

DAY21-SQL

☰ 태그	
📅 날짜	@2022년 10월 28일

Oracle DB

SQL Developer 설치

oracle express version 설치

oracle live 사이트

<https://livesql.oracle.com/apex/f?p=590:1000>

SQL

1. SELECT

테이블에서 전체데이터 혹은 일부데이터만 추출하여 조회하는 구문 즉 ,데이터베이스에 있는 데이터를 조회할수 있다.

select은 가장 많이 사용하는 가장 중요한 sql 구문이다.

```
SELECT    열이름 또는 * (모든 열)
FROM      테이블이름;
```

1. SQL 은 대문자와 소문자를 구별하지 않는다. 하지만 반드시 구문은 대문자로 나머지는 전부 소문자로 작성하자.
2. SQL 문은 한줄로 또는 여러줄로 작성할 수 있다. 하지만 여러줄로 작성함
3. SQL 문은 들여쓰기를 상관하지 않지만 들여쓰기를 적절하

SELECT * // *은 모든 열이라는 의미
FROM hr.employees;

SELECT 열이름, 열이름 : 입력한 열이름만 출력된다.

ORDER BY 열이름 [ASC or DESC]:

출력의 결과를 반대로 정렬 (게시물 정렬 등)

ASC 오름차순 123456, abcde, 가나다라마

DESC 내림차순 54321, edcba, 마라다나가

SQL Worksheet

```
1 SELECT employee_id, first_name, last_name
2 FROM hr.employees
3 ORDER BY employee_id DESC;
4
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
206	William	Gietz
205	Shelley	Higgins
204	Hermann	Baer
203	Susan	Mavris
202	Pat	Fay
201	Michael	Hartstein
200	Jennifer	Whalen

DISTINCT : 중복 값 제거 후 출력

before

1	SELECT job_id
2	FROM hr.employees
3	
4	

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD PRES
AD_VP
AD_VP
FI_ACCOUNT
FI_ACCOUNT

after

1	SELECT DISTINCT job_id
2	FROM hr.employees
3	
4	

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD PRES
AD_VP
FI_ACCOUNT
FI_MGR
HR_REP

중복값이 사라진것을 확인 할 수 있다.

AS : 별칭 / 별명 사용하기

필드명을 임시로 줄이거나 편하게 보기 원할때 AS를 사용하여 변경할 수 있다.

AS자체를 생략할 수 도 있다.

SELECT 열이름 AS 별명

별명에 띄어쓰기나 특수문자 등을 사용하길 원할때는 ""을 붙여준다.

SQL Worksheet

```
1 SELECT DISTINCT job_id AS 업무아이디, first_name AS 이름, last_name AS 성
2 FROM hr.employees
3
4
```

업무아이디	이름	성
AD_ASST	Jennifer	Whalen
IT_PROG	Valli	Pataballa
PU_MAN	Den	Raphaely
SA_MAN	Karen	Partners
SA_REP	David	Bernstein
SA_REP	Sundita	Kumar
SA_REP	Jonathon	Taylor
SH_CLERK	Vance	Jones

|| : 연결연산자

연결 연산자를 사용하면 각 열의 결과를 연결하여 하나의 열로 결과를 임시로 표현할 수 있다.

새로운 열을 임시로 생성할 수 도 있다.

1	SELECT DISTINCT job_id, first_name ' ' last_name
2	FROM hr.employees
3	
4	

JOB_ID	FIRST_NAME ' ' LAST_NAME
AC_MGR	Shelley Higgins
IT_PROG	Diana Lorentz
PU_CLERK	Guy Himuro
SA_REP	Lindsey Smith
SA_REP	Clara Vishney

진짜로 붙은게 아니라 임시로 붙은 것이다.

|| ' ' || : 두 열 사이에 띄어쓰기를 넣은것 같음

열에다가 직접 글자도 붙힐 수 있다.

1	SELECT DISTINCT job_id,
2	first_name ' ' last_name,
3	email '@' 'company.com' AS 회사이메일주소
4	FROM hr.employees

JOB_ID	FIRST_NAME ' ' LAST_NAME	회사이메일주소
AD_PRES	Steven King	SKING@company.com
AD_VP	Lex De Haan	LDEHAAN@company.com
IT_PROG	Alexander Hunold	AHUNOLD@company.com
ST_MAN	Shanta Vollman	SVOLLMAN@company.com
ST_CLERK	Mozhe Atkinson	MATKINSO@company.com
ST_CLERK	Stephen Stiles	SSTILES@company.com
ST_CLERK	Joshua Patel	JPATEL@company.com
SA_REP	Christopher Olsen	COLSEN@company.com
SA_REP	Allan McEwen	AMCEWEN@company.com
SA_REP	Sundar Ande	SANDE@company.com
SA_REP	Jack Livingston	JLIVINGS@company.com

INSERT 행 삽입

insert 명령어를 사용하여 테이블에 데이터를 넣게 된다.

