**3과목 데이터베이스 구축**

1장 논리 데이터베이스 설계

076 데이터베이스 설계

- 사용자의 요구를 분석하여 데이터베이스의 구조에 맞게 변형한 후 DBMS로 데이터베이스를 구현하여 일반 사용자들이 사용하게 하는 것

무결성 / 일관성 / 회복/ 보안 / 효율성 / 데이터베이스 확장

요구 조건 분석 -> 개념적 설계 -> 논리적 설계 -> 물리적 설계 -> 구현

개념적 설계 – E-R다이어그램 논리적 설계 – 트랜잭션 인터페이스

077 데이터 모델의 개념

- 현실 세계의 정보를 컴퓨터에 표현하기 위해 단순화, 추상화하여 체계적으로 표현

데이터 모델 구성 요소 : 개체, 속성, 관계

데이터 모델 종류 : 개념적 데이터 모델, 논리적 데이터 모델, 물리적 데이터 모델

데이터 모델에 표시할 요소 : 구조, 연산, 제약조건

구조 – 데이터 구조 및 정적 성질 / 연산 – 데이터베이스를 조작하는 기본 도구 / 제약 조건 – 논리적인 제약 조건

078 데이터 모델의 구성 요소 – 개체

- 데이터베이스에 표현하려는 것, 사람이 생각하는 개념이나 정보 단위 같은 현실 세계의 대상체

유형, 무형의 정보 / 영속적 / 유일한 식별자로 구별 가능 / 하나 이상의 관계

업무에서 사용하는 용어로 지정 / 약어 사용 X / 단수 명사 사용 / 개체명 유일 / 의미에 따라 이름 부여

079 데이터 모델의 구성 요소 – 속성

- 데이터 베이스를 구성하는 가장 작은 논리적 단위

데이터 항목 또는 데이터 필드

개체를 구성하는 항목

개체의 특성을 기술

속성의 수 – 디그리 or 차수

기본 속성 / 설계 속성 / 파생 속성

업무에서 사용하는 용어로 지정 / 서술형 지정 X / 약어 사용 X / 개체명은 속성명으로 사용X / 개체에서 유일하게 식별 가능

080 데이터 모델의 구성 요소 – 관계

- 개체와 개체 사이의 논리적인 연결

개체 간의 관계 / 속성 간의 관계

일 대 일(1:1) / 일 대 다(1:n) / 다 대 다(n:m)

종속 관계 / 중복 관계 / 재귀 관계 / 배타 관계

식별 관계 – B 개체의 존재 여부가 A 개체의 존재 여부에 의존적일 때 / 실선

비식별 관계 – B 개체의 존재 여부가 A 개체의 존재 여부와 관계없을 때 / 점선

081 식별자

- 각각의 인스턴스를 유일하게 구분할 수 있는 구분자 모든 개체는 한 개 이상의 식별자를 반드시 가져야 함

대표성 여부 : 주 식별자 / 보조 식별자

스스로 생성 여부 : 내부 식별자 / 외부 식별자

단일 속성 여부 : 단일 식별자 / 복합 식별자

대체 여부 : 원조 식별자 / 대리 식별자

주 식별자 – 개체를 대표하는 유일한 식별자

보조 식별자 – 주 식별자를 대신하여 개체를 식별할 수 있는 속성

유일성 / 최소성 / 불변성 / 존재성 -> 주 식별자의 특성

내부 식별자 – 개체 내에서 스스로 만들어지는 식별자

외부 식별자 – 외부 개체의 식별자를 가져와 사용하는 식별자

단일 식별자 – 주 식별자가 한 가지 속성으로만 구성된 식별자

복합 식별자 – 두 개 이상의 속성으로 구성된 식별자

원조 식별자 – 업무에 의해 만들어지는 가공되지 않은 원래의 식별자 (본질 식별자)

대리 식별자 – 속성이 두 개 이상인 경우 속성들을 하나의 속성으로 묶어 사용하는 식별자 (인조 식별자)

