

I 데이터 모델링의 이해

제 1장 데이터 모델링의 이해

제 1절 데이터 모델의 이해

데이터 모델링이란?

- (1) 정보시스템을 구축하기 위한 데이터 관점의 업무 분석 기법
- (2) 현실 세계의 데이터를 약속된 표기법으로 표현하는 과정
- (3) 데이터베이스 구축을 위한 분석 및 설계의 과정

데이터 모델링의 특징: 추상화, 단순화, 명확성

관점: 데이터, 프로세스(업무), 상관(처리 방식)

유의사항: 유연성, 유일성, 일관성

데이터 모델링의 3단계:

개념적(추상) – 논리적(정규화) – 물리적 모델링(DB)

데이터 베이스 스키마 구조:

외부(뷰) – 개념(통합된 사용자) - 내부 스키마(물리)

데이터 독립성: 논리적 독립성, 물리적 독립성

데이터 모델링의 3요소: 엔터티, 관계, 속성

ERD 작성 순서: (1) 엔터티 도출 (2) 배치 (3) 관계 설정 (4) 관계명 기술 (5) 관계 차수 설정 (6) 선택사항 기술

제 2절 엔터티 [Noun]

정의: 업무에서 관리해야 하는 데이터의 집합, 단수명사, 인스턴스의 집합

특징: (1) 업무에서 필요로 함 (2) 유일한 식별자 (3) 2개 이상의 인스턴스 집합 (4) 업무 프로세스에서 이용됨 (5) 2개 이상의 속성 (6) 관계 가짐

유무형에 따른 분류: 유형 엔터티, 무형 엔터티

발생 시점에 따른 분류: 기본, 중심, 행위 엔터티

엔터티 명명 규칙: (1) 협업에서 사용되는 용어 (2) 약어 지양 (3) 단수 명사 (4) 유일성 보장 (5) 의미 명확

제 3절 속성

정의: 업무에서 필요로 하는 인스턴스에서 관리하고자 하는 의미상 더 분리되지 않는 최소의 데이터 단위

특징: (1) 업무에서 필요 (2) 주식별자에 함수적으로 종속(기본키 변경되면 속성도 변경) (3) 1개의 속성은 1개의 속성값 가짐 (4) 속성도 집합

특성에 따른 분류: 기본, 설계, 파생 속성

분해 가능 여부: 단일, 복합, 단일값, 다중값 속성

도메인: 속성이 가질 수 있는 값의 범위

제 4절 관계 [Verb]

관계의 표기법: 관계명, 관계차수(多是 삼발 표시), 관계 선택사항(필수는!, 선택은 O로 표시)

ERD: 존재관계, 행위관계 (표기 구분 안 함)

UML: 연관관계(실선, 멤버변수), 의존관계(점선, 파라미터)

식별관계(실선)	비식별관계(점선)
강한 연결관계	약한 연결관계
자식 식별자의 구성에 포함	자식 일반 속성에 포함
부모 엔터티에 종속	

제 5절 식별자

정의: 엔터티를 대표할 수 있는 유일성을 만족하는 속성 (자주 이용되는 속성, 명칭이나 내역 이름은 X)

특징: (1) 유일성 (2) 최소성 (3) 불변성 (4) 존재성

대표성	주식별자	보조식별자
스스로 생성	내부식별자	외부식별자

속성의 수	단일 식별자	복합식별자
대체 여부	본질식별자	인조식별자

제 2장 데이터 모델과 SQL

제 1절 정규화

제 1정규형: 모든 속성은 반드시 하나의 값을 가져야 한다 [원자성]:

제 2정규형: 엔터티의 일반 속성은 주식별자 전체에 종속이어야 한다 [부분함수종속성]:

제 3정규형: 엔터티의 일반속성 간에는 서로 종속적이지 않아야 한다 [이행함수종속성]:

보이스코드 정규형: 후보키가 기본키 속성 중 일부에 함수적 종속일 때 다수의 주식별자를 분리함

제 2절 관계와 조인의 이해

(1) 조인이란 식별자를 상속하고, 상속된 속성을 매핑 키로 활용하여 데이터를 결합하는 것

(2) 부모의 식별자를 자식의 식별자에 포함하면 식별 관계, 부모의 식별자를 자식의 일반속성으로 상속하면 비식별관계 (비식별관계에서 조인 많이 발생)

