ddl-for-tables/](https://siguniang.wordpress.com/2013/09/04/oracle-dbms_metadata-get_ddl-fetch-ddl-for-tables/)

<https://www.toadworld.com/platforms/oracle/b/weblog/archive/2015/11/25>

[/get-ddl-and-set-transform-param-in-dbms-metadata](https://www.toadworld.com/platforms/oracle/b/weblog/archive/2015/11/25/get-ddl-and-set-transform-param-in-dbms-metadata)

<https://stackoverflow.com/questions/6782278>

[/generating-ddl-script-for-object-without-schema-name-baked-in-using-dbms-metadata](https://stackoverflow.com/questions/6782278/generating-ddl-script-for-object-without-schema-name-baked-in-using-dbms-metadata)

### スキーマ名を省略する

```
execute DBMS_METADATA.SET_TRANSFORM_PARAM(dbms_metadata.SESSION_TRANSFORM, 'EMIT_SCHEMA', false);
```

### storage セクションを省略する

```
execute dbms_metadata.set_transform_param( DBMS_METADATA.SESSION_TRANSFORM, 'STORAGE', false );
```

### 外部キーを alter table にする

```
execute dbms_metadata.set_transform_param( DBMS_METADATA.SESSION_TRANSFORM, 'CONSTRAINTS_AS_ALTER', true );
```

### セグメント属性を省く

```
execute dbms_metadata.set_transform_param(DBMS_METADATA.SESSION_TRANSFORM, 'SEGMENT_ATTRIBUTES', false );
```

### DDL 単位でセミコロンをつける

```
execute DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM, 'SQLTERMINATOR', true);
```

## 各オブジェクトを出力するバッチ

### getDDL.sh

```
DBCCONNECT=user/pass@host:port/service  
OWNER= 取得スキーマ  
OUT_PATH= 出力先
```

```
OBJECTS="  
TABLE  
VIEW  
SEQUENCE  
FUNCTION  
PROCEDURE  
INDEX  
DATABASE LINK  
"
```

```
WORKFILE=temp.tmp # 拡張子は必ず付けること
```

```
source setenvora.sh
```

```

LINE_SEP='
'
OLD_IFS=$IFS
IFS=$LINE_SEP

# 各オブジェクト取得
for TYPE in ${OBJECTS}
do
    sqlplus64 ${DBCONNECT} @getName.sql "${TYPE}" ${OWNER} ${WORKFILE}

    mkdir -p "${OUT_PATH}/${TYPE}/_/"
    rm -rf "${OUT_PATH}/${TYPE}/_/"/*

    cat ${WORKFILE} | while read line
    do
        sqlplus64 ${DBCONNECT} @getObject.sql "${TYPE}" ${OWNER} "${line}" "${OUT_PATH}/${TYPE}/
_}/${line}/_".sql"
    done
done

# 表領域取得
sqlplus64 ${DBCONNECT} @getTablespace.sql "${OUT_PATH}/TABLESPACE.sql"

# ユーザ
mkdir -p "${OUT_PATH}/USERS"
rm -rf "${OUT_PATH}/USERS/*"

sqlplus64 ${DBCONNECT} @getUsersName.sql ${WORKFILE}
cat ${WORKFILE} | while read line
do
    sqlplus64 ${DBCONNECT} @getUser.sql "${line}" "${OUT_PATH}/USERS/${line}/_".sql"
done
IFS=$OLD_IFS
# find ${OUT_PATH} -name "*.sql" -exec sed -i 's/ *$//g' {} ¥;
rm -f ${WORKFILE}