후보 식별자 – 각 인스턴스를 유일하게 식별할 수 있는 속성

082 E-R 모델

- 개념적 데이터 모델의 가장 대표적인 것

개체 - 사각형 / 관계 - 마름모 / 속성 – 타원

083 관계형 데이터 모델

- 2차원적인 표를 이용해서 데이터 상호 관계를 정의하는 DB 구조

대표적 언어 : SQL

1 : 1 / 1: N / N : M

084 관계형 데이터베이스의 구조

개체나 관계를 릴레이션이라는 표로 표현

튜플 – 행, 레코드 기수/대응수

속성 – 열, 데이터 항목/필드, 디그리/차수

도메인 – 같은 타입의 원자값들의 집합

릴레이션의 특징

튜플 사이에는 순서가 없음

릴레이션은 시간에 따라 변함

속성들 가느이 순서는 중요하지 않음

085 관계형 데이터베이스의 제약조건 – 키

키 – 튜플을 찾거나 순서대로 정렬할 때 튜플들을 서로 구분할 수 있는 기준이 되는 속성

후보키 – 튜플을 유일하게 식별하기 위해 사용 / 기본키로 사용할 수 있는 속성들

유일성 / 최소성 만족

기본키 – 후보키 중 주키로 중복죈 값을 가질 수 없음 / 널 값을 가질 수 없음

유일성 / 최소성 만족

대체키 – 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키

슈퍼키 – 릴레이션을 구성하는 모든 튜플들 중 슈퍼키로 구성된 속성의 잡합과 동일한 값은 나타나지 않음

유일성만 만족

외래키 – 다른 릴레이션의 기본키를 참조하는 속성

086 관계형 데이터베이스의 제약 조건 – 무결성

무결성 – 데이터베이스에 저장된 데이터 값과 그것이 표현하는 현실 세계의 실제값이 일치하는 정확성

개체 무결성 – 기본키를 구성하는 어떤 속성도 널값이나 중복값을 가질 수 없음

도메인 무결성 – 주어진 속성 값이 정의된 도메인에 속한 값이어야 함

참조 무결성 – 외래키 값은 널값이거나 참조 릴레이션의 기본키 값과 동일해야 함

사용자 정의 무결성 – 속성 값들이 사용자가 정의한 제약조건에 만족해야 함

087 관계대수 및 관계해석

관계대수 – 원하는 정보와 정보를 검색하기 위해 어떻게 유도하는가를 기술하는 절차적 언어

순수 관계 연산자 : select, project, join, division

일반 집합 연산자 : union, intersection, difference, Cartesian product

select – 선택 조건을 만족하는 튜플의 부분집합을 구하여 새로운 릴레이션을 만드는 연산

수평 연산, 시그마 사용

project – 속성 리스트에 제시된 속성 값만을 추출하여 새로운 릴레이션을 만드는 연산

수직 연산, 파이 사용

join – 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐 새로운 릴레이션을 만드는 연산

division – r의 속성이 s의 속성값을 모두 가진 튜플에서 s가 가진 속성을 제외한 속성만을 구하는 연산

union – 합집합 intersection – 교집합 difference – 차집합

관계 해석 – 원하는 정보가 무엇이라는 것만 정의하는 비절차적 특성

088 정규화

- 종속성 이론을 이용하여 잘못 설계된 관계형 스키마를 더 작은 속성의 세트로 쪼개어 바람직한 스키마로 만들어 가는 과정

일관성 / 정확성 / 단순성 / 비중복성 / 안전성

삽입 이상 / 삭제 이상 / 갱신 이상

제1정규형 – 릴레이션 속한 모든 도메인이 원자값만으로 되어 있는 정규형

제2정규형 – 기본키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속을 만족하는 정규형

제3정규형 – 기본키가 아닌 모든 속성이 기본키에 대해 이행적 종속을 만족하지 않는 정규형

BCNF – 결정자가 모두 후보키인 정규형

제4정규형 – 다치 종속이 성립하는 경우 r이 모든 속성이 a에 함수적 종속 관계를 만족하는 정규형

제5정규형 – 모든 조인 종속이 r의 후보키를 통해서만 성립되는 정규형

089 반정규화

- 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로 의도적으로 정규화 원칙을 위배하는 행위

테이블 통합 / 테이블 분할 – 수평 분할, 수직 분할 / 중복 테이블 추가 – 집계 테이블 추가, 진행 테이블 추가, 특정 부분만을 포함하는 테이블 추가 / 중복 속성 추가

