

```

1 DATA DICTIONARY
2 -테이블의 종류는 2가지
3 1. Data Dictionary
4 -데이터베이스를 생성할 때 기본적으로 만들어지는 자료사전테이블
5 -데이터베이스 내의 모든 상태 정보를 제공해 주는 테이블
6 -사용자가 생성한 테이블이 어떤 것이 있는지, 어떤 구조로 생성되어 있는지, 데이터베이스의 성능이 어떤지 등을
  보여주는 테이블
7
8 2. 일반적인 테이블 - 사용자의 데이터를 저장하고 관리하기 위해서 생성하는 테이블
9
10 3. 모든 자료사전 정보출력하기
11 SELECT * FROM DICTIONARY;
12
13 4. 종류
14 a. ALL XXXXX 와 USER XXXXX
15 -주로 개발자들이 자신이 만든 테이블 또는 객체들에 대한 정보 수집 및 분석시 사용하는 자료사전
16 -일반 사용자로 접속해서 조회 가능
17 b. DBA XXXXX
18 -데이터베이스 관리자가 데이터베이스의 관리를 목적으로 참조하는 테이블
19 -SYS 또는 SYSTEM 사용자로 데이터베이스로 접속한 후 참조가능
20 c. X$ XXXXX
21 -데이터베이스의 성능과 튜닝 시에 주로 사용하는 테이블이지만 일반 사용자가 DBA도 직접적으로 참조할 수 없다.
22 -V$ XXXXX 성능 동적 뷰를 통해서만 참조가능
23 5. 자료사전 테이블의 각 컬럼에 대한 설명을 출력
24
25 SELECT * FROM DICT_COLUMNS;
26
27 6. 예제
28 a. 사용자가 생성한 모든 테이블을 출력
29 SELECT * FROM user_tables;
30 b. 사용자가 생성한 모든 인덱스를 출력
31 SELECT * FROM user_indexes;
32 c. 사용자가 생성한 모든 뷰를 출력
33 SELECT * FROM user_views;
34 d. 사용자가 생성한 모든 제약조건을 출력
35 SELECT * FROM user_constraints;
36
37 USER 데이터 덱서너리
38 a. 현재 접속한 사용자의 계정이 소유한 모든 테이블 정보 조회
39 DESC USER TABLES;
40 b. 현재 사용자 조회
41 SHOW USER; <--- sqlplus 로 확인해야 함.
42 c. 현재 사용자 계정인 scott 이 사용할 수 있는 테이블의 이름 조회
43 SELECT TABLE_NAME FROM USER_TABLES ORDER BY TABLE_NAME DESC;
44
45 ALL 데이터 덱서너리
46 a. 타 계정의 객체 조회가능
47 DESC ALL TABLES;
48 b. SELECT OWNER, TABLE NAME FROM ALL TABLES;
49
50 DBA 데이터 덱서너리
51 SELECT TABLE_NAME, OWNER FROM DBA_TABLES;<---error, scott 는 DBA가 아니어서 권한문제 발생
52 CONN SYSTEM/MANAGER
53 SELECT TABLE_NAME, OWNER FROM DBA_TABLES;<---sqlplus 로 접속해야 함.
54
55 -----
56 DDL(Data Definition Language)
57 -데이터베이스의 Object 구조를 생성, 변경, 삭제하는 명령어
58 -CREATE, ALTER, DROP, RENAME, COMMENT, TRUNCATE
59
60 REM 테이블의 생성
61 --1. Naming Convension
62 --1). 각 이름은 문자로 시작해야 하며, 1 ~ 30자리까지 사용할 수 있다.
63 --2). 이름은 A ~ Z, a ~ z, 0 ~ 9, (밑줄), $, # (적합한 문자이지만, 가능하면 사용하지 말것) 등을 사용할 수
  있다.
64 --3). 이름은 동일한 아이디(소유자)의 다른 객체와 같지 않아야 한다.
65 --4). 대소문자를 구별하지 않는다.
66 --5). 한글버전에서만 한글을 사용할 수 있다.
67 --6). 예약어는 사용 불가능
68 --7) 공백허용 안함
69
70 --2. CREATE TABLE
71 -----테이블을 생성할 때
72 -----Syntax
73 ----- CREATE TABLE table name
74 ----- (
75 ----- column_name data_type [DEFAULT expression] [,,,]
76 ----- )
77
78 a. 일반적 생성 방법
79 CREATE TABLE table_name ( column_name datatype, ...);
80 CREATE TABLE dept1
81 (deptno NUMBER(2),
82  dname VARCHAR2(13),
83  loc VARCHAR2(14)
84 );
85
86 b. SUB_QUERY 에 의해 검색된 테이블과 동일한 구조로 생성 --> 테이블 복사
87 CREATE TABLE dept2
88 AS
89 SELECT * FROM dept;
90
91 CREATE TABLE emp2
92 AS
93 SELECT EMPNO, ENAME FROM EMP;

```

```

94
95 CREATE TABLE emp3
96 AS
97 SELECT * FROM emp
98 WHERE DEPTNO = 10;
99
100 CREATE TABLE emp4
101 AS
102 SELECT * FROM emp where 1=0; <-- 구조만 복사
103
104 --Guide Lines
105 --지정된 열 이름을 사용하여 테이블을 생성하며 SELECT 문에 의해 검색된 행을 테이블에 삽입.
