SQL Intermediate
02 JOIN

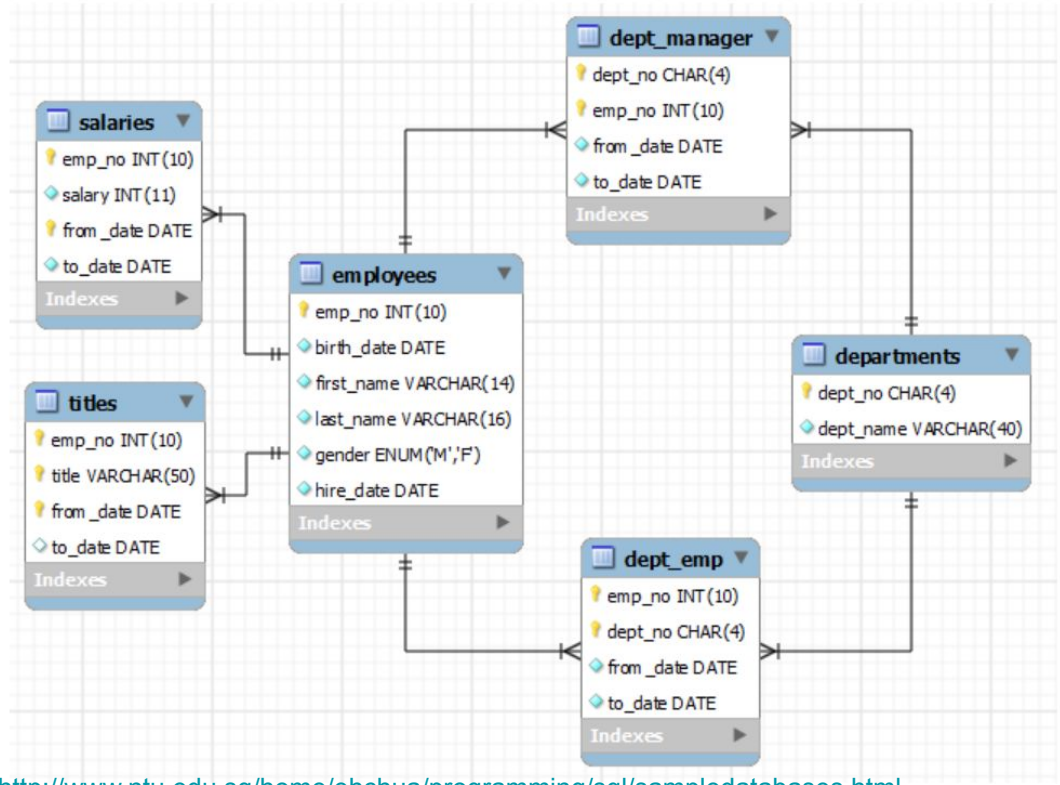
SQL Intermediate	

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	**1. Alias**
	2. Joins
	  a. INNER JOIN
	  b. OUTER JOIN
	  c. LEFT JOIN
	  d. RIGHT JOINT

1	1. SubQuery 2. Aggregate Functions
.5	1. Optimization 소개 a. EXPLAIN
	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. Alias

테이블, 컬럼 등에 별칭(별명)을 붙인다. Alias는 해당 **SQL**에서만 유효하다.
하나의 **SQL문** 내에서 테이블 이름과 별명을 혼용할 수 없다.

- 컬럼명이 길어서 바꿔서 출력하고 싶은 경우 사용.
- 조인 시, 테이블에 별칭을 붙여서 조회 할 데이터가 어느 테이블에 있는지 쉽게 알수 있도록 사용. (에러방지)

```
SELECT concat(first_name, ' ', last_name) AS emp_name FROM employees;
```

```
1   select concat(first_name, ' ', last_name) as emp_name from employees;
```

