

```

1  REM Oracle Objects
2  --TABLE, VIEW, SEQUENCE, INDEX, SYNONYM,...
3  --Dictionary --> USER XXXs
4  --USER TABLES, USER_CONSTRAINTS, USER_VIEWS, USER_TAB_COMMENTS, USER_COL_COMMENTS
5  --DESC USER OBJECTS;
6  --SELECT object_name, object_type, created, last_ddl_time
7  --FROM USER_OBJECTS ORDER BY object_type DESC;
8
9  REM View
10 --1. 테이블 뷰를 통한 데이터의 논리적 부분 집합 또는 조합
11 --2. 논리 테이블
12 --3. 자체적으로 데이터를 갖고 있지 않다.
13 --4. 데이터를 보거나 변경할 수 있는 창이다.
14 --5. 뷰의 기반이 되는 테이블을 기본 테이블이라 한다.
15 --6. Data Dictionary에 SELECT문으로 저장
16 --7. 공간을 차지하지도 않는다.
17
18 REM View 의 목적
19 --1. 데이터베이스의 선택적인 내용을 보여줄 수 있기 때문에 데이터베이스에 대한 액세스를 제한한다. --> 보안에
20 도움이 된다.
21 --2. 복잡한 질의어를 통해 얻을 수 있는 결과를 간단한 질의어를 써서 구할 수 있게 한다. --> 성능향상
22 --3. 데이터 독립성을 허용한다.
23 --4. 동일한 데이터의 다른 VIEW를 나타낸다.
24 --5. 조인을 한 것처럼 여러 테이블에 대한 데이터를 VIEW를 통해 볼 수 있다.
25 --6. 한개의 VIEW로 여러 테이블에 대한 데이터를 검색할 수 있다.
26
27 REM View 종류
28 --1. 단순뷰(Simple View)
29 --1) 오직 하나의 테이블에서만 데이터를 가져온다.
30 --2) 그룹 이나 다중행 함수를 포함하지 않는다.
31 --3) 뷰를 이용해서 DML 을 수행할 수 있다.
32 --4) DISTINCT 사용 불가능.
33 --2. 복합뷰(Complex View)
34 --1) 다중 테이블에서 데이터를 가져온다.
35 --2) 그룹, 다중행 함수를 포함한다.
36 --3) DML 문장을 수행할 수 없다.
37 --4) DISTINCT 사용 가능.
38
39 REM View 생성 권한
40 --기본적으로 SCOTT 는 CREATE VIEW 권한이 없다.
41 --CREATE VIEW 권한이 필요하다.
42 -- conn SYS as SYSDBA
43 -- GRANT CREATE VIEW TO scott;
44 -- conn scott/tiger;
45
46 REM View Syntax
47 --CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW view_name(alias,...)
48 --AS
49 --Subquery
50 --[WITH CHECK OPTION [CONSTRAINT constraint]]
51 --[WITH READ ONLY]
52
53 --OR REPLACE : 기존에 존재하는 뷰가 있다면 삭제하고 새로 만든다.
54 --FORCE : 기본 테이블의 존재여부에 상관없이 뷰를 생성한다.
55 --NO FORCE : 기본 테이블이 존재 할 때만 뷰 생성. 기본값
56 --WITH READ ONLY : 읽기 전용으로 생성
57 --WITH CHECK OPTION : 서브쿼리 내의 조건을 만족하는 행만 변경 가능
58 --CONSTRAINT constraint : 제약 조건을 추가한다.
59
60 REM VIEW Guide Lines
61 --1. 뷰를 정의하는 하위 질의는 조인, 그룹, 하위 질의 등의 복합 SELECT 구문을 포함할 수 있다.
62 --2. 뷰를 정의하는 하위 질의는 ORDER BY 절을 포함할 수 없다. ORDER BY 절은 뷰에서 데이터를 검색할 때 지정.
63 --3. View 를 수정할 때에는 ALTER를 사용하지 않고, OR REPLACE를 사용한다.
64 --4. VIEW 의 구조를 볼 때는 DESC 사용.