```
INSERT INTO hr.departments VALUES (272, 'TEST DEPT', 200, 1700)
```

온라인이어서 안됨.

전체 필드에 데이터를 삽입 할때는 모든 열의 이름을 적거나 또는 생략 가능하다.

하지만 일부 열에만 데이터를 삽입할 때는 그 일부 열이름을 적어주어야만 한다.

```
INSERT INTO hr.departments(DEPARTMENT_ID, DEPARTMENT_NAME, MANAGER_ID, LOCATION_ID)
VALUES (272, 'TEST DEPT', 200, 1700);
```

```
INSERT INTO hr.departments(DEPARTMENT_ID, DEPARTMENT_NAME, MANAGER_ID, LOCATION_ID)
VALUES (272, 'TEST DEPT', 200, 1700);
```

UPDATE : 행 갱신

기존의 데이터 값을 다른 데이터 값으로 변경할 때 사용된다.

```
UPDATE 테이블 이름
SET 열이름 = 데이터값
(변경하려는 열) (변경하려는 값)
WHERE 조건 열이름 = 데이터값;
```

```
UPDATE hr.departments
SET Manager_id = 201,
    Location_id = 1800
WHERE department_name = 'Marketing';
```

departments 테이블에서 department_name 이 Marketing 인 데이터를 찾아서
manager_id 값을 201로 변경하고 Location_id 값은 1800으로 변경하여라

```
UPDATE 회원정보 테이블
SET addr = "서울시 서초구 서초동 1번지"
WHERE 회원아이디 = 290;
```

DELETE 행 삭제

DELETE 명령어는 테이블의 데이터를 삭제할 때 사용된다.

UPDATE와 마찬가지로 WHERE 절을 사용한다.

주의 : WHERE을 생략하면 한번에 모든 데이터가 삭제 됨으로 주의해야한다.

```
DELETE FROM 테이블명  
  
WHERE 조건식;
```

```
DELETE FROM departments  
WHERE department_name = 'Marketing;  
  
DELETE FROM 회원  
WHERE 회원_아이디 = 101;
```

데이터베이스 생성

```
CREATE DATABASES 데이터베이스명;
```

테이블 생성

```
CREATE TABLE 테이블명 (  
    필드명, 데이터타입(크기),  
    필드명, 데이터타입(크기),  
    필드명, 데이터타입(크기),  
);
```

```
CREATE TABLE Member (
```

```
mid, varchar(255),  
name , varchar(255),  
age, varchar(255),  
);
```

테이블 삭제

