

# ORACLE OBJECT INDEX

## ▶ INDEX

SQL명령문의 처리 속도를 향상시키기 위해서 컬럼에 대해 생성하는  
오라클 객체로 내부 구조는 B\*트리 형식으로 구성되어 있음

### ✓ 장점

검색 속도가 빨라지고 시스템에 걸리는 부하를 줄여  
시스템 전체 성능 향상

### ✓ 단점

인덱스를 위한 추가 저장 공간이 필요하고 인덱스를 생성하는데  
시간이 걸림  
데이터의 변경 작업(INSERT/UPDATE/DELETE)이 자주 일어날 경우  
오히려 성능 저하

## ▶ INDEX 표현식

**CREATE [UNIQUE] INDEX** 인덱스 명

**ON** 테이블 명(컬럼 명, 컬럼 명 | 함수 명, 함수 계산식);

**SELECT \* FROM** USER\_IND\_COLUMNS;

INDEX_NAME	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION	COLUMN_LENGTH	CHAR_LENGTH	DESCEND
1 SYS_C007182	USER_UNIQUE	USER_ID	1	20	20 ASC	
2 SYS_C007184	USER_UNIQUE2	USER_ID	1	20	20 ASC	
3 SYS_C007186	USER_UNIQUE3	USER_ID	2	20	20 ASC	
4 SYS_C007186	USER_UNIQUE3	USER_NO	1	22	0 ASC	
5 SYS_C007188	USER_PRIMARYKEY	USER_NO	1	22	0 ASC	
6 SYS_C007189	USER_PRIMARYKEY	USER_ID	1	20	20 ASC	
7 SYS_C007196	USER_PRIMARYKEY2	USER_ID	2	20	20 ASC	
8 SYS_C007196	USER_PRIMARYKEY2	USER_NO	1	22	0 ASC	
9 SYS_C007210	USER_GRADE	GRADE_CODE	1	22	0 ASC	
10 SYS_C007212	USER_FOREIGNKEY	USER_NO	1	22	0 ASC	
11 SYS_C007213	USER_FOREIGNKEY	USER_ID	1	20	20 ASC	
12 SYS_C007226	USER_CHECK	USER_NO	1	22	0 ASC	
13 SYS_C007227	USER_CHECK	USER_ID	1	20	20 ASC	
14 엔터티1_PK	EMPLOYEE	EMP_ID	1	3	3 ASC	
15 엔터티1_PK1	JOB	JOB_CODE	1	2	2 ASC	
16 엔터티1_PK2	DEPARTMENT	DEPT_ID	1	2	2 ASC	
17 엔터티1_PK3	LOCATION	LOCAL_CODE	1	2	2 ASC	
18 엔터티1_PK4	NATIONAL	NATIONAL_CODE	1	2	2 ASC	
19 엔터티2_PK	SAL_GRADE	SAL_LEVEL	1	2	2 ASC	

## ▶ INDEX 구조

```
SELECT ROWID, EMP_ID, EMP_NAME
FROM EMPLOYEE;
```

### ✓ IDX\_EMPID

KEY	ROWID
200	AAAE7UAABAAALC5AAA
201	AAAE7UAABAAALC5AAB
202	AAAE7UAABAAALC5AAC

**AAAE7UAABAAALC5AAA**

데이터 오브젝트 번호    파일 번호    BLOCK 번호    ROW 번호

ROWID	EMP_ID	EMP_NAME
1 AA AE7UAABAAALC5AAA	200	선동일
2 AA AE7UAABAAALC5AAB	201	송종기
3 AA AE7UAABAAALC5AAC	202	노용철
4 AA AE7UAABAAALC5AAD	203	송은희
5 AA AE7UAABAAALC5AAE	204	유재식
6 AA AE7UAABAAALC5AAF	205	정중하
7 AA AE7UAABAAALC5AAG	206	박나라
8 AA AE7UAABAAALC5AAH	207	하미유
9 AA AE7UAABAAALC5AAI	208	김해술
10 AA AE7UAABAAALC5AAJ	209	심봉선
11 AA AE7UAABAAALC5AAK	210	윤은해
12 AA AE7UAABAAALC5AAL	211	전형돈
13 AA AE7UAABAAALC5AAM	212	장프위
14 AA AE7UAABAAALC5AAN	213	하동운
15 AA AE7UAABAAALC5AAO	214	방명수
16 AA AE7UAABAAALC5AAP	215	대복혼
17 AA AE7UAABAAALC5AAQ	216	차태연
18 AA AE7UAABAAALC5AAR	217	전지연
19 AA AE7UAABAAALC5AAS	218	미오리
20 AA AE7UAABAAALC5AAT	219	임시환
21 AA AE7UAABAAALC5AAU	220	미종석
22 AA AE7UAABAAALC5AAV	221	유하진
23 AA AE7UAABAAALC5AAW	222	이태림
24 AA AE7UAABAAALC5AAX	900	장채현

