

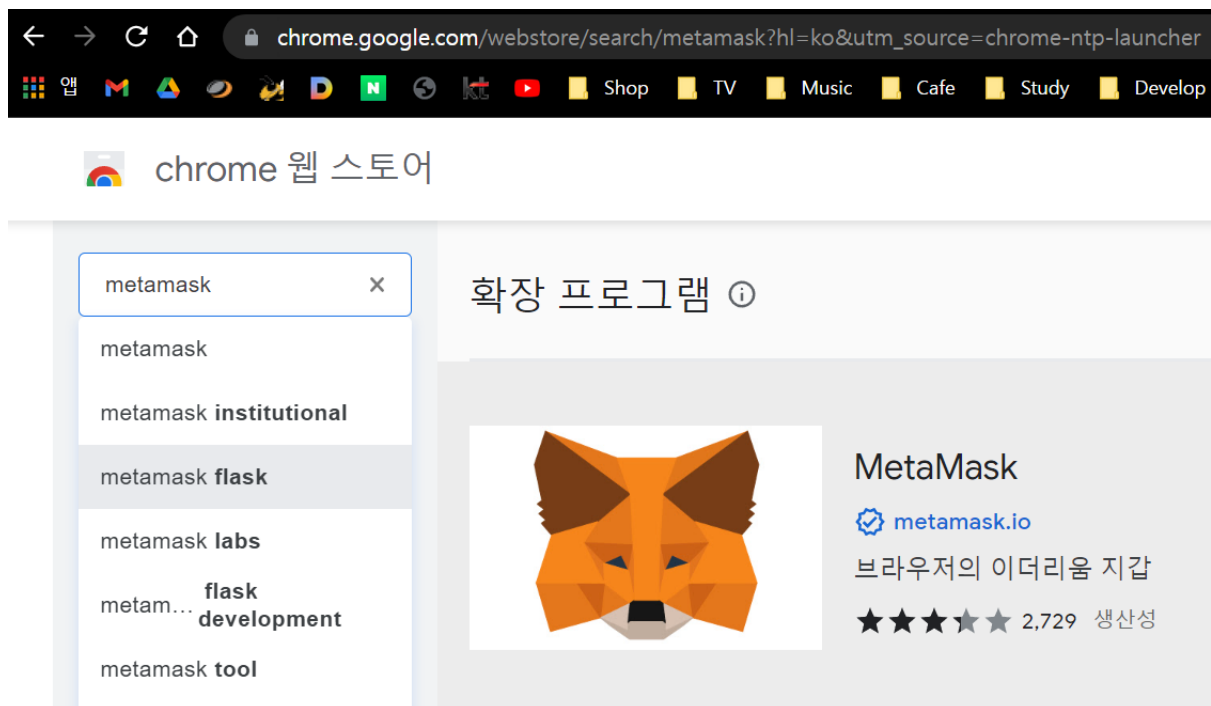
# Day21

## NFT(Non Fungible Token)

- 블록체인으로 디지털 자산의 소유 증명을 위한 토큰
- 거래내역을 블록체인상에 남겨 영구적으로 그 고유성을 인정받음

### 1. 지갑 설치

크롬 브라우저 - Metamask





# Welcome to MetaMask

Connecting you to Ethereum and the Decentralized Web.

We're happy to see you.

Get started



## New to MetaMask?



No, I already have a Secret Recovery Phrase

Import your existing wallet using a Secret Recovery Phrase

Import wallet



Yes, let's get set up!

This will create a new wallet and Secret Recovery Phrase

Create a wallet



METAMASK

< Back

# Secret Recovery Phrase

Your Secret Recovery Phrase makes it easy to back up and restore your account.

