SQL 기본

SQL의 개요

- SQL 이란?
  - Structured Query Language (구조적 쿼리언어) 의 약자 'Sequel'로 읽음
  - 데이터베이스와의 통신을 위해서 특별히 고안된 언어
- 장점
  - 특정 데이터베이스 회사에만 사용하는 점유물이 아님
  - 대부분의 DBMS에서 SQL을 지원함.
  - SQL 을 알면 모든 데이터 베이스에서 활용할 수 있음
  - 배우기 쉬움
  - 간단하지만 매우 강력한 언어
  - 효과적으로 각 요소를 사용하면 매우 복잡하고 정교한 데이터 베이스 작업을 수행할 수 있음.
- 기타
  - 표준 SQL은 ANSI 표준위원회에서 관리, ANSI-SQL 이라고 부름

SQL 문장의 종류

명령어 종류	명령어	설명
데이터조작어 (Data Manipulation Language)	select	데이터 베이스에 들어있는 데이터를 조회, 검색
	Insert Update delete	테이블에 들어있는 데이터에 변형을 가함 새로운 행을 입력, 행을 변경, 행을 삭제
데이터정의어 (Data Definition Language)	Create Alter Drop Rename	테이블 같은 데이터 구조를 정의 구조를 생성, 변경, 삭제, 이름 변경
데이터제어어 (Data Control Language)	GRANT REVOKE	데이터베이스에 접근하고 객체를 사용하도록 권한을 주고 회수
트랜젝션 제어어 (Transaction Control Language)	COMMIT ROLLBACK	논리적 작업의 단위를 모아서 DML에 의해 조작된 결과를 작업단위(트랜잭션)별로 제어

Normalization – 테이블의 정규화

- Primary key 와 Foreign key

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	30
.....						
7698	BLAKE	MANAGER		01-MAY-81	3850	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	10

PK FK

DEPTNO	DNAME	LOC
10	STORE	CHICAGO
20	RESEARCH	DALLAS
30	SALES	NEW YORK
40	MARKETING	BOSTON

PK

Sqlite3 Datatype

Storage Class	Meaning
NULL	NULL values mean missing information or unknown.
INTEGER	Integer values are whole numbers (either positive or negative). An int eger can have variable sizes such as 1, 2,3, 4, or 8 bytes.
REAL	Real values are real numbers with decimal values that use 8-byte floa ts.
TEXT	TEXT is used to store character data. The maximum length of TEXT is unlimited. SQLite supports various character encodings.
BLOB	BLOB stands for a binary large object that can store any kind of data. The maximum size of BLOB is, theoretically, unlimited.

DDL SQL의 실습

➢ 테이블 생성

```
create table EMP (
    EMPNO number(4) not null,
    ENAME varchar2(30) not null,
    JOB varchar2(10),
    MGR number(4),
    HIREDATE date,
    SAL number(7,2),
    DEPTNO number(2),
    PRIMARY KEY  (EMPNO));
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	30
.....						
7698	BLAKE	MANAGER		01-MAY-81	3850	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	10

EMP 테이블

DDL SQL의 실습

- 아래와 같은 테이블을 생성해 보세요.
  - DEPTNO : number(4)
  - DNAME:char (10)
  - LOC: varchar2(30)

DEPTNO	DNAME	LOC
10	STORE	CHICAGO
20	RESEARCH	DALLAS
30	SALES	NEW YORK
40	MARKETING	BOSTON

DEPT 테이블

- Project 테이블을 생성해 보세요.

