1 SQL에서 기본 질의

01 SQL에서 기본 질의

1 기본 질의 문

- SQL은 데이터베이스로부터 정보를 검색하는 문장을 가짐

- 질의 문(Query Statement)은 다음 3개의 절(Clause)로 구성됨
  - Select 〈attribute list〉
  - From 〈table list〉
  - Where 〈condition〉

01 SQL에서 기본 질의

1 기본 질의 문

- Where절에 조인 조건을 기술할 수 있음
  - 테이블들끼리의 연결은 외래키를 이용함

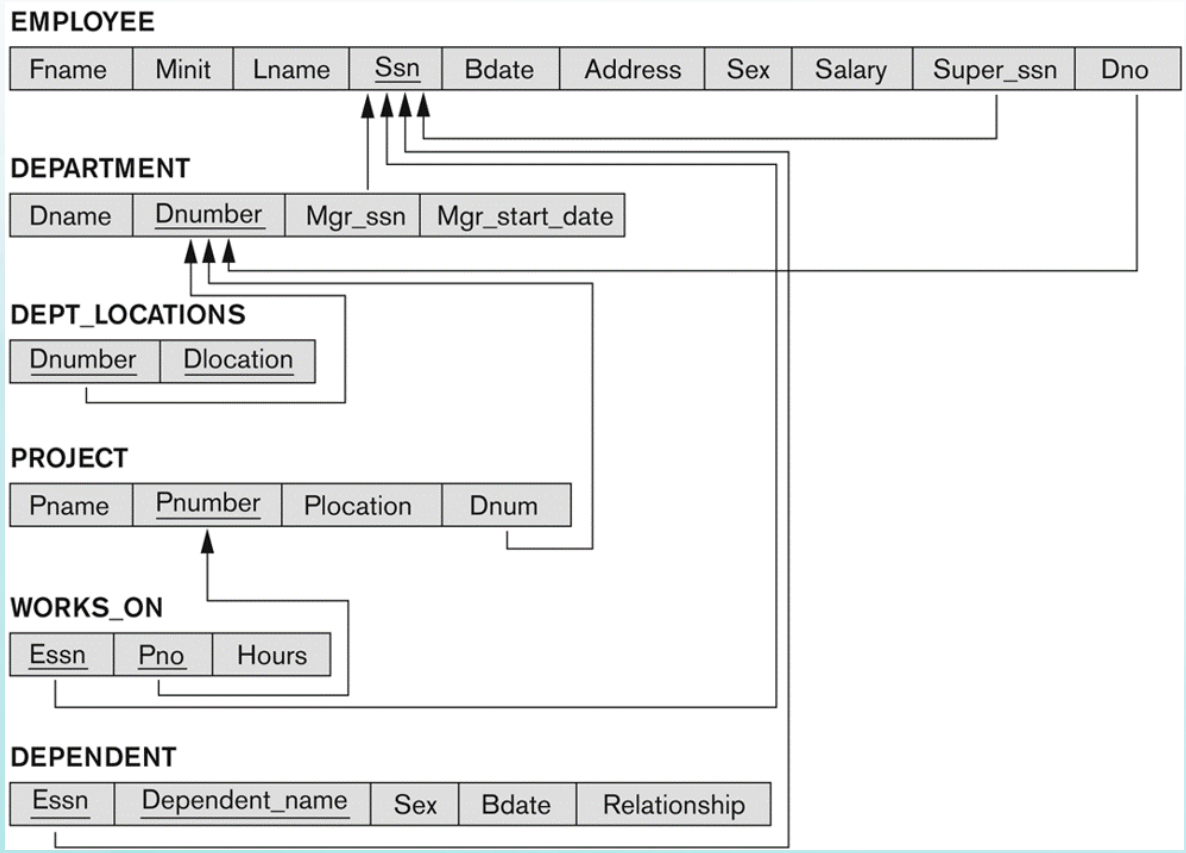
- 관계 대수와 SQL의 중요한 차이점
  - 관계 대수의 결과 릴레이션은 일종의 집합이기 때문에 중복 투플을 가질 수 없지만
  - SQL의 결과 테이블은 중복 투플을 가질 수 있음
  - 중복 투플을 가질 수 있기 때문에 집합이 아니라 다중집합(Multi-set 또는 bag) 임
  - DISTINCT 를 사용하여 중복을 제거할 수 있음