**WARNING:** Never disclose your Secret Recovery Phrase. Anyone with this phrase can take your Ether forever.

split traffic swing ecology leg  
suggest biology wide number  
episode legend fence

Remind me later

Next

# Confirm your Secret Recovery Phrase

Please select each phrase in order to make sure it is correct.

split	traffic	swing	ecology
leg	suggest	biology	wide
number	episode	legend	fence

biology	ecology	episode	fence
leg	legend	number	split
suggest	swing	traffic	wide



0 ETH

\$0.00 USD



Buy



Send



Swap

Assets

Activity

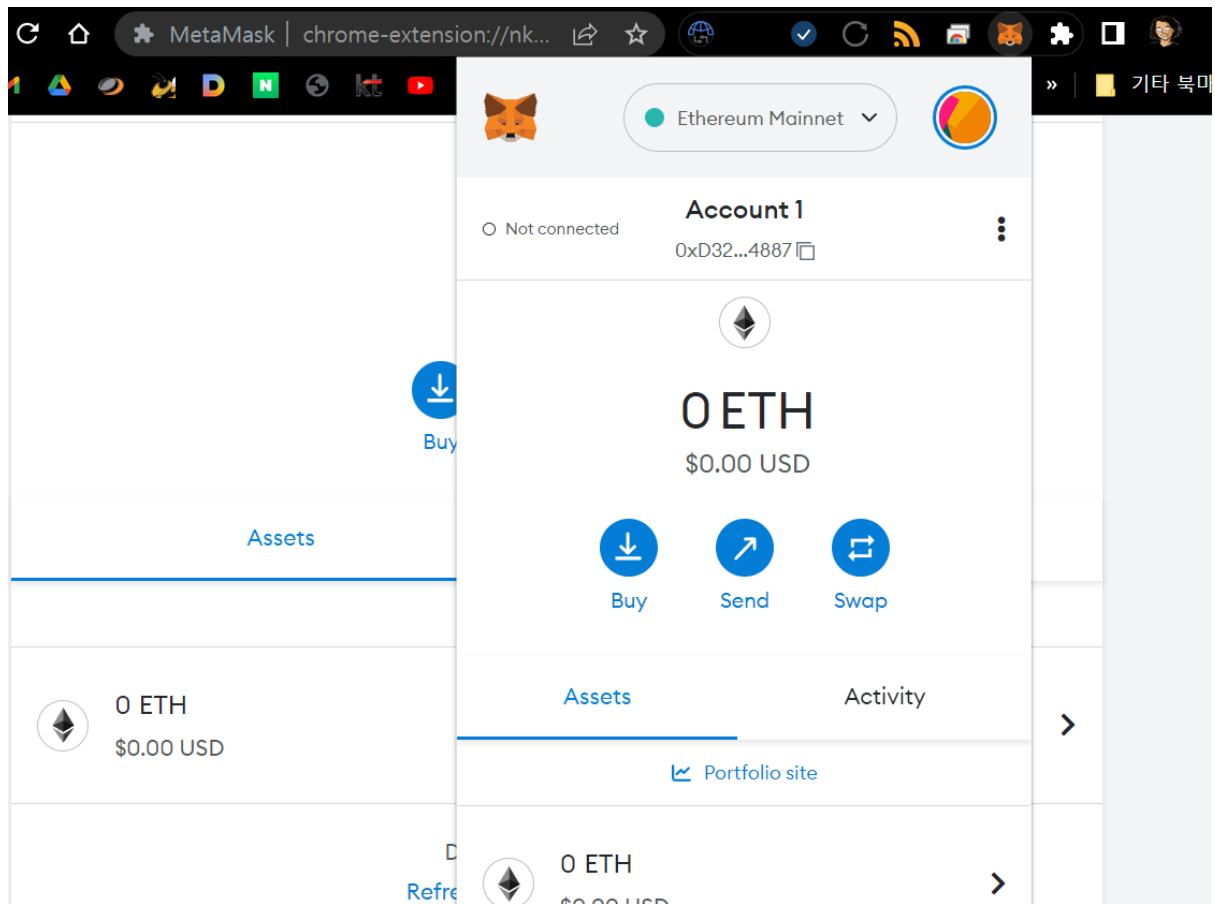
[Portfolio site](#)