## ▶ INDEX 종류

1. 고유 인덱스(UNIQUE INDEX)  
중복 값이 포함될 수 없음  
PRIMARY KEY 제약조건을 생성하면 자동으로 생성됨
2. 비고유 인덱스(NONUNIQUE INDEX)  
빈번하게 사용되는 일반 컬럼을 대상으로 생성  
주로 성능 향상을 위한 목적으로 생성
3. 단일 인덱스(SINGLE INDEX)  
한 개의 컬럼으로 구성된 인덱스
4. 결합 인덱스(COMPOSITE INDEX)  
두 개 이상의 컬럼으로 구성된 인덱스
5. 함수 기반 인덱스(FUNCTION-BASED INDEX)  
SELECT절이나 WHERE절에 산술 계산식/함수식이 사용된 경우  
계산식은 인덱스의 적용을 받지 않음

## ▶ INDEX 종류

### ✓ 고유 인덱스(UNIQUE INDEX)

```
CREATE UNIQUE INDEX IDX_EMPNO
ON EMPLOYEE(EMP_NO);
```

Unique index IDX\_EMPNO이 (가) 생성되었습니다.

```
SELECT * FROM USER_IND_COLUMNS
WHERE TABLE_NAME = 'EMPLOYEE';
```

INDEX_NAME	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION	COLUMN_LENGTH	CHAR_LENGTH	DESCEND
1 엔터티1_PK	EMPLOYEE	EMP_ID	1	3	3	ASC
2 IDX_EMPNO	EMPLOYEE	EMP_NO	1	14	14	ASC

```
CREATE UNIQUE INDEX IDX_DEPTCODE
ON EMPLOYEE(DEPT_CODE);
```

\* UNIQUE INDEX는 중복 값이 있는 컬럼에 생성 시 에러 발생

오류 보고 -

SQL 오류: ORA-01452: cannot CREATE UNIQUE INDEX; duplicate keys found  
01452. 00000 - "cannot CREATE UNIQUE INDEX; duplicate keys found"

\*Cause:

\*Action:

## ▶ INDEX 종류

### ✓ 비고유 인덱스(NONUNIQUE INDEX)

```
CREATE INDEX IDX_DEPTCODE  
ON EMPLOYEE(DEPT_CODE);
```

Index IDX\_DEPTCODE이 (가) 생성되었습니다.

```
SELECT * FROM USER_IND_COLUMNS  
WHERE TABLE_NAME = 'EMPLOYEE';
```

\* NONUNIQUE INDEX는 중복 값이 있는 컬럼에도 생성 가능

	INDEX_NAME	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION	COLUMN_LENGTH	CHAR_LENGTH	DESCEND
1	엔터티1_PK	EMPLOYEE	EMP_ID	1	3	3	ASC
2	IDX_EMPNO	EMPLOYEE	EMP_NO	1	14	14	ASC
3	IDX_DEPTCODE	EMPLOYEE	DEPT_CODE	1	2	2	ASC

## ▶ INDEX 종류

### ✓ 결합 인덱스(COMPOSITE INDEX)

```
CREATE INDEX IDX_DEPT
ON DEPARTMENT(DEPT_ID, DEPT_TITLE);
```

Index IDX\_DEPT이 (가) 생성되었습니다.

```
SELECT * FROM USER_IND_COLUMNS
WHERE TABLE_NAME = 'DEPARTMENT';
```

\* COMPOSITE INDEX는 두 개 이상의 컬럼을 하나의 인덱스로 생성 가능

\* COLUMN\_POSITION의 순서에 의해 성능이 차이날 수 있음

	INDEX_NAME	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION	COLUMN_LENGTH	CHAR_LENGTH	DESCEND
1	엔터티1 PK2	DEPARTMENT	DEPT_ID	1	2		2 ASC
2	IDX_DEPT	DEPARTMENT	DEPT_ID	1	2		2 ASC
3	IDX_DEPT	DEPARTMENT	DEPT_TITLE	2	35		35 ASC



## ▶ INDEX 종류

### ✓ 함수 기반 인덱스(FUNCTION-BASED INDEX)

```
CREATE TABLE EMP_SAL
```

Table EMP\_SAL이 (가) 생성되었습니다.

```
AS SELECT EMP_ID, EMP_NAME, SALARY, BONUS,
```

```
       (SALARY + (SALARY + NVL(BONUS, 0))) * 12 연봉
```

```
FROM EMPLOYEE;
```

```
CREATE INDEX IDX_SALCALC
```

Index IDX\_SALCALC이 (가) 생성되었습니다.

```
ON EMP_SAL ((SALARY + SALARY * NVL(BONUS, 0))) * 12);
```

```
SELECT * FROM USER_IND_COLUMNS
```

```
WHERE TABLE_NAME = 'EMP_SAL';
```

INDEX_NAME	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION	COLUMN_LENGTH	CHAR_LENGTH	DESCEND
1 IDX_SALCALC	EMP_SAL	SYS_NC00006\$	1	22		0 ASC

