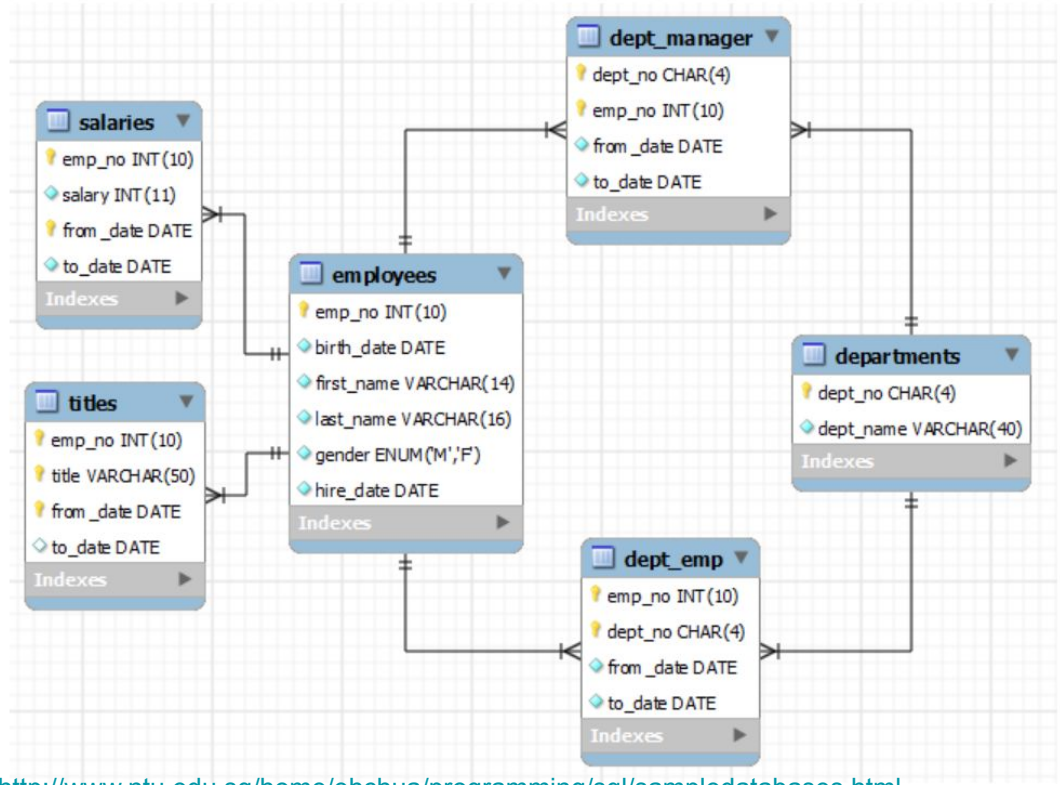
SQL Intermediate
01 연산자

SQL Intermediate	
1	**1. 연산자** **1. 비교 연산자** **2. 산술** **3. 논리연산자** **4. 기타연산자 (Like : 특정 패턴을 검색하기 위한 연산자)** **와일드카드 : 문자열 내에서 임의의 문자나 문자열을 대체하기 위해서 사용하는 기호** **2. 타입변환** **3. 기타함수**
1	- Alias - Joins (Natural, Inner/Outer) -- https://www.w3schools.com/sql/sql_join.asp - -- http://tcpschool.com/mysql/mysql_multipleTable_join - SubQuery
1	- Aggregate Functions
	- Optimization 소개 - https://www.eversql.com/sql-query-optimizer?utm_source=blog
	과제 연습

