

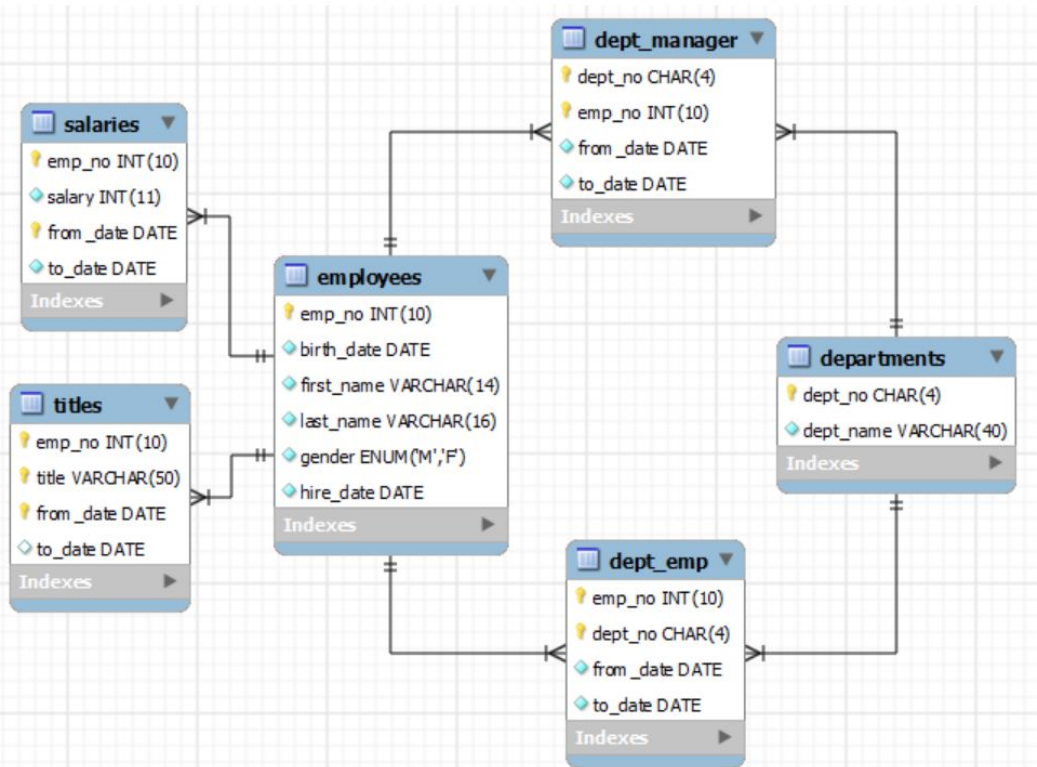
SQL Intermediate

01 연산자

SQL Intermediate

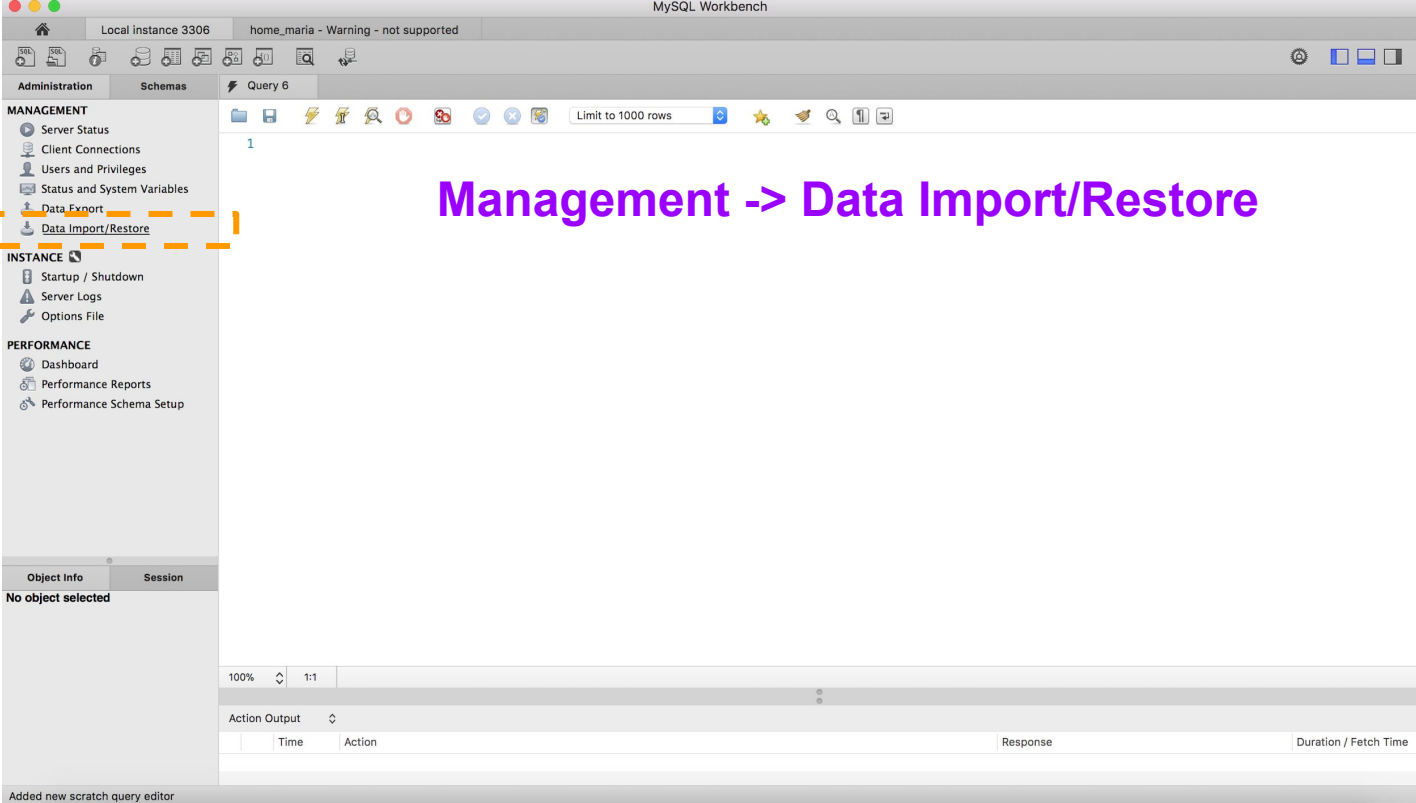
1	<ul style="list-style-type: none">1. 연산자<ul style="list-style-type: none">1. 비교 연산자2. 산술3. 논리연산자4. 기타연산자 (Like : 특정 패턴을 검색하기 위한 연산자) 와일드카드 : 문자열 내에서 임의의 문자나 문자열을 대체하기 위해서 사용하는 기호2. 타입변환3. 기타함수
1	<ul style="list-style-type: none">1. Alias2. Joins<ul style="list-style-type: none">a. INNER JOINb. OUTER JOINc. LEFT JOINd. RIGHT JOINT
1	<ul style="list-style-type: none">- SubQuery- Aggregate Functions
	<ul style="list-style-type: none">- Optimization 소개<ul style="list-style-type: none">- https://www.eversql.com/sql-query-optimizer?utm_source=blog
	과제 연습

오늘 활용할 테이블



- employees [직원]
- salaries [연봉]
- titles [직급]
- departments [부서]
- dept_emp [부서-직원]
- dept_manager [부서-매니저]

1. SQL파일로 테이블 및 데이터 Import



The screenshot shows the MySQL Workbench interface. The left sidebar contains a tree view with categories: MANAGEMENT, INSTANCE, and PERFORMANCE. Under MANAGEMENT, 'Data Import/Restore' is highlighted with an orange dashed box. The main area displays a query editor with a single line '1'. A large purple text overlay reads 'Management -> Data Import/Restore'. The bottom status bar shows 'No object selected' and 'Added new scratch query editor'.