emp_name
Georgi Facello
Bezalel Simmel
Parto Bamford
Chirstian Koblick

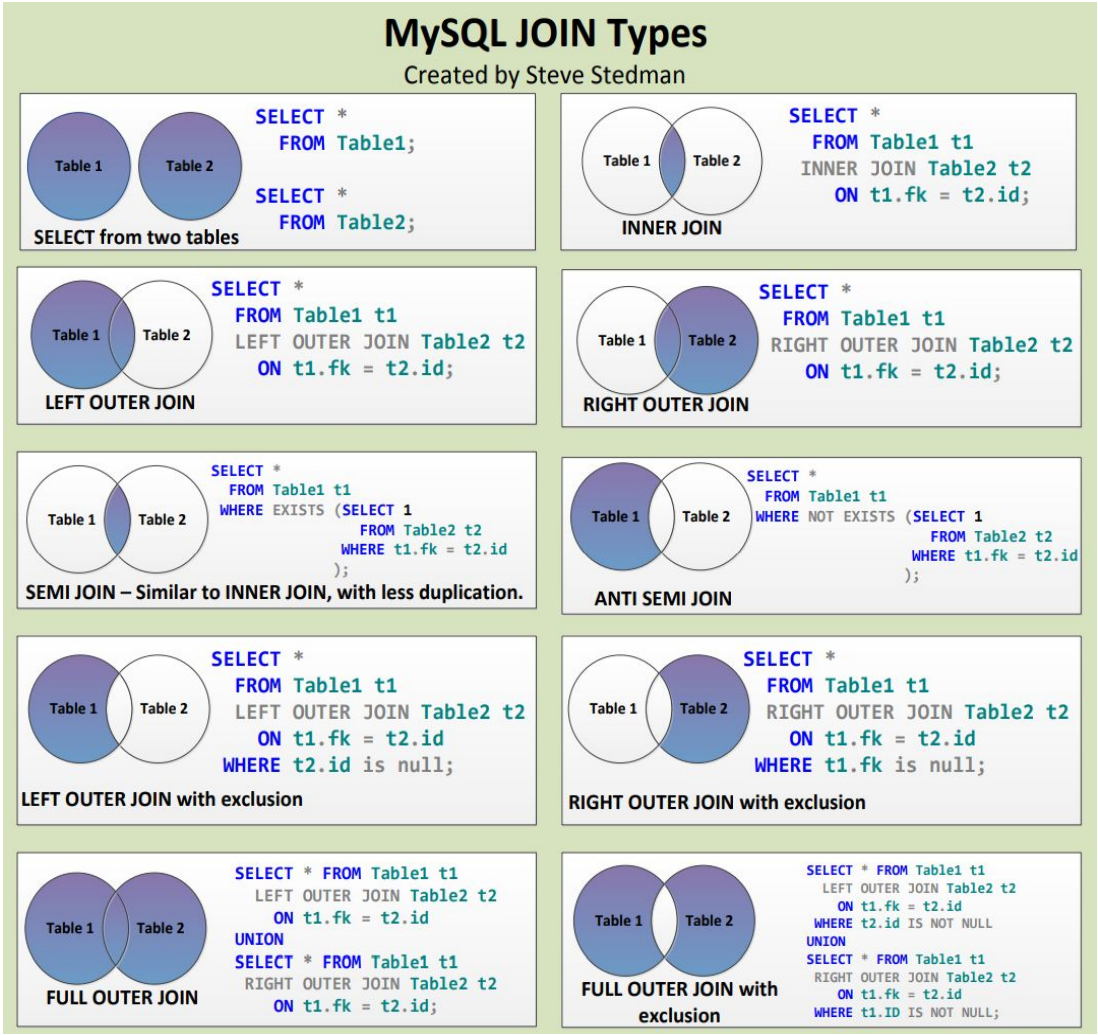
```
SELECT emp_no FROM dept_emp AS sales_emps WHERE dept_no='6007' ;
```

```
5   select emp_no from dept_emp as sales_emps where dept_no='d007';
```

2. JOIN

데이터베이스 내 여러 테이블에서의 로우를 조합하여 하나의 결과로 표현해준다.
레코드를 조합하는 방식에 따라서 크게 다음과 같이 구분한다.

