TABLESPACE

TABLE

TABLESPACE

TABLE

TABLESPACE

TABLE

USER

USER

USER

USER

USER

USER

USER

USER

USER

USER

USER

USER

SID

SID

SID

SID

LOCAL

DB

DB

DB

DB

DBMS

|  |
| --- |
| **[오라클] 데이터 저장구조, 테이블스페이스 or 테이블생성**http://blogimgs.naver.net/imgs/nblog/spc.gif[SQL 함수](http://blog.naver.com/PostList.nhn?blogId=oasis1570&categoryNo=72&from=postList)/ [오라클](http://blog.naver.com/PostList.nhn?blogId=oasis1570&categoryNo=74&parentCategoryNo=74&from=postList)http://blogimgs.naver.net/imgs/nblog/spc.gif  2011/05/24 17:47  복사<http://blog.naver.com/oasis1570/40130041702> |

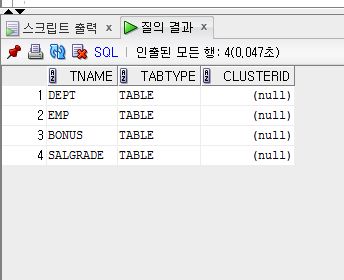
**<EMP 라는 테이블이 저장된 파일명을 조회해 보자>**

**-테이블 = 실체 (Entity)  
-컬럼(Colum) = Field = 열  
-ROW(로우) = Record = 행**

**SELECT USER FROM DUAL;**

**SHOW USER;**

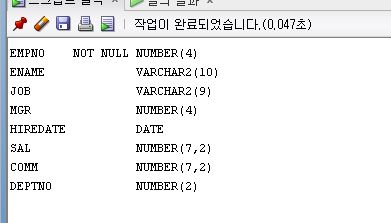
**SELECT \* FROM TAB; -현재 오라클 서버에 접속한 사용자 소유의 테이블과 뷰명을 조회해 해준다.**

****

**SELECT \* FROM EMP; -emp테이블 조회**

****

**DESCTIBE EMP;  
DESC EMP; --emp테이블의 구조정보**

****

**================================================================================================================**

**\*\*오라클의 논리적인 데이터저장구조\*\***

**1 .BLOCK : 오라클에서 말하는 데이터 입,출력의 최소단위**

**BLOCK 크기종류 : 2KB,4KB,8KB,16KB,32KB**

**2. EXTENT : SEGMENT의 할당단위 이다.  
1 EXTENT 는 8 BLOCK 이다.**

**3. SEGMENT : 테이블(TABLE),인덱스(INDEX)를 합쳐서 SEGMENT 라고 부른다.**

**4. TABLESPACE : SEGMENT 를 저장해 놓은 공간을 TABLESPACE 라고 부른다.   
즉, 운영체제의 물리적인 파일 명에 대한 오라클의 논리적인 이름을 TABLESPACE 라고 명칭한다.**

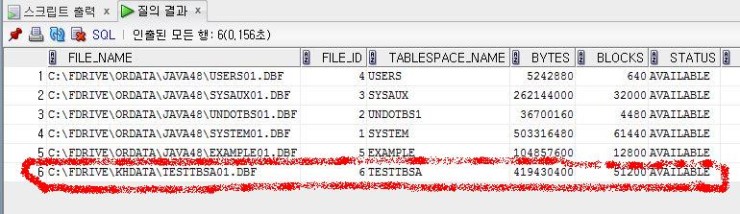
**5. DATABASE : 여러개의 TABLESPACE 를 모아놓은 것을 DATABASE 라고 부른다.**

**<테이블 스페이스 생성하기>**

**CREATE TABLESPACE TESTTBSA --TESTTBSA라는 테이블 스페이스 생성  
DATAFILE 'C:\Fdrive\khdata\TESTTBSA01.DBF' --실제 데이터가 저장되는 테이블스페이스파일의 경로**

**SIZE 400M --테이블스페이스의 사이즈 (명시하지 않으면 기본8블럭으로 생성됨)  
EXTENT MANAGEMENT LOCAL  
SEGMENT SPACE**

**SELECT \*  
FROM DBA\_DATA\_FILES; --생성된 데이터파일 확인**

****

**400MB의 TESTTBSA01 생성**

**SELECT \*  
FROM DBA\_TABLESPACES; --생성된 테이블스페이스 확인**

****

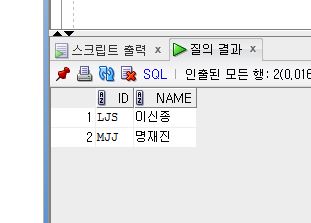
**<테이블만들기>**

**CREATE TABLE TBL\_MEMBER --TBL\_MEMBER 이라는 테이블생성  
(ID VARCHAR2(10) --두개의 컬럼생성  
,NAME VARCHAR2(20)  
) TABLESPACE TESTTBSA; --저장되어질 테이블스페이스(이전에 이미 생성한 TESTTBSA 테이블스페이스)**

**INSERT INTO TBL\_MEMBER VALUES('LJS','이신종'); --데이터 입력  
INSERT INTO TBL\_MEMBER VALUES('MJJ','명재진'); --실제데이터스페이스에 입력된것은 아니고 메모리상에만 올라간것**