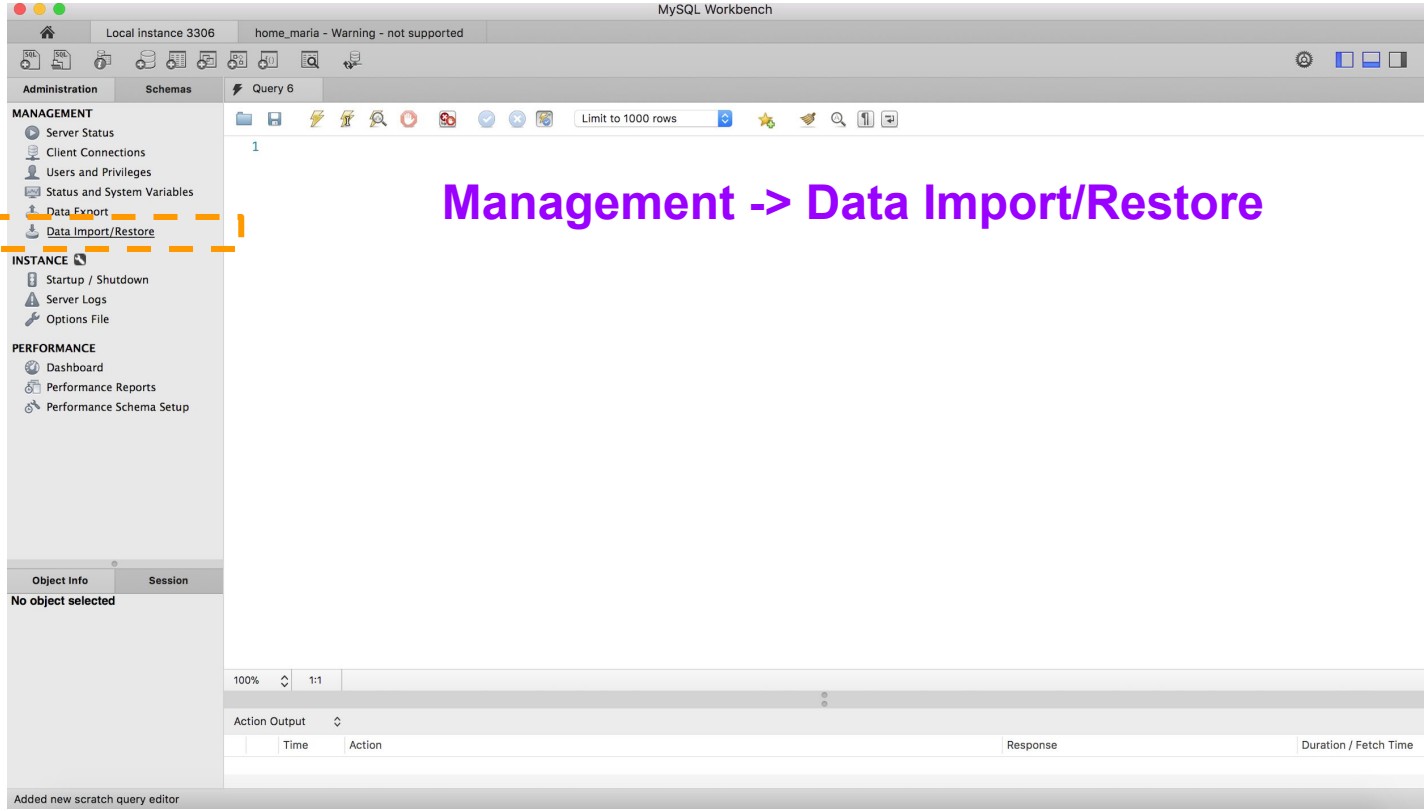
오늘 활용할 테이블



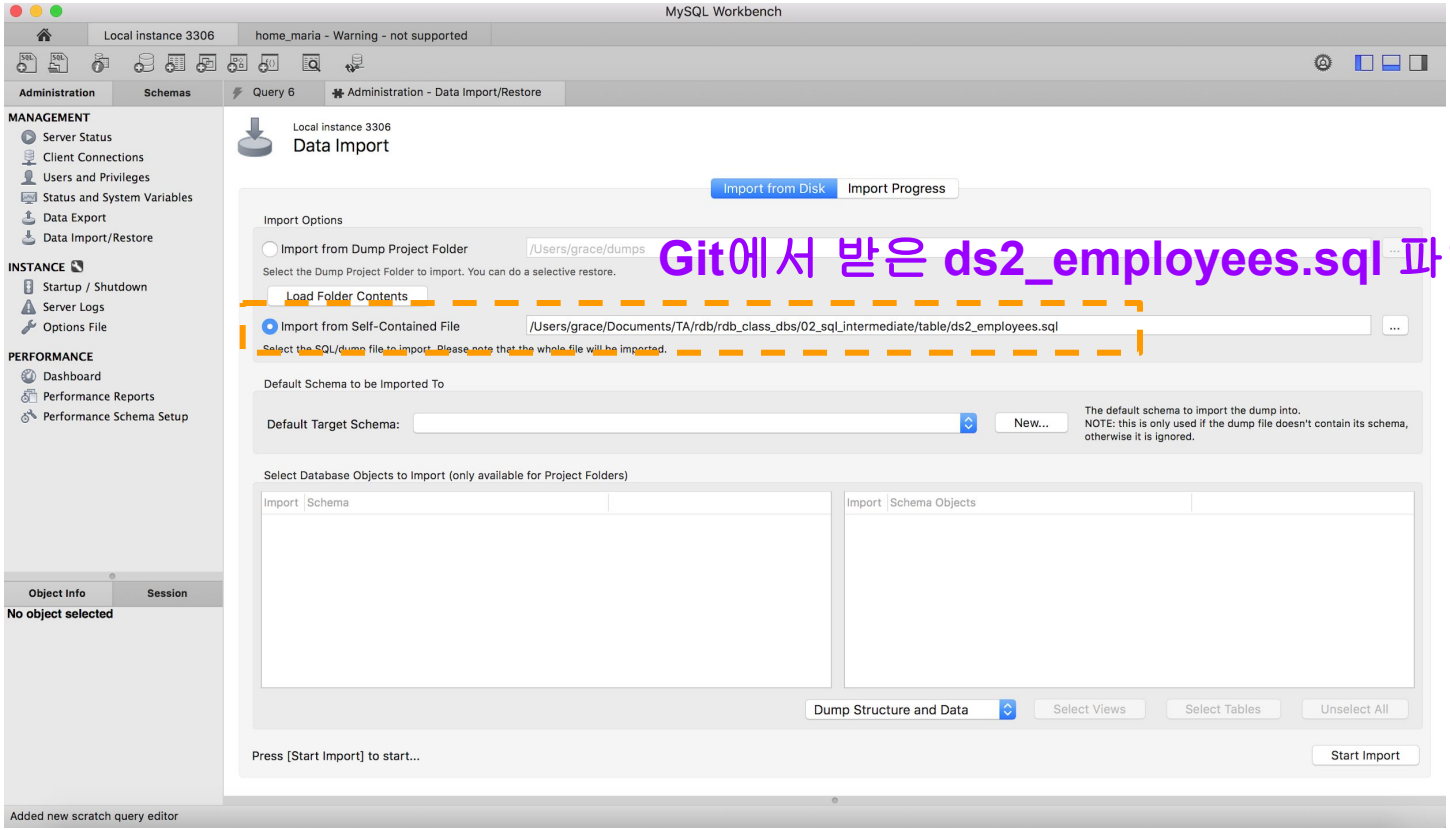
http://www.ntu.edu.sg/home/ehchua/programming/sql/sampledatabases.html