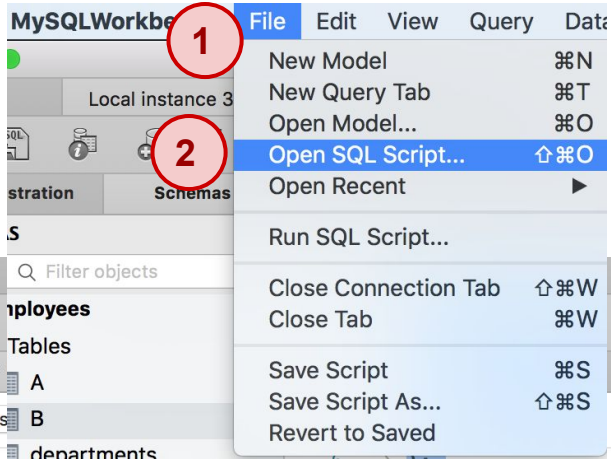
- **INNER JOIN**
- **OUTER JOIN**
- **LEFT JOIN**
- **RIGHT JOINT**

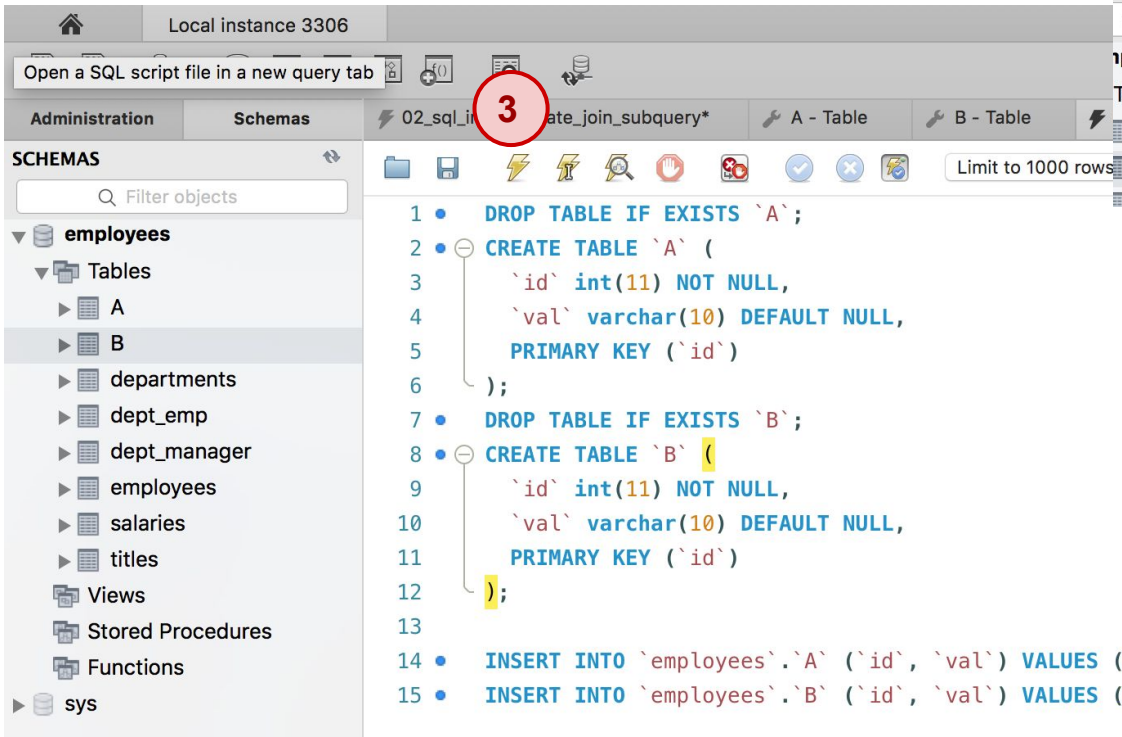


https://stevestedman.com/2015/03/mysql-join-types-poster/

JOIN

1. File > Open SQL Script
2. 02_sql_intermediate > table > **02_join_ab.sql**
3. **번개 모양 버튼 클릭**





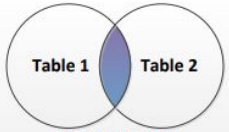
A

id	val
1	A
2	B
3	C
4	D
5	E
6	F

B

id	val
5	E
6	F
7	G
8	H
9	I
10	J

INNER JOIN




```
SELECT *
  FROM Table1 t1
 INNER JOIN Table2 t2
    ON t1.fk = t2.id;
```