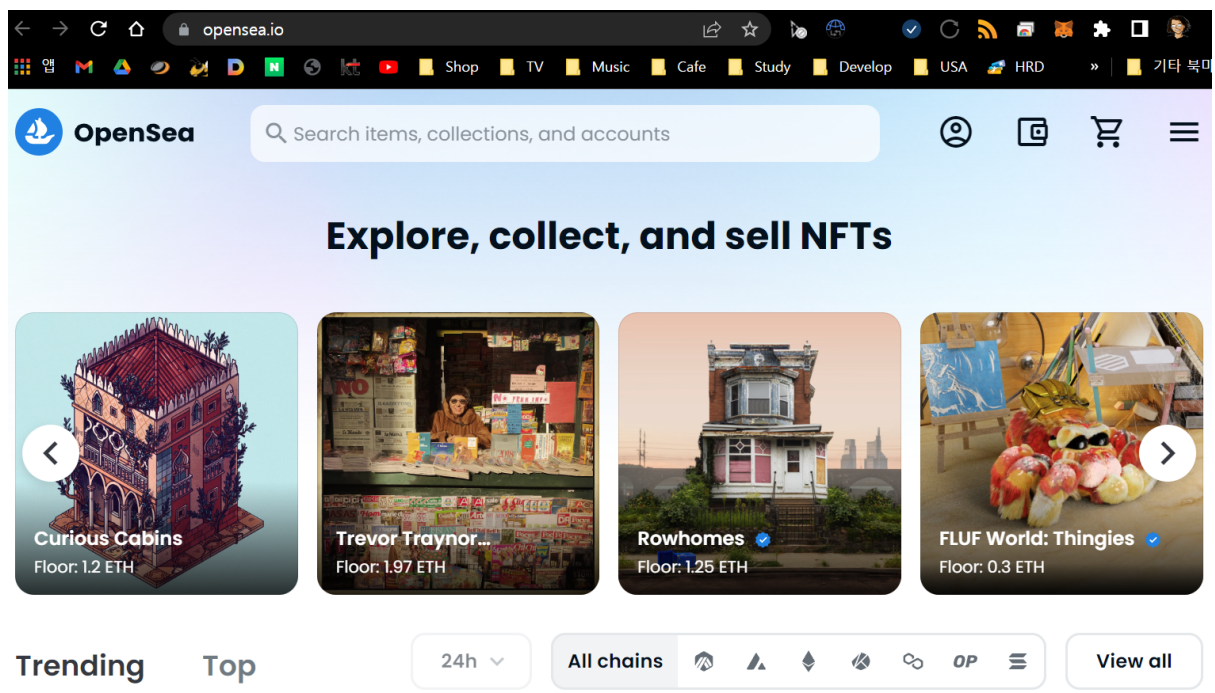
0 ETH

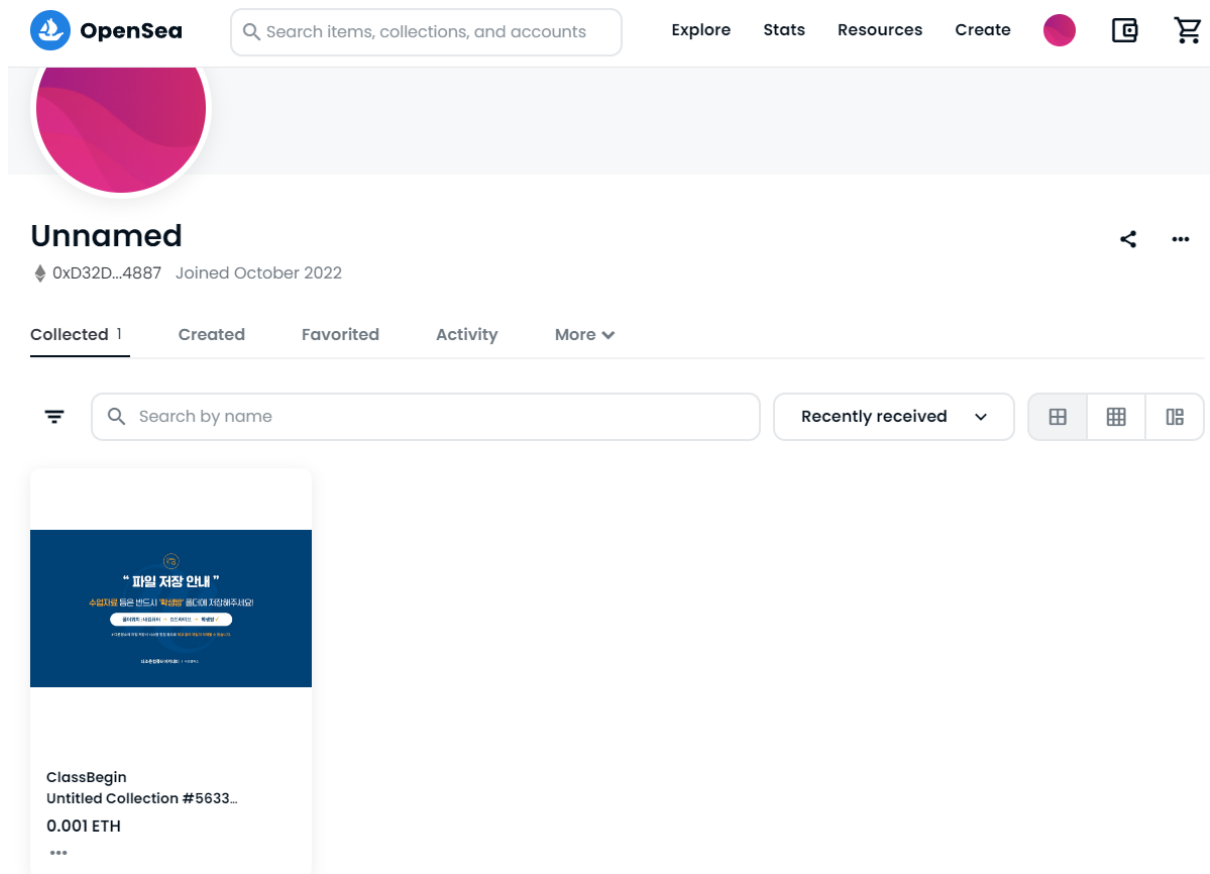
\$0.00 USD





## 2. OpenSea 가입





## SQL

### 환경 구성

#### 1) 온라인 SQL

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

#### 2) Oracle Database 11g

<https://www.oracle.com/database/technologies/xe-prior-release-downloads.html>

#### 3) Oracle Developer

<https://www.oracle.com/database/sqldeveloper/technologies/download/>

- SQL은 대소문자를 구별하지 않으나 구문은 대문자로 작성 권고
- SQL문은 한줄 또는 여러줄로 작성 가능
- 들여쓰기를 상관하지 않지만 들여쓰기를 적절하게 사용하면 구문을 보기 좋게 작성하게 됨

## 1. SELECT

```
SELECT 열이름 or *(모든열)
FROM 테이블 이름;
```

- 데이터베이스에 있는 데이터 조회

### 1) 모든열 조회

1	SELECT *	
2	FROM hr.employees;	

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID
100	Steven	King	SKING	515.123.4567	17-JUN-03	AD_PRES	24000	-	-
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-05	AD_VP	17000	-	100
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-01	AD_VP	17000	-	100
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-06	IT_PROG	9000	-	102
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-07	IT_PROG	6000	-	103