MySQL Workbench

Local instance 3306 home_maria - Warning - not supported

Administration Schemas Query 6

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore**

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Object Info Session

No object selected

1

Limit to 1000 rows

Management -> Data Import/Restore

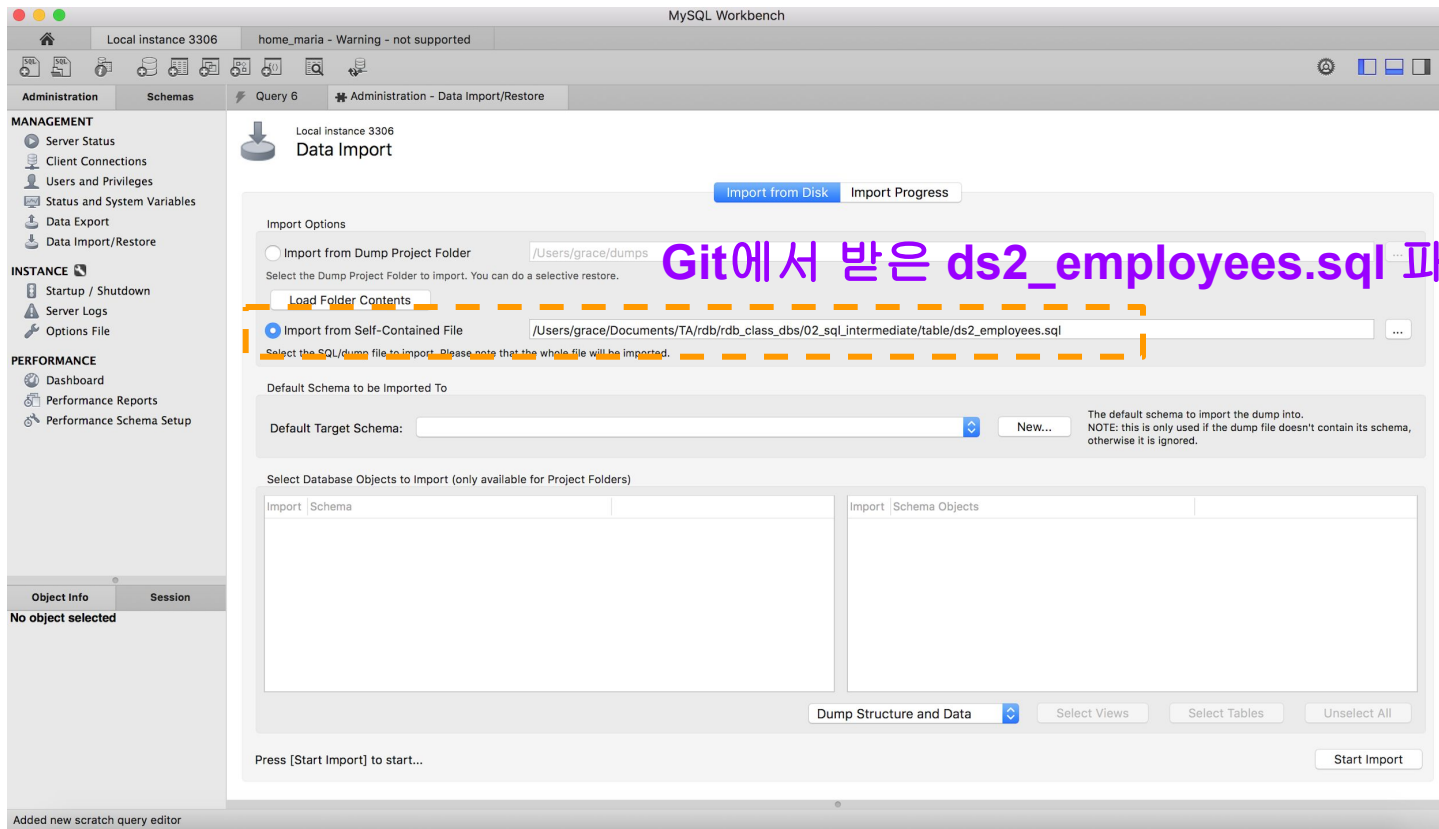
100% 1:1

Action Output

Time	Action	Response	Duration / Fetch Time
------	--------	----------	-----------------------

Added new scratch query editor

2. SQL파일로 테이블 및 데이터 Import



The screenshot shows the MySQL Workbench interface with the 'Data Import' wizard open. The 'Import from Self-Contained File' option is selected, and the file path is set to `/Users/grace/Documents/TA/rdb/rdb_class_dbs/02_sql_intermediate/table/ds2_employees.sql`. A purple text overlay reads 'Git에서 받은 ds2_employees.sql 파일 위치 선택'. The 'Default Target Schema' is set to 'home_maria'. The 'Dump Structure and Data' checkbox is checked. The 'Start Import' button is visible at the bottom right.

MySQL Workbench

Local instance 3306 home_maria - Warning - not supported

Administration Schemas Query 6 Administration - Data Import/Restore

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Object Info Session

No object selected

Local instance 3306

Data Import

Import from Disk Import Progress

Import Options

- Import from Dump Project Folder /Users/grace/dumps
- Load Folder Contents
- Import from Self-Contained File /Users/grace/Documents/TA/rdb/rdb_class_dbs/02_sql_intermediate/table/ds2_employees.sql

Select the SQL dump file to import. Please note that the whole file will be imported.

Default Schema to be Imported To

Default Target Schema: New...

The default schema to import the dump into. NOTE: this is only used if the dump file doesn't contain its schema, otherwise it is ignored.

Select Database Objects to Import (only available for Project Folders)