(3) 관계를 맺는다는 것은 식별자를 상속시키고 해당 식별자를 매핑하여 데이터를 결합하는 것

제 3절 모델이 표현하는 트랜잭션의 이해

IE: 필수적인 관계(실선), 선택적인 관계(원)

바커: 필수적인 관계(실선), 선택적인 관계(점선)

제 4절 Null 속성의 이해

Null의 특성: (1) 아직 정의되지 않은 값으로, 0이나 ' '이 아님 (2) NOT NULL 또는 PRIMARY KEY 외 모든 데이터 유형에 포함 가능 (3) NVL, ISNULL로 다른 결

과값을 얻음 (4) 집계 함수에서는 제외됨

Null의 연산:

(1) NULL값과의 연산은 Null을 리턴

(2) 모든 비교는 알 수 없음(Unknown) 리턴

(3) 집계함수는 Null을 제외하고 계산

제 5절 본질식별자 vs. 인조식별자

(1) 인조식별자는 대체로 본질식별자가 복잡한 구성을 가질 때 만들어진다

(2) 인조식별자를 사용하면 중복 데이터를 막기 어려워진다

(3) 인조식별자를 사용하면 본질식별자를 사용할 때와 비교하여 추가적인 인덱스가 필요해진다

(4) 인조 식별자는 단점도 존재하므로 꼭 필요한 경우에만 사용하는 것이 바람직하다

II SQL 기본 및 활용

제 1장 SQL 기본

제 1절 관계형 데이터베이스 개요

데이터베이스: 효율적인 데이터 관리와 데이터 손상을 피하고 데이터 복구를 위한 시스템(DBMS)

명령어 종류	명령어	설명
DML	SELECT	조회 및 검색 (= RETRIEVE)
	INSERT UPDATE DELETE	데이터 변형
DDL	CREATE ALTER DROP RENAME	데이터 구조 정의
DCL	GRANT REVOKE	권한 부여 및 회수
TCL	COMMIT ROLLBACK	트랜잭션 제어

일반 집합 연산자: UNION, INTERSECTION, DIFFERENCE, PRODUCT(CROSS JOIN)

순수 관계 연산자: SELECT, PROJECT, (NATURAL) JOIN, DIVIDE(현재 사용되지 않음)

CHARCTER(s), VARCHAR(s), NUMERIC, DATETIME

제 2절 SELECT문

```
SELECT [ALL/DISTICT/(*)] 칼럼명1, 칼럼명2 ...  
FROM 테이블명
```

ALIAS: (1) 칼럼명 바로 뒤에 위치 (2) 칼럼명과 ALIAS 사이에 AS 키워드 사용 가능 (3) 이중 인용부호는 ALIAS가 공백, 특수문자를 포함하는 경우나 대소문자 구분이 필요할 때 사용

합성연산자:

- (1) 수직 바 || (Oracle)
- (2) 플러스 + (SQL server)
- (3) CONCAT (string1, string2)

```
SELECT first_name || last_name AS full_name  
FROM employees;
```

실행순서: FROM – WHERE – GROUP BY – HAVING – SELECT – ORDER BY

제 3절 함수

함수 종류	내용
문자형 함수	CONCAT, UPPER, LOWER, SUBSTR, LENGTH, TRIM, REPLACE, INSTR, LEFT, RIGHT, MID 등
숫자형 함수	ROUND, ABS, POWER, CEIL, FLOOR, MOD, SIGN, SQRT, EXP, LOG 등
날짜형 함수	DATEADD, DATEDIFF, YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, DATEPART, GETDATE 등
변환형 함수	CAST, CONVERT, TO_CHAR, TO_NUMBER, TO_DATE, NUMTOYMINTERVAL, TO_TIMESTAMP 등

SIGN: 숫자가 양수인지, 음수인지, 0인지 구별

CEIL(숫자): 보다 크거나 작은 최소 정수를 리턴

FLOOR(숫자): 보다 작거나 같은 최대 정수를 리턴

TRUNC(숫자,[m]): 숫자를 m자리에서 반올림해 리턴, m생략 시 디폴트값은 0

EXP: 지수 값 리턴, POWER: 거듭제곱 값 리턴, SQRT: 제곱근값 리턴, LN: 자연 로그 값 리턴

참고) WEHRE절에는 집계 함수를 사용할 수 없음

단일행 NULL 관련 함수의 종류:

NVL(표현식1, 표현식2): 1이 NULL이면 2 출력
NVL2(표현식1, 표현식2, 표현식3): 1이 NULL이면 3, 아니면 2
NULLIF(표현식 1, 표현식 2): 1=2면 NULL 리턴, 1<>2면 표현식 1 리턴
COALESCE(표현식 1, 표현식2): 임의의 개수 표현식에서 NULL이 아닌 최소의 표현식 나타냄

제 4절 WHERE 절

조건식: (1) 칼럼명 (2) 비교 연산자 (3) 문자, 숫자, 표현식 (4) 비교 칼럼명 (JOIN 사용 시)

구분	연산자	비고
비교	=, >, >=, <, <=	
SQL	BETWEEN a AND B IN (list) LIKE '비교문자열' IS NULL	비교문자열에 '%', '_' 사용
논리	AND, OR, NOT	
부정비교	!=, ^=, <>, NOT=	
부정 SQL	NOT BETWEEN a AND B NOT IN (list) IS NOT NULL	

참고) 오라클에 ' ' 입력하면 NULL로 입력되어 조회하려면 IS NULL 조건으로 조회하여야 함. SQL에서는 '' 로 저장 및 조회 가능

- (1) 검색 CASE 표현식: 개별 조건 확인하고 반환
- (2) 단순 CASE 표현식: 표현식 값 기준, 여러 조건을 확인
- (3) DECODE: 여러 조건 비교하고 일치하는 조건의 결과를 반환

제 5절 GROUP BY, HAVING절

일반적으로 GROUP BY절과 같이 사용되지만 테이블 전체가 하나의 그룹이 되는 경우에는 단독으로 사용

이 가능함

집계함수: COUNT(*), COUNT, SUM, AVG, MAX, MIN, STDDEV, CARIANCE/VAR, 기타

특성:

- (1) GROUP BY절을 통해 소그룹별 기준을 정한 후, SELECT절에 집계 함수를 사용
- (2) 집계함수의 통계 정보는 NULL 제외하고 수행
- (3) SELECT절과 달리 ALIAS 사용 불가
- (4) HAVING절은 GROUP BY절의 기준항목이나 소그룹의 집계함수를 이용한 조건 표시
- (5) GROUP BY절에 의한 소그룹별로 만들어진 집계 데이터 중, HAVING절에서 제한 조건을 두어 만족하는 내용만 출력
- (6) HAVING절은 일반적으로 GROUP BY절 뒤에 위치하지만 GROUP BY 없어도 사용 가능

제 6절 ORDER BY 절

- (1) 기본적인 정렬 순서는 오름차순(ASC)
- (2) 오라클에서 NULL은 최댓값, SQL에서는 최솟값
- (3) SELECT절에서 오직 한 개만 올 수 있음

제 7절 조인

- (1) 일반적으로 조인은 PK와 FK의 연관성에 의해 성립된다 (어떤 경우 논리적인 값들의 연관만으로도 성립됨)
- (2) DBMS 옵티마이저는 FROM절에 나열된 데이터들을 항상 2개로 묶어서 처리한다
- (3) EQUU JOIN은 조인에 관여하는 테이블들의 값이 정확하게 일치할 때 (=) 사용된다 이외는 NON EQUI JOIN임 (설계상 불가능한 경우 있음)

INNER JOIN	동일한 값만 반환, 디폴트 값, 선택표 혹은 조건절로 수행
NATURAL JOIN	동일한 이름의 칼럼에 대해 수행
USING 조건절 / ON 조건절	원하는 칼럼 조건
CROSS JOIN	카타시안 조합
OUTER 조인	(+ 표기

제 2장 SQL 활용

제 1절 서브쿼리

연관 서브쿼리: 서브쿼리가 메인 쿼리 칼럼을 가짐

비연관 서브쿼리: 메인 쿼리에 값을 제공하기 위한 목적으로 사용됨.