65
66 /*
67 CREATE VIEW VIEW_TEST
68 AS
69 SELECT * FROM TEST; --ERROR : TEST 테이블이 없음.
70
71 CREATE FORCE VIEW VIEW TEST
72 AS
73 SELECT * FROM TEST; --WARNING
74
75 DESC VIEW TEST;
76
77 CREATE VIEW empview10
78 AS
79 SELECT empno, ename, job
80 FROM emp
81 WHERE deptno = 10;
82
83 DESC empview10 --VIEW 구조보기
84
85 SELECT * FROM empview10; --View를 이용한 데이터 조회
86
87 CREATE VIEW EMP20
88 AS
89 SELECT EMPNO, ENAME, SAL
90 FROM EMP
91 WHERE DEPTNO = 20;
92
93 DESC EMP20;
94

```

```

95      SELECT * FROM EMP20;
96
97      CREATE OR REPLACE EMP20(ENO, NAME, PAYROLL)
98      AS
99      SELECT EMPNO, ENAME, SAL
100     FROM EMP
101     WHERE DEPTNO = 20;
102  */
103
104  REM Data Dictionary에서 View정보보기
105  --SELECT view_name, text FROM USER_VIEWS;
106
107  REM VIEW 의 원리
108  --1. 사용자가 뷰에 대해서 질의를 하면 USER VIEWS 에서 뷰에 대한 정의를 조회한다.
109  --2. 기본 테이블에 대한 뷰의 권한을 살핀다.
110  --3. 뷰에 대한 질의를 기본 테이블에 대한 질의로 변환한다. 이때, USER VIEWS의 TEXT 의 값을 참조한다.
111  --4. 기본 테이블에 대한 질의를 통해 데이터를 검색한다.
112  --5. 검색된 결과를 출력한다.
113
114  /*
115      CREATE VIEW EMP_30_VU
116      AS
117      SELECT EMPNO, ENAME, SAL, DEPTNO
118      FROM EMP
119      WHERE DEPTNO = 30;
120
121      DESC EMP 30 VU;
122
123      INSERT INTO EMP_30_VU
124      VALUES(1111, 'Jimin', 500, 30);
125
126      SELECT * FROM EMP_30_VU;
127      SELECT * FROM EMP;  --VIEW에 추가한 것이 실제 기본 테이블에도 반영됨.
128  */
129  REM View 수정
130  --1. OR REPLACE 옵션을 사용
131  --2. 이미 뷰가 있더라도 뷰를 생성하여 해당 뷰를 대체
132  --CREATE OR REPLACE VIEW empview10
133  --(employee number, employee name, job title)
134  --AS
135  --SELECT empno, ename, job
136  --FROM emp
137  --WHERE deptno = 10;
138
139  REM VIEW 실습
140  --1. 부서별로 부서명, 최소 급여, 최대 급여, 부서의 평균 급여를 포함하는 DEPT SUM View 를 생성하라.
141  --CREATE OR REPLACE VIEW dept_sum
142  --(deptno, tmin, tmax, tavg) --컬럼이름을 주지 않으면 오류발생
143  --AS
144  --SELECT deptno, MIN(sal), MAX(sal), AVG(sal)
145  --FROM emp
146  --GROUP BY deptno;
147
148  --2. emp table에서 사원번호, 이름, 업무를 포함하는 emp_view VIEW를 생성하시오.
149  --CREATE OR REPLACE VIEW emp_view
150  --(사원번호, 이름, 업무)
151  --AS
152  --SELECT empno, ename, job
153  --FROM emp;
154
155  --3. 위 2번에서 생성한 VIEW를 이용하여 10번 부서의 자료만 조회하시오
156  --CREATE OR REPLACE VIEW emp_view
157  --(사원번호, 이름, 업무)
158  --AS
159  --SELECT empno, ename, job
160  --FROM emp
161  --WHERE deptno = 10;
162
163  --4. 위 2번에서 생성한 VIEW를 Data Dictionary 에서 조회하시오.
164  --SELECT view_name, text
165  --FROM user v_iews
166  --WHERE view name = UPPER('emp view');
167
168  --5. 이름, 업무, 급여, 부서명, 위치를 포함하는 emp_dept_name 이라는 VIEW를 생성하시오.
169  --CREATE OR REPLACE VIEW emp_dept name
170  --AS
171  --SELECT ename, job, sal, dname, loc
172  --FROM emp, dept
173  --WHERE emp.deptno = dept.deptno;
174
175  --6. 87년에 입사한 사람을 볼 수 있는 뷰
176  --CREATE OR REPLACE VIEW view_emp_83
177  --(sabun, name, hiredate)
178  --AS
179  --SELECT empno, ename, hiredate
180  --FROM emp
181  --WHERE TO CHAR(hiredate, 'YY') = '83';
182
183  --7.부서별로 부서명, 최소급여, 최대급여, 부서별 평균급여를 포함하는 view_dept_sum 뷰를 생성하시오.