105	David	Austin	DAUSTIN	590.423.4569	25-JUN-05	IT_PROG	4800	-	103
106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-06	IT_PROG	4800	-	103
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-07	IT_PROG	4200	-	103
108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-02	FI_MGR	12008	-	101
109	Daniel	Faviet	DFAVIET	515.124.4169	16-AUG-02	FI_ACCOUNT	9000	-	108
110	John	Chen	JCHEN	515.124.4269	28-SEP-05	FI_ACCOUNT	8200	-	108
111	Ismael	Sciarra	ISCIARRA	515.124.4369	30-SEP-05	FI_ACCOUNT	7700	-	108
112	Jose Manuel	Urman	JMURMAN	515.124.4469	07-MAR-06	FI_ACCOUNT	7800	-	108
113	Luis	Popp	LPOPP	515.124.4567	07-DEC-07	FI_ACCOUNT	6900	-	108
114	Den	Raphaely	DRAPHEAL	515.127.4561	07-DEC-02	PU_MAN	11000	-	100
115	Alexander	Khoo	AKHOO	515.127.4562	18-MAY-03	PU_CLERK	3100	-	114
116	Shelli	Baida	SBAIDA	515.127.4563	24-DEC-05	PU_CLERK	2900	-	114
117	Sigal	Tobias	STOBIAS	515.127.4564	24-JUL-05	PU_CLERK	2800	-	114
118	Guy	Himuro	GHIMURO	515.127.4565	15-NOV-06	PU_CLERK	2600	-	114
119	Karen	Colmenares	KCOLMENA	515.127.4566	10-AUG-07	PU_CLERK	2500	-	114