단일 행 서브쿼리: 실행결과가 항상 1건 이하, 단일 행 비교 연산자(=,<,>,<=,>=,<>)와 사용

다중 행 서브 쿼리: 실행 결과가 여러 건인 서브쿼리. 다중행 비교 연산자와 함께 사용

연산자	다중행 비교 연산자 설명
IN	결과에 값이 포함되는지 확인 (OR 조건)
ANY	결과 중 하나라도 조건을 만족하는지
ALL	모든 값이 조건을 만족하는지
EXISTS	결과가 존재하는지 여부를 확인

다중 칼럼 서브 쿼리: 여러 칼럼 반환, 메인 쿼리 조건절에 따라 여러 칼럼 동시에 비교 가능

- (1) 스칼라 서브쿼리: SELECT 절에서 사용, 한행, 한 칼럼만을 반환하는 서브쿼리
- (2) 인라인 뷰 (동적 뷰) FROM 절에서 사용, 서브 쿼리를 임시 테이블처럼 사용
- (3) HAVING절, ORDER BY절 등에서도 사용 가능

제 2절 집합 연산자

UION(중복 제거), UNION ALL(결과 전부 합침), INTERSECT(교집합, 중복 제거), EXCEPT(차집합, 중복 제거)

제 3절 그룹함수

NULL 빼고 집계, 결과값 없는 행 출력 안 함

표현식	출력값
ROLL UP(1,2)	1과 2별 소계, 1별 소계, 총 합계 (계층 구조, 순서 바뀌면 결과 값 바뀜)
CUBE(1,2)	1과 2별 소계, 1별 소계, 2별 소계, 총 합계(순서 무관)
GROUPING SETS(1,2)	1별 소계, 2별 소계(순서 무관)

제 4절 윈도우 함수

결과에 대한 처리로, 결과 건수에 영향 미치지 않음

구분	함수	비고
순위	ROW_NUMBER	

	RANK DENSE_RANK	
집계	SUM, AVG, MAX, MIN COUNT	
행 순서	FIRST_VALUE LAST_VALUE LAG LEAD	
비율	PERCENT_RANK CUME_DIST NTILE	행 순서별 백분율, 건수 누적 백분율, 전체 건수 주어진 인자로 N등분

문법: SELECT 윈도우함수 (A) OVER (PARTITION BY 칼럼 ORDER BY칼럼 윈도우절) FROM 테이블명

윈도우절	설명
BETWEEN a AND b	프레임 범위 지정합니다.
UNBOUNDED PRECEDING/FOLLOWING	프레임 시작/끝을 현재 윈도우 그룹의 첫 번째/마지막 행으로 설정
N PRECEDING/FOLLOWING	현재 행을 기준으로 지정된 수만큼 이전/이후의 행을 나타냄
CURRENT ROW	현재 행을 기준으로 윈도우 프레임을 설정

제 5절 Top N 쿼리

- (1) ORDER BY: 데이터 정렬
- (2) LIMIT: 정렬된 결과에서 상위 N개 행 선택
- (3) FETCH: 결과 집합에서 상위 N개 행 선택
- TOP(n) WITH TIES: 값이 동일한 경우 함께 출력

제 6절 계층형 질의와 셀프 조인

- CONNECT BY: 트리형태의 구조로 쿼리 수행
- START WITH: 계층 구조 전개의 시작 위치 (최상위 행) 지정
- CONNECT_BY_ROOT/ISLEAF: 최상위/하위 계층값
- SYS_CONNECT_BY_PATH: 계층 구조의 전개 경로
- ORDER BY SIBLINGS BY: 형제 노드 사이에서 정렬
- (1) SQL Server에서 계층형 질의문은 CTE(Common Table EXPRESSION)를 재귀호출함으로써 계층구조를 전개한다

- (2) "" 앵커 멤버를 실행하여 기본 결과 집합을 만들고 이후 재귀 멤버를 지속적으로 실행한다
- (3) 오라클의 계층형 질의문에서 WHERE절은 모든 전개를 진행한 후 필터 조건으로서 조건을 만족하는 데이터만을 추출하는 데 활용된다
- (4) "" PRIOR 키워드는 CONNECT BY절 뿐만 아니라 SELECT, WHERE절에서도 사용할 수 있다.