090 시스템 카탈로그

- 시스템 자체에 관련이 있는 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스

시스템 카탈로그 = 데이터 사전

시스템 카탈로그에 저장된 정보 -> 메타 데이터

카탈로그의 특징

일반 이용자가 SQL을 이용하여 내용 검색 가능

카탈로그를 갱신하는 것은 허용되지 않음

DBMS가 스스로 생성하고 유지 -> 시스템 자동으로 갱신

문제정리

트랜잭션 인터페이스 설계, 스키마 평가 및 정제 -> 논리적 설계 단계

트랜잭션 모델링 -> 개념적 설계 단계

허가 받은 사용자만이 갱신할 수 있다는 설명 -> 소프트웨어 개발 시 충족시켜야 할 보안 요소인 무결성

도메인 -> 속성들이 가질 수 있는 모든 값들의 집합

개체 타입 -> 같은 개체를 갖는 속성들의 집합

오너-멤버 관계 -> 네트워크(망)형 모델

테이블에 속한 튜플 수 -> 카디널리티 / 속성의 수 -> 차수

E-R모델에서 관계 -> 개체 간의 관계 표현

릴레이션에 포함된 각 튜플 사이에는 순서가 없음

후보키 – 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합 / 기본키로 사용할 수 있는 속성들

대체키 – 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키를 의미

외래키 – 다른 릴레이션의 기본키를 참조하는 속성

수학의 Predicate Calculus에 기반을 두고 관계 데이터베이스를 위해 제안된 것 -> 관계 해석

3NF에서 모든 이행 종속을 제거하지 못한 경우 정보의 중복 문제가 존재

정규형들은 차수가 높아질수록 만족시켜야 할 제약조건이 증가

무결성 종류 – 개체 무결성 / 참조 무결성 / 도메인 무결성 / 사용자 정의 무결성

1NF : 릴레이션에 속한 모든 도메인이 원자값만으로 되어 있는 정규형, 릴레이션의 모든 속성 값이 원자 값으로만 되어 있는 정규형

2NF : 릴레이션 R이 1NF이고 기본키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속을 만족하는 정규형

3NF: 릴레이션 R이 2NF이고 기본키가 아닌 모든 속성이 기본키에 대해 이행적 종속을 만족하지 않는 정규형

BCNF : 릴레이션 R에서 결정자가 모두 후보키인 정규형

4NF : 릴레이션 R에 다치 종속 AB가 성립하는 경우 R의 모든 속성이 A에 함수적 종속 관계를 만족하는 정규형

5NF : 릴레이션 R의 모든 조인 종속이 R의 후보키를 통해서만 성립되는 정규형

키워드 : 1. 개체 2. 속성 3. 관계 4. 튜플 5. 도메인 6. 키 7. 무결성 8. 관계대수 9. 정규화 10. 시스템 카탈로그

3장 SQL 응용

107 SQL의 개념

- 국제 표준 데이터베이스 언어 / 관계대수와 관계해석을 기초로 한 혼합 데이터 언어

- 질의어(질의기능), 데이터 구조 정의, 데이터 조작, 데이터 제어 기능

SQL의 분류

DDL(데이터 정의어) -> SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경, 삭제할 때 사용하는 언어

CREATE / ALTHER / DROP

DML(데이터 조작어) -> 데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는 데 사용되는 언어

SELECT / INSERT / DELETE / UPDATE

DCL(데이터 제어어) -> 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정의하는데 사용되는 언어

COMMIT / ROLLBACK / GRANT / REVOKE

108 DDL

DB 구조, 데이터 형식, 접근 방식 등 DB를 구축하거나 수정할 목적으로 사용되는 언어

CREATE SCHEMA 스키마명 AUTHORIZATION 사용자\_id;

-> 스키마를 정의하는 명령문

CREATE DOMAIN 도메인명 [AS] 데이터\_타입

[DEFAULT 기본값]

[CONSTRAINT 제약조건명 CHECK (범위값)];

-> 도메인을 정의하는 명령문

CREATE TABLE 테이블명

(속성명 데이터\_타입 [DEFAULT 기본값] [NOT NULL],

[,PRIMARY KEY(기본키\_속셩명, )]

[,UNIQUE(대체키\_속셩명, )

[,FOREIGN KEY(외래키\_속성명, )]

[REFERENCES 참조테이블(기본키\_속성명, )]

[ON DELETE 옵션]

[ON UPDATE 옵션]