- SELECT 뒤에 \* 사용

## 2) 선택 열 조회



## SQL WORKSHEET

```
1 SELECT first_name, last_name
2 FROM hr.employees;
```

FIRST_NAME	LAST_NAME
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer
Shelli	Baida
Amit	Banda
Elizabeth	Bates
Sarah	Bell
David	Bernstein
Laura	Bissot
Harrison	Bloom
Alexis	Bull
Anthony	Cabrio
Gerald	Cambrault
Nanette	Cambrault
John	Chen
Kelly	Chung
Karen	Colmenares
Curtis	Davies
Lex	De Haan

- SELECT 뒤에 선택할 열 이름 선택

## ORDER BY

- 출력의 결과를 반대로 정렬할 필요가 있을때 사용

SELECT 열이름 or \*(모든열)  
FROM 테이블 이름  
ORDER BY 원하는 열  
DESC(내림차순) or ASC(오름차순)

```
1 SELECT employee_ID,first_name,last_name  
2 FROM hr.employees  
3 order by employee_ID desc;
```

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
199	Douglas	Grant
198	Donald	OConnell
197	Kevin	Feeney
196	Alana	Walsh
195	Vance	Jones
194	Samuel	McCain
193	Britney	Everett
192	Sarah	Bell
191	Randall	Perkins
190	Timothy	Gates
189	Jennifer	Dilly
188	Kelly	Chung
187	Anthony	Cabrio
186	Julia	Dellinger

© 2022 Oracle. All rights reserved. Oracle Database 19c Enterprise Edition

## DISTINCT

```
SELECT DISTINCT 원하는 열  
FROM 테이블 이름;
```

- 중복값 제거 후 출력

### 1) 중복제거 전

## SQL Worksheet

```
1 SELECT job_id
2 FROM hr.employees;
```

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
AD_VP
FI_ACCOUNT
FI_ACCOUNT
FI_ACCOUNT
FI_ACCOUNT
FI_ACCOUNT
FI_MGR
HR_REP
IT_PROG
IT_PROG
IT_PROG
IT_PROG
IT_PROG
MK_MAN
MK_REP
PR_REP
PU_CLERK
PU_CLERK

2) 중복 제거 후

```

1 SELECT DISTINCT job_id
2 FROM hr.employees;

```

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD PRES
AD_VP
FI_ACCOUNT
FI_MGR
HR_REP
IT_PROG
MK_MAN
MK_REP
PR_REP
PU_CLERK
PU_MAN
SA_MAN
SA_REP
SH_CLERK
ST_CLERK
ST_MAN

[Download CSV](#)  
 19 rows selected.

## AS

SELECT 열이름 AS 별명

- 별칭/ 별명 사용
- 필드명을 임시로 줄이거나 편하게 보기 원할때 사용
- AS를 생략할수 있음

1	SELECT job_id AS 업무아이디, first_name AS 이름, last_name AS 성
2	FROM hr.employees;

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
SH_CLERK	Kevin	Feeney
SH_CLERK	Donald	OConnell
ST_CLERK	Steven	Markle

## || (연결연산자)

```
SELECT 열이름, 열이름 || 열이름
```

- 각 열의 결과를 연결하여 결과를 하나의 열로 임시 표현

```

1 SELECT job_id, first_name || ' ' || last_name
2 FROM hr.employees;