Import	Schema
--------	--------

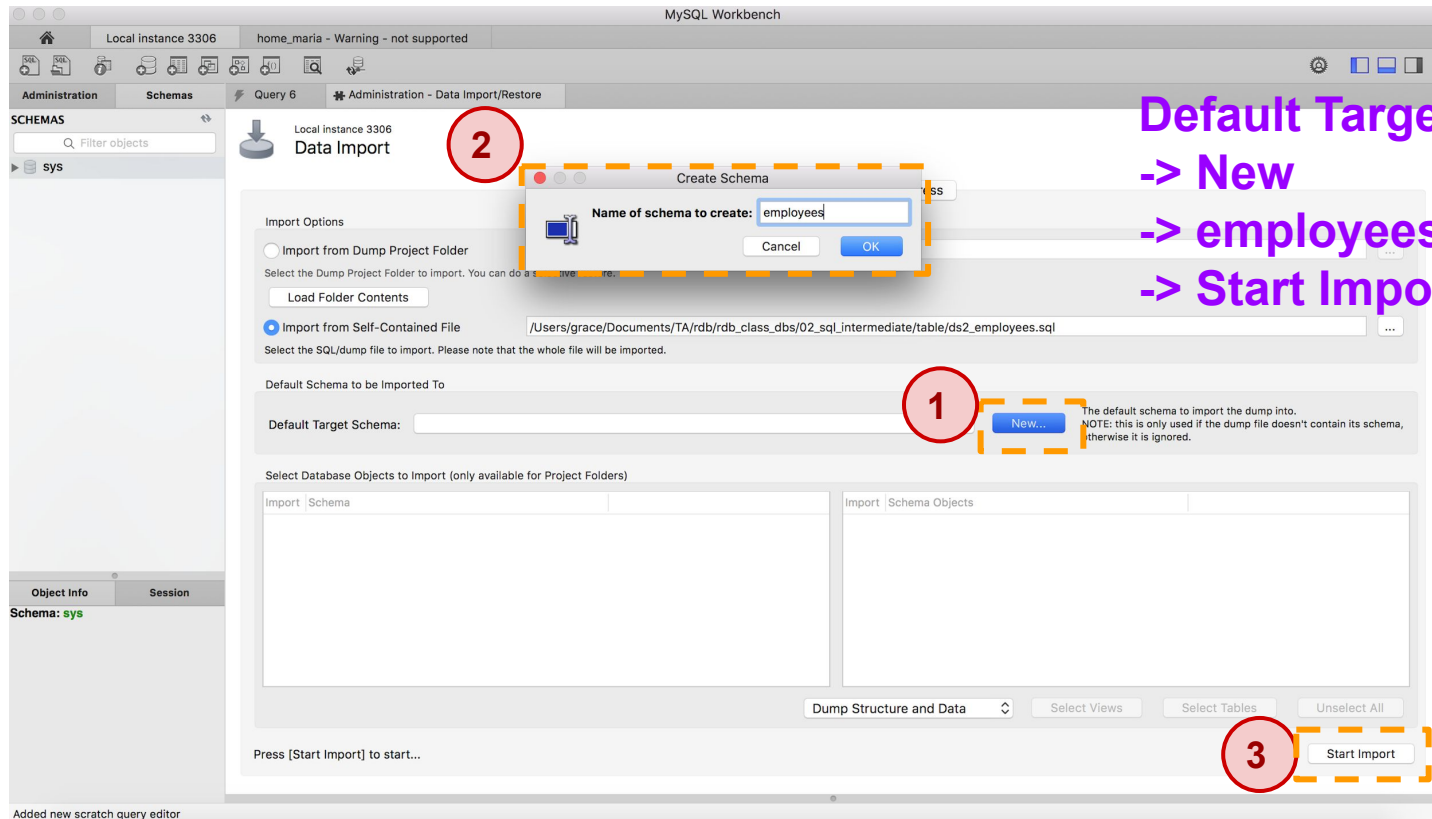
Import	Schema	Objects
--------	--------	---------

Dump Structure and Data Select Views Select Tables Unselect All

Press [Start Import] to start...

Start Import

3. SQL파일로 테이블 및 데이터 Import



Default Target Schema

-> New

-> employees 입력

-> Start Import 클릭

4. 데이터 확인

The screenshot displays the MySQL Workbench interface. The top toolbar includes icons for home, local instance, schemas, query, and administration. The main window is titled "Local instance 3306" and "home_maria - Warning - not supported". The "Administration - Data Import/Restore" tab is active. On the left, the "SCHEMAS" panel shows a search bar and a tree view with "sys" expanded. The main area shows the "Data Import" wizard for "Local instance 3306". The "Import Progress" button is highlighted. Below the progress bar, the status is "Import Completed" and "1 of 1 imported." The log shows the creation of the "employees" schema and the restoration of the "ds2_employees.sql" file. The command used for restoration is: `Running: /Applications/MySQLWorkbench.app/Contents/MacOS/mysql --defaults-file="/var/folders/gb/y7rkqw7s7_z517_ps2kqgx0000gn/T/ImprOGzz9/extraparams.cnf" --protocol=tcp --host=localhost --user=root --port=3306 --default-character-set=utf8 --comments --database=employees < "/Users/grace/Documents/TA/rdb/rdb_class_dbs/02_sql_intermediate/table/ds2_employees.sql"`. The log also indicates that the import of the file has finished.

Local instance 3306 home_maria - Warning - not supported

Administration Schemas Query 6 Administration - Data Import/Restore

SCHEMAS

Filter objects

sys

Local instance 3306

Data Import

Import from Disk Import Progress

Import Completed

Status:
1 of 1 imported.

Log:

Creating schema employees

16:56:12 Restoring /Users/grace/Documents/TA/rdb/rdb_class_dbs/02_sql_intermediate/table/ds2_employees.sql
Running: /Applications/MySQLWorkbench.app/Contents/MacOS/mysql --defaults-file="/var/folders/gb/y7rkqw7s7_z517_ps2kqgx0000gn/T/ImprOGzz9/extraparams.cnf" --protocol=tcp --host=localhost --user=root --port=3306 --default-character-set=utf8 --comments --database=employees < "/Users/grace/Documents/TA/rdb/rdb_class_dbs/02_sql_intermediate/table/ds2_employees.sql"
16:56:12 Import of /Users/grace/Documents/TA/rdb/rdb_class_dbs/02_sql_intermediate/table/ds2_employees.sql has finished

4. 데이터 확인

Local instance 3306

home_maria - Warning - not supported

Query 6 Administration - Data Import/Restore

Data Import

Import from Disk Import Progress

Import Completed

Status:
1 of 1 imported.