01 SQL에서 기본 질의

1 기본 질의 문

질의 예제를 위한
Company RDB 스키마

[COMPANY 관계
데이터베이스 스키마]



※ 출처 : 데이터베이스 시스템 6판, Elmasri, Navathe 저, 황규영 외 역, 홍릉과학출판사, 2016년

01 SQL에서 기본 질의

1 기본 질의 문

예제 Q0)

이름(Fname)이 'John' 인 사원의 생일과 주소를 검색하시오

```
SELECT   bdate, address
FROM     EMPLOYEE
WHERE    fname = 'John' ;
```

- 관계대수 연산의 SELECT-PROJECT 쌍과 유사함
- SELECT 절은 선택되는 속성들을 표시하고, WHERE절은 행(투플) 선택 조건을 표시함
- 질의 결과 테이블이 중복 투플을 포함할 수 있음

01 SQL에서 기본 질의

1 기본 질의 문

예제 Q1)

'Research' 부서에서 일하는
종업원들의 이름과 주소를 검색하시오

```
SELECT   FNAME, LNAME, ADDRESS
FROM     DEPARTMENT, EMPLOYEE
WHERE    DNAME='Research' AND DNUMBER=DNO ;
```

- 관계대수 연산의 SELECT-JOIN-PROJECT 과 유사함
- (DNAME='Research')은 행(투플) 선택 조건이고 관계대수에서 SELECT 연산에 해당함
- (DNUMBER=DNO)은 조인조건이고 관계대수의 JOIN 연산에 해당함

01 SQL에서 기본 질의

1 기본 질의 문

예제 Q2)

위치 'Stafford' 에서 진행되는 프로젝트의 번호(PNUMBER), 담당부서 번호(DNUM), 부서 관리자의 성(LNAME), 주소(ADDRESS), 생일(BDATE)을 검색하시오

```
SELECT   PNUMBER, DNUM, LNAME, BDATE, ADDRESS
FROM     PROJECT, DEPARTMENT, EMPLOYEE
WHERE    PLOCATION='Stafford' AND
         DNUM=DNUMBER AND  MGRSSN=SSN ;
```

- 조인 조건 DNUM=DNUMBER는 프로젝트(PROEJCT)와 담당 부서(DEPARTMENT)를 조인하고, MGRSSN=SSN은 담당 부서(DEPARTMENT)와 그 관리자(EMPLOYEE)를 조인함

01 SQL에서 기본 질의

2 alias, *, DISTINCT, WHERE 절의 생략

- (테이블 명).(속성 명)
  : 서로 다른 테이블이 동일한 속성 명을 가질 수 있음,
  이 경우 테이블 명과 함께 속성 명을 사용함으로써
  모호함을 방지해야 함

- 테이블 명이 너무 길 때 표시하기가
  불편하기 때문에 별명(alias)을 사용함,
  (테이블 명).(속성 명) 대신 (별명).(속성 명)을
  사용할 수 있음

- SELECT 절에서 * 을 사용하면
  결과 테이블에 모든 속성들을 표시함

01 SQL에서 기본 질의

2 alias, *, DISTINCT, WHERE 절의 생략

- 결과 테이블에서 중복된 투플을 제거하고자 할 때는 SELECT절에서 DISTINCT 를 사용하면 됨

- WHERE 절을 생략하면 투플 선택에 대한 조건이 없다는 것을 의미함, 즉, FROM 절에 있는 테이블의 모든 투플이 선택됨

01 SQL에서 기본 질의

2 alias, *, DISTINCT, WHERE 절의 생략

예제)

질의 예제를 위한 Sailor RDB 스키마

Sailor(sid, sname, rating ,age) PK={sid}
Boat(bid, bname, color) PK={bid}
Res(sid, bid) PK={sid, bid}, FK={sid}, {bid}

01 SQL에서 기본 질의

2 alias, *, DISTINCT, WHERE 절의 생략

예제 S0) alias 의 예

300번 배를 승선 예약한 선원들의 이름을 조회하시오

```
select   s.sname
from     Res r, Sailor s
where    r.bid = 300 and r.sid = s.sid ;
```