[,CONSTRAINT 제약조건명][CHECK(조건식)]];

-> 테이블을 정의하는 명령문

ON DELETE 옵션 – NO ACTION / CASCADE / SET NULL / SET DEFAULT

ON UPDATE 옵션 – NO ACTION / CASCADE / SET NULL / SET DEFAULT

CREATE VIEW 뷰명[(속성명[, 속성명, ])]

AS SELECT문;

-> 뷰를 정의하는 명령문

CREATE [UNIQUE] INDEX 인덱스명

ON 테이블명(속성명 [ASC/DESC] [,속성명 [ASC/DESC]]

[CLUSTER];

-> 인덱스를 정의하는 명령문

ALTER TABLE 테이블명 ADD 속성명 데이터\_타입 [DEFAULT‘기본값’];

ALTER TABLE 테이블명 ALTER 속성명 [SET DEFAULT‘기본값’];

ALTER TABLE 테이블명 DROP COLUMN 속성명 [CASCADE];

-> 테이블에 대한 정의를 변경하는 명령문

DROP SCHEMA 스키마명 [CASCADE | RESTRICT];

DROP DOMAIN 도메인명 [CASCADE | RESTRICT];

DROP TABLE 테이블명 [CASCADE | RESTRICT];

DROP VIEW 뷰명 [CASCADE | RESTRICT];

DROP INDEX 인덱스명 [CASCADE | RESTRICT];

DROP CONSTRAINT 제약조건명;

-> 스키마, 도메인, 기본 테이블, 뷰 테이블, 인덱스, 제약 조건 등을 제거하는 명령문

109 DCL

- 데이터 보안, 무결성, 회복, 병행 제어 등을 정의하는 데 사용하는 언어

GRANT -> 권한 부여를 위한 명령어

REVOKE -> 권한 취소를 위한 명령어

GRANT 사용자등급 TO 사용자\_ID\_리스트 [IDENTIFIED BY 암호];

REVOKE 사용자등급 FROM 사용자\_ID\_리스트;

-> 사용자등급 지정 및 해제

GRANT 권한\_리스트 ON 개체 TO 사용자 [WITH GRANT OPTION];

REVOKE [GRANT OPTION FOR] 권한\_리스트 ON 개체 FROM 사용자 [CASCADE];

-> 테이블 속성에 대한 권한 부여 및 취소

COMMIT -> 트랜잭션이 성공적으로 끝날 때 일관성 상태를 가지기 위해 변경된 모든 내용을 데이터베이스에 반영하는 명령

ROLLBACK -> 아직 COMMIT 되지 않은 변경된 모든 내용들을 취소하고 데이터베이스를 이전 상태로 되돌리는 명령어

SAVEPOINT -> 트랜잭션 내에 ROLLBACK 할 위치인 저장점을 지정하는 명령어

110 DML

- 데이터베이스 사용자가 응용 프로그램이나 질의어를 통해 저장된 데이터를 실질적으로 관리하는데 사용되는 언어

INSERT INTO 테이블명([속성명1, 속성명2, ])

VALUES (데이터1, 데이터2, );

-> 기본 테이블에 새로운 튜플을 삽입할 때 사용

DELETE

FROM 테이블명

[WHERE 조건];

-> 기본 테이블에 있는 튜플 중에서 특정 튜플(행)을 삭제할 때 사용

UPDATE 테이블명

SET 속성명=데이터[,속성명=데이터, ]

[WHERE 조건];

-> 기본 테이블에 있는 튜플들 중에서 특정 튜플의 내용을 변경할 때 사용

111 DML – SELECT – 1

SELECT [PREDICATE] [테이블명.]속성명 [AS 별칭][, [테이블명.]속성명, ]

[, 그룹함수(속성명) [AS 별칭]]

[, Window함수 OVER (PARTITION BY 속성명1, 속성명2,

ORDER BY 속성명3, 속성명4, )]

FROM 테이블명[, 태이블명, ]

[WHERE 조건]

[GROUP BY 속성명, 속성명, ]

[HAVING 조건]

[ORDER BY 속성명[ASC | DESC]];