Log:

Creating schema employees

16:56:12 Restoring /Users/grace/Documents/TA/rdb/rdb_class_dbs/02_sql_intermediate/table/ds2_employees.sql
Running: /Applications/MySQLWorkbench.app/Contents/MacOS/mysql --defaults-file="/var/folders/gb/y7rkqw7s7_z517__ps2kqg/port=3306--default-character-set=utf8--comments--database=employees <"/Users/grace/Documents/TA/rdb/rdb_class_dbs/02_employees.sql
16:56:12 Import of /Users/grace/Documents/TA/rdb/rdb_class_dbs/02_sql_intermediate/table/ds2_employees.sql has finished

Administration Schemas

SCHEMAS

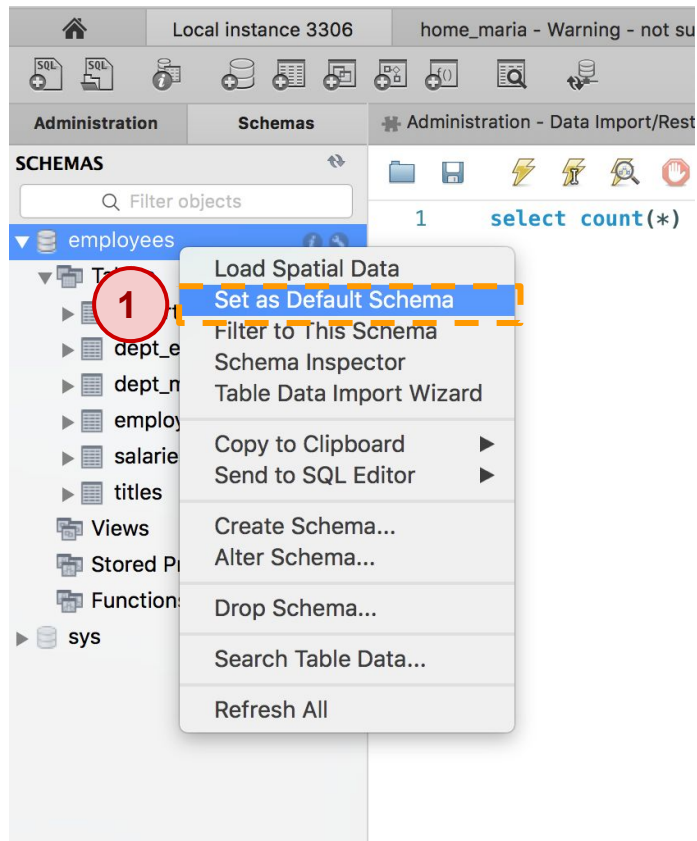
Filter objects

employees

Tables

- departments
- dept_emp
- dept_manager
- employees
- salaries
- titles

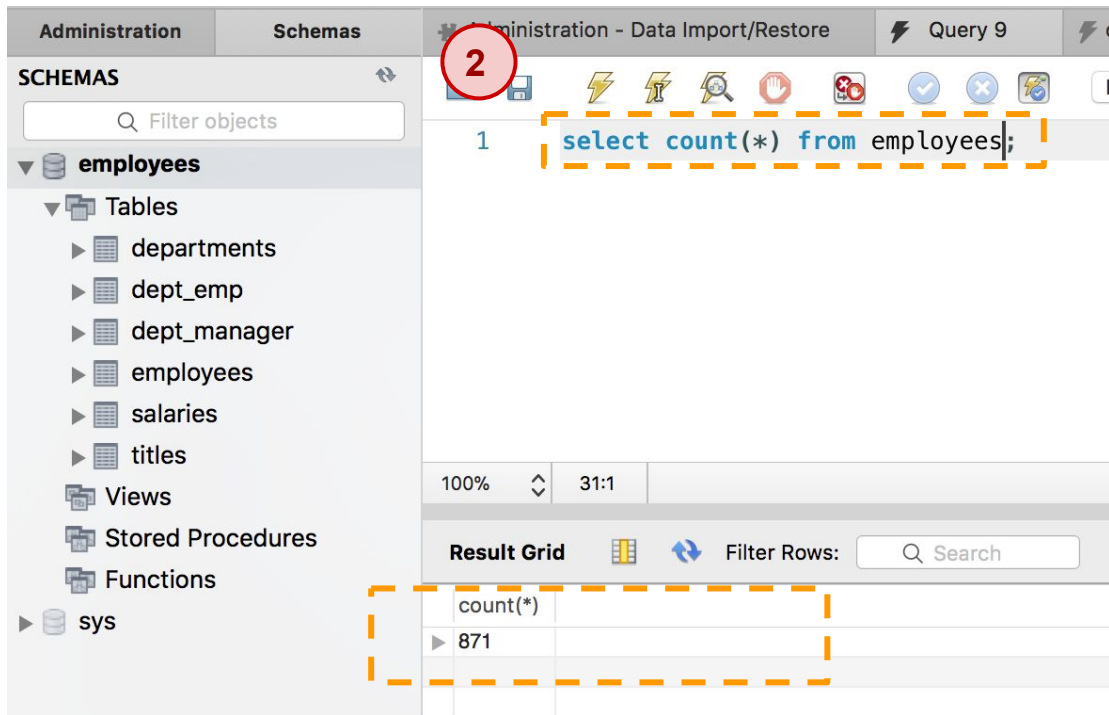
4. 데이터 확인



select count(*) from employees;

select count(*) from departments;

select count(*) from dept_manager;



1. 비교 연산자

A = B	A와 B가 같으면 True 반환
A != B A <> B	A와 B가 다르면 True 반환
A IS NULL	A값이 NULL이면 True 반환
A IS NOT NULL	A값이 NULL이 아니면 True 반환
A < B (<=)	A가 B보다 작으면 True 반환 (작거나 같으면)
A > B (>=)	A가 B보다 크면 True 반환 (크거나 같으면)
A BETWEEN min AND max	A값이 min 값보다 크거나 같고, max 값보다 작거나 같으면 True 반환
A NOT BETWEEN min AND max	A값이 min 값보다 작거나 max 크면 True 반환
A IN()	A값이 리스트에 존재하면 True 반환
A NOT IN()	A값이 리스트에 존재하지 않으면 True 반환

비교 연산자

A = B	A와 B가 같으면 True 반환
A != B A <> B	A와 B가 다르면 True 반환
A IS NULL	A값이 NULL이면 True 반환
A IS NOT NULL	A값이 NULL이 아니면 True 반환

SELECT 3=3;
SELECT 1 IS TRUE;
SELECT 0 IS FALSE;

SELECT * FROM employees
WHERE gender='M';

SELECT * FROM departments
WHERE dept_name='Development';

SELECT * FROM dept_manager
WHERE dept_no='d009';

The screenshot displays three SQL queries in the SQL Developer interface, each with its corresponding result grid. Blue dashed lines connect the queries on the left to the SQL statements in the screenshot.

Query 1: `Select * from employees where gender='M';`
Result Grid:

emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26
10003	1959-12-03	Parto	Bamford	M	1986-08-28

Query 2: `select * from departments where dept_name='Development';`
Result Grid:

dept_no	dept_name
d005	Development

Query 3: `SELECT * FROM dept_manager where dept_no!='d009';`

비교 연산자

A < B (<=)	A가 B보다 작으면 True 반환 (작거나 같으면)
A > B (>=)	A가 B보다 크면 True 반환 (크거나 같으면)
A BETWEEN min AND max	A값이 min 값보다 크거나 같고, max 값보다 작거나 같으면 True 반환
A IN()	A값이 리스트에 존재하면 True 반환

SELECT 3 BETWEEN 2 AND 7;
SELECT 5 IN(2,3,4,5);

SELECT * FROM salaries
WHERE salary > 120000;

SELECT * FROM salaries
WHERE salary BETWEEN 30000 AND 39999;

SELECT * FROM dept_manager
WHERE dept_no IN ('d001', 'd002');

19 **select * from salaries where salary > 120000;**

21 **select * from salaries where salary between 30000 and 39999;**

23 **select * from dept_manager where dept_no in ('d001', 'd002');**

Result Grid

emp_no	sal...	from_date	to_date
10282	39265	1998-05-02	1999-05-02
10048	39507	1986-02-24	1987-01-27
10048	39520	1996-04-01	1997-04-01
10406	39528	1997-01-25	1998-01-25

Result Grid

emp_no	dept_no	from_date	to_date
110022	d001	1985-01-01	1991-10-01

2. 산술연산자

DIV	Integer division
/	Division operator
-	Minus operator
%, MOD	Modulo operator
+	Addition operator
*	Multiplication operator

SELECT 25 + 5, 25 * 5, 25 DIV 2, 25 % 2;

35 • `select 25+5, 25*5, 25 DIV 2, 25 % 2;`

100% 36:35

Result Grid Filter Rows:

	25+5	25*5	25 DIV 2	25 % 2
▶	30	125	12	1

**SELECT emp_no,
YEAR(CURRENT_DATE()) - YEAR(birth_date)
FROM employees;**

39 • `select emp_no, year(current_date())-year(hire_date) from employees;`

40

100% 1:38

Result Grid Filter Rows:

	emp_no	(year(current_date())-year(hire_date))
▶	10001	33
	10002	34

3. 논리연산자

AND, &&	논리식이 모두 True 이면 True 반환
OR, 	논리식 중에서 하나라도 True 이면 True 반환
XOR	논리식이 서로 다르면 True 반환
NOT	논리식의 결과가 True 이면 거짓을, 거짓이면 True 반환
!	논리식의 결과가 True 이면 거짓을, 거짓이면 True 반환