01 SQL에서 기본 질의

2 alias, *, DISTINCT, WHERE 절의 생략

예제 S1) * 의 예

100번 배의 모든 정보를 조회하시오

```
select *
from Boat
where bid = 100 ;
```

01 SQL에서 기본 질의

2 alias, *, DISTINCT, WHERE 절의 생략

예제 S2) DISTINCT 의 예

선원들의 이름을 조회하시오
단, 같은 이름이 여러 개 있으면 한 번만 출력하시오

```
select DISTINCT(sname)
from Sailor ;
```

01 SQL에서 기본 질의

2 alias, *, DISTINCT, WHERE 절의 생략

예제 S3) WHERE 절 생략의 예
선원들의 모든 정보를 조회하시오

```
select *
from Sailor ;
```

01 SQL에서 기본 질의

3 질의 문의 집합 연산

- SQL은 일부 집합 연산들을 수용함
- SQL에서는 합집합(UNION) 연산, 차집합(EXCEPT) 연산, 교집합(INTERSECT) 연산을 제공함
- 테이블에 대한 집합 연산의 결과는 투플들의 집합임, 즉, 중복된 투플은 결과에서 제거됨
- 집합 연산들은 합집합 호환성을 갖는 테이블에만 적용, 즉, 두 개의 테이블은 동일한 속성들을 가지며 이 속성들은 같은 순서로 나타나야 함

01 SQL에서 기본 질의

4 중첩 질의(Nested Query)

- 질의는 중첩될 수 있음
- 보통 Where절에
  또 다른 완전한 조회문이 올 수 있음
- IN, NOT IN, EXISTS, NOT EXISTS, ANY, ALL 등의
  연산 키워드가 있음
- From 절도 중첩 질의를 포함할 수 있음

01 SQL에서 기본 질의

4 중첩 질의(Nested Query)

예제 Q3)

Research 부서에서
근무하는 사원의 이름과 주소를 검색하시오

```
SELECT      FNAME, LNAME, ADDRESS
FROM        EMPLOYEE
WHERE       DNO IN  (
                SELECT  DNUMBER
                FROM    DEPARTMENT
                WHERE   DNAME='Research' ) ;
```

01 SQL에서 기본 질의

5 SQL 질의에서의 NULL

- NULL 은 알려지지 않는 값,
  이용할 수 없는 값, 적용할 수 없는 값을 의미

- 속성 값이 NULL 인지 검사하는 연산들이 있음,
  IS 또는 IS NOT 을 사용

- 예제 Q4)
  감독관이 없는 모든 종업원들의 이름을 검색하시오

```
SELECT    FNAME, LNAME
FROM      EMPLOYEE
WHERE     SUPERSSN IS NULL ;
```

01 SQL에서 기본 질의

6 조인 조건의 또 다른 형태

- 보통 Where 절에 조인조건을 명시함

- 새로운 SQL 표준에서는
  From 절에 명시하는 형태를 제안,
  또한 여러 가지 유형의 조인을 명시할 수 있도록 함
  - JOIN, NATURAL JOIN, LEFT OUTER JOIN,
    RIGHT OUTER JOIN, CROSS JOIN

- 예제 Q5)

```
SELECT   FNAME, LNAME, ADDRESS
FROM     (EMP JOIN DEPT ON DNUMBER=DNO)
WHERE    DNAME='Research ;
```

01 SQL에서 기본 질의

7 집단 함수(Aggregate Function)

- SQL에서는 COUNT, SUM, MAX, MIN, AVG 등을 포함

- 예제 Q6)
  사원들 봉급의 합과 평균 봉급을 구하시오

```
SELECT  SUM(SALARY), AVG(SALARY)
FROM  EMP ;
```

- 예제 S4)

```
Select count(*)
From Sailor ;
```

01 SQL에서 기본 질의

8 그룹화

- 많은 경우에, 테이블 내에 있는 투플들을 그룹화 속성을 기준으로 여러 그룹으로 나눌 수 있음
- 각 그룹 별로 독립적으로 집단 함수들을 적용할 수 있음
- SQL은 SELECT 절에 나타나는 속성들 중에서 그룹화 속성을 GROUP BY절에 명시하도록 함

01 SQL에서 기본 질의

8 그룹화

예제 Q7)

각 부서별 부서번호, 부서 내 사원 수, 평균 봉급을 검색하시오

```
SELECT  DNO, COUNT(*), AVG(SALARY)
FROM  EMPLOYEE
GROUP BY  DNO ;
```