SELECT * FROM A
INNER JOIN B
ON A.id = B.id;

```
7 •  select * from A inner join B on A.id=B.id;
```

id	val	id	val
5	E	5	E
6	F	6	F

SELECT * FROM A
JOIN B
ON A.id = B.id;

A

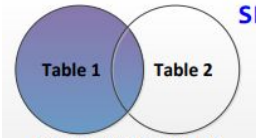
id	val
1	A
2	B
3	C
4	D
5	E
6	F

B

id	val
5	E
6	F
7	G
8	H
9	I
10	J

MySQL에서는 JOIN, INNER JOIN이 동일한 결과를 반환함.

LEFT JOIN



LEFT OUTER JOIN

```
SELECT *
  FROM Table1 t1
  LEFT OUTER JOIN Table2 t2
    ON t1.fk = t2.id;
```

SELECT * FROM A
LEFT OUTER JOIN B
ON A.id = B.id;

```
11 • select * from A left outer join B on A.id=B.id;
```

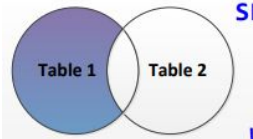
id	val	id	val
1	A	NULL	NULL
2	B	NULL	NULL
3	C	NULL	NULL
4	D	NULL	NULL
5	E	5	E
6	F	6	F

A

id	val
1	A
2	B
3	C
4	D
5	E
6	F

B

id	val
5	E
6	F
7	G
8	H
9	I
10	J



LEFT OUTER JOIN with exclusion

```
SELECT *
  FROM Table1 t1
  LEFT OUTER JOIN Table2 t2
    ON t1.fk = t2.id
  WHERE t2.id is null;
```

SELECT * FROM A
LEFT OUTER JOIN B
ON A.id = B.id
WHERE B.id is null;

```
12 • select * from A left outer join B on A.id=B.id where B.id is null;
```

id	val	id	val
1	A	NULL	NULL
2	B	NULL	NULL
3	C	NULL	NULL
4	D	NULL	NULL

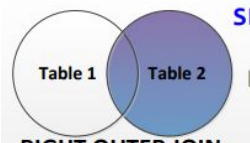
A

id	val
1	A
2	B
3	C
4	D
5	E
6	F

B

id	val
5	E
6	F
7	G
8	H
9	I
10	J

RIGHT JOIN




```
SELECT *
  FROM Table1 t1
RIGHT OUTER JOIN Table2 t2
  ON t1.fk = t2.id;
```

RIGHT OUTER JOIN

SELECT * FROM A
RIGHT OUTER JOIN B
ON A.id = B.id;

```
17 •  select * from A right outer join B on A.id=B.id;
```

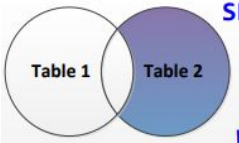
id	val	id	val
5	E	5	E
6	F	6	F
NULL	NULL	7	G
NULL	NULL	8	H
NULL	NULL	9	I
NULL	NULL	10	J

A

id	val
1	A
2	B
3	C
4	D
5	E
6	F

B

id	val
5	E
6	F
7	G
8	H
9	I
10	J




```
SELECT *
  FROM Table1 t1
RIGHT OUTER JOIN Table2 t2
  ON t1.fk = t2.id
WHERE t1.fk is null;
```

RIGHT OUTER JOIN with exclusion

SELECT * FROM A
RIGHT OUTER JOIN B
ON A.id = B.id
WHERE A.id is null;

```
19 •  select * from A right outer join B on A.id=B.id where A.id is null;
```

id	val	id	val
NULL	NULL	7	G
NULL	NULL	8	H
NULL	NULL	9	I
NULL	NULL	10	J

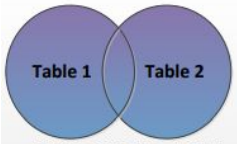
A

id	val
1	A
2	B
3	C
4	D
5	E
6	F

B

id	val
5	E
6	F
7	G
8	H
9	I
10	J

FULL JOIN = LEFT JOIN + RIGHT JOIN




```
SELECT * FROM Table1 t1
  LEFT OUTER JOIN Table2 t2
    ON t1.fk = t2.id
UNION
SELECT * FROM Table1 t1
  RIGHT OUTER JOIN Table2 t2
    ON t1.fk = t2.id;
```

```
SELECT * FROM A
LEFT OUTER JOIN  B
ON A.id = B.id
UNION
SELECT * FROM A
RIGHT OUTER JOIN  B
ON A.id = B.id;
```