106 --열 사양이 제공되는 경우 열 수는 하위 질의 SELECT 목록의 열 수와 동일해야 한다.
107 --열 사양이 제공되지 않는 경우 해당 테이블의 열 이름은 하위 질의의 열 이름과 동일.
108
109 c. 조회
110 --SELECT table_name
111 --FROM user_tables;
112
113 --SELECT object name, object type
114 --FROM user_objects;
115
116 --CREATE TABLE dept30
117 --AS
118 --SELECT empno, ename, sal * 12 "Annual Salary", hiredate
119 --FROM emp
120 --WHERE deptno = 30;
121
122 --부서별로 인원수, 평균급여, 급여의 합, 최소급여, 최대급여를 포함하는 emp_calcu table을 생성하라.
123 --CREATE TABLE emp_calcu
124 --(deptno, tcount, tavg, tsum, tmin, tmax)
125 --AS
126 --SELECT deptno, COUNT(*), AVG(sal), SUM(sal), MIN(sal), MAX(sal)
127 --FROM emp
128 --GROUP BY deptno;
129
130 --사원번호, 이름, 업무, 입사일자, 부서번호만 포함하는 emp_temp table을 생성하는데, 자료는 포함하지 않고
131 --스키마만 생성하시오.
132 --CREATE TABLE emp temp
133 --(사원번호, 이름, 업무, 입사일자, 부서번호)
134 --AS
135 --SELECT empno, ename, job, hiredate, deptno
136 --FROM emp
137 --WHERE 1 < 0;
138
139 --1. 테이블을 다음차트를 기반으로 DEPARTMENT 테이블을 생성하고, 생성한 후 테이블 구조를 확인하시오.
140 --열이름 : id name
141 --데이터유형 : NUMBER VARCHAR2
142 --길이 : 7 25
143 -- CREATE TABLE department
144 -- (id NUMBER(7),
145 -- name VARCHAR2(25));
146 --DESC department
147
148 --2. 테이블을 다음차트를 기반으로 EMPLOY 테이블을 생성하고, 생성한 후 테이블 구조를 확인하시오.
149 --열이름 : id last_name first_name dept_id
150 --데이터유형 : NUMBER VARCHAR2 VARCHAR2 NUMBER
151 --길이 : 7 25 25 7
152 -- CREATE TABLE employ
153 -- (id NUMBER(7),
154 -- last_name VARCHAR2(25),
155 -- first_name VARCHAR(25),
156 -- dept_id NUMBER(7)
157 --);
158 --DESC employ
159
160 REM ALTER TABLE
161 --테이블의 스키마 변경
162
163 REM column add
164 --열 추가
165 --새 열은 마지막 열, 위치 지정 불가능
166 --이미 행을 포함하고 있는 테이블이라면 새로 들어갈 열의 모든 행은 초기에 널 값을 갖는다.
167 --ALTER TABLE table_name
168 --ADD column datatype [DEFAULT expr]
169 --[, column datatype]...);
170
171 --ALTER TABLE dept30
172 --ADD (job VARCHAR2(9));
173
174 ALTER TABLE dept1
175 ADD
176 (bigo VARCHAR2(15)
177 );
178
179 --BONUS table에 etc column을 추가하라. 단 자료형은 VARCHAR2(50)을 사용하라.
180 --ALTER TABLE bonus
181 --ADD (etc VARCHAR2(50));
182
183 REM column modify
184 --ALTER TABLE table_name
185 --MODIFY (column datatype [DEFAULT expr]
186 --[, column datatype]...);
187

```

```

188 --MODIFY Guide Lines
189 -- 숫자 열의 너비 또는 전체 자릿수를 증가가능.
190 -- 열이 널 값만 포함하고 테이블에 행이 없는 경우 열의 너비를 줄일 수 있다.
191 -- 열이 널 값을 포함하면 데이터 유형을 변경할 수 있다.
192 -- 열의 기본값을 변경하면 변경 이후에 테이블에 삽입되는 항목에만 영향을 준다.
193 테이블의 구조 변경(데이터 타입, 길이), 만일 기존 데이터가 있을 경우에는 CHAR와 VARCHAR2 사이의 타입 변경만
가능
194 컬럼의 크기 변경 역시 기존에 저장된 데이터의 길이와 같거나 클 경우에만 변경이 가능
195 즉 해당 컬럼에 자료가 없을 경우에는 데이터타입과 컬럼의 크기를 변경가능
196 해당 컬럼에 자료가 있을 경우에는 데이터타입 변경 불가능하고 크기는 늘릴 수만 있음.
197 --ALTER TABLE dept30
198 --MODIFY (ename VARCHAR2(15));
199
200 ALTER TABLE dept1
201 MODIFY
202 (bigo VARCHAR2(30)
203 );
204
205
206 --1. 위 예제에서 생성한 EMPLOY테이블의 last name열에 긴 성을 가진 사원의성을 저장할 수 있도록 길이를 50으로
수정한 후, 내용을 확인하시오.