```
create table PROJECT (
        PNO number(3) primary key,
        PNAME varchar2(60) unique, PMGR number(4) not null,
        PERSONS number(5), BUDGET number(8,2) not null,
        PSTART date, PEND date);
```

DML SQL의 실습

- Select
  - 특정 행을 조회한다.
- Select 구문
  - select [distinct] <column(s)>
  - from <table>
  - [where <condition>]
  - [order by <column(s) [asc|desc]>]
- 예제
  - select LOC, DEPTNO from DEPT;
  - select * from EMP;
  - select ENAME, DEPTNO, SAL * 1.55 from EMP;

DML SQL의 실습

- Select
  - for numbers: abs, cos, sin, exp, log, power, mod, sqrt, +, −, *, /, ...
  - for strings: chr, concat(string1, string2), lower, upper, replace(string, search string,replacement string), translate, substr(string, m, n), length, to date, . . .
  - for the date type: add month, month between, next day, to char,
- 실습
  - select ENAME, DEPTNO, HIREDATE from EMP;
  - Select * from EMP

    order by DEPTNO [asc], HIREDATE desc;

ENAME	DEPTNO	HIREDATE
FORD	10	03-DEC-81
SMITH	20	17-DEC-80
BLAKE	30	01-MAY-81
WARD	30	22-FEB-81
ALLEN	30	20-FEB-81
.....		

DML SQL의 실습

- Selection of Tuples
  - where 안에 비교 연산자 =, != or <>, , <=, =>
- 실습
  - select JOB, SAL from EMP

    where (MGR = 7698 or MGR = 7566) and SAL > 1500;
  - select EMPNO, ENAME, SAL from EMP

    where SAL between 1500 and 2500;
  - select ENAME from EMP

    where HIREDATE between '02-APR-81' and '08-SEP-81';

DML SQL의 실습

- Insert
  - insert into <table> [(<column i, . . . , column j>)]
  - values (<value i, . . . , value j>);
- 실습
  - insert into PROJECT(PNO, PNAME, PERSONS, BUDGET, PSTART) values(313, 'DBS', 4, 150000.42, '10-OCT-94');
  - insert into PROJECT values(313, 'DBS', 7411, null, 150000.42, '10-OCT-94', null);

DML SQL의 실습

- Update
  - update <table> set
  - <column i> = <expression i>, . . . , <column j> = <expression j>
  - [where <condition>];
- 실습
  - update EMP set JOB = 'MANAGER', DEPTNO = 20, SAL = SAL +1000 where ENAME = 'JONES';
  - update EMP set SAL = SAL * 1.15 where DEPTNO in (10,30);

DML SQL의 실습

- Delete
  - delete from <table> [where <condition>];
- 실습
  - delete from PROJECT where PEND < sysdate;

Group by

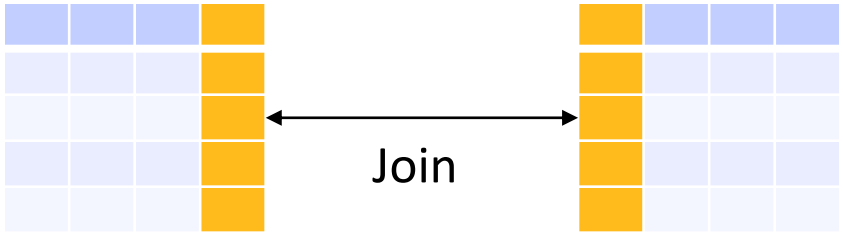
- Aggregation

OrderID	CustomerID	Amount	OrderDate	ShipperID
10308	2	23000	1996-09-18	3
10309	37	78000	1996-09-19	1
10310	77	39000	1996-09-20	2

- 고객별 주문금액 합계
  - SELECT customerId, sum(amount) FROM order GROUP BY customerId;
- 날짜별 매출합계
  - SELECT orderdate, sum(amount) FROM order GROUP BY orderdate;
  - min()함수, max()함수, count(), mean() 함수 사용가능

Join

- 두 개 이상의 테이블 들을 연결 또는 결합하여 데이터를 출력하는 것