- EMPLOYEE 투플들을 그룹화 속성 DNO 값이 같은 것끼리 모아서 여러 그룹으로 나눔
- 각 그룹의 투플들에 대하여 COUNT와 AVG 함수를 적용함
- SELECT 절은 그룹화 속성과 각 그룹에 속하는 투플들에 적용할 집단 함수들만 포함

01 SQL에서 기본 질의

9 그룹 조건을 명시하는 Having

- 때때로 어떤 조건들을 만족하는 그룹들에 대해서만 집단함수들을 적용하기도 함
- HAVING절은 집단함수를 적용할 그룹들을 선택하는 데 사용됨

01 SQL에서 기본 질의

9 그룹 조건을 명시하는 Having

예제 Q8)

두 명 이상의 사원이 근무하는 각 프로젝트 별로 프로젝트 번호, 프로젝트 이름, 프로젝트에서 근무하는 사원의 수를 검색하시오

```
SELECT   PNUMBER, PNAME, COUNT(*)
FROM     PROJECT, WORKS_ON
WHERE  PNUMBER=PNO
GROUP BY    PNUMBER, PNAME
HAVING        COUNT(*) > 2 ;
```

01 SQL에서 기본 질의

10 그룹 조건을 명시하는 Having

- LIKE 비교 연산자는
  문자열의 일부에 대해 비교 조건을 명시

- 부분 문자열은 두 개의 예약 문자를 사용
  - '%'(또는 '*')은 0보다 큰 임의 개수의 문자로 대체
  - '-'는 임의의 한 개 문자로 대체

01 SQL에서 기본 질의

10 그룹 조건을 명시하는 Having

예제 Q9)

주소가 'Houston, Texas'인 사원을 검색하시오

```
SELECT   *
FROM     EMPLOYEE
WHERE    ADDRESS LIKE '%Houston,TX%' ;
```

01 SQL에서 기본 질의

11 Order By 절

- ORDER BY 절은 하나 이상의 속성을 기준으로 결과 투플들을 정렬할 수 있음

- 예제 Q10)

  사원이 수행하는 프로젝트들을 검색하는 데, 프로젝트 이름 순서대로, 그리고 각 프로젝트 내에서는 사원 이름 순서대로 구하시오

```
SELECT   PNAME, LNAME
FROM     EMP, WORKS_ON, PROJECT
WHERE    SSN=ESSN AND  PNO=PNUMBER
ORDER BY  PNAME, LNAME ;
```

01 SQL에서 기본 질의

11 Order By 절

- 디폴트 정렬 순서는 오름차순임
- 내림차순으로 정렬하고자 한다면 키워드 DESC로 지정
- 키워드 ASC 는 오름차순 정렬을
  명시적으로 지정할 때 사용함
- 예제)

  ORDER BY PNAME **DESC**, LNAME **ASC** ;

01 SQL에서 기본 질의

12 SQL 질의 문 요약

- SQL에서 하나의 질의는 6개의 절로 구성
- 필수적으로 질의에 나타내야 하는
  두 개의 절은 SELECT 와 FROM 절임
- 6개의 절은 다음 순서로 명시함
  - SELECT <attribute list>
    FROM <table list>
    [WHERE <condition>]
    [GROUP BY <grouping attribute(s)>]
    [HAVING <group condition>]
    [ORDER BY <attribute list>] ;

01 SQL에서 기본 질의

12 SQL 질의 문 요약

- SELECT 절은 결과에 포함될 속성들이나 함수를 나열함
- FROM 절은 질의에서 필요한 모든 테이블들을 명시함
- WHERE 절은 조인 조건을 포함하여 FROM 절에 명시된 테이블로부터 투플들을 선택하기 위한 조건들을 명시
- GROUP BY 절은 그룹화 속성들을 명시
- HAVING 절은 그룹들에 대한 조건을 명시

01 SQL에서 기본 질의

12 SQL 질의 문 요약

- ORDER BY 절은 질의 결과를 출력하는 순서를 명시
- 질의는 WHERE절, GROUP BY 절과 HAVING 절의 순서로 적용함으로써 평가됨