A

id	val
1	A
2	B
3	C
4	D
5	E
6	F

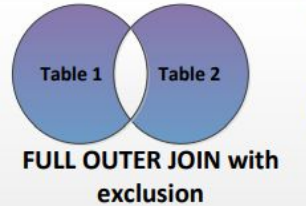
B

id	val
5	E
6	F
7	G
8	H
9	I
10	J

```
21 •  select * from A left outer join B on A.id=B.id
22    union
23    select * from A right outer join B on A.id=B.id;
```

id	val	id	val
1	A	NULL	NULL
2	B	NULL	NULL
3	C	NULL	NULL
4	D	NULL	NULL
5	E	5	E
6	F	6	F
NULL	NULL	7	G
NULL	NULL	8	H
NULL	NULL	9	I
NULL	NULL	10	J

FULL JOIN = LEFT JOIN + RIGHT JOIN




```
SELECT * FROM A
LEFT OUTER JOIN  B
ON A.id = B.id WHERE B.id IS NULL
UNION
SELECT * FROM A
RIGHT OUTER JOIN  B
ON A.id = B.id WHERE A.id IS NULL;
```

```
25 •  select * from A left outer join B on A.id=B.id where B.id is null
26    union
27    select * from A right outer join B on A.id=B.id where A.id is null;
```

100% 68:27

Result Grid | Filter Rows: Search | Export:

id	val	id	val
1	A	NULL	NULL
2	B	NULL	NULL
3	C	NULL	NULL
4	D	NULL	NULL
NULL	NULL	7	G
NULL	NULL	8	H
NULL	NULL	9	I
NULL	NULL	10	J

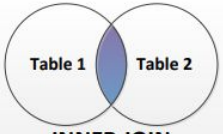
A

id	val
1	A
2	B
3	C
4	D
5	E
6	F

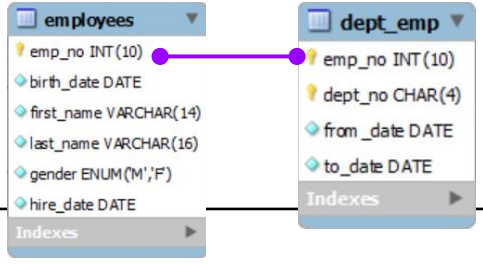
B

id	val
5	E
6	F
7	G
8	H
9	I
10	J

= JOIN




```
SELECT *
  FROM Table1 t1
INNER JOIN Table2 t2
   ON t1.fk = t2.id;
```



1. 직원 정보와 부서 번호를 함께 보고 싶다.

SELECT * FROM employees AS e INNER JOIN dept_emp AS d ON e.emp_no = d.emp_no;

```
33 •  select * from employees as e inner join dept_emp as d on e.emp_no=d.emp_no;
```

emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	dept_no	from_date	to_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26	10001	d005	1986-06-26	9999-01-01
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21	10002	d007	1996-08-03	9999-01-01
10003	1959-12-03	Parto	Bamford	M	1986-08-28	10003	d004	1995-12-03	9999-01-01
10004	1954-05-01	Chirstian	Koblick	M	1986-12-01	10004	d004	1986-12-01	9999-01-01
10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12	10005	d003	1989-09-12	9999-01-01
10006	1953-04-20	Anneke	Preusig	F	1989-06-02	10006	d005	1990-08-05	9999-01-01
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10	10007	d008	1989-02-10	9999-01-01

2. 여러 부서를 거친 직원들은 어떻게 반영되었을까.

