

9장 SQL (Structured Query Language)



9장 SQL (Structured Query Language)

- **SQL(Structured Query Language)**
 - 데이터베이스에 접근할 때 사용하는 언어
 - 크게 DCL, DML과 DDL로 나눌 수 있다.



9.1 DML(데이터 조작용어)



9.1 DML(데이터 조작용어)

- **SELECT 문**

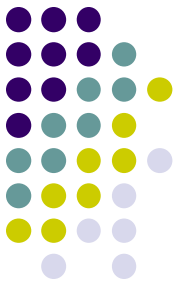
- SQL에서 데이터 검색

SELECT
FROM

[DISTINCT | ALL]추출할_애틀리뷰트
추출에_필요한_테이블

[WHERE
GROUP BY
HAVING
ORDER BY

추출하기_위한_조건
그룹별로_뭉을_기준이_되는_애틀리뷰트
그룹이_만족해야_할_조건
추출한_데이터의_정렬]



9.1 DML(데이터 조작어)

- **DISTINCT:중복되는 컬럼의 값 중 하나씩만 추출**

```
SELECT DISTINCT JOBID  
FROM EMP;
```

```
SELECT DISTINCT JOBID,DEPTID  
FROM EMP;
```

```
SELECT ALL JOBID  
FROM EMP;
```

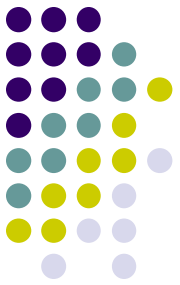


9.1 DML(데이터 조작어)

- **ORDER BY : 정렬**

```
SELECT *  
FROM EMP  
ORDER BY SALARY [ASC | DESC];
```

```
SELECT EMPID, FNAME  
FROM EMP  
ORDER BY HDATE; //ASC, 빠른날이 먼저 출력
```



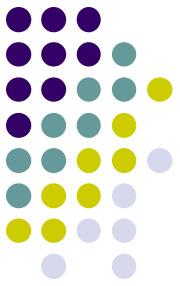
9.1 DML(데이터 조작어)

- **ORDER BY : 정렬**

EX)

1. 부서코드가 빠른 사람 순으로 사번,이름,부서번호
출력

2. SELECT *
FROM EMP
ORDER BY EMPID,FNAME;



9.1 DML(데이터 조작어)

- **ALIAS (as): 별칭, AS 생략가능, 한글가능**

```
SELECT EMPID AS "사원번호"  
FROM EMP;
```

```
SELECT EMPID "사원번호"  
FROM EMP;
```




9.1 DML(데이터 조작어)

- 연산자 : 산술연산자,논리연산자,NOT

1. select EMPID, SALARY, SALARY*1.1
from emp;

2. select *
from emp
where DEPTID=30 and SALARY > 2000;

Ex) 사번,이름,급여,커미션, 급여+커미션 출력



9.1 DML(데이터 조작어)

- 연산자 : **in,between,is,like**

1. select EMPID, FNAME, SALARY
from emp
where SALARY between 1000 and 2000;

2. select EMPID,FNAME
from emp
Where EMPID in (129,139,149);

3. select EMPID,FNAME
from emp
where FNAME like 'A%';



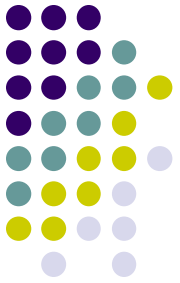
9.1 DML(데이터 조작용어)

- 연산자 : **in,between,is,like**

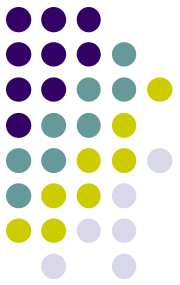
1. select EMPID,FNAME
from emp
where FNAME like 'MILLE_'; //한글자

2. select EMPID,FNAME
from emp
Where comm is null;

3. select EMPID,FNAME
from emp
where comm is not null;



9.2 FUNCTION(함수)



9.2 함수(function)

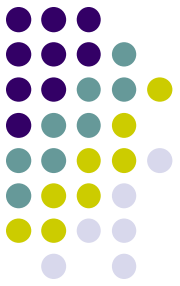
- **함수(function)**
 - 특정 기능을 수행
 - Data에 대한 계산을 수행
 - Data 를 다른 형태로 변환



9.2 집계함수(function)

- count,sum,min,max,avg,concat...