논리 연산자

AND, &&	논리식이 모두 True이면 True 반환
OR,	논리식 중에서 하나라도 True이면 True 반환

```
SELECT * FROM dept_emp
WHERE dept_no='d001'
AND from_date >
DATE_FORMAT('2002-01-01', '%Y-%m-%d');
```

```
SELECT * FROM employees
WHERE first_name='Mary'
OR first_name='Akemi';
```

```
25 select * from dept_emp
26 where dept_no='d001'
27 and from_date> date_format('2002-01-01', '%Y-%m-%d');
```

100% 54:27

Result Grid Filter Rows: Search Edit:

emp_no	dept_no	from_date	to_date
▶ 10681	d001	2002-01-08	9999-01-01

```
29 select * from employees
30 where first_name='Mary' or first_name='Akemi';
```

100% 47:30

Result Grid Filter Rows: Search Edit:

emp_no	birth_date	first_name	last_name	gender	hire_date
▶ 10011	1953-11-07	Mary	Sluis	F	1990-01-22
10103	1953-11-26	Akemi	Birch	M	1986-12-02
10145	1956-03-30	Akemi	Esposito	F	1987-08-01

논리 연산자

NOT	논리식의 결과가 True이면 거짓을, 거짓이면 True 반환
!	논리식의 결과가 True이면 거짓을, 거짓이면 True 반환

SELECT * FROM employees
WHERE YEAR(hire_date) != 2002;

SELECT * FROM dept_emp
WHERE dept_no
NOT IN ('d001', 'd002', 'd003');

32 • **select * from employees**

33 **where year(hire_date) != 2002;**

emp_no	birth_date	first_name	last_name	gender	hire_date	
▶ 10001	1953-09-02	Georgi	Facello	M	1986-06-26	
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21	
10003	1959-12-03	Parto	Bamford	M	1986-08-28	
10004	1954-05-01	Cherish	Kelly	M	1986-10-01	

44 • **select * from dept_emp**

45 **where dept_no not in ('d001', 'd002', 'd003');**

emp_no	dept_no	from_date	to_date	
▶ 10003	d004	1995-12-03	9999-01-01	
10004	d004	1986-12-01	9999-01-01	
10010	d004	1996-11-24	2000-06-26	

4. 기타 연산자

와이드 카드 문자	%	0개 또는 1개 이상의 여러 개 문자와 대응되는 포맷문자 bl% -> bl, black, blue, blob....
	-	1개의 문자와 대응되는 포맷문자 h_t -> hot, hat, hit
LIKE		와일드카드를 사용하여, 문자 혹은 숫자의 형태가 일치하는지 비교

기타 연산자

LIKE	와일드카드를 사용하여, 문자 혹은 숫자의 형태가 일치하는지 비교
%	0개 또는 1개 이상의 여러 개 문자와 대응되는 포맷문자
_	1개의 문자와 대응되는 포맷문자

**SELECT * FROM employees
WHERE first_name LIKE 'M%';**

41 • `select * from employees where first_name like 'M%';`

42

emp_no	birth_date	first_name	last_name	gender	hire_date	
▶ 10011	1953-11-07	Mary	Sluis	F	1990-01-22	
10020	1952-12-24	Mayuko	Warwick	M	1991-01-26	
10042	1956-02-26	Magy	Stamatiou	F	1993-03-21	

**SELECT * FROM employees
WHERE emp_no LIKE '111%';**

42 • `select * from employees where emp_no like '111%';`

43

emp_no	birth_date	first_name	last_name	gender	hire_date	
▶ 11100	1955-10-14	Berna	Cochrane	M	1986-10-13	
11102	1959-03-14	Bernardo	Carrera	M	1991-01-03	
11107	1958-06-19	Martijn	Meriste	F	1998-05-02	

**SELECT * FROM employees
WHERE emp_no LIKE '1110_';**

43 • `select * from employees where emp_no like '1110_';`

44

emp_no	birth_date	first_name	last_name	gender	hire_date	
▶ 11100	1955-10-14	Berna	Cochrane	M	1986-10-13	
11102	1959-03-14	Bernardo	Carrera	M	1991-01-03	
11107	1958-06-19	Martijn	Meriste	F	1998-05-02	
11108	1962-04-15	Danae	Grabner	F	1992-06-12	

5. 타입 변환

MySQL은 비교나 검색을 수행할 때 데이터의 타입이 서로 다를 경우, 내부적으로 타입이 같아지도록 변환하여 처리합니다.

하지만 명시적으로 타입을 변환할 수 있습니다.