207 --2. DEPARTMENT 및 EMPLOYEE 테이블 모두 Data Dictionary에 저장되었는지 확인하시오.
208
209 REM column remove
210 --열 삭제
211 --Guide Lines
212 -- 열은 데이터를 포함하거나 포함하지 않거나 삭제 할 수 있다.
213 -- 한 번에 하나의 열만 삭제할 수 있다.
214 -- 테이블에는 최소 하나의 열이 있어야 하므로 열을 삭제하려면 테이블에 열이 둘 이상 있어야 한다.
215 -- 삭제된 열은 복구할 수 없다.
216
217 ALTER TABLE emp1
218 DROP COLUMN JOB;
219
220 REM TABLE 삭제
221 --table의 모든 구조와 데이터가 삭제
222 --DDL이기 때문에 TRANSACTION 이 COMMIT된다.
223 --영구히 삭제, 되돌릴 수 없다.
224 --DROP TABLE table name [CASCADE CONSTRAINT];
225 --DROP TABLE emp30;
226 --DROP TABLE dept;
227 -삭제하려는 테이블의 기본 키나 고유 키를 다른 테이블에서 참조해서 사용하는 경우에는 해당 테이블을 제거할 수
없다.
228 -이런 경우에는 참조하는 테이블을 먼저 제거한 후 해당 테이블을 삭제해야 한다.
229
230
231 REM TABLE TRUNCATE
232 --모든 행을 제거하고 해당 테이블이 사용하는 저장공간을 해제
233 --ROLLBACK 불가능
234 --DELETE는 모든 행을 제거할 수 있지만, 저장공간을 해제할 수 없다.
235 --모든 행을 제거할 때는 DELETE보다 TRUNCATE TABLE문을 사용하자.
236 TRUNCATE table_name
237 --TRUNCATE TABLE emp30;
238 -DROP TABLE은 테이블 자체를 제거하지만, TRUNCATE는 테이블은 존재하면서 데이터만 제거하기에 구조는 남아있다.
239
240 7. 이름변경
241 --RENAME old table name TO new table name
242 RENAME dept2 TO dept3;
243
244 REM 테이블에 주석 문 추가
245 --COMMENT 문을 사용하여 열, 테이블, 뷰에 2000 BYTES까지 주석을 추가가능
246 --Data Dictionary View (ALL_COL_COMMENTS, USER_COL_COMMENTS, ALL_TAB_COMMENTS, USER_TAB_COMMENTS)를 통하여
확인가능
247 --COMMENT ON TABLE table_name | COLUMN table.column IS 'text';
248 COMMENT ON TABLE dept1 IS '부서 정보 테이블입니다';
249 SELECT * FROM user_tab_comments WHERE table_name='DEPT1';
250 COMMENT ON COLUMN dept1.deptno IS '부서코드';
251 SELECT * FROM user_col_comments WHERE table_name = 'DEPT1';
252
253
254 REM DATA TYPE
255 1. CHAR : 고정길이의 문자 데이터. 기본값 및 최소크기 : 1, 최대크기 2000. 나머지 공간을 여백으로 채워서 처음
정의된 공간을 모두 사용하는 타입
256 NCHAR 는 다양한 언어의 문자값을 모두 저장 조회 가능, 한글/영어이외 모두
257 2. VARCHAR2 : 가변길이의 문자 데이터, 최대크기를 지정해야 한다. 기본값 및 최소 크기 : 1, 최대크기 : 4000.
여백으로 채우지 않고 필요한 공간만 사용
258 3. NUMBER : 숫자값을 -38자리 ~ 38자리 저장
259 4. NUMBER(p, s) : 전체자리수(precision) 중 소수점 이하 자리수(scale), 전체자리수는 십진수의 총 갯수이고,
소수점 이하 자릿수는 소수점의 오른쪽에 있는 숫자 갯수.
260 전체자리수의 범위는 1에서 38까지이고, 소수점 이하 자릿수의 범위는 -84에서 127까지.
261 ABCDE --> char(8) --> ABCDE
262 ABCDE --> varchar(8) --> ABCDE
263 12345 --> NUMBER(8) --> 12345
264 123456.12 --> NUMBER(8,2) --> 123456.12
265 2002-Feb-14 --> DATE --> 2002-Feb-14
266 5. DATE : B.C. 4712년 1월 1일 ~ A.D. 9999년 12월 31일까지의 날짜 및 시간 값.
267 SELECT SYSDATE FROM dual;
268 -현재 세션의 날짜 타입 변경
269 ALTER SESSION SET NLS DATE FORMAT = 'DD-MON-YYYY'; <---sqlplus에서 수행할 것
270 6. LONG : 가변길이. 최대 2Gbyte
271 7. CLOB : 단일 바이트 단위의 가변길이. 최대 4Gbyte.
272 8. RAW : 원시 이진 데이터(1 ~ 2000)
273 9. LONG RAW : (1 ~ 2Gbyte)
274 10. BLOB : 가변 이진 데이터(1 ~ 4Gbyte)
275
276

```