```

JOB_ID	FIRST_NAME    ' '    LAST_NAME
AC_ACCOUNT	William Gietz
AC_MGR	Shelley Higgins
AD_ASST	Jennifer Whalen
AD_PRES	Steven King
AD_VP	Neena Kochhar
AD_VP	Lex De Haan
FI_ACCOUNT	Daniel Faviet
FI_ACCOUNT	John Chen
FI_ACCOUNT	Ismael Sciarra
FI_ACCOUNT	Jose Manuel Urman
FI_ACCOUNT	Luis Popp
FI_MGR	Nancy Greenberg
HR_REP	Susan Mavris
IT_PROG	Alexander Hunold
IT_PROG	Bruce Ernst
IT_PROG	David Austin
IT_PROG	Valli Pataballa
IT_PROG	Diana Lorentz
MK_MAN	Michael Hartstein
MK_REP	Pat Fay
PR_REP	Hermann Baer
PU_CLERK	Alexander Khoo
PU_CLERK	Shelli Baida
PU_CLERK	Sigal Tobias

```

1 SELECT job_id,
2        first_name || ' ' || last_name AS 이름,
3        email || '@company.com' AS 이메일
4 FROM hr.employees;

```

JOB_ID	이름	이메일
AD_PRES	Steven King	SKING@company.com
AD_VP	Neena Kochhar	NKOCHHAR@company.com
AD_VP	Lex De Haan	LDEHAAN@company.com
IT_PROG	Alexander Hunold	AHUNOLD@company.com
IT_PROG	Bruce Ernst	BERNST@company.com
IT_PROG	David Austin	DAUSTIN@company.com
IT_PROG	Valli Pataballa	VPATABAL@company.com
IT_PROG	Diana Lorentz	DLORENTZ@company.com
FI_MGR	Nancy Greenberg	NGREENBE@company.com
FI_ACCOUNT	Daniel Faviet	DFAVIET@company.com
FI_ACCOUNT	John Chen	JCHEN@company.com
FI_ACCOUNT	Ismael Sciarra	ISCIARRA@company.com
FI_ACCOUNT	Jose Manuel Urman	JMURMAN@company.com
FI_ACCOUNT	Luis Popp	LPOPP@company.com
PU_MAN	Den Raphaely	DRAPHEAL@company.com

- 문자열을 붙일수 있음

## 행 데이터 삽입

```

INSERT INTO 테이블명(열1, 열2, 열3)
VALUES(열1의 데이터 값, 열2의 데이터 값, 열3의 데이터 값);

```

### 1) SELECT 구문으로 데이터형태 확인



```

1 SELECT *
2 FROM hr.departments;

```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
100	Finance	108	1700

2) INSERT 문으로 hr.departments 테이블 데이터 형식에 맞게 value값 삽입

```

1 INSERT INTO hr.departments
2 VALUES(272,'TEST DEPT',200,1700);

```

- 전체 필드가 아닌 일부 필드만 데이터 넣을 경우는  
테이블명 뒤에 열이름 명시해야함

```

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

```

```

INSERT INTO hr.departments(department_id, manager_id)
VALUES(272, 200);

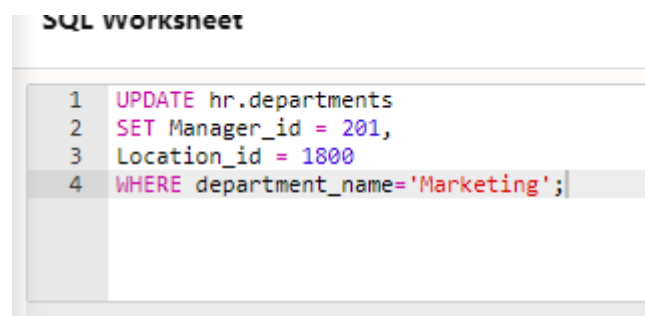
```

## 데이터 변경

```
UPDATE 테이블 이름  
SET 변경하려는 열이름 = 바꿀 데이터 값  
WHERE 조건;
```

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

hr.departments 테이블에서 department\_name의 값이 Marketing 인 값을 찾아서 Manager\_id의 값을 201로 변경하고, Location\_id의 값을 1800 변경해라