184  --CREATE OR REPLACE VIEW view_dept sum(v dname, v min sal, v max sal, v avg sal)
185  --AS
186  --SELECT dname, MIN(sal), MAX(sal), AVG(sal)
187  --FROM emp, dept
188  --WHERE emp.deptno = dept.deptno
189  --GROUP BY dept.dname;

```

```

190
191 REM 복합뷰
192 --1. 두개 이상의 테이블로 부터 값을 출력하고, 그룹함수를 포함하는 복잡한 VIEW
193 --2. 뷰의 어떤 컬럼이 함수나 표현식에서 유래했다면 반드시 별칭을 사용해야함
194
195 --사원이름, 업무, 급여, 부서명, 위치를 포함하는 view_emp_dept 뷰를 생성하시오.
196 --CREATE OR REPLACE VIEW view_emp_dept
197 --("사원이름", "업무", "급여", "부서명", "부서의 위치")
198 --AS
199 --SELECT ename, job, sal, dname, loc
200 --FROM emp, dept
201 --WHERE emp.deptno = dept.deptno AND emp.deptno = 10;
202
203 REM 단순뷰에서 DML 사용하기
204 --1. 뷰가 다음과 같은 사항을 포함하고 있다면 DELETE 는 할 수 없다.
205 --1) 그룹함수
206 --2) GROUP BY절
207 --3) DISTINCT 키워드
208 --2. 뷰가 다음과 같은 사항을 포함하고 있다면 UPDATE를 할 수 없다.
209 --1) 그룹함수
210 --2) GROUP BY절
211 --3) DISTINCT 키워드
212 --4) 표현식으로 표현된 열
213 --5) ROWNUM 의사열
214 --3. 뷰가 다음과 같은 사항을 포함하고 있다면 INSERT를 할 수 없다.
215 --1) 그룹함수
216 --2) GROUP BY절
217 --3) DISTINCT 키워드
218 --4) 표현식으로 표현된 열 과 ROWNUM 의사열
219 --5) 뷰에 의해 선택되지 않은 NOT NULL 이 기본 테이블에 있는 경우
220
221 /*
222 CREATE TABLE dept_clone
223 AS
224 SELECT *
225 FROM dept;
226
227 ALTER TABLE dept clone
228 ADD CONSTRAINT dept clone deptno pk PRIMARY KEY(deptno);
229
230 ALTER TABLE dept_clone
231 MODIFY (dname VARCHAR2(14) NOT NULL);
232
233 CREATE OR REPLACE VIEW view_dept_clone
234 AS
235 SELECT deptno, loc
236 FROM dept_clone;
237
238 INSERT INTO VIEW DEPT CLONE
239 VALUES(50, 'SEOUL'); --ERROR
240 */
241
242 REM WITH CHECK OPTION 절 사용하기
243 --1. 뷰를 통해 뷰에서 지정한 참조 무결성을 체크할 수 있다.
244 --2. 뷰를 통해서 실행되는 INSERT 나 UPDATE는 WITH CHECK OPTION 절이 있으면 뷰를 가지고 검색할 수 없는 행의
    생성을 허용하지 않는다.
245
246 --사원테이블과 동일한 emp_20(20번부서만)이라는 뷰를 생성하되, WITH CHECK OPTION 을 사용해서 생성하시오.
247 /*
248 CREATE OR REPLACE VIEW emp_20
249 AS
250 SELECT * FROM emp
251 WHERE deptno = 20
252 WITH CHECK OPTION CONSTRAINT emp_20 ck;
253
254 UPDATE emp_20
255 SET deptno = 30
256 WHERE empno = 7566;
257 */
258
259 REM WITH READ ONLY 사용하기
260 --1. 단순뷰는 DML 연산을 수행할 수 있다.
261 --2. 하지만, WITH READ ONLY 절을 사용해서 생성된 뷰는 DML 연산을 할 수 없다.
262
263 --사원테이블에서 emp_10(10번부서의 사원중 사번, 이름, 직무만 포함)이라는 뷰를 생성해서 WITH READ ONLY
    절을 이용해서 생성하시오.