**SELECT \* --생성한 TBL\_MEMBER테이블 조회  
FROM TBL\_MEMBER;**

**﻿**

****

**COMMIT; --메모리상에 올라온 데이터를 실제 하드디스크에 저장(TESTTBSA 테이블스페이스**

**의 'C:\Fdrive\khdata\TESTTBSA01.DBF' 에 저장 )**

**이명령이 실행되지 않으면 메모리상에 올라온 데이터는 데이터스페이스에 저장**

**되어지지않는다.**

**SELECT \*   
FROM USER\_TABLES   
WHERE TABLE\_NAME = 'TBL\_MEMBER';** -- 테이블네임과 테이블 스페이스 확인

## [테이블스페이스 생성, 계정 생성 및 권한부여](http://sudili.tistory.com/600)

[Oracle](http://sudili.tistory.com/category/Oracle) 2011/11/30 09:30

|  |  |
| --- | --- |
|  |  |

-- 테이블스페이스 확인  
SELECT \* FROM DBA\_DATA\_FILES;  
  
﻿﻿-- 테이블스페이스 생성

CREATE TABLESPACE TESTDB -- 테이블스페이스 명을 TESTDB로 지정

DATAFILE 'D:\oradata\TESTDB.DBF' -- ﻿실제 데이터가 저장되는 테이블스페이스 파일 지정

SIZE 20480M -- 테이블스페이스 사이즈(Default 8블럭)

DEFAULT STORAGE

(INITIAL 10K -- 테이블스페이스의 맨 첫번째 Extents의 크기

NEXT 10K -- 다음 Extents의 크기

MINEXTENTS 2 -- 생성할 Extents의 최소값

MAXEXTENTS 50 -- 생성할 Extents의 최대값

PCTINCREASE 50) -- Extents의 증가율(Default 50)

;

-- 계정생성

CREATE USER testuser IDENTIFIED BY testuser -- ID/PW 지정

DEFAULT TABLESPACE TESTDB -- 기본 테이블스페이스 지정

TEMPORARY TABLESPACE TEMP01 -- 임시 테이블스페이스 지정(생략가능)  
;

\* 임시 테이블스페이스를 지정해 주지 않으면 시스템 테이블스페이스가 기본으로 지정되지만,  
시스템 테이블스페이스에 단편화가 발생할 수 있으므로 사용자를 생성할 때   
임시 테이블스페이스를 따로 지정해 주는 것이 좋다.

-- 권한부여

GRANT connect, resource TO testuser;

Oracle의 백업과 복구와 관련되서 많은 tool들이 나왔지만 Oracle에서 기본적으로 지원되는 exp, imp 명령어를 통해서도 백업과 복구의 수행이 가능하다.

이번 포스트는 exp, imp 명령어의 사용예를 몇가지 정리해 보았다.

**1. 백업**

- full backup

statistics 옵션은 보통 주지 않아도 되지만.. 이럴경우 questionable한 statistic을 export 중이라는 warning이 뜨는 경우가 많으며, DB에서 자동으로 업데이트 되는 것으로 크게 상관이 없으므로 none으로 하는게 속 편하다

exp dbid/dbpasswd@tnsname file=filename.dmp log=filename.log full=y statistics=none

- table만 backup

DB에 data량이 많은 경우 table shema와 data를 분리해서 백업해야 할 경우가 있다. 그럴경우에 유용하다.

exp dbid/dbpasswd@tnsname owner=dbid file=filename.dmp log=filename.log **﻿**indexes=n constraints=n triggers=n rows=n compress=n statistics=none

- data를 제외한 table, index, trigger등을 backup

exp dbid/dbpasswd@tnsname owner=dbid file=filename.dmp log=filename.log rows=n compress=n statistics=none

- data만을 backup

exp dbid/dbpasswd@tnsname owner=dbid file=filename.dmp log=filename.log statistics=none indexes=n constraints=n triggers=n buffer=100000000 compress=n grants=n

- 특정 table만을 export

특정 몇개 table만을 export 할 경우에는 아래와 같이 tables 옵션을 사용하면 된다.

exp dbid/dbpasswd@tnsname file=filename.dmp log=filename.log tables=(table1, table2)

**2. 복구**

- full restore

백업을 할 때 계획적으로 잘 하였다면 일반적으로 백업 파일을 full restore 하면 된다. 즉 exp를 table schema 만을 골라서 했다면 full로 imp 해도 table schema만 복구되므로 백업할 때부터 계획을 세우고 하는 것이 편하다. ignore 옵션은 import를 하는 중간에 error가 발생해도 나머지 작업을 계속 진행하는 옵션이다. 이미 존재하는 개체로 인한 에러가 발생할 때 유용한 옵션이므로 복구시에는 기본적으로 달아주는 것이 좋다. 그렇지 않으면 import를 수행하기 전에 import 될 table, index, trigger 등을 미리 drop 해야한다.

imp dbid/dbpasswd@tnsname file=filename.dmp log=filename.log full=y ignore=y

- data만 복구

복구의 경우 기본적으로 백업할 때의 옵션을 사용하면 되며 ignore=y만 붙여주면 된다.

imp dbid/dbpasswd@tnsname file=filename.dmp log=filename.log grants=n ignore=y buffer=100000000 rows=y indexes=n constraints=n

**[출처]** [[오라클] 데이터 저장구조, 테이블스페이스 or 테이블생성](http://blog.naver.com/oasis1570/40130041702)|**작성자** [Yellow9](http://blog.naver.com/oasis1570)