```

## getName.sql

```

--1 ページの行数 0 の場合はページヘッダ等を全て非表示にする
set pagesize 0
--1 行のバイト数。これを超える場合は SET WRAP の設定に従う
set linesize 32767
--1 行の幅を超えた場合に折り返す
set wrap on
-- 問合せ結果の件数を表示しない
set feedback off
--LONG、CLOB、NCLOB のチャンクサイズ
set LONGCHUNKSIZE 500
--LONG、CLOB、NCLOB の表示
set long 2000000
-- 行末のスペースを削除
SET TRIMS ON

VAR objecttype VARCHAR2(30)
exec :objecttype := '&1'
VAR owner VARCHAR2(30)
exec :owner := '&2'

spool &3
SELECT
    OBJECT_NAME
FROM
    DBA_OBJECTS
WHERE
    OWNER = :owner
    AND OBJECT_TYPE = :objecttype
ORDER BY OBJECT_NAME;
spool off;

exit;

```

## getObject.sql

```

--1 ページの行数 0 の場合はページヘッダ等を全て非表示にする
set pagesize 0
--1 行のバイト数。これを超える場合は SET WRAP の設定に従う

```

```

set linesize 32767
--1 行の幅を超えた場合に折り返す
set wrap on
-- 問合せ結果の件数を表示しない
set feedback off
--LONG、CLOB、NCLOB のチャンクサイズ
set LONGCHUNKSIZE 2048
--LONG、CLOB、NCLOB の表示
set long 20000000
-- 行末のスペースを削除
SET TRIMS ON

VAR objecttype VARCHAR2(30)
exec :objecttype := '&1'
VAR owner VARCHAR2(30)
exec :owner := '&2'
VAR objectname VARCHAR2(50)
exec :objectname := '&3'

-- 最後にセミコロンを付ける
EXECUTE DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM, 'SQLTERMINATOR', TRUE);
-- STORAGE 属性を省略
EXECUTE DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM, 'STORAGE', FALSE );
-- セグメント属性を省略
EXECUTE DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM, 'SEGMENT_ATTRIBUTES',
FALSE );
-- キー情報は出力しない
-- EXECUTE DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM, 'CONSTRAINTS', FALSE
);
-- キー情報は alter table にする
EXECUTE DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM, 'CONSTRAINTS_AS_ALTER',
TRUE );
-- スキーマ名を省略
EXECUTE DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM, 'EMIT_SCHEMA', FALSE);

spool &4
SELECT
  DBMS_METADATA.GET_DDL(
    CASE
      WHEN OBJECT_TYPE = 'DATABASE LINK' THEN 'DB_LINK'
      ELSE OBJECT_TYPE
    END,
    OBJECT_NAME, OWNER) DDL
FROM
  DBA_OBJECTS
WHERE
  OWNER = :owner
  AND OBJECT_TYPE = :objecttype
  AND OBJECT_NAME = :objectname;
spool off;

exit;

```

## getTablespace.sql

```

--1 ページの行数 0 の場合はページヘッダ等を全て非表示にする
set pagesize 0
--1 行のバイト数。これを超える場合は SET WRAP の設定に従う
set linesize 32767
--1 行の幅を超えた場合に折り返す
set wrap on
-- 問合せ結果の件数を表示しない
set feedback off
--LONG、CLOB、NCLOB のチャンクサイズ
set LONGCHUNKSIZE 2048
--LONG、CLOB、NCLOB の表示
set long 20000
-- 行末のスペースを削除
SET TRIMS ON

spool &1
SELECT DBMS_METADATA.GET_DDL('TABLESPACE', TABLESPACE_NAME) AS DDL FROM DBA_TABLESPACES
ORDER BY TABLESPACE_NAME;

spool off;

exit;