264 /*
265 CREATE OR REPLACE VIEW emp_10
266 AS
267 SELECT empno, ename, job
268 FROM emp
269 WHERE deptno = 10
270 WITH READ ONLY;
271
272 UPDATE EMP_10
273 SET JOB = 'MANAGER'; --오류
274 */
275
276 REM WITH CHECK OPTION 과 WITH READ ONLY 의 차이
277 --WITH CHECK OPTION 은 조건에 사용한 컬럼의 값을 수정하지 못하게 하는 것
278 --WITH READ ONLY 는 기본 테이블의 모든 컬럼 값을 수정하지 못하게 하는 것
279
280 REM Top-N Query
281 --1. 테이블에서 조건에 대한 최상위 레코드 n개 또는 최하위 레코드 n개를 표시
282 --2. Syntax

```

```

283      --SELECT [column list], ROWNUM
284      --FROM      (SELECT [column list] FROM table
285      --              ORDER BY Top-N_column)
286      --WHERE ROWNUM <= N;
287  /*
288      SELECT ROWNUM, ENAME, SAL
289      FROM EMP;      --ROWNUM : 각 행에 대한 일련번호
290
291      SELECT ROWNUM, EMPNO, ENAME, HIREDATE
292      FROM EMP
293      ORDER BY HIREDATE;
294  */
295
296  REM ROWNUM
297  --1. ORACLE 에서 내부적으로 부여하는 칼럼
298  --2. INSERT 문을 이용해서 입력하면 입력한 순서에 따라 1씩 증가하면서 값이 지정
299  --3. 데이터가 입력된 시점에서 결정되는 ROWNUM 칼럼값은 바뀌지 않는다.
300  --4. 만일 정렬된 순서대로 ROWNUM 칼럼값이 매겨지게 하려면 쿼리문을 새로운 테이블에 저장해야 한다.
301
302  /*
303      CREATE OR REPLACE VIEWS VIEW_HIREDATE
304      AS
305      SELECT EMPNO, ENAME, HIREDATE
306      FROM EMP
307      ORDER BY HIREDATE;
308
309      SELECT ROWNUM, EMPNO, ENAME, HIREDATE
310      FROM VIEW_HIREDATE;
311
312      SELECT ROWNUM, EMPNO, ENAME, HIREDATE
313      FROM VIEW HIREDATE
314      WHERE ROWNUM <=5;  --입사일이 빠른 5명
315
316      SELECT ROWNUM rank, ename, sal
317      FROM      (SELECT ename, sal
318      --              FROM emp
319      --              ORDER BY sal DESC) e
320      WHERE ROWNUM <= 3;
321
322      --emp table에서 가장 최근에 입사한 5명의 사원번호, 사원명, 입사날짜를 출력하시오.
323      SELECT ROWNUM AS "랭킹", empno, ename, hiredate
324      FROM      (SELECT empno, ename, hiredate
325      --              FROM emp
326      --              ORDER BY hiredate DESC) aaa
327      WHERE ROWNUM <= 5;
328  */
329
330  REM View 제거
331  --1.뷰가 삭제돼도 기본 테이블의 데이터에는 영향이 없음
332  --2. Syntax
333      --DROP VIEW view_name;
334
335      --DROP VIEW empview10;
336
337
338  REM SEQUENCE
339  --1. SEQUENCE ?
340      --1) 자동으로 생성되는 중복되지 않은 유일한 값
341      --2) 테이블의 행에 대해 자동적으로 번호를 부여하기 위해 사용하는 객체
342      --3) 각 레코드는 유일하기 위한 PRIMARY KEY를 부여하기 위함이다.
343      --4) 값의 증가 또는 감소를 할 수 있다.
344      --5) 응용 프로그램 코드의 양을 줄일 수 있다.
345      --6) 테이블과 별도로 저장 및 생성되기 때문에 여러 테이블에 동일한 시퀀스 적용 가능
346
347  --2. 특징
348      --1) 자동적으로 유일 번호를 생성한다.
349      --2) 공유가능한 객체이다.
350      --3) 주로 기본키 생성에 사용된다.
351      --4) 자바같은 어플리케이션의 코드를 대체할 수 있다.
352      --5) 메모리에 CACHE되면 보다 효율성이 높아진다.
353
354  --3. Syntax
355      --CREATE SEQUENCE sequence_name
356      --      [START WITH n]      --생략하면 기본 1
357      --      [INCREMENT BY n]    --생략하면 기본 1
358      --      [MAXVALUE n | NOMAXVALUE]    --최대값은 10의 27승
359      --      [MINVALUE n | NOMINVALUE ]    --최소값은 -10의 26승
360      --      [CYCLE | NOCYCLE]      --기본값은 NOCYCLE
361      --      [CACHE | NOCACHE]      --기본값은 20개
362
363      --INCREMENT BY n : 시퀀스 번호 간격. 여기서 n은 정수. (이 절을 생략하면 시퀀스는 1씩 증가.)
364      --START WITH n : 첫 번째로 생성할 시퀀스 번호를 지정. (이 절을 생략하면 시퀀스는 1에서 시작.)
365      --MAXVALUE n : 시퀀스가 생성할 수 있는 최대값.