```
select ENAME, E.DEPTNO, DNAME
from EMP E, DEPT D
where E.DEPTNO = D.DEPTNO
and JOB = 'SALESMAN';
```

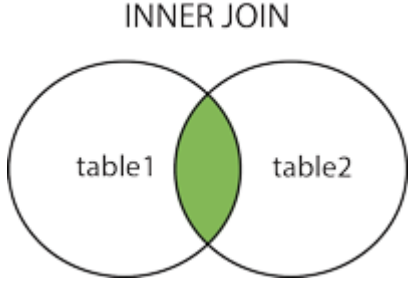
```
select E1.ENAME, E2.ENAME
from EMP E1, EMP E2
where E1.MGR = E2.EMPNO;
```

Inner Join

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	John	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Joe	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Jolly	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
INNER JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

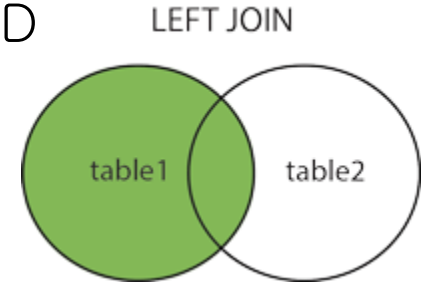


Left Join

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	John	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Joe	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Jolly	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

- SELECT Customers.CustomerName, Orders.OrderID
  FROM Customers
  LEFT JOIN Orders
  ON Customers.CustomerID=Orders.CustomerID
  ORDER BY Customers.CustomerName;



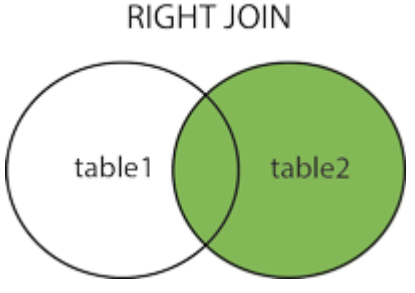
Right Join

- Right Join

EmployeeID	LastName	FirstName	BirthDate	Photo	Notes
1	Davolio	Nancy	12/8/1968	EmpID1.pic	Education includes a BA in psychology.....
2	Fuller	Andrew	2/19/1952	EmpID2.pic	Andrew received his BTS commercial and....
3	Leverling	Janet	8/30/1963	EmpID3.pic	Janet has a BS degree in chemistry....

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

```
SELECT Orders.OrderID, Employees.FirstName
FROM Orders
RIGHT JOIN Employees
ON Orders.EmployeeID=Employees.EmployeeID
ORDER BY Orders.OrderID;
```

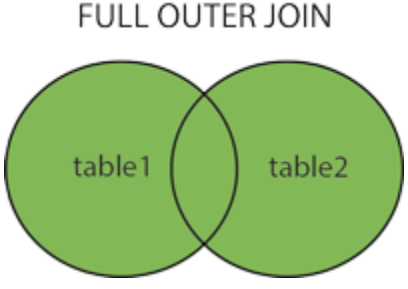


Full Join

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	John	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Joe	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Jolly	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09-18	3
10309	37	3	1996-09-19	1
10310	77	8	1996-09-20	2

```
SELECT Customers.CustomerName, Orders.OrderID
 FROM Customers
 FULL OUTER JOIN Orders
 ON Customers.CustomerID=Orders.CustomerID
 ORDER BY Customers.CustomerName;
```



Full Join

- Full Join 결과

CustomerName	OrderID
Alfreds Futterkiste	
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
	10382
	10351

Subquery

- 쿼리 결과가 다른 쿼리의 where 조건에 쓰이는 것
- <expression> [not] in (<subquery>)

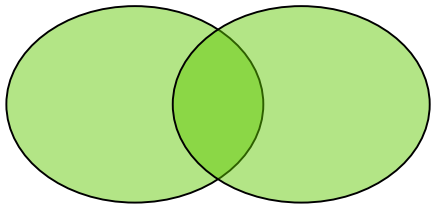
  - select ENAME, SAL from EMP
  - where EMPNO in
  - (select PMGR from PROJECT
  - where PSTART < '31-DEC-90')
  - and DEPTNO =20;

```
select  from EMP
where DEPTNO in
(select DEPTNO from DEPT
where LOC = 'BOSTON');
```

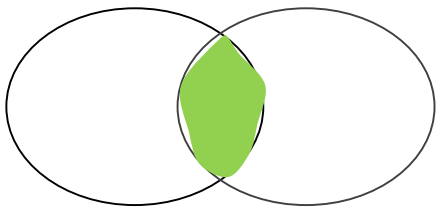
```
select * from EMP
where SAL > all
(select SAL from EMP
where DEPTNO = 30)
and DEPTNO <> 30;
```

Operation on Result Sets (집합연산자)

- Union : 합집합
  - select EMPNO, ENAME from EMP
  - union
  - select EMPNO, ENAME from EMP2;



- Intersect : 교집합
  - select from EMP
  - intersect
  - select from EMP2;



- minus : 차집합
  - select from EMP
  - minus
  - select from EMP2;