-> SELECT의 일반 형식

기본검색 – SELECT 절에 원하는 속성을 지정하여 검색

조건 지정 검색 – WHERE 절에 조건을 지정하여 조건에 만족하는 튜플만 검색

정렬 검색 – ORDER BY 절에 특정 속성을 지정하여 지정된 속성으로 자료를 정렬하여 검색

하위 질의 – 조건절에 주어진 질의를 먼저 수행하여 그 검색 결과를 조건절의 피연산자로 사용

복수 테이블 검색 – 여러 테이블을 대상으로 검색 수행

112 DML – SELECT – 2

WINDOW 함수 이용 검색 – GROUP BY절을 이용하지 않고 함수의 인수로 지정한 속성을 범위로 하여 속성의 값을 집계

ROW\_NUMBER() / RANK() /DENSE\_RANK()

그룹 지정 검색 – GROUP BY절에 지정한 속성을 기준으로 자료를 그룹화하여 검색

SELECT 속성명1, 속성명2,

FROM 테이블명

UNION | UNION ALL | INTERSECT | EXCEPT

SELECT 속성명1, 속성명2,

FROM 테이블명

[ORDER BY 속성명 [ASC | DESC]];

-> 집합 연산자를 이용한 통합 질의

113 DML – JOIN

- 2개의 테이블에 대해 연관된 튜플들을 결합하여 하나의 새로운 릴레이션을 반환

INNER JOIN – EQUI JOIN / NON-EQUI JOIN

OUTER JOIN – LEFT OUTER JOIN / RIGHT OUTER JOIN / FULL OUTER JOIN

INNER JOIN

EQUI JOIN

SELECT [테이블명1.]속성명,[테이블명2.]속성명,

FROM 테이블명1, 테이블명2,

WHERE 테이블명1.속성명 = 테이블명2.속셩명;

-> WHERE절을 이용한 EQUI JOIN의 표기 형식

SELECT [테이블명1.]속성명,[테이블명2.]속성명,

FROM 테이블명1 NATURAL JOIN 테이블명2;

-> NATURAL JOIN절을 이용한 EQUI JOIN의 표기 형식

SELECT [테이블명1.]속성명,[테이블명2.]속성명,

FROM 테이블명1 JOIN 테이블명2 USING(속셩명);

-> JOIN~USING절을 이용한 EQUI JOIN의 표기 형식

NON-EQUI

SELECT [테이블명1.]속성명,[테이블명2.]속성명,

FROM 테이블명1,테이블명2,

WHERE (NON-EQUI JOIN 조건);

비교 연산자 - > / < / <> / >= / <= 연산자 이용

OUTER JOIN

LEFT OUTER JOIN

SELECT [테이블명1.]속성명,[테이블명2.]속성명,

FROM 테이블명1 LEFT OUTER JOIN 테이블명2

ON 테이블명1.속성명 = 테이블명2.속성명;

SELECT [테이블명1.]속성명,[테이블명2.]속성명,

FROM 테이블명1, 테이블명2

WHERE 테이블명1.속성명 = 테이블명2.속성명(+);

RIGHT OUTER JOIN

SELECT [테이블명1.]속성명,[테이블명2.]속성명,

FROM 테이블명1 RIGHT OUTER JOIN 테이블명2

ON 테이블명1.속성명 = 테이블명2.속성명;

SELECT [테이블명1.]속성명,[테이블명2.]속성명,

FROM 테이블명1, 테이블명2

WHERE 테이블명1.속성명(+) = 테이블명2.속성명;

FULL OUTER JOIN

SELECT [테이블명1.]속성명,[테이블명2.]속성명,

FROM 테이블명1 FULL OUTER JOIN 테이블명2

ON 테이블명1.속성명 = 테이블명2.속성명;

SELF JOIN

- 같은 테이블에서 2개의 속성을 연결하여 EQUI JOIN을 하는 JOIN 방법

SELECT [별칭1.]속성명,[별칭1.]속성명,

FROM 테이블명1 [AS] 별칭1 JOIN 테이블명1 [AS] 별칭2

ON 별칭1.속성명 = 별칭2.속성명;

SELECT [별칭1.]속성명,[별칭1.]속성명,

FROM 테이블명1 [AS] 별칭1, 테이블명1 [AS] 별칭2

WHERE 별칭1.속성명 = 별칭2.속성명;