366      --NOMAXVALUE : 오름차순 시퀀스의 최대값으로 10의 27승을 지정하고 내림차순 시퀀스의 최대값으로 -1을 지정.(기본 옵션)
367      --MINVALUE n : 시퀀스의 최소값.
368      --NOMINVALUE : 오름차순 시퀀스의 최소값으로 1을 지정하고 내림차순 시퀀스의 최소값으로 -10의 26승을 지정.(기본 옵션)
369      --CYCLE | NOCYCLE : 시퀀스가 최대값 또는 최소값에 도달한 후 추가 값의 생성 여부를 지정.(기본 옵션: NOCYCLE)
370      --CACHE n | NOCACHE : Oracle Server가 미리 할당하여 메모리에 저장할 값의 개수지정.(기본적으로 Oracle Server는 20개 지정)
371
372  --4. Sequence 생성
373      --CREATE SEQUENCE dept deptno seq
374      --INCREMENT BY 1

```

```

375      --START WITH 91
376      --NOCACHE
377      --NOCYCLE;
378
379 --5. SEQUENCE 확인
380 --SELECT sequence_name, min_value, max_value, increment_by, last_number
381 --FROM USER_SEQUENCES
382 --WHERE sequeunce_name = UPPER('ddd');
383
384 REM Sequence 사용
385 --1. NEXTVAL 및 CURRVAL 의사 열을 사용하여 시퀀스 값 참조.
386 --1) CURRVAL 에 새로운 값이 할당되기 위해서는 NEXTVAL 로 새로운 값을 생성해야 한다.
387 --2) NEXTVAL 로 새로운 값을 생성한 다음 이 값을 CURRVAL 로 대체한다.
388
389 --3) NEXTVAL 의사 열
390 --a. 현재 시퀀스 값의 다음 값을 반환
391 --b. sequence.NEXTVAL을 참조하면 새 시퀀스 번호가 생성
392 --SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM dual;
393
394 --4) CURRVAL 의사 열
395 --a. 현재 사용자가 방금 생성한 시퀀스 번호를 참조하는 데 사용.
396 --b. CURRVAL 참조 전에 NEXTVAL을 사용하여 현재 사용자 세션에서 시퀀스 번호를 생성
397 --c. CURRVAL에 시퀀스 이름을 지정해야 한다.
398 --d. sequence.CURRVAL을 참조하면 해당 사용자 프로세스에 마지막으로 반환된 값이 표시.
399
400 --2. NEXTVAL 및 CURRVAL 사용 규칙
401 --1) 하위 질의에 속하지 않은 SELECT 문의 SELECT 목록
402 --2) INSERT 문에 있는 하위 질의의 SELECT 목록
403 --3) INSERT 문의 VALUES 절
404 --4) UPDATE 문의 SET 절
405
406 --3. GuideLine
407 --1) 사용할 수 있는 경우
408 --a. subquery 아닌 SELECT 문에서
409 --b. INSERT 문장에서 VALUES 절
410 --c. UPDATE 문장에서 SET 절
411 --2) 사용할 수 없는 경우
412 --a. 뷰에서의 SELECT문
413 --b. DISTINCT가 들어있는 SELECT문
414 --c. GROUP BY, HAVING, ORDER BY 를 사용하는 SELECT문
415 --d. subquery
416 --e. CREATE TABLE, ALTER TABLE 에서 DEFAULT 옵션 사용할 때
417
418 --4. 사용 예
419 /*
420      CREATE SEQUENCE DEPT DEPTNO_SEQ
421      INCREMENT BY 10
422      START WITH 10;
423
424      SELECT DEPT DEPTNO_SEQ.NEXTVAL FROM DUAL; --10
425      SELECT DEPT_DEPTNO_SEQ.CURRVAL FROM DUAL; --10
426
427      SELECT DEPT DEPTNO_SEQ.NEXTVAL FROM DUAL; --20
428      SELECT DEPT_DEPTNO_SEQ.CURRVAL FROM DUAL; --20
429
430      SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL; --30
431      SELECT DEPT DEPTNO_SEQ.CURRVAL FROM DUAL; --30
432
433      SELECT DEPT_DEPTNO_SEQ.NEXTVAL FROM DUAL; --40
434      SELECT DEPT DEPTNO_SEQ.CURRVAL FROM DUAL; --40
435
436      --INSERT INTO dept(deptno, dname, loc)
437      --VALUES (DEPT_DEPTNO_SEQ.NEXTVAL, 'MARKETING', 'SAN DIEGO');
438 */
439 --5. 시퀀스 현재 값 확인
440 --SELECT DEPT_DEPTNO_SEQ.CURRVAL FROM dual;