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

1. SQL파일로 테이블 및 데이터 Import

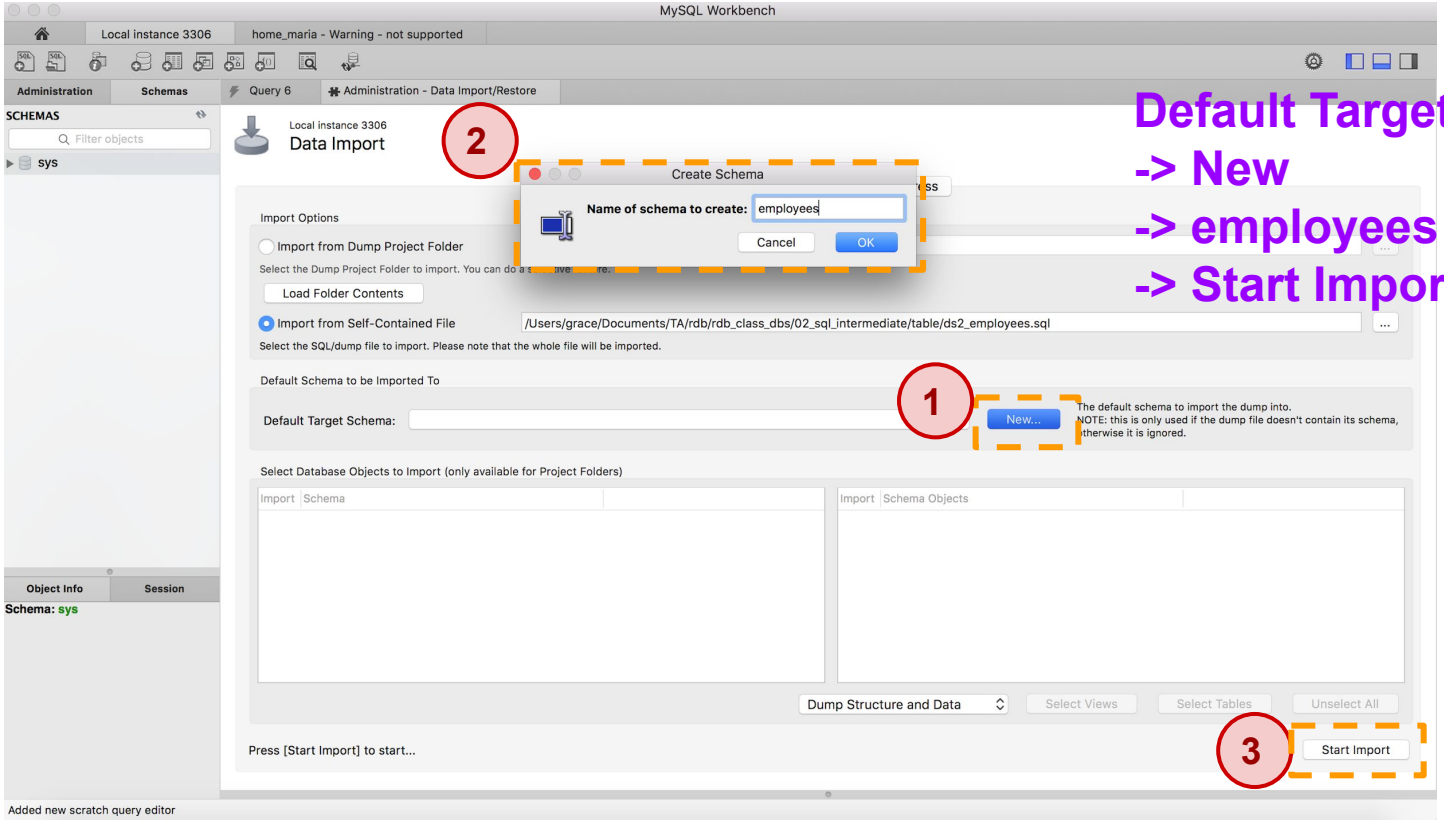


2. SQL파일로 테이블 및 데이터 Import



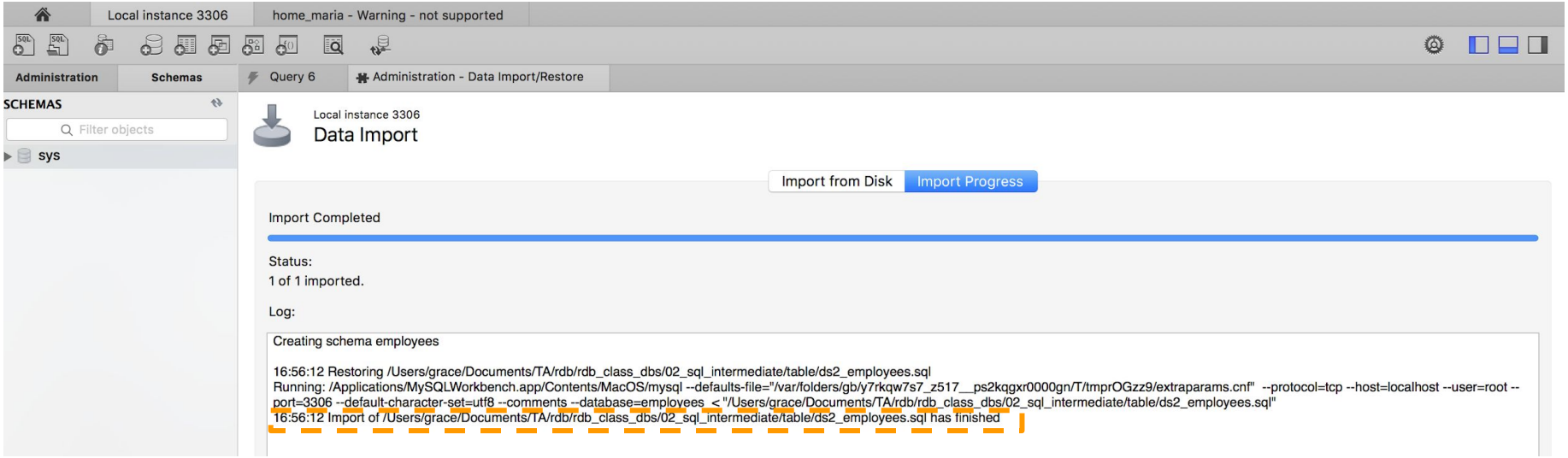