문제 정리

DECIMAL(m,n) – 10진 소수

INTEGER – 4Byte 정수

SMALLINT – 2Byte 정수

FLOAT – 부동 소수점 소수

CHAR(n) – 문자의 수가 n인 스트링

VARCHAR(n) – 문자의 수가 최대 n인 스트링

CLUSTER – 동일 인덱스 값을 갖는 튜플들을 그룹으로 묶을 때 사용

CASCADE 옵션을 사용하면 제거될 테이블을 참조하는 모든 제약과 뷰도 자동으로 스키마로부터 삭제됨

RESTRICT 옵션이 사용되면 테이블이 제약이나 뷰로부터 참조되지 않는 경우에만 삭제

SELECT절은 질의 결과에 포함된 데이터 열들을 기술하며 DB로부터 데이터 열 또는 계산 열이 될 수 있음

명령 하나로 한 개의 테이블에만 삽입시킬 수 있음

한 개의 DELETE문에는 한 개의 테이블명만 사용할 수 있음

레코드를 삭제할 때 로그를 남기는 DELETE문은 ROLLBACK 명령어로 되살릴 수 있지만 TRUNCATE문은 로그를 남기지 않아 되살릴 수 없음

테이블 생성에 대한 사용자등급 – CREATE TABLE

권한을 부여하는 사용자 앞에는 TO를 붙임

키워드 : 1. SQL 2. DDL 3. DROP .4 DCL 5. GRANT 6. DML 7. DELETE 8. SELECT 9. DISTINCT 10. JOIN

4장 SQL 활용

114 프로시저

- 절차형 SQL을 활용하여 특정 기능을 수행하는 일종의 트랜잭션 언어 / 호출을 통해 실행되어 미리 저장해 놓은 SQL 작업을 수행

프로시저의 구성도

DECLARE(필수)

BEGIN(필수)

CONTROL

SQL

EXCEPTION

TRANSACTION

END(필수)

CREATE [OR REPLACE] PROCEDURE 프로시저명(파라미터)

[지역 변수 선언]

BEGIN

프로시저 BODY;

END;

-> 프로시저 생성

EXECUTE 프로시저명;

EXEC 프로시저명;

CALL 프로시저며이

-> 프로시저 실행

DROP PROCEDURE 프로시저명;

-> 프로시저 제거

115 트리거

- 삽입, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행되는 절차형 SQL

트리거의 구성도

DECLARE(필수)

EVENT(필수)

BEGIN(필수)

CONTROL

SQL

EXCEPTION

END(필수)

CREATE [OR REPLACE] TRIGGER 트리거명 동작시기 동작 ON 테이블명

[REFERENCING NEW | OLD AS 테이블명]

[FOR EACH ROW [WHEN 조건식]]

BEGIN

프로시저 BODY;

END;

-> 트리거 생성

DROP TRIGGER 트리거명;

-> 트리거 제거

116 사용자 정의 함수

- SQL을 사용하여 일련의 작업을 연속적으로 처리 / 종료 시 처리 결과를 단일값으로 반환하는 절차형 SQL

사용자 정의 함수의 구성도

DECLARE(필수)

BEGIN(필수)

CONTROL

SQL

EXCEPTION

RETURN(필수)

END(필수)

CREATE [OR REPLACE] FUNCTION 사용자 정의 함수명(파라미터)

[지역변수 선언]

BEGIN

사용자 정의 함수 BODY;

RETURN 반환값;

END;

-> 사용자 정의 함수 생성

SELECT 사용자 정의 함수명 FROM 테이블명;

INSERT INTO 테이블명(속성명) VALUES (사용자 정의 함수명);

DELETE FROM 테이블명 WHERE 속성명 = 사용자 정의 함수명;

UPDATE 테이블명 SET 속성명 = 사용자 정의 함수명;

-> 사용자 정의 함수 실행

DROP FUNCTION 사용자 정의 함수명;

-> 사용자 정의 함수 제거

117 DBMS 접속 기술

DBMS 접속 – 사용자가 데이터를 사용하기 위해 응용 시스템을 이용하여 DBMS에 접근하는 것을 의미

응용 시스템 – 사용자로부터 매개 변수를 전달받아 SQL을 실행하고 DBMS로부터 전달받은 결과를 사용자에게 전달하는 매개체 역할을 수행

웹 응용 시스템 – 웹 서버와 웹 애플리케이션 서버(WAS)