## ▶ INDEX 재생성

DML 작업(특히 DELETE 명령)을 수행한 경우,  
해당 인덱스 엔트리가 논리적으로만 제거되고  
실제 엔트리는 남아있게 되므로 제거된 인덱스가 필요 없는 공간을  
차지하고 있지 않도록 인덱스를 재생성 해야 함

### ✓ 표현식

```
ALTER INDEX 인덱스명 REBUILD;
```

# ▶ INDEX 활용한 정렬

**SELECT \* FROM EMPLOYEE  
ORDER BY EMP\_NO;**

**SELECT \* FROM EMPLOYEE  
WHERE EMP\_NO > '0';**

\* 둘 다 주민번호 기준 오름차순 정렬이지만 ORDER BY로 정렬하는 것보다  
인덱스를 활용하는 것이 더 좋은 성능을 보임

EMP_ID	EMP_NAME	EMP_NO	EMAIL	PHONE	DEPT_CODE	JOB_CODE	SAL_LEVEL	SALARY	BONUS	MANAGER_ID	HIRE_DATE	ENT_DATE	ENT_YN
1 200	선동일	621235-1985634	sun_di@kh.or.kr	01099546325	D9	J1	S1	8000000	0.3 (null)		90/02/06	(null)	N
2 201	송중기	631156-1548654	song_jk@kh.or.kr	01045686656	D9	J2	S1	6000000	(null) 200		01/09/01	(null)	N
3 202	노홍철	861015-1356452	no_hc@kh.or.kr	01066656263	D9	J2	S4	3700000	(null) 201		01/01/01	(null)	N
4 203	송은희	631010-2653546	song_eh@kh.or.kr	01077607879	D6	J4	S5	2800000	(null) 204		96/05/03	(null)	N
5 204	유재석	660508-1342154	yoo_js@kh.or.kr	01099999129	D6	J3	S4	3400000	0.2 200		00/12/29	(null)	N
6 205	정종하	770102-1357951	jung_jh@kh.or.kr	01036654875	D6	J3	S4	3900000	(null) 204		99/09/09	(null)	N
7 206	박나라	630709-2054321	pack_nr@kh.or.kr	01096935222	D5	J7	S6	1800000	(null) 207		08/04/02	(null)	N
8 207	하미유	690402-2040612	ha_iy@kh.or.kr	01036654488	D5	J5	S5	2200000	0.1 200		94/07/07	(null)	N
9 208	김해솔	870927-1313564	kim_hs@kh.or.kr	01078634444	D5	J5	S5	2500000	(null) 207		04/04/30	(null)	N
10 209	심봉선	750206-1325546	sim_bs@kh.or.kr	0113654485	D5	J3	S4	3500000	0.15 207		11/11/11	(null)	N
11 210	윤은혜	650505-2356985	youn_eh@kh.or.kr	0179964233	D5	J7	S5	2000000	(null) 207		01/02/03	(null)	N
12 211	전형돈	830807-1121321	jun_hd@kh.or.kr	01044432222	D8	J6	S5	2000000	(null) 200		12/12/12	(null)	N
13 212	장프위	780923-2234542	jang_zw@kh.or.kr	01066682224	D8	J6	S5	2550000	0.25 211		15/06/17	(null)	N
14 213	하동운	621111-1785463	ha_dh@kh.or.kr	01158456632	(null)	J6	S5	2320000	0.1 (null)		99/12/31	(null)	N
15 214	방영수	856795-1313513	bang_ms@kh.or.kr	01074127545	D1	J7	S6	1380000	(null) 200		10/04/04	(null)	N
16 215	대복혼	881130-1050911	dae_bh@kh.or.kr	01088808584	D5	J5	S4	3760000	(null) (null)		17/06/19	(null)	N
17 216	차태연	770808-1364897	cha_ty@kh.or.kr	01064643212	D1	J6	S5	2780000	0.2 214		13/03/01	(null)	N
18 217	전지연	770808-2665412	jun_jy@kh.or.kr	01033624442	D1	J6	S4	3660000	0.3 214		07/03/20	(null)	N
19 218	미오리	870427-2232123	loo_or@kh.or.kr	01022306545	(null)	J7	S5	2890000	(null) (null)		16/11/28	(null)	N
20 219	임시환	660712-1212123	im_sw@kh.or.kr	(null)	D2	J4	S6	1550000	(null) (null)		99/09/09	(null)	N
21 220	이종석	770823-1113111	lee_js@kh.or.kr	(null)	D2	J4	S5	2490000	(null) (null)		14/09/18	(null)	N
22 221	유하진	800808-1123341	yoo_hj@kh.or.kr	(null)	D2	J4	S5	2480000	(null) (null)		94/01/20	(null)	N
23 222	이태림	760918-2854697	lee_tr@kh.or.kr	01033000002	D8	J6	S5	2436240	0.35 100		97/09/12	17/09/12	Y
24 900	장채현	901123-1080503	jang_ch@kh.or.kr	01055569512	D1	J8	S3	4300000	0.2 200		17/09/19	(null)	N