Git에서 받은 ds2_employees.sql 파일 위치 선택

3. SQL파일로 테이블 및 데이터 Import

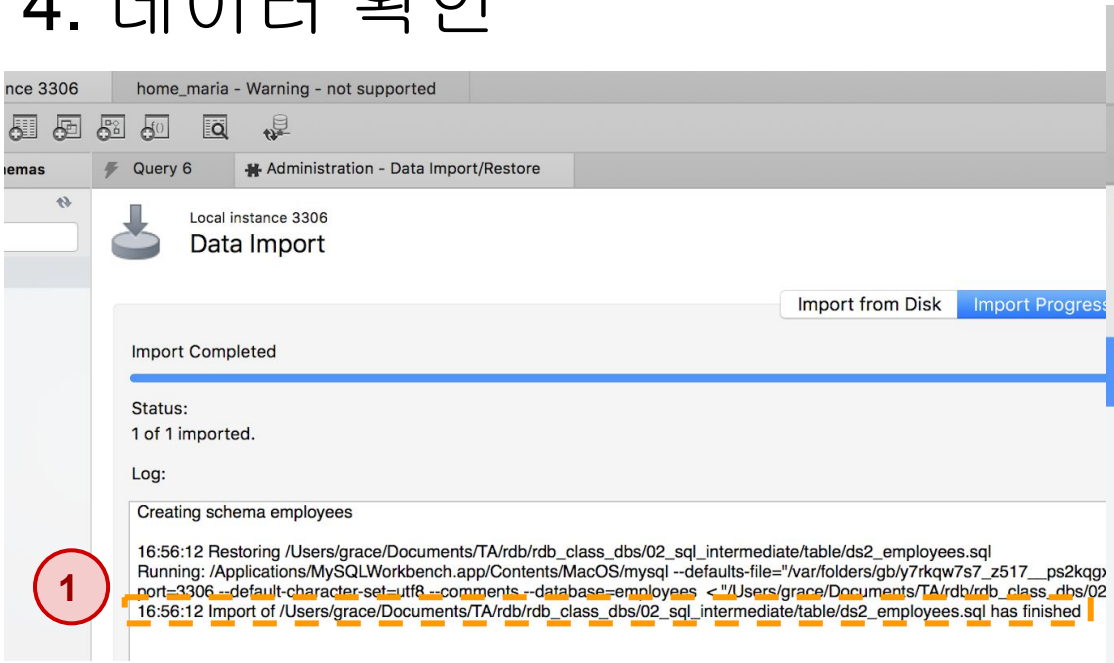


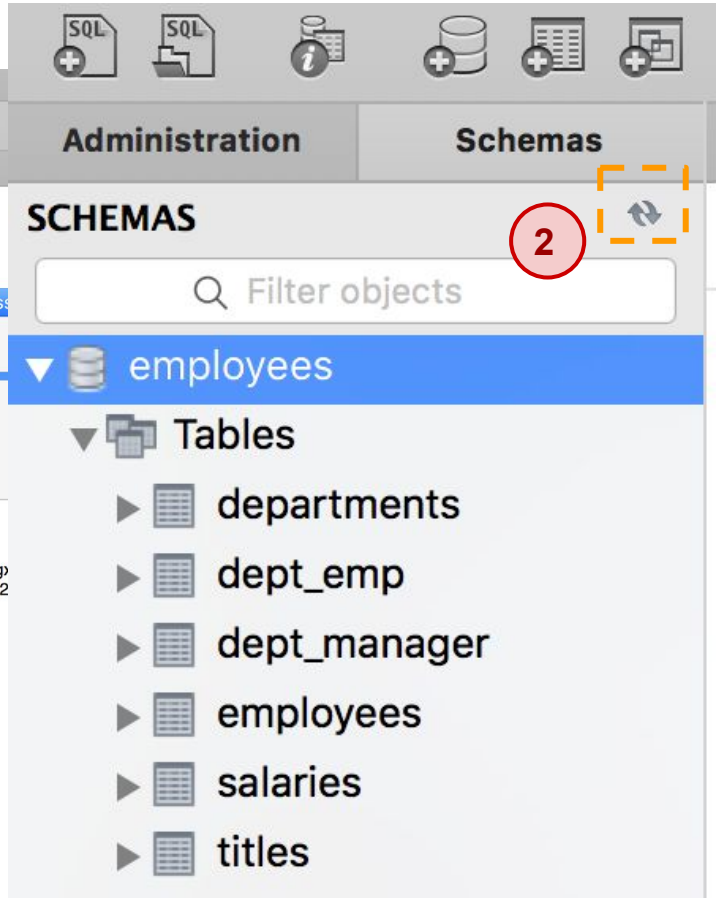
Default Target Schema
-> New
-> employees 입력
-> Start Import 클릭