```

## getUser.sql

```
--1 ページの行数 0 の場合はページヘッダ等を全て非表示にする
set pagesize 0
--1 行のバイト数。これを超える場合は SET WRAP の設定に従う
set linesize 32767
--1 行の幅を超えた場合に折り返す
set wrap on
-- 問合せ結果の件数を表示しない
set feedback off
--LONG、CLOB、NCLOB のチャンクサイズ
set LONGCHUNKSIZE 2048
--LONG、CLOB、NCLOB の表示
set long 20000
-- 行末のスペースを削除
SET TRIMS ON

VAR objectname VARCHAR2(50)
exec :objectname := '&1'

spool &2
SELECT DBMS_METADATA.GET_DDL('USER', USERNAME) AS DDL FROM dba_users
where USERNAME = :objectname
ORDER BY USERNAME;

select 'GRANT ' || PRIVILEGE || ' TO ' || GRANTEE || case when ADMIN_OPTION = 'YES' then ' WITH
ADMIN OPTION' end case
from dba_sys_privs
where GRANTEE = :objectname
order by GRANTEE;

select 'GRANT ' || GRANTED_ROLE || ' TO ' || GRANTEE || case when ADMIN_OPTION = 'YES' then ' WITH
ADMIN OPTION' end case
from dba_role_privs
where GRANTEE = :objectname
order by GRANTEE;

spool off;

exit;
```

## getUsersName.sql

```
--1 ページの行数 0 の場合はページヘッダ等を全て非表示にする
set pagesize 0
--1 行のバイト数。これを超える場合は SET WRAP の設定に従う
set linesize 32767
--1 行の幅を超えた場合に折り返す
set wrap on
-- 問合せ結果の件数を表示しない
set feedback off
--LONG、CLOB、NCLOB のチャンクサイズ
set LONGCHUNKSIZE 2048
--LONG、CLOB、NCLOB の表示
set long 20000
-- 行末のスペースを削除
SET TRIMS ON

spool &1
SELECT USERNAME FROM dba_users
ORDER BY USERNAME;
spool off;

exit;
```

## setenvora.sh

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/:${LD_LIBRARY_PATH}
#export NLS_LANG=JAPANESE_JAPAN.JA16SJIS
export NLS_LANG=Japanese_Japan.UTF8
export ORACLE_SID=XE
```

## 以下参考

### 各オブジェクト

```
set pagesize 0
set linesize 200
set feedback off

VAR objecttype VARCHAR2(30)
exec :objecttype := 'VIEW' --TABLE とか取得したいオブジェクトのタイプを指定する

SELECT
  DBMS_METADATA.GET_DDL(OBJECT_TYPE, OBJECT_NAME, OWNER) DDL
FROM
  ALL_OBJECTS
WHERE
  OWNER = :owner
  AND OBJECT_TYPE = :objecttype
  --AND OBJECT_NAME = :objectname;
```

### 表領域

```
SELECT DBMS_METADATA.GET_DDL('TABLESPACE', TABLESPACE_NAME) AS DDL FROM DBA_TABLESPACES
ORDER BY TABLESPACE_NAME;
```

### ユーザ情報

```
SELECT DBMS_METADATA.GET_DDL('USER', USERNAME) AS DDL FROM dba_users
ORDER BY USERNAME;
```

### ユーザの権限

```
select * from dba_sys_privs
select * from dba_role_privs
```

以下のような感じで権限付与の SQL を作成できる

```
select 'GRANT ' || PRIVILEGE || ' TO ' || GRANTEE || case when ADMIN_OPTION = 'YES' then ' WITH
ADMIN OPTION' end case
from dba_sys_privs
where grantee = 'hoge'

select 'GRANT ' || GRANTED_ROLE || ' TO ' || GRANTEE || case when ADMIN_OPTION = 'YES' then ' WITH
ADMIN OPTION' end case
from dba_role_privs
where grantee = 'hoge'
```

## 以下、古い情報

### テーブル

```
set pagesize 0
set linesize 200
set feedback off

VAR owner VARCHAR2(10)
exec :owner := 'USER'

-- テーブル
spool src/TABLE.sql
SELECT
  TO_CHAR(DBMS_METADATA.GET_DDL('TABLE', TABLE_NAME, OWNER)) DDL
FROM
```

```
ALL_TABLES
WHERE
  OWNER = :owner
ORDER BY TABLE_NAME;
spool off;

exit;
```

## インデックス

```
set pagesize 0
set linesize 200
set feedback off

VAR owner VARCHAR2(10)
exec :owner := 'USER'