셀프조인: 동일 테이블 사이의 조인. 식별을 위해 반드시 테이블 별칭(Alias) 사용

```
SELECT ALIAS명1.칼럼명,ALIAS명2.칼럼명...
FROM 테이블 ALIAS명1, 테이블 ALIAS명2
WHERE ALIAS명 1.칼럼명2 = ALIAS명2.칼럼명1;
```

뷰 사용의 장점:

- (1) 독립성: 테이블 구조가 변경되어도 뷰를 사용하는 응용프로그램 변경하지 않아도 된다
- (2) 편리성: 복잡한 질의 단순하게 작성할 수 있다
- (3) 보안성: 숨기고 싶은 정보 빼고 생성할 수 있다

제 3장 SQL 관리 구문

제약조건의 종류: PRIMARY KEY(기본키), UNIQUE KEY(고유키), NOT NULL, CHECK, FOREIGN KEY(외래키)

기본키 할당:

```
ALTER TABLE 테이블명 ADD CONSTRAINT
constraint_name PRIMART KEY (칼럼명1, 칼럼명2)
```

트랜잭션의 특성:

- (1) 원자성: 연산은 모두 실행되거나 전혀 실행되지 않음
- (2) 일관성: 연산 이전에 데이터에 잘못 없다면 연산 이후에도 잘못이 있으면 안됨
- (3) 고립성: 연산 도중 다른 트랜잭션 영향 받지 않음
- (4) 지속성: 트랜잭션이 성공적으로 수행되면 영구 저장됨

DELETE(MODIFY) ACTION:

- (1) Cascade: Master 삭제 시 Child 같이 삭제
- (2) SET NULL / SET Default: "" NULL값 처리 / 기본값
- (3) Restrict: Child 테이블에 PK값 없는 경우에만 Master 삭제 가능

(4) No Action: 참조무결성 위반하는 삭제나 수정 액션 x

INSERT ACTION:

- (1) Automatic: Master PK 자동으로 생성 후 Child 입력
- (2) SET NULL / Default: PK 없으면 Null값 처리 / 기본값
- (3) Dependent: Master 테이블에 PK가 존재할 때만 Child 입력 허용
- (4) No Action: 참조무결성 위반하는 액션 x

DROP	TRUNCATE	DELETE
ROLLBACK 불가 (Auto Commit)	ROLLBACK 불가 (Auto Commit)	사용자 Commit 이전ROLLBACK 가능
테이블이 사용했던 Storage를 모두 Release	최초 테이블 생성 시 할당된 Storage 남기고 Release	데이터 모두 Delete해도 Storage Release 되지 않음
테이블 정의 자체를 완전히 삭제	테이블을 최초 생성된 초기 상태로 만들	데이터만 삭제

DB 키의 종류:

종류	설명
기본키	엔터티를 대표하는 키 (Null값 불가)
후보키	유일성과 최소성 만족
슈퍼키	유일성만 만족
대체키	기본키 제외 나머지
외래키	여러 테이블의 기본 키 필드, 참조 무결성 확인하기 위해 사용 (Null값 가능)
고유키	고유한 값 보장 (Null값 단 1개만 가능)

연산자의 우선순위: 괄호 - NOT - 비교 연산자 및 SQL 연산자 - AND - OR

[DML]

SELECT 컬럼명 FROM 테이블명;

INSERT INTO 테이블명 VALUES (컬럼 명시 안 하면 모든 컬럼에 들어갈 값 입력해야 함)

UPDATE 테이블명 SET 컬럼명 = 필드값;

DELETE FROM 테이블명 (WHERE 조건절);

[DDL] 자동으로 커밋되며 롤백이 불가함

CREATE 테이블명 (컬럼명 데이터타입 제약조건...);

[Oracle]

ALTER TABLE 테이블명 MODIFY (컬럼명 1 데이터 유형 [DEFAULT 식][NOT NULL], 컬럼명2 데이터 유형...);

[SQL Server]

ALTER TABLE 테이블명 ALTER (컬럼명1 데이터 유형 [DEFAULT 식][NOT NULL], 컬럼명2 데이터 유형...);

DROP(TRUNCATE) 테이블명

[DCL]

GRANT(REVOKE) 권한 ON 프로젝트 TO 유저명;