4. 데이터 확인

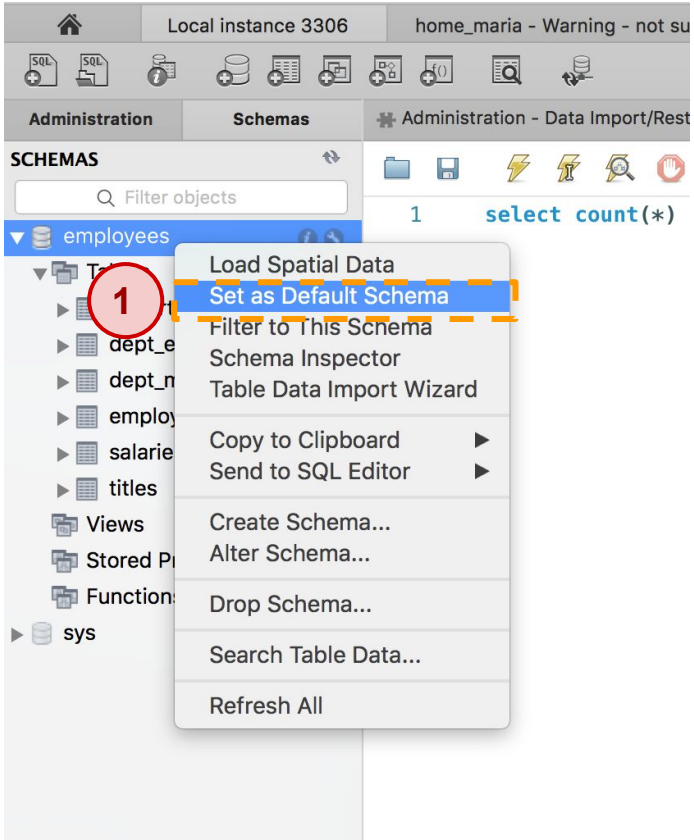

Local instance 3306
home_maria - Warning - not supported
Administration
Schemas
Query 6
Administration - Data Import/Restore
SCHEMAS
Filter objects
sys
Local instance 3306
Data Import
Import from Disk
Import Progress
Import Completed
Status:
1 of 1 imported.
Log:
Creating schema employees
16:56:12 Restoring /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__ps2kqgxr0000gn/T/tmprOGzz9/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


nce 3306
home_maria - Warning - not supported
Query 6
Administration - Data Import/Restore
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__ps2kqg
port=3306 --default-character-set=utf8 --comments --database=employees < "/Users/grace/Documents/TA/rdb/rdb_class_dbs/02
16:56:12 Import of /Users/grace/Documents/TA/rdb/rdb_class_dbs/02_sql_intermediate/table/ds2_employees.sql has finished
1

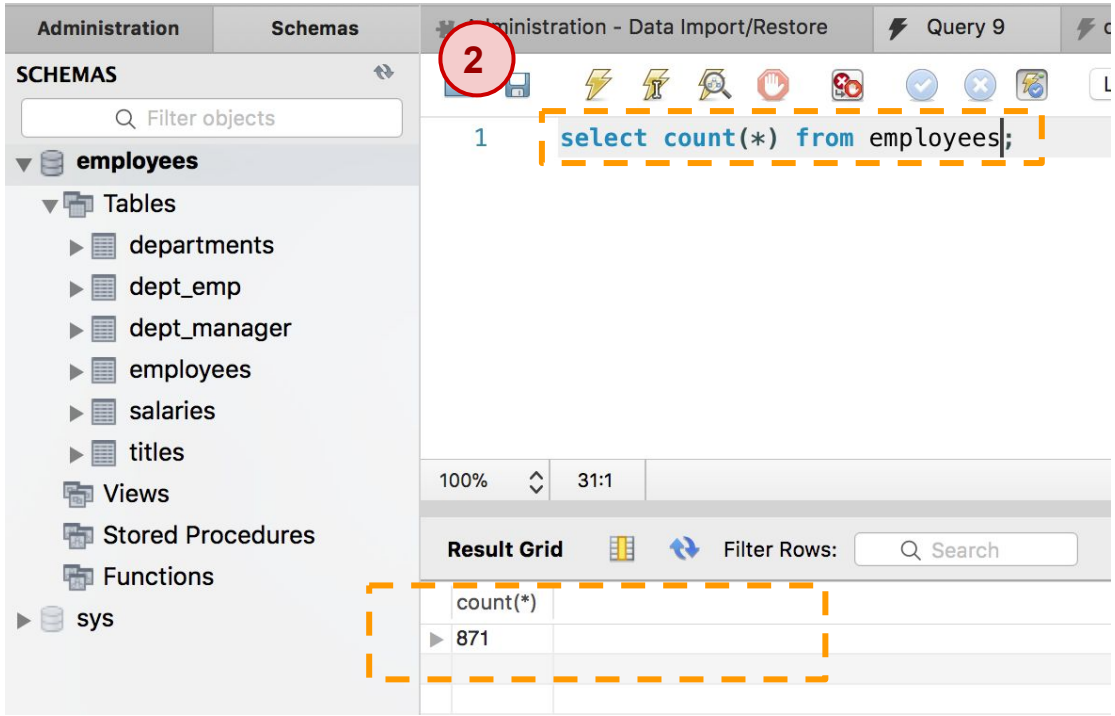

Administration
Schemas
SCHEMAS
2
Filter objects
employees
Tables
departments
dept_emp
dept_manager
employees
salaries
titles