## 회원정보중 주소 변경

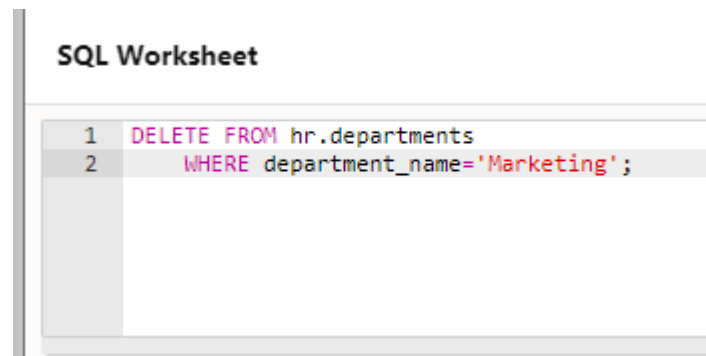
```
UPDATE 회원정보 테이블  
SET 주소 = '서울시 서초구 서초동 1번지'  
WHERE 이름 = '홍길동';
```

## 행 삭제

```
DELETE FROM 테이블 명  
WHERE 조건식;
```

- WHERE 절을 생략할 경우 테이블에 모든 데이터가 삭제

```
DELETE FROM hr.departments  
WHERE department_name='Marketing';
```



```
DELETE FROM 회원정보 테이블  
WHERE 회원_아이디 = '홍길동';
```

## 데이터베이스 생성

```
CREATE DATABASES 데이터베이스 명;
```

## 테이블 생성

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

```
필드명 데이터타입(크기));
```

- 회원 정보 테이블 생성

```
CREATE TABLE member(  
  id varchar(255),  
  name varchar(255),  
  age varchar(255),  
  addr varchar(255));
```

## 테이블 삭제

```
DROP TABLE 테이블명;
```

## 테이블 수정

### 1) 열 추가

```
ALTER TABLE 테이블명  
ADD 필드명 데이터 타입;
```

### 2) 열 삭제

```
ALTER TABLE 테이블명  
DROP COLUMN 필드명;
```

## Maria DB

### 1. Maria DB 접속

```
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: ****
```

### 2. 데이터베이스 검색

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

### 3. 데이터베이스 생성

```
MariaDB [(none)]> create database univ;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| univ |
+-----+
5 rows in set (0.001 sec)
```

#### 4. 생성한 데이터베이스 접속

```
MariaDB [(none)]> use univ;  
Database changed  
MariaDB [univ]>
```

#### 5. 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 (0.011 sec)
```

#### 6. 테이블 확인

```
MariaDB [univ]> show tables;  
+-----+  
| Tables_in_univ |  
+-----+  
| student        |  
+-----+  
1 row in set (0.001 sec)  
  
MariaDB [univ]>  
MariaDB [univ]> select * from student;  
Empty set (0.021 sec)
```

```
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 (0.010 sec)
```

## 7. 데이터 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 (0.003 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [univ]>
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)

## heidi SQL

데이터베이스 필터 테이블 필터

호스트: 127.0.0.1 데이터베이스: univ 테이블: st

**MariaDB**

- information\_schema
- mysql
- performance\_schema
- sys
- univ (32.0 KiB)
  - prof (16.0 KiB)
  - student (16.0 KiB)

univ.student: 4 행 (총) (대략적)

hakbun	name	year	dept	addr
16,001	한국인	4	컴공	서울
17,992	터키인	1	경영	터키
19,999	미국인	3	건축	LA
29,999	일본인	2	컴공	인천

## 행추가

1  
2  
3

SELECT \* FROM student;

KIB  
KIB  
KIB

student (4r × 5c)

hakbun	name	year	dept	addr
16,001	한국인	4	컴공	서울
17,992	터키인	1	경영	터키
19,999	미국인	3	건축	LA
29,999	일본인	2	컴공	인천

복사(C) Ctrl+C

Copy as ▶

붙여넣기(P) Ctrl+V

값 삽입 ▶

TEXT/BLOB 필드에 파일 삽입...

BLOB을 파일로 저장...

격자 뷰 옵션 ▶

URL 열기

+

행 삽입(I) Ins

+

행 복제 (키 미포함) Ctrl+Ins

+

행 복제 (키 포함)

테이블 추가