1. `select count(empid) from emp;`
2. `select max(empid) from emp;`
3. `select concat(concat(fname,' is a '), JOBID) from emp;`
4. `select lpad(fname,10,'*') from emp;`
5. `select substr(fname,3,2) from emp;`



9.2 함수(function)

1. `select round (15.193,1) from emp;`
2. `select floor(15.7) from emp;`
3. `select fname, to_char(HDATE,'yyyy/mm/dd') from emp;`
4. `select fname,to_char(SALARY,'$999,999') from emp;`



9.2 함수(function)

1. `select avg(SALARY), deptid from emp;`
2. `select avg(SALARY), deptid from emp group by deptid;`



9.2 함수(function)

3. `select deptid, avg(SALARY) from emp
where avg(SALARY) > 2500 group by deptid;`

4. `select DEPTID, avg(SALARY) from emp
group by DEPTID
having avg(SALARY) > 2500;`

Ex) 부서별 인원이 두 명 이상인 부서의 부서번호, 인원수,
급여총액을 출력



9.1 DML(데이터 조작어)

- **INSERT 문**

- 테이블에 새로운 튜플 삽입
- 자식 테이블의 외래키 칼럼에는 부모 테이블의 칼럼의 데이터들 중 하나가 입력되어야만 함
- INSERT INTO TABLE_NAME(COLUMN_NAMES...) VALUES (VALUES...)



9.1 DML(데이터 조작용어)

- **DELETE 문**

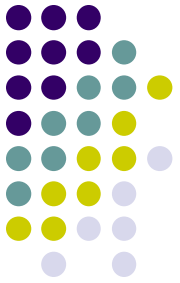
- 테이블에서 데이터 삭제
- Delete FROM 절과 WHERE 절로 이루어 짐
- DELETE FROM TABLE_NAME;
- CF) DROP TABLE TABLE_NAME;



9.1 DML(데이터 조작어)

- **UPDATE 문**

- 테이블의 행 값을 수정할 때 사용
- 외래키나 기본키일 때 참조 무결성 제약조건을 위반하면 오류 발생
- UPDATE TABLE_NAME
SET COLUMN_NAME = 'NEW VALUE'
WHERE CONDITION;



9.3 DDL(데이터 정의어)



9.3 DDL(데이터 정의어)

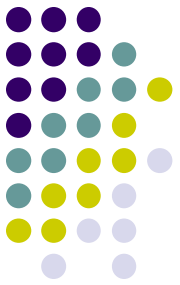
- **DDL**

- CREATE table

- : 테이블 생성

- : 테이블 이름, 칼럼 이름, 칼럼의 데이터 타입과 크기 반드시 명시해야 함

- create table test(id INT(3),
name varchar(10) not null);



9.3 DDL(데이터 정의어)

• 데이터 타입

| 데이터 타입 | 정의 |
|---------------|-------------------------------|
| VARCHAR(size) | 가변 길이의 문자열 타입 (1-255) |
| CHAR(size) | 고정 길이의 문자열 타입(1-65535) |
| DECIMAL(p, s) | 길이가 p인 숫자에서 소수 아래 숫자가 s개인 십진수 |
| DATE | 날짜와 시간 |
| INT(INTEGER) | -21억 ~ 21억 |
| LONGTEXT | 최대 4GB의 문자 데이터 타입 |
| FLOAT | 소수점 7자리까지 표현 |
| BLOB | 4GB이상의 이진 데이터 타입 |
| REAL, DOUBLE | 소수점 15자리까지 표현 |
| JSON | JSON 형식의 데이터 |



9.3 DDL(데이터 정의어)

- **DDL**

- ALTER table

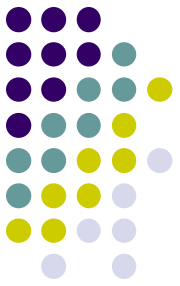
- : 테이블 생성 후 테이블의 구조 변경

- : 칼럼을 추가(ADD), 수정(MODIFY), 삭제(DROP), 새로운 칼럼의 값 정의 등을 할 수 있음

- alter table test add tel varchar2(10);

- alter table test modify tel varchar2(13);

- alter table test drop tel;



9.3 DDL(데이터 정의어)

- **DDL**

- DROP table : 테이블 삭제

- drop table test;

cf)truncate (DDL) /delete (DML)

Truncate table table_name;



9.4 제약조건

• 제약조건

- 테이블 수준에서 이루어지는 작업에 대한 규칙
- 모든 제약조건은 데이터 카탈로그에 저장

| 제약조건 | 의미 |
|-------------|---|
| NOT NULL | NULL 값이 입력될 수 없음을 명시한다. |
| UNIQUE | 테이블 내의 모든 행은 컬럼 또는 컬럼 조합의 값이 반드시 유일해야 한다. Null 허용 |
| PRIMARY KEY | 테이블에서 각 행을 식별할 수 있는 특별한 칼럼이다. Null 불가 |
| FOREIGN KEY | 어떠한 칼럼과 그 칼럼에 관계를 가지고 있는 테이블의 칼럼이다. |
| CHECK | 조건을 명시하여 참인 경우에만 값을 받는다. |