DBMS 접속 기술 – DBMS에 접근하기 위해 사용하는 API 또는 API의 사용을 편리하게 도와주는 프레임워크을 의미

JDBC

- JAVA 언어로 다양한 종류의 데이터베이스에 접속하고 SQL문을 수행할 때 사용되는 표준 API

ODBC

- 데이터베이스에 접근하기 위한 표준 개방형 API로 개발 언어에 관계 없이 사용할 수 있음

MyBatis

- JDBC 코드를 단순화하여 사용할 수 있는 SQL Mapping 기반 오픈 소스 접속 프레임워크

동적 SQL – 개발 언어에 삽입되는 SQL 코드를 문자열 변수에 넣어 처리하는 것으로 조건에 따라 SQL 구문을 동적으로 변경하여 처리

118 SQL 테스트

- SQL이 작성 의도에 맞게 원하는 기능을 수행하는지 검증하는 과정

단문 SQL 테스트 – SQL과 TCL을 테스트하는 것

DESCRIBE 명령어로 DDL로 작성된 테이블이나 뷰의 속성, 자료형, 옵션들을 확인할 수 있음

DCL로 설정된 사용자 권한은 사용자 권한 정보가 저장된 테이블을 SELECT로 조회하거나 SHOW 명령어로 확인할 수 있음

Oracle – SELECT \* FROM DBA\_ROLE\_PRIVES WHERE GRANTEE = 사용자;

MySQL – SHOW GRANTS FOR 사용자@호스트;

절차형 SQL 테스트 – 디버깅을 통해 기능의 적합성 여부를 검증하고 실행을 통해 결과를 확인하는 테스트를 수행 SHOW ERROS

Oracle 출력 형식 – DBMS\_OUTPUT.ENABLE; : 화면에 출력하기 위해 DBMS\_OUTPUT 패키지를 불러옴

- DBMS\_OUTPUT.PUT\_LINE(데이터) : 데이터에 넣은 변수나 값을 화면에 출력

MySQL 출력 형식 – SELECT 데이터; : 데이터에 넣은 변수나 값을 화면에 출력

119 ORM

- 객체 지향 프로그래밍의 객체와 관계형 데이터베이스의 데이터를 연결하는 기술

- 가상의 객체지향 데이터베이스를 만들어 프로그래밍 코드와 데이터 연결

- 독립적이므로 재사용 및 유지보수 가능

- 직관적이고 간단하게 데이터 조작 가능

ORM 프레임워크 – ORM을 구현하기 위한 구조와 구현을 위해 필요한 여러 기능들을 제공하는 소프트웨어

JAVA – JPA, Hibernate, EclipaseLink, DataNucleus, Ebean

C++ - ODB, QxQrm

Python – Django, SQLAlchemy, Storm

IOS – DatabaseObjects, Core Data

.NET – Nhibernate, DatabaseObjects, Dapper

PHP – Doctrine, Propel, RedBean

120 쿼리 성능 최적화

- 데이터 입 출력 애플리케이션의 성능 향상을 위해 SQL 코드를 최적화하는 것

성능 측정 도구인 APM을 사용하여 최적화 할 쿼리를 선정해야 함

옵티마이저가 수립한 실행 계획을 검토하고 SQL 코드와 인덱스 재구성

실행 계획 – 옵티마이저가 수립한 SQL 코드의 실행 절차와 방법을 의미

EXPLAIN 명령어를 통해 확인 가능 / 실행 계획에는 연산 순서가 적혀있음

쿼리 성능 최적화 – 실행 계획을 참고하여 SQL문이 더 빠르고 효율적으로 작동하도록 SQL 코드와 인덱스를 재구성하는 것을 의미

SQL 코드 재구성

- WHERE 절을 추가하여 일부 레코드만 조회하도록 하여 조회에 들어가는 비용을 줄임

- WHERE 절에 연산자 사용 자제

- IN 보다 EXISTS를 활용

- 실행 계획 잘못되었다고 판단되는 경우 힌트를 활용하여 변경

인덱스 재구성

- SQL 코드에서 조회되는 속성과 조건들을 고려하여 인덱스 구성

인덱스의 추가 및 변경은 다른 SQL문에도 영향을 줄 수 있으므로 신중히 결정

- 단일 인덱스로 쓰거나 수정 없이 읽기로만 사용되는 테이블의 경우 IOT로 구성하는 것을 고려