- CAST(값 AS 타입) - CONVERT(값, 타입)	CHAR	문자열
	DATE	날짜
	DATETIME	날짜시간
	DECIMAL	숫자. An exact fixed-point number. The total number of digits is specified in size. (소수점 이하 갯수 지정)

타입 변환

CAST(값 AS 타입)

CONVERT(값, 타입)

SELECT CAST('2019-06-10' AS DATE);

48 • `select cast('2019-06-10' as date);`

49

cast('2019-06-10' as date)

▶ 2019-06-10

SELECT emp_no,

CAST(hire_date AS DATETIME) FROM employees;

49 • `select emp_no, cast(hire_date as datetime) from employees;`

50

emp_no cast(hire_date as datetime)

▶ 10001 1986-06-26 00:00:00

10002 1985-11-21 00:00:00

10003 1986-08-28 00:00:00

10004 1986-12-01 00:00:00

SELECT CONVERT(150, CHAR);

6. 기타 함수

CONCAT(expr1, expr2, ...) CONCAT_WS(separator, expr1, ..)	문자열 합치기
REPLACE(string, from_str, new_str)	부분문자열 변경
CURRENT_DATE(), CURRENT_TIME()	
YEAR(), MONTH()	
ADDDATE(date, INTERVAL value addunit)	시간 더하기 SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, YEAR.....
DATEDIFF(date1, date2)	시간 차 구하기

String Functions

CONCAT (expr1, expr2, ...) CONCAT_WS (separator, expr1, ..)	문자열 합치기
REPLACE (string, from_str, new_str)	부분문자열 변경
<pre>SELECT CONCAT(first_name, ' ', last_name) FROM employees WHERE first_name LIKE 'M%';</pre> <pre>SELECT CONCAT_WS(' ', first_name, last_name) FROM employees WHERE first_name LIKE 'M%';</pre> <pre>SELECT REPLACE(dept_name, 'Service', 'Care') FROM departments;</pre>	<div>45 • <pre>select concat(first_name, ' ', last_name)</pre></div> <div>46 <pre>from employees where first_name like 'M%';</pre></div> <div>47 • <pre>select concat_ws(' ', first_name, last_name)</pre></div> <div>48 <pre>from employees where first_name like 'M%';</pre></div> <div>49 <pre>select concat_ws(' ', first_name, last_name)</pre></div> <div>▶ Mary Sluis</div> <div>Mayuko Warwick</div> <div>Magy Stamatou</div> <div>Mingsen Casley</div> <div>Moss Shanhogue</div> <div>Mingsen Casley</div> <div>Moss Shanhogue</div> <div>50 • <pre>select replace(dept_name, 'Service', 'Care') from departments;</pre></div> <div>1 <pre>replace(dept_name, 'Service', 'Ca...</pre></div> <div>▶ Customer Care</div> <div>Development</div>

DateTime

CURRENT_DATE(), CURRENT_TIME()	
Year(), Month()	
ADDDATE(date, INTERVAL value addunit)	SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, YEAR.....

```
SELECT emp_no, first_name,  
YEAR(birth_date) FROM employees;
```

```
SELECT emp_no, first_name,  
MONTH(birth_date) FROM employees;
```

```
SELECT ADDDATE('2019-05-31', 30);
```

```
SELECT emp_no, hire_date,  
ADDDATE(hire_date, INTERVAL 30 DAY)  
FROM employees;
```

The screenshot shows a SQL IDE with three queries and their results. The first query selects employee details and the year of their birth date. The second query selects employee details and the month of their birth date. The third query adds 30 days to the hire date of each employee.

```
52 • select emp_no, first_name, year(birth_date) from employees;  
53 • select emp_no, first_name, month(birth_date) from employees;
```

emp_no	first_name	month(birth_date)
10001	Georgi	9
10002	Bezalel	6
10003	Parto	12
10004	Chirstian	5

```
55 • select adddate('2019-05-31', 30);
```

```
56 • select emp_no, hire_date, adddate(hire_date, INTERVAL 30 DAY) from employees;
```

emp_no	hire_date	adddate(hire_date, INTERVAL 30 DAY)
10001	1986-06-26	1986-07-26
10002	1985-11-21	1985-12-21
10003	1986-08-28	1986-09-27

More..

https://www.w3schools.com/sql/func_mysql_substr.asp

w3schools.com

THE WORLD'S LARGEST WEB D



HTML

CSS

JAVASCRIPT

SQL

PHP

BOOTSTRAP

HOW TO

PYTHON

MORE ▾

REFERENCES ▾

EXER

MySQL Functions

String Functions

ASCII

CHAR_LENGTH

CHARACTER_LENGTH

CONCAT

CONCAT_WS

FIELD

FIND_IN_SET

FORMAT

INSERT

INSTR

LCASE

LEFT

LENGTH

LOCATE

LOWER

LPAD

LTRIM

RIGHT

RPAD

MySQL SUBSTR() Function

◀ MySQL Functions ▶

Example

Extract a substring from a string (start at position 5, extract 3 characters):

```
SELECT SUBSTR("SQL Tutorial", 5, 3) AS ExtractString;
```

Try it Yourself »

Definition and Usage

The SUBSTR() function extracts a substring from a string (starting at any position).

Note: The SUBSTR() and MID() functions equals to the SUBSTRING() function.

Syntax