441
442 --6. 실습
443 /*
444      CREATE SEQUENCE SAMPLE_SEQ;
445      SELECT SAMPLE_SEQ.CURRVAL FROM DUAL; --ERROR,
446      --NEXTVAL을 사용하지 않고 바로 CURRVAL 을 사용하기 때문
447
448      CREATE TABLE dept_temp
449      AS
450      SELECT * FROM dept
451      WHERE 1 < 0;
452
453      ALTER TABLE dept temp
454      ADD CONSTRAINT dept_temp_pk PRIMARY KEY(deptno);
455
456      CREATE SEQUENCE dept_copy_deptno
457      START WITH 10
458      INCREMENT BY 10
459      MAXVALUE 50;
460
461      INSERT INTO dept temp
462      VALUES (dept_copy_deptno.NEXTVAL, 'Sales', 'Seoul');
463      INSERT INTO dept_temp
464      VALUES (dept_copy_deptno.NEXTVAL, 'Accounting', 'Pusan');
465      INSERT INTO dept_temp
466      VALUES (dept_copy_deptno.NEXTVAL, 'Marking', 'Kuro');
467      INSERT INTO dept temp
468      VALUES (dept_copy_deptno.NEXTVAL, 'IT', 'Kangnam');
469 */

```

```

470
471 REM Sequence 수정
472 --1. 시퀀스가 MAXVALUE 한계에 도달하면 해당 시퀀스의 추가 값을 할당할 수 없다.
473 --2. 시퀀스가 MAXVALUE를 초과함을 알리는 오류 메시지가 나타난다.
474 --3. 시퀀스를 계속 사용하려면 ALTER SEQUENCE 문을 사용하여 수정한다.
475 --4. GuideLine
476 --1) Sequence 에 대한 수정권한, 소유자여야 한다.
477 --2) 변경이후부터 적용된다.
478 --3) 시작값을 변경하려면 제거하고 새로 생성해야 한다.
479 --4) INCREMENT BY, MAXVALUE, MINVALUE, CYCLE, CACHE를 변경가능
480 --5) 일부 검증이 수행. 예를 들어, 새로운 MAXVALUE 값을 현재 시퀀스 번호보다 작게 지정할 수 없다.
481
482 --ALTER SEQUENCE sequence name
483 --      [INCREMENT BY n]
484 --      [{MAXVALUE n | NOMAXVALUE}]
485 --      [{MINVALUE n | NOMINVALUE}]
486 --      [{CYCLE | NOCYCLE}]
487 --      [{CACHE n | NOCACHE}];
488
489 /*
490 ALTER SEQUENCE dept deptno seq
491 INCREMENT BY 10
492 MAXVALUE 990;
493
494 ALTER SEQUENCE dept copy deptno
495 MAXVALUE 99;
496 */
497 REM Sequence 제거
498 --DROP SEQUENCE sequence_name;
499
500 --DROP SEQUENCE dept deptno seq;
501
502 REM SEQUENCE 실습
503 --1. DEPT table의 기본 키 열로 사용할 시퀀스를 다음 조건에 맞게 생성하시오.
504 --      1) 시퀀스 값은 60에서 시작하여 10씩 증가하며 최대 200까지 가능하도록 한다.
505 --      2) 시퀀스 이름은 dept_deptno_seq로 지정한다.
506
507 --2. 위 1에서 만든 시퀀스의 이름, 최대값, 증가분, 마지막 번호의 정보를 아래와 같이 출력되도록 Data
508 Dictionary에서 조회하시오
509 --SEQUENCE_NAME          MAX_VALUE          INCREMENT_BY          LAST_NUMBER
510 -----
511 --DEPT_DEPTNO_SEQ      200                  10                      60
512
513 --3. 위 1에서 만든 시퀀스를 이용하여 부서이름은 education, 부서 위치는 seoul로 새 행을 dept 테이블에
514 추가하고, 아래와 같이 조회하시오.
515 --      60      EDUCATION      SEOUL
516
517 REM INDEX
518 --1. 행에 대한 빠른 참조를 위해서, 테이블에 인덱스를 생성할 수 있다.
519
520 --2. 장/단점
521 --1) 장점
522 --검색 속도가 빨라진다.
523 --시스템에 부하를 줄여서 시스템 전체 성능을 향상시킨다.
524 --2) 단점
525 --인덱스를 위한 추가 공간이 필요하다.
526 --인덱스를 생성하는 데 시간이 걸린다.