4. 데이터 확인



select count(*) from employees;

select count(*) from departments;
select count(*) from dept_manager;



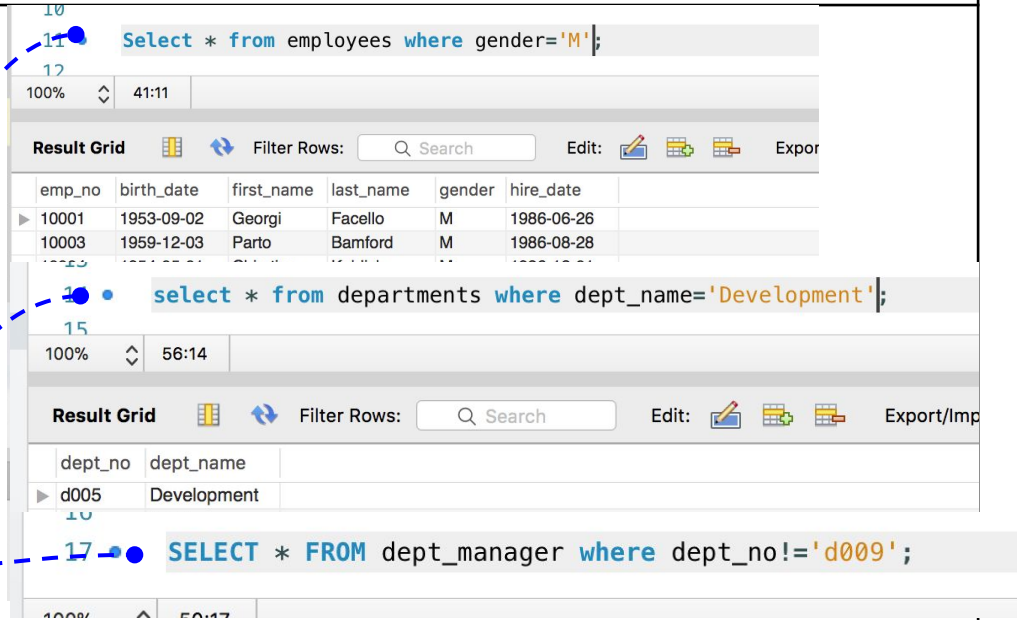
1. 비교 연산자

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

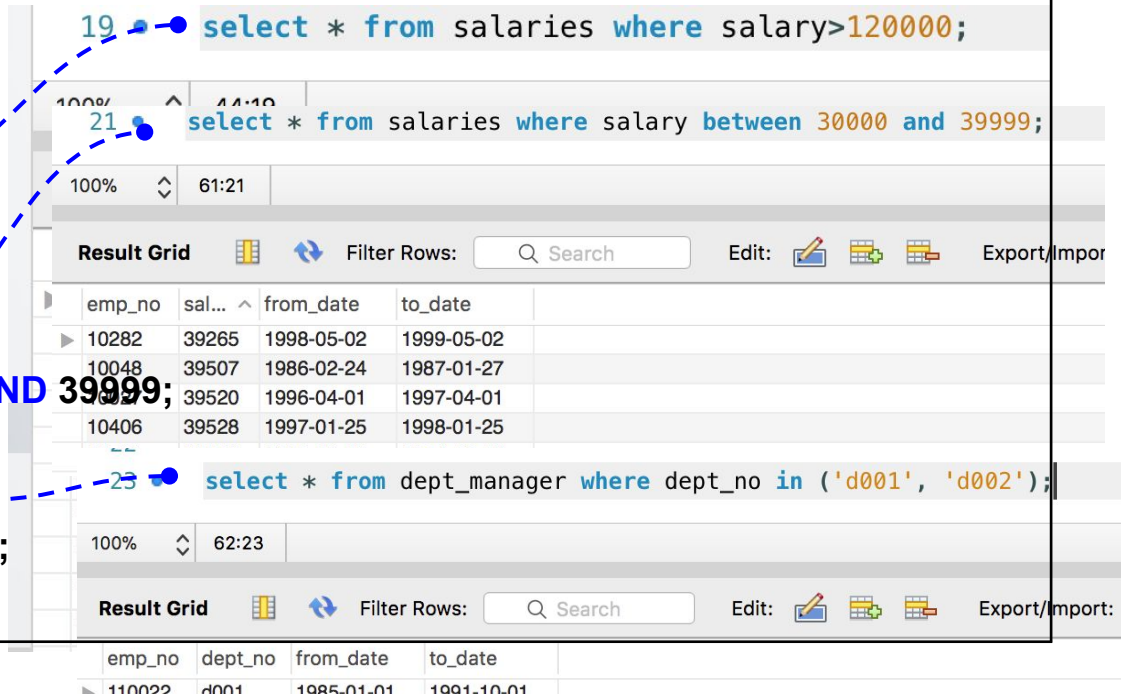
```
SELECT 3=3;
SELECT 1 IS TRUE;
SELECT 0 IS FALSE:

SELECT * FROM employees
WHERE gender='M';

SELECT * FROM departments
WHERE dept_name='Development';

SELECT * FROM dept_manager
WHERE dept_no='d009';
```