```
DROP TABLE 테이블명;
```

테이블 수정

```
열 추가  
ALTER TABLE 테이블명  
ADD 필드명 데이터타입;
```

```
-----  
열 삭제  
ALTER TABLE 테이블명  
DROP column(필드명);
```

[https://mariadb.org/download/?
t=mariadb&p=mariadb&r=10.9.3&os=windows&cpu=x86_64&pkg=msi&m=blendbyte](https://mariadb.org/download/?t=mariadb&p=mariadb&r=10.9.3&os=windows&cpu=x86_64&pkg=msi&m=blendbyte)

[Download](#)
[Documentation](#)
[Contribute](#)
[Server Fest](#)
[Events](#)
[Sponsor](#)
[Blog](#)
[Planet MariaDB](#)
[About](#)

Latest releases
 [10.11.0 \(Alpha\)](#)
[10.10.1 \(RC\)](#)
[10.9.3](#),
 [10.8.5](#),
 [10.7.6](#),
 [10.6.10](#),
 [10.5.17](#),
 [10.4.26](#),
 [10.3.36](#).
 [Vote in our poll](#)

Download MariaDB Server
 REST API
 Release Schedule
 Reporting Bugs
 MariaDB Server Statistics
 Download MariaDB Server Documentation
 View all releases for:
 MariaDB Server
 MariaDB Galera
 Connector/C
 Connector/J
 Connector/ODBC

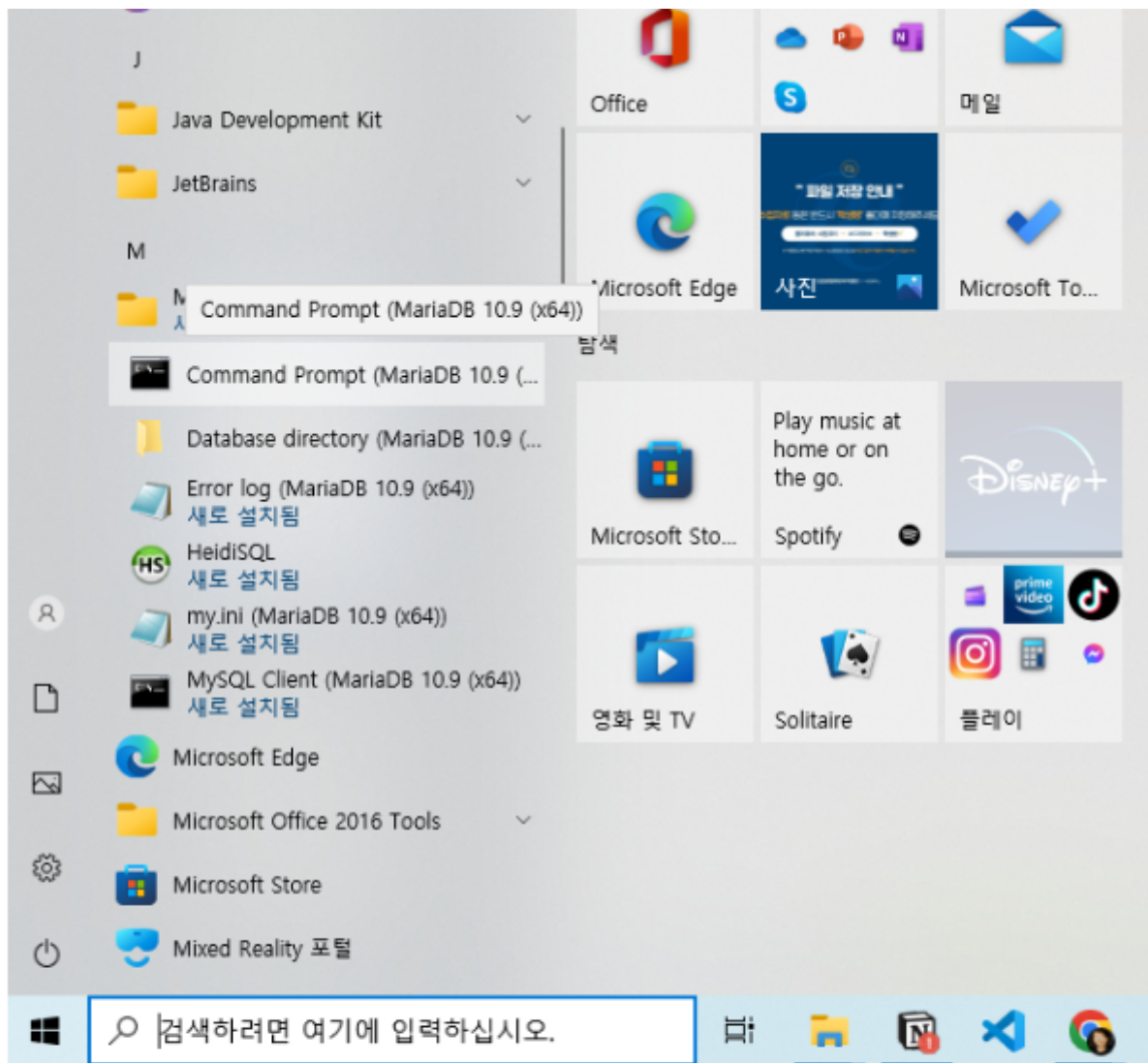
MariaDB Server Version
 MariaDB Server 10.9.3
 Display older releases: ☐
 Operating System
 Windows
 Architecture
 x86_64
 Package Type
 MSI Package
 Download
 Mirror
 Blendbyte - Taipei
 Release date: 2022-09-20
 File name: mariadb-10.9.3-win64.msi

User settings

Default instance properties
 MariaDB 10.9 (x64) database configuration

☒ **Modify password for database user 'root'**
 New root password: Enter new root password
 Confirm: Retype the password
☒ **Enable access from remote machines for 'root' user**
☒ **Use UTF8 as default server's character set**

Data directory



```
Command Prompt (MariaDB 10.9 (x64)) - echo Setting environment for MariaDB 10.9 (x64)
Setting environment for MariaDB 10.9 (x64)
C:\Windows\System32>
```

```
Command Prompt (MariaDB 10.9 (x64)) - mysql -u root -p
Setting environment for MariaDB 10.9 (x64)