527 --데이터의 변경작업이 자주 일어나는 경우에는 오히려 성능이 더 떨어진다.
528
529 --3. 생성해야 할 조건
530 --1) 테이블의 행의 수가 많다. 아주작은 크기의 테이블에는 오히려 성능이 떨어진다.
531 --2) 사용자의 SQL 문에서 WHERE 조건절에 자주 사용되는 칼럼이 대상이 된다.
532 --3) WHERE 조건에 의한 결과가 전체 행수의 비율(분포도) 2~4% 인 경우에 효과가 있다.
533 --4) JOIN 에 자주 사용되는 칼럼이나 NULL 을 포함하는 칼럼이 많은 경우
534
535 --4. 생성하지 않아야 할 조건
536 --1) 테이블에 행이 적은 경우
537 --2) 칼럼이 WHERE 조건에 자주 사용되지 않을 때
538 --3) WHERE 조건에 의한 결과가 전체 행에 대해 10~15%의 결과보다 높게 리턴될 때
539 --4) 테이블이 자주 입력, 수정 , 삭제 될 때는 오히려 검색 속도가 더 떨어진다.
540
541 --5. INDEX Type
542 --1) UNIQUE index : 지정된 열의 값이 고유해야 한다.
543 CREATE UNIQUE INDEX I_emp_ename ON emp(ename);
544
545 /*
546 CREATE TABLE DEPT1
547 AS
548 SELECT * FROM DEPT
549 WHERE 0 = 1;
550
551 INSERT INTO DEPT1 VALUES(10, 'ACCOUNTING', 'SEOUL');
552 INSERT INTO DEPT1 VALUES(20, 'SALES', 'PUSAN');
553 INSERT INTO DEPT1 VALUES(30, 'OPERATION', 'PUSAN');
554 INSERT INTO DEPT1 VALUES(40, 'IT', 'DAEJUN');
555
556 CREATE UNIQUE INDEX IDX_DEPT1_DEPTNO ON DEPT1(DEPTNO); --SUCCESS
557 CREATE UNIQUE INDEX IDX_DEPT1_LOC ON DEPT1(LOC); --ERROR
558 CREATE INDEX IDX_DEPT1_DEPTNO_ON DEPT1(DEPTNO); --SUCCESS. UNIQUE 를 빼면 됨.
559 */
560 --2) Non-unique index : 가장 빠름을 보장하는 칼럼, 칼럼의 값이 고유하지 않을 때
561 CREATE INDEX I_emp_ename ON emp(ename);
562 --3) Single column index : 하나의 칼럼에만 인덱스를 부여
563 CREATE INDEX I_emp_ename ON emp(ename);

```



```

562      --4) Composite Index : 여러 컬럼에 인덱스부여가능 (16개까지 가능)
563          CREATE INDEX I_emp empno ename ON emp(empno, ename);
564  /*
565      CREATE INDEX IDX_DEPT1_COM ON DEPT1(DEPTNO, DNAME);
566  */
567  --5) 함수 기반 인덱스
568      CREATE INDEX I_emp_ename ON emp(sal * 12);
569
570  /*
571      ----조회 속도 비교하기
572      CREATE TABLE EMP_CLONE
573      AS
574      SELECT * FROM EMP; --PRIMARY KEY 가 들어오지 않음.
575
576      INSERT INTO EMP_CLONE SELECT * FROM EMP;
577      INSERT INTO EMP_CLONE SELECT * FROM EMP;
578      INSERT INTO EMP_CLONE SELECT * FROM EMP;
579      --아주 여러번 수행해서 레코드의 갯수를 늘린다.
580
581      --검색용 레코드 한 개 추가
582      INSERT INTO EMP_CLONE(EMPNO, ENAME)
583      VALUES(1111, 'Jimin');
584
585      --시작 시간 설정
586      SET TIMING ON
587
588      SELECT DISTINCT EMPNO, ENAME FROM EMP_CLONE
589      WHERE ENAME = 'JIMIN';
590      --경과된 시간 표시...메모할 것
591
592  */
593  --6. INDEX 생성
594  --CREATE INDEX I_emp_ename ON emp(ename);
595
596  /*
597      --위에서 생성한 테이블에 인덱스 부여
598      CREATE INDEX IDX_EMP_ENAME ON EMP_CLONE(ENAME);
599      --다시 검색한다.
600      SELECT DISTINCT EMPNO, ENAME FROM EMP_CLONE
601      WHERE ENAME = 'JIMIN';
602      --경과된 시간 표시...비교한다.