2 SQL에서 삽입, 삭제, 갱신문

02 SQL에서 삽입, 삭제, 갱신문

1 삽입 문(Insert)

- 테이블에 투플을 추가하는 데 사용
- 속성 값들의 순서는 CREATE TABLE 명령에서 명시한 속성들의 순서와 같아야 함
- 질의로 검색되는 결과를 삽입할 수도 있음

02 SQL에서 삽입, 삭제, 갱신문

1 삽입 문(Insert)

예제) 단순삽입

```
INSERT  INTO  Sailor(sid, sname, rating, age) VALUES
(1, '홍길동', 1, 52 ) ;
```

예제) 질의의 결과를 삽입

```
INSERT INTO  DEPTS_INFO
SELECT  DNAME, COUNT(*), SUM(SALARY)
FROM    DEPT, EMP
WHERE  DNUMBER=DNO
GROUP BY  DNAME ;
```

02 SQL에서 삽입, 삭제, 갱신문

2 삭제 문(Delete)

- 릴레이션에서 투플들을 제거하는 명령
- 삭제할 투플들의 조건을 나타내는 WHERE절을 포함함
- 한 번에 한 테이블 내의 투플들만 삭제함
- WHERE 절을 생략한 경우에는 테이블 내의 모든 투플을 삭제하고 테이블은 빈 테이블로 남게 됨

02 SQL에서 삽입, 삭제, 갱신문

2 삭제 문(Delete)

예제)

```
DELETE FROM EMP WHERE SSN='123456789' ;
```

예제)

```
DELETE FROM EMP
WHERE DNO  IN (
      SELECT  DNUMBER
      FROM    DEPT
      WHERE  DNAME='Research') ;
```

예제)

```
DELETE FROM  EMP ;
```

02 SQL에서 삽입, 삭제, 갱신문

3 갱신 문(Update)

- 투플의 속성 값을 수정하기 위해 사용
- WHERE 절은 테이블에서 수정할 투플들을 선택하는데 사용됨
- SET 절은 변경할 속성과 그들의 새로운 값을 명시함
- UPDATE 명령은 같은 테이블 내에서 여러 투플을 수정할 수 있음

02 SQL에서 삽입, 삭제, 갱신문

3 갱신 문(Update)

예제)

10번(PNUMBER=10) 프로젝트의 PLOCATION 을 'Bellaire'로 변경하고 담당부서를 5번 부서(DNUM=5)로 변경하시오

```
UPDATE  PROJECT
SET     PLOCATION = 'Bellaire', DNUM = 5
WHERE   PNUMBER=10 ;
```

02 SQL에서 삽입, 삭제, 갱신문

3 갱신 문(Update)

예제)

'Research' 부서에 있는 사원들의
봉급을 10% 인상하시오

```
UPDATE   EMP
SET      SALARY = SALARY *1.1
WHERE    DNO  IN (
         SELECT   DNUMBER
         FROM     DEPT
         WHERE  DNAME='Research') ;
```

3 SQL에서 뷰의 구성

03 SQL에서 뷰의 구성

1 뷰의 개념

- SQL에서 뷰는 다른 테이블들에서 유도되는 가상 테이블
- 뷰에 적용할 수 있는 갱신 연산들은 제한되지만 질의 연산은 아무런 제한을 받지 않음
- 물리적으로 존재하지 않더라도 자주 참조할 필요가 있는 테이블을 명시하는 한 가지 방법

03 SQL에서 뷰의 구성

1 뷰의 개념

- 다수의 테이블을 조인하는 질의를 사용하는 대신에, 자주 검색하고자 하는 속성들을 포함하고 있는 조인의 결과를 미리 뷰로 정의할 수 있음

- 그러면 다수의 테이블을 조인하는 질의 대신에 뷰로 구성된 하나의 테이블을 검색함으로써 원하는 속성을 검색할 수 있음

03 SQL에서 뷰의 구성

2 뷰의 명시

- 뷰를 정의하는 명령 : CREATE VIEW
- 뷰의 정의는
  - 뷰 이름,
  - 속성 이름의 리스트,
  - 뷰의 내용을 나타내는 질의 로 구성됨