C:\Windows\System32>mysql -u root -p
Enter password: ****_
```

현재 사용 가능한 DB목록 보기

SHOW DATABASES;

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.000 sec)

MariaDB [(none)]>
```

univ 데이터베이스 생성

```

MariaDB [(none)]> CREATE DATABASES univ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
comes with your MySQL version for the right syntax to use near 'DATABASES'
MariaDB [(none)]> CREATE DATABASE univ;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| univ |
+-----+
5 rows in set (0.000 sec)

MariaDB [(none)]> _

```

student 학생 테이블 생성

```

MariaDB [univ]> CREATE TABLE student (
  -> hakbun          int          not null,
  -> name            varchar(5),
  -> year            tinyint,
  -> dept            varchar(10),
  -> addr            varchar(50),
  -> primary key(hakbun));
Query OK, 0 rows affected (0.015 sec)

MariaDB [univ]> _

```

생성한 테이블의 구조 확인하기

```

MariaDB [univ]> desc student;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| hakbun | int(11)       | NO   | PRI | NULL    |       |
| name   | varchar(5)    | YES  |     | NULL    |       |
| year   | tinyint(4)    | YES  |     | NULL    |       |
| dept   | varchar(10)   | YES  |     | NULL    |       |
| addr   | varchar(50)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.008 sec)

MariaDB [univ]> _

```

세명의 데이터를 테이블에 삽입

```

MariaDB [univ]> INSERT INTO student
-> VALUES(16001,"한국인",4,"컴공","서울"),
->          (19999,"미국인",3,"건축","LA"),
->          (29999,"일본인",2,"컴공","인천");
Query OK, 3 rows affected (0.002 sec)
Records: 3  Duplicates: 0  Warnings: 0

```

```

MariaDB [univ]> SELECT * FROM student;
+-----+-----+-----+-----+-----+
| hakbun | name      | year | dept  | addr  |
+-----+-----+-----+-----+-----+
| 16001  | 한국인   | 4    | 컴공  | 서울  |
| 19999  | 미국인   | 3    | 건축  | LA    |
| 29999  | 일본인   | 2    | 컴공  | 인천  |
+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [univ]>

```

학년이 3학년 이상 인 학생들의 목록만 출력

hakbun	name	year	dept	addr
16001	한국인	4	컴공	서울
19999	미국인	3	건축	LA

2 rows in set (0.000 sec)

학과가 컴공인 학생들의 이름과 학년만 출력하세요

학년은 내림차순 이름은 오름차순으로 정렬

```
mar@abb [univ]> select name,
```

name	year
한국인	4
일본인	2

2 rows in set (0.000 sec)

```

MariaDB [univ]> SELECT name, year
-> FROM student
-> WHERE dept="컴공"
-> ORDER BY year DESC, name ASC;
+-----+-----+
| name      | year |
+-----+-----+
| 한국인    | 4    |
| 일본인    | 2    |
+-----+-----+
2 rows in set (0.001 sec)

MariaDB [univ]>

```

대학에 개설된 학과들의 이름만 중복을 제거하고 조회하시오

```

MariaDB [univ]> SELECT DISTINCT dept FROM student;
+-----+
| dept |
+-----+
| 컴공 |
| 건축 |
+-----+
2 rows in set (0.000 sec)

MariaDB [univ]>

```

학생 수를 출력하시오.

```
SELECT COUNT(*) FROM 테이블;
```

```

MariaDB [univ]> SELECT COUNT(*) FROM student;
+-----+
| COUNT(*) |
+-----+
|        3 |
+-----+
1 row in set (0.000 sec)

MariaDB [univ]>

```

대학에 개설된 전공의 갯수를 구하시오.

```

MariaDB [univ]> SELECT COUNT(DISTINCT(dept)) FROM student;
+-----+
| COUNT(DISTINCT(dept)) |
+-----+
|                2 |
+-----+
1 row in set (0.001 sec)

MariaDB [univ]>

```

19학번만 추출

```

MariaDB [univ]> SELECT * FROM student WHERE hakbun LIKE '19%';
+-----+-----+-----+-----+-----+
| hakbun | name   | year | dept | addr |
+-----+-----+-----+-----+-----+
| 19999 | 미국인 | 3    | 건축 | LA   |
+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [univ]>

```

지금까지 위에서 사용한 기본 프롬프트 틀은 너무 많은 에러 발생으로 불편함.