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

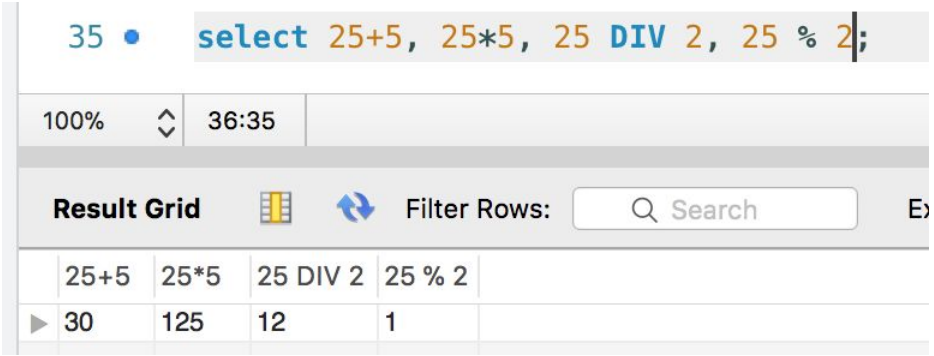
```
SELECT 3 BETWEEN 2 AND 7;
SELECT 5 IN(2,3,4,5);

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');
```



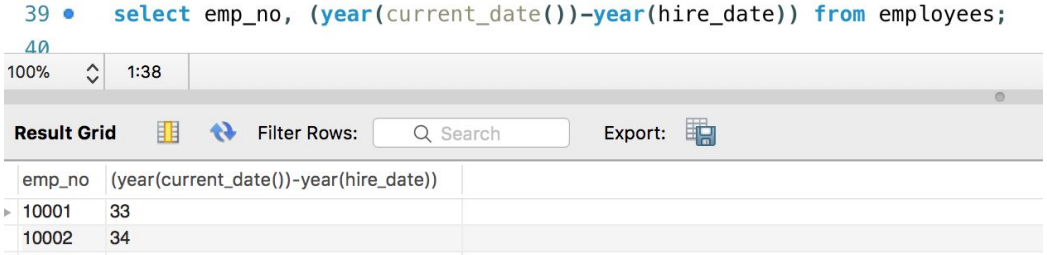
2. 산술연산자

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

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




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



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');
```

```
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

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

```
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 **True**이면 거짓을, 거짓이면 **True** 반환
!	논리식의 결과가 **True**이면 거짓을, 거짓이면 **True** 반환

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

```
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

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

```
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. 기타연산자

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. 타입변환

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)

타입변환

CAST(값 AS 타입)	
CONVERT(값, 타입)	

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

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

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;
```

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. 기타함수

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

```
SELECT CONCAT(first_name, ' ', last_name)
FROM employees
WHERE first_name LIKE 'M%';

SELECT CONCAT_WS(' ', first_name, last_name)
FROM employees
WHERE first_name LIKE 'M%';

SELECT
REPLACE(dept_name, 'Service', 'Care')
FROM departments;
```

```
45  select concat(first_name, ' ', last_name)
46  from employees where first_name like 'M%';
47
47  select concat_ws(' ', first_name, last_name)
48  from employees where first_name like 'M%';
49
```

concat_ws(' ', first_name, last_name)
Mary Sluis
Mayuko Warwick
Magy Stamatiou
Mingsen Casley
Moss Shanbhogue
Mingsen Casley
Moss Shanbhogue

```
50  select replace(dept_name, 'Service', 'Care') from departments;
```

replace(dept_name, 'Service', 'Ca...
Customer Care
Development

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;

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

100% 34:55

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

100% 78:55

Result Grid Filter Rows: Search Export:

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