03 SQL에서 뷰의 구성

2 뷰의 명시

🔎 예제) WORKS_ON1 뷰 생성

```
CREATE VIEW  WORKS_ON1(FN, LN, PN, H)
AS
SELECT    FNAME, LNAME, PNAME, HOURS
FROM      EMPLOYEE, WORKS_ON, PROJECT
WHERE   SSN=ESSN AND PNO=PNUMBER ;
```

03 SQL에서 뷰의 구성

2 뷰의 명시

- 새로 생성된 뷰 를 사용해서 질의를 수행

```
SELECT    *
FROM      WORKS_ON1
WHERE     PNAME = 'Project-X' ;
```

- 기본 테이블들을 사용해서
  위의 질의를 나타내려면 두 번의 조인이 필요함

- 뷰의 주요 장점은 질의들을
  간단하게 작성할 수 있다는 것임

03 SQL에서 뷰의 구성

2 뷰의 명시

- 뷰는 항상 최신 정보를 반영해야 하므로
  만약, 뷰의 정의에 사용된 테이블이 수정되면
  뷰는 자동적으로 변경사항들을 반영해야 함

- 따라서 뷰는 뷰의 정의 시점이 아니라
  뷰에 대해 질의를 할 때 구체화 됨

- 뷰를 최신 정보로 유지하는 것은 사용자가 아니라
  DBMS의 책임

- 어떤 뷰가 더 이상 필요하지 않으면 DROP VIEW 를
  사용하여 뷰를 제거함, DROP VIEW WORKS_ON1

03 SQL에서 뷰의 구성

3 효율적인 뷰 구현

- 질의를 위해 뷰를 효율적으로 구현하는 문제는 복잡하고 어려운 문제임, 두 가지 접근방식이 있음

- 질의 수정(Query modification) 접근법
  - 뷰에 대한 질의를 기본 테이블들에 대한 질의로 변환하여 처리
  - 문제점 : 복잡한 질의로 정의된 뷰는 특히 짧은 시간 내에 뷰에 많은 질의가 적용될 때 비효율적임

03 SQL에서 뷰의 구성

3 효율적인 뷰 구현

- 뷰 실체화(View materialization) 접근법
  - 뷰에 대해 처음 질의 요구가 있을 때 임시 뷰 테이블을 물리적으로 생성하고 뷰에 대한 다른 질의들이 이어질 것이라는 가정하에 테이블을 유지하는 방식
  - 문제점 : 뷰를 최신 상태로 유지하기 위해서 기본 테이블이 갱신되면 뷰 테이블도 변경되어야 함
  - 일정 기간 동안 뷰에 질의가 없으면 물리적 뷰 테이블을 삭제함
  - 다음에 그 뷰를 참조하는 질의가 있을때 처음부터 뷰 테이블을 재구성함

03 SQL에서 뷰의 구성

4 뷰의 갱신

- 뷰에 대한 갱신은 기본 테이블에 대한 갱신으로 모호함이 없이 대응시킬 수 있을 때만 가능

- 갱신할 수 있는 뷰
  - 집단함수를 사용하지 않는 단일 뷰
  - 조인을 포함하는 뷰 중에 기본 테이블들에 대한 여러 개의 갱신으로 대응될 수 있는 뷰

03 SQL에서 뷰의 구성

4 뷰의 갱신

- 갱신할 수 없는 뷰
  - 그룹화와 집단함수를 사용하여 정의된 뷰
  - 일반적으로 다수의 테이블을 조인하여 정의된 뷰

- SQL에서 뷰의 갱신을 가능하게 하려면 뷰 정의의 끝에 WITH CHECK OPTION 절을 추가해야 함
  - 이렇게 하면 DBMS가 뷰를 갱신할 수 있는가를 검사할 수 있고, 뷰의 갱신을 위한 실행 계획을 찾아낼 수 있음

4 상용 DBMS 상에서 작성 실습

04 상용 DBMS 상에서 작성 실습

1 실습 참고 사이트

- Oracle 사이트 https://www.oracle.com/kr/
- MySQL 사이트 https://dev.mysql.com/
- MariaDB 사이트 https://mariadb.org/
- PostgreSQL 사이트 https://www.postgresql.org/

- DBA 커뮤니티 구루비 http://www.gurubee.net/
- DB 사랑 넷 http://database.sarang.net/

- 소프트캠퍼스 오라클(Oracle) 데이터 베이스11g youtube.com
- 한빛미디어 이것이 MySQL이다 youtube.com