-- インデックス
spool_src/INDEX.sql
SELECT
  TO_CHAR(DBMS_METADATA.GET_DDL('INDEX', INDEX_NAME, OWNER)) DDL
FROM
  ALL_INDEXES
WHERE
  OWNER = :owner
ORDER BY INDEX_NAME;
spool off;

exit;
```

## ビュー

```
set pagesize 0
set linesize 200
set feedback off

VAR owner VARCHAR2(10)
exec :owner := 'USER'

-- ビュー
spool_src/VIEW.sql
SELECT
  TO_CHAR(DBMS_METADATA.GET_DDL('VIEW', VIEW_NAME, OWNER)) DDL
FROM
  ALL_VIEWS
WHERE
  OWNER = :owner
ORDER BY VIEW_NAME;
spool off;

exit;
```

## シーケンス

```
set pagesize 0
set linesize 200
set feedback off

VAR owner VARCHAR2(10)
exec :owner := 'USER'

-- シーケンス
spool_src/SEQUENCE.sql
SELECT
  TO_CHAR(DBMS_METADATA.GET_DDL('SEQUENCE', SEQUENCE_NAME, SEQUENCE_OWNER)) DDL
FROM
  ALL_SEQUENCES
WHERE
  SEQUENCE_OWNER = :owner
ORDER BY SEQUENCE_NAME;
spool off;

exit;
```

## DB LINK

```
set pagesize 0
set linesize 200
set feedback off

VAR owner    VARCHAR2(10)
exec :owner := 'USER'

-- DB LINK
spool src/DB_LINK.sql
SELECT
  TO_CHAR(DBMS_METADATA.GET_DDL('DB_LINK', DB_LINK, OWNER)) DDL
FROM
  ALL_DB_LINKS
WHERE
  OWNER = :owner
ORDER BY DB_LINK;
spool off;

exit;
```

## PROCEDURE

```
set pagesize 0
set linesize 200
set feedback off
set LONGCHUNKSIZE 200
set long 20000

VAR owner    VARCHAR2(10)
exec :owner := 'USER'

-- Procedure
spool src/PROCEDURE.sql
SELECT
  DBMS_METADATA.GET_DDL('PROCEDURE', OBJECT_NAME, OWNER) DDL
FROM
  ALL PROCEDURES
WHERE
  OWNER = :owner
  and OBJECT_TYPE='PROCEDURE'
ORDER BY PROCEDURE_NAME;
spool off;

exit;
```

## FUNCTION

```
set pagesize 0
set linesize 200
set feedback off
set LONGCHUNKSIZE 200
set long 20000

VAR owner    VARCHAR2(10)
exec :owner := 'USER'

-- Function
spool src/FUNCTION.sql
SELECT
  DBMS_METADATA.GET_DDL('FUNCTION', OBJECT_NAME, OWNER) DDL
FROM
  ALL PROCEDURES
WHERE
  OWNER = :owner
  and OBJECT_TYPE='FUNCTION'
ORDER BY PROCEDURE_NAME;
spool off;

exit;
```

## 参考

[http://oracle.se-free.com/ddl/A1\\_get\\_ddl.html](http://oracle.se-free.com/ddl/A1_get_ddl.html)

DBMS\_METADATA.GET\_DDL( オブジェクトタイプ , オブジェクト名 , [ スキーマ ] )

SQL> set long 2000

SQL> set heading off

SQL>

SQL> select

```
2      dbms_metadata.get_ddl('TABLE','EMP')
3      from dual;
```

CREATE TABLE "TEST"."EMP"

( "EMP\_ID" CHAR(5),

"DEPT\_ID" CHAR(5),

"EMP\_NAME" VARCHAR2(10),

"EMP\_DATE" DATE,

"SALARY" NUMBER(5,0),

PRIMARY KEY ("EMP\_ID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ENABLE

) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS"