

NULL 속성의 이해 개념서

이 문서는 데이터베이스에서 NULL 값의 개념, 특징, 처리 방법 및 관련 주의사항에 대해 포괄적으로 설명합니다. NULL의 기본 개념부터 실전 활용 예제와 SQLD 시험 대비 팁까지 다룹니다. 데이터베이스 설계, 쿼리 작성, 성능 최적화에 있어 NULL의 중요성을 강조하며, 실무에서 자주 마주치는 NULL 관련 문제들을 해결하는 방법을 제시합니다.

1. NULL의 기본 개념

1.1 NULL의 정의

- 아직 정의되지 않은 값
- 알 수 없는(unknown) 값
- 해당되는 값이 없음
- 무한대의 가능한 값

1.2 NULL의 특징

1. 비교 연산

```
-- NULL은 = 으로 비교 불가
WHERE 컬럼 = NULL    -- (X)
WHERE 컬럼 IS NULL   -- (O)
```

2. 산술 연산

```
-- NULL과의 산술 연산 결과는 NULL
SELECT 100 + NULL;  -- 결과: NULL
SELECT 100 * NULL;  -- 결과: NULL
```

3. 집계 함수

```
-- NULL은 집계함수에서 제외됨
SELECT COUNT(*)      -- NULL 포함
SELECT COUNT(컬럼명) -- NULL 제외
```

2. NULL 처리 함수

2.1 NVL/ISNULL 함수

```
-- Oracle
SELECT NVL(컬럼명, '대체값')
FROM 테이블;

-- SQL Server
SELECT ISNULL(컬럼명, '대체값')
FROM 테이블;
```

```
-- 사용 예시
SELECT 사원명, NVL(급여, 0) AS 급여
FROM 사원;
```

2.2 COALESCE 함수

```
-- 첫 번째로 NULL이 아닌 값 반환
SELECT COALESCE(컬럼1, 컬럼2, 컬럼3, '기본값')
FROM 테이블;

-- 사용 예시
SELECT 사원명,
       COALESCE(핸드폰, 집전화, 회사전화, '연락처없음') AS 연락처
FROM 사원;
```

2.3 NULLIF 함수

```
-- 두 값이 같으면 NULL, 다르면 첫 번째 값 반환
SELECT NULLIF(컬럼1, 컬럼2)
FROM 테이블;

-- 사용 예시
SELECT NULLIF(입력값, '0') AS 결과
FROM 테이블;
```

3. NULL과 조인

3.1 INNER JOIN

```
-- NULL 값은 조인 조건에서 제외됨
SELECT e.사원명, d.부서명
FROM 사원 e
INNER JOIN 부서 d ON e.부서번호 = d.부서번호;
```

3.2 OUTER JOIN

```
-- NULL 값도 결과에 포함
SELECT e.사원명, d.부서명
FROM 사원 e
LEFT OUTER JOIN 부서 d ON e.부서번호 = d.부서번호;

-- NULL 값 처리
SELECT e.사원명,
       COALESCE(d.부서명, '부서미배정') AS 부서명
FROM 사원 e
LEFT OUTER JOIN 부서 d ON e.부서번호 = d.부서번호;
```

4. NULL과 인덱스

4.1 인덱스 처리

```
-- Oracle에서는 NULL 값도 인덱스에 포함 가능
CREATE INDEX idx_col1 ON 테이블(컬럼1);

-- 복합 인덱스에서의 NULL 처리
CREATE INDEX idx_composite
ON 테이블(컬럼1, 컬럼2, 컬럼3);
```

4.2 성능 고려사항

```
-- NULL 포함 컬럼의 인덱스 사용
SELECT *
FROM 테이블
WHERE 컬럼1 IS NULL;  -- 인덱스 스캔 가능

-- 효율적인 인덱스 활용
SELECT *
FROM 테이블
WHERE NVL(컬럼1, '대체값') = '찾는값';  -- 인덱스 사용 불가
```