9.4 제약조건

- **NOT NULL**

- 널이 칼럼에 삽입되는 것을 막음

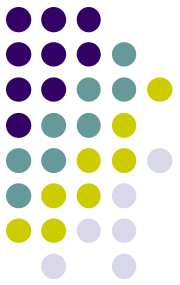
| 학생 | | | |
|-----|------------|-----|---------------|
| 이름 | 학번 | ... | 핸드폰번호 |
| 곽나리 | 2006150004 | ... | 010-9763-6639 |
| 김승현 | 2007140013 | ... | 010-3345-4520 |
| 이건희 | 2008130032 | ... | (null) |
| 이민구 | 2006130030 | ... | 010-2772-3640 |
| 정유나 | 2007140045 | ... | 010-7756-1124 |
| 하정헌 | 2005150048 | ... | (null) |
| 한우경 | 2008130050 | ... | (null) |



9.4 제약조건

- **UNIQUE : null 허용**
 - 한 테이블의 두 개의 행이 중복되지 않음

| 학생 | | | |
|--------------|-----------------|-----|----------|
| 이름 | email | ... | 학과 |
| 곽나리 | nari@db.co.kr | ... | 컴퓨터공학과 |
| 김승현 | seung@db.co.kr | ... | 산업디자인공학과 |
| 이건희 | gun@db.co.kr | ... | 전자공학과 |
| 이한솔 | hansol@db.co.kr | ... | 신소재공학과 |
| 행 삽입 | ... | ... | ... |
| 김남수 삽입 성공 | nams00@db.co.kr | ... | 컴퓨터공학과 |
| 김남수 삽입 실패 | nams00@db.co.kr | ... | 전자공학과 |



9.4 제약조건

- **PRIMARY KEY : null 불가**
 - 테이블에는 유일한기본키 값이 존재해야 함

| 학생 | | | |
|-----|------------|-----|---------------|
| 이름 | 학번 | ... | 핸드폰번호 |
| 곽나리 | 2006150004 | ... | 010-9763-6639 |
| 김승현 | 2007140013 | ... | 010-3345-4520 |
| 이건희 | 2008130032 | ... | (null) |
| ... | ... | ... | ... |

행 삽입

| | | | |
|-----|--------|-----|---------------|
| 이민구 | (null) | ... | 010-3066-3140 |
|-----|--------|-----|---------------|

삽입 실패

| | | | |
|-----|------------|-----|---------------|
| 이영재 | 2008130039 | ... | 010-7026-0012 |
|-----|------------|-----|---------------|

삽입 성공



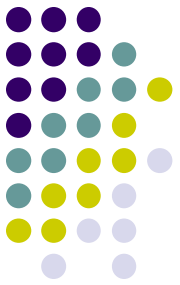
9.4 제약조건

- **FOREIGN KEY**

- 같은 테이블이나 다른 테이블과 관계를 맺을 때,
그 테이블의 PRIMARY KEY나 UNIQUE KEY 간에 성립

PRIMARY KEY 제약조건

| 교수 | | | |
|-----|-----|-----|--------|
| ID | 이름 | ... | 연구실 |
| 100 | 오용철 | ... | 504 |
| 101 | 이상혁 | ... | 306 |
| 102 | 김진환 | ... | (null) |
| 103 | 윤미래 | ... | 207 |



9.4 제약조건

- FOREIGN KEY

FOREIGN KEY 제약조건

| 학생 | | | |
|-----|------------|-----|---------|
| 이름 | 학번 | ... | 지도교수_ID |
| 곽나리 | 2006150004 | ... | 100 |
| 김승현 | 2007140013 | ... | 101 |
| 이건희 | 2008130032 | ... | 102 |
| ... | ... | ... | ... |

행 삽입

| | | | |
|-----|------------|-----|----|
| 이영재 | 2008130036 | ... | 12 |
|-----|------------|-----|----|

삽입 실패

| | | | |
|-----|------------|-----|-----|
| 이영재 | 2008130036 | ... | 102 |
|-----|------------|-----|-----|

삽입 성공

그림 (Ch09-003)



9.4 제약조건

- **CHECK**

- 칼럼에 대해 값의 범위를 지정함으로써, 범위 밖의 값은 삽입을 제한
- `payment varchar2(12) not null`
`constraint check_payment check(`
`payment in ('credit card','point','money wire'));`