따라서 HeidiSQL을 사용하면 아주 쾌적하게 MySQL은 물론 MariaDB까지 다룰 수 있다.

heidisql.com/download.php

HS HeidiSQL

개발자 채용 성공 전략

- 네이버에 합격자 목록
- 코딩 테스트
- 기술 면접
- 마케팅과 스택

코딩테스트 기술면
개발자 취업 하루 3시간 30분
패스트캠퍼스

Home Downloads Images Forum Donate Bugtracker Help

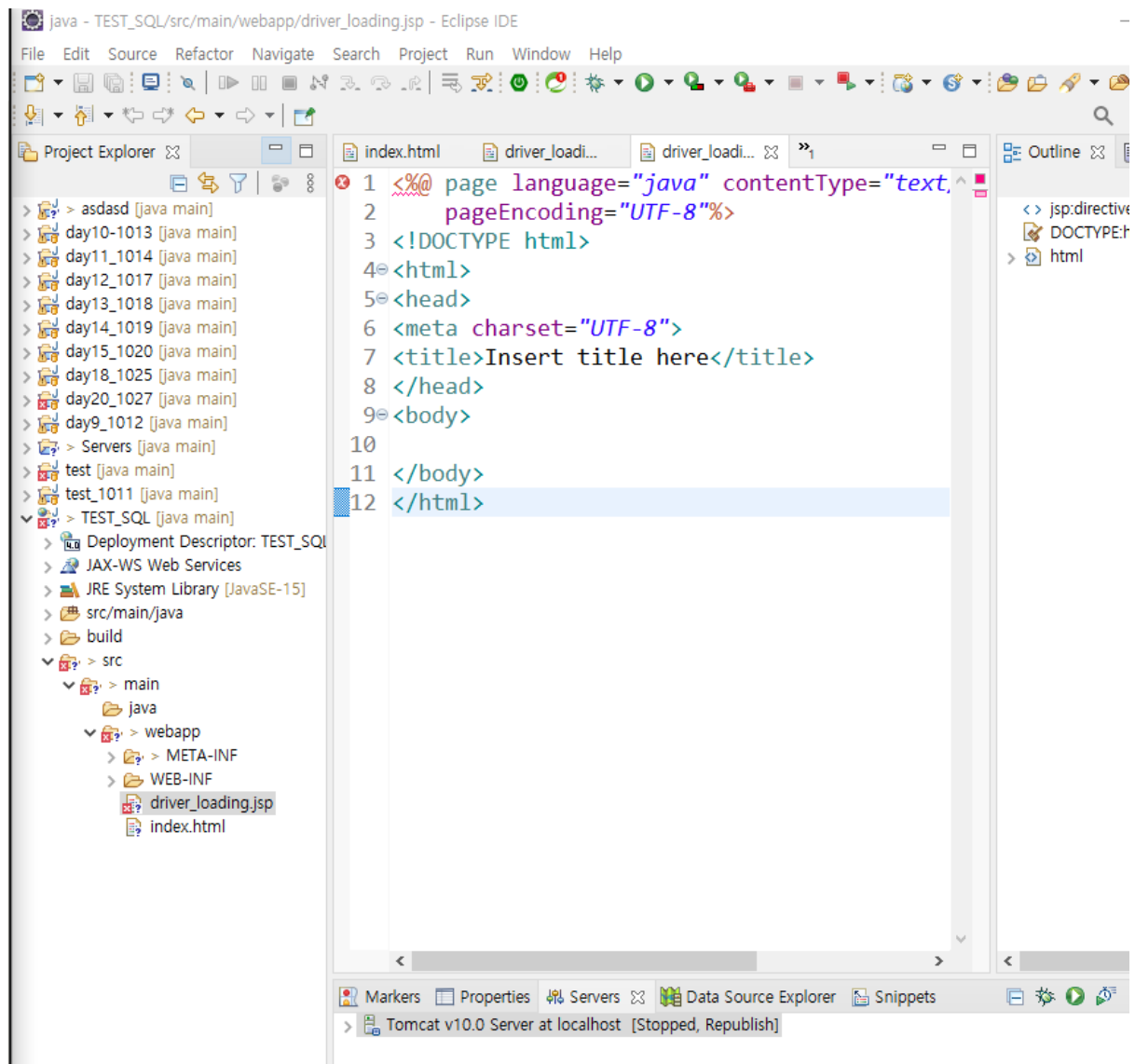


Download MariaDB Version 10.9

MariaDB

Download HeidiSQL 12.1, released on 13 Aug 2022

Installer, 32/64 bit combined (SHA1 checksum)



흔히 있는 에러 : 프로젝트면 우클릭 -> properties → java build path → libraries →