603  */
604  --7. INDEX 확인
605  --SELECT index_name, table_name FROM USER_INDEXES;
606  --SELECT index_name, table_name, column_name FROM USER_IND_COLUMNS;
607  --SELECT index_name, column_name
608  --FROM USER_IND_COLUMNS
609  --WHERE table_name = UPPER('zipcode');
610
611  --SELECT index_name, index_type, table_name
612  --FROM USER_INDEXES
613  --WHERE table_name = UPPER('zipcode');
614
615  --DESC USER_COLUMNS;
616
617  --7. INDEX 의 제거
618  --DROP INDEX index_name
619  --DROP INDEX i_emp_ename;
620
621  --8. INDEX 의 수정
622  --1) 인덱스는 수정할 수 없다.
623  --2) 수정하기 위해서는 제거하고 새로 생성해야 한다.
624
625
626  REM SYNONYM 동의어
627  --1. What?
628      --테이블, 시퀀스, 뷰, 프로그램 유닛에 대한 별칭
629      --다른 사용자가 소유한 객체를 참조하기 위해서는 유저명.객체명으로 접근해야 한다.
630      --다른 이름으로 접근하기 위한 객체를 동의어라 한다.
631      --사용할 수 있는 객체는 TABLE, VIEW, SEQUENCE, Procedure.
632      --비공개 동의어 : 객체에 대한 접근 권한을 부여 받은 사용자가 정의한 동의어로, 해당 사용자만 사용가능
633      --공개 동의어 : 권한을 주는 사용자가 정의한 동의어로 누구나 사용할 수 있는 동의어. 공개 동의어는 DBA
        권한을 가지 사용자만이 생성할 수 있다. 또는 CREATE PUBLIC SYNONYM 권한이 있는 사용자만 생성 가능.
        SYNONYM 앞에 PUBLIC 을 붙인다.
634      --DUAL 은 원래 SYS 가 소유하는 테이블인데, 다른 사용자가 접근하기 위해서는 원래 SYS.DUAL로 해야 하지만,
        지금까지 모든 사용자가 접근할 수 있는 이유는 DUAL 은 PUBLIC 이기 때문이다.
        --SELECT * FROM DUAL;
635
636
637  --2. Syntax
638      --CREATE [PUBLIC] SYNONYM synonym_name
639      --FOR user_name.object_name;
640
641  --3. 실습
642  /*
643      CONN SCOTT/tiger
644      SHOW USER --SCOTT
645      CREATE SYNONYM e FOR SCOTT.EMP;
646      --ERROR. 기본적으로 SYNONYM 생성 권한 없음.
647
648      CONN SYS AS SYSDBA
649      SHOW USER --SYS
650          GRANT CREATE SYNONYM TO scott;
651
652      CONN SCOTT/tiger
653      SHOW USER --SCOTT

```

```

654
655 CREATE SYNONYM e FOR SCOTT.EMP;  --SUCCESS
656
657 SELECT * FROM e;
658
659 SELECT * FROM USER_SYNONYMS;  --확인
660
661 -----PUBLIC SYNONYM 생성하기
662 CONN SYS AS SYSDBA
663 SHOW USER
664
665 CREATE USER JAVA IDENTIFIED BY java
666 DEFAULT TABLESPACE USERS
667 TEMPORARY TABLESPACE TEMP;
668
669 GRANT RESOURCE, CONNECT TO JAVA;
670 GRANT SELECT ANY TABLE TO JAVA;
671 --다른 사용자의 테이블을 SELECT 하기 위한 권한설정
672
673 CONN SCOTT/tiger
674 CREATE PUBLIC SYNONYM D_TEMP FOR SCOTT.DEPT;
675 --ERROR. 기본적으로 유저는 CREATE SYNONYM 권한이 없다.
676
677 CONN SYSTEM/javaoracle
678 GRANT CREATE PUBLIC SYNONYM TO scott;
679 --PUBLIC 권한은 DBA만 가능, 혹은 CREATE PUBLIC SYNONYM 권한 사용자만 가능.
680
681 CONN SCOTT/tiger
682 CREATE PUBLIC SYNONYM D_TEMP FOR SCOTT.DEPT;  --SUCCESS
683
684 CONN JAVA/java
685
686 SELECT * FROM D_TEMP;  --SUCCESS
687 */
688
689 --3. 동의어 삭제
690 --DROP [PUBLIC] SYNONYM synonym name;
691 --단 PUBLIC SYNONYM 은 DBA의 권한만이 가능
692
693 --4. 확인
694 --SELECT * FROM USER_SYNONYMS;

```