5. NULL과 제약조건

5.1 NOT NULL 제약조건

```
-- 테이블 생성 시
CREATE TABLE 사원 (
    사원번호 NUMBER PRIMARY KEY,
    사원명 VARCHAR2(100) NOT NULL,
    이메일 VARCHAR2(200) NOT NULL,
    전화번호 VARCHAR2(20)
);

-- 기존 테이블 변경
ALTER TABLE 사원
MODIFY (사원명 NOT NULL);
```

5.2 UNIQUE 제약조건

```
-- NULL 값은 UNIQUE 제약조건에서 중복 허용
CREATE TABLE 회원 (
    회원번호 NUMBER PRIMARY KEY,
    이메일 VARCHAR2(200) UNIQUE,
    닉네임 VARCHAR2(100) UNIQUE
);
```

6. NULL 값 처리 전략

6.1 데이터 모델링 단계

```
-- 기본값 정의
CREATE TABLE 주문 (
    주문번호 NUMBER PRIMARY KEY,
    주문상태 VARCHAR2(10) DEFAULT 'PENDING',
    할인을 NUMBER DEFAULT 0
);

-- NOT NULL 제약조건 활용
CREATE TABLE 고객 (
    고객번호 NUMBER PRIMARY KEY,
    고객명 VARCHAR2(100) NOT NULL,
    가입일자 DATE DEFAULT SYSDATE NOT NULL
);
```

6.2 애플리케이션 단계

```
-- NULL 값 표시 전략
SELECT 사원명,
       CASE WHEN 급여 IS NULL THEN '미입력'
            ELSE TO_CHAR(급여, '999,999')
       END AS 급여
FROM 사원;

-- NULL 값 변환 처리
INSERT INTO 사원 (사원명, 급여)
VALUES (
    :사원명,
    NVL(:급여, 0)
);
```

7. NULL 관련 주의사항

7.1 비교 연산

```
-- 잘못된 사용
SELECT * FROM 사원 WHERE 급여 = NULL;  -- 결과 없음

-- 올바른 사용
SELECT * FROM 사원 WHERE 급여 IS NULL;
SELECT * FROM 사원 WHERE 급여 IS NOT NULL;
```

7.2 집계 함수

```
-- 잘못된 이해
SELECT AVG(급여) FROM 사원;  -- NULL 제외하고 평균 계산

-- 올바른 처리
SELECT AVG(NVL(급여, 0)) FROM 사원;  -- NULL을 0으로 처리하고 평균 계산
```

8. 실전 활용 예제

8.1 리포트 작성

```
-- NULL 처리를 포함한 리포트 쿼리
SELECT 부서명,
       COUNT(*) AS 총사원수,
       COUNT(급여) AS 급여받는사원수,
       AVG(NVL(급여, 0)) AS 평균급여,
       MAX(급여) AS 최대급여,
       MIN(NVL(급여, 0)) AS 최소급여
FROM 사원
GROUP BY 부서명;
```

8.2 데이터 정제

```
-- NULL 데이터 정제 쿼리
UPDATE 고객
SET 이메일 = NULL
WHERE 이메일 IN ('', ' ', 'NULL', 'null');

UPDATE 고객
SET 전화번호 = NULLIF(전화번호, '000-0000-0000');
```

SQLD 시험 대비 TIP

주요 출제 포인트

- NULL의 개념**
 - 정의와 특징
 - 연산 규칙
- NULL 처리 함수**
 - NVL/ISNULL
 - COALESCE
 - NULLIF
- NULL과 조인/인덱스**
 - 조인에서의 동작
 - 인덱스 활용

학습 전략

- NULL의 기본 개념 이해
- NULL 처리 함수의 용도 숙지
- 조인과 인덱스에서의 영향 파악
- 실제 사례 학습

실전 문제 유형

- NULL 연산 결과 예측
- 적절한 NULL 처리 함수 선택
- 조인 결과 예측
- 성능 관련 문제