- 여러부서 기록이 있는 직원 **emp_no : 10817**

```
37 •  select * from dept_emp where emp_no=10817;
```

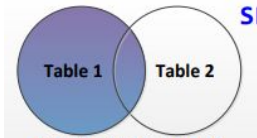
emp_no	dept_no	from_date	to_date
10817	d007	1990-12-26	2000-01-24
10817	d009	2000-01-24	9999-01-01

SELECT * FROM employees AS e INNER JOIN dept_emp AS d ON e.emp_no = d.emp_no WHERE e.emp_no=10817;

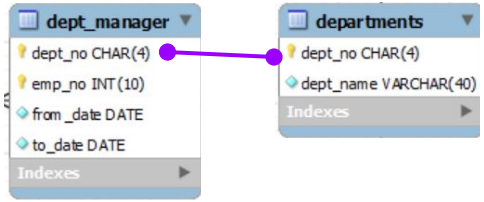
```
38 •  select * from employees as e inner join dept_emp as d on e.emp_no=d.emp_no where e.emp_no=10817;
```

emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	dept_no	from_date	to_date
10817	1958-10-02	Uri	Rullman	F	1990-12-26	10817	d007	1990-12-26	2000-01-24
10817	1958-10-02	Uri	Rullman	F	1990-12-26	10817	d009	2000-01-24	9999-01-01

= JOIN




```
SELECT *
  FROM Table1 t1
  LEFT OUTER JOIN Table2 t2
    ON t1.fk = t2.id;
```



3. 새로운 부서가 개설되었다.

```
INSERT INTO departments values ('d000', 'Task Force');
```

dept_no	dept_name
d009	Customer Service
d005	Development
d002	Finance
d003	Human Resources
d001	Marketing
d004	Production
d006	Quality Management
d008	Research
d007	Sales
d000	Task Force

4. 아직 새로운 부서의 매니저는 결정되지 않았다.
부서를 기준으로 매니저들을 보고 싶다.

```
SELECT * FROM departments as d
LEFT OUTER JOIN dept_manager as m
ON d.dept_no = m.dept_no;
```

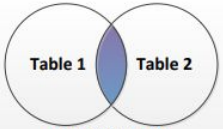
```
43 •  select * from departments as d left outer join dept_manager as m on d.dept_no=m.dept_no;
44
```

100% 89:43

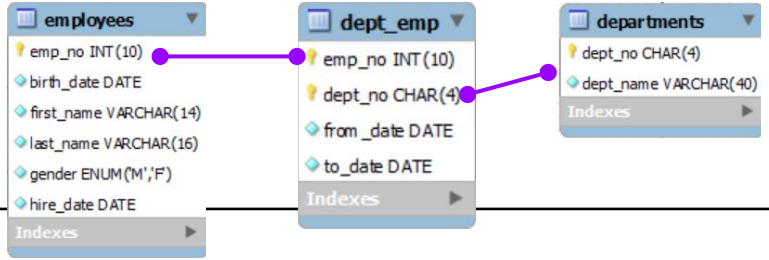
Result Grid | Filter Rows: Search | Export:

dept_...	dept_name	emp_no	dept_no	from_date	to_date
d000	Task Force	NULL	NULL	NULL	NULL
d001	Marketing	110022	d001	1985-01-01	1991-10-01
d001	Marketing	110039	d001	1991-10-01	9999-01-01
d002	Finance	110085	d002	1985-01-01	1989-12-17
d002	Finance	110114	d002	1989-12-17	9999-01-01
d003	Human Resources	110183	d003	1985-01-01	1992-03-21
d003	Human Resources	110228	d003	1992-03-21	9999-01-01
d004	Production	110303	d004	1985-01-01	1988-09-09

= JOIN




```
SELECT *
  FROM Table1 t1
INNER JOIN Table2 t2
   ON t1.fk = t2.id;
```



5. 직원-부서명을 함께 보고 싶다.

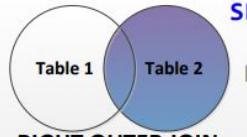
```
SELECT * FROM dept_emp AS de
INNER JOIN employees AS e ON de.emp_no = e.emp_no
INNER JOIN department AS d ON de.dept_no = d.dept_no;
```

```
44 •  select * from dept_emp as de
45    inner join employees as e on de.emp_no=e.emp_no
46    inner join departments as d on de.dept_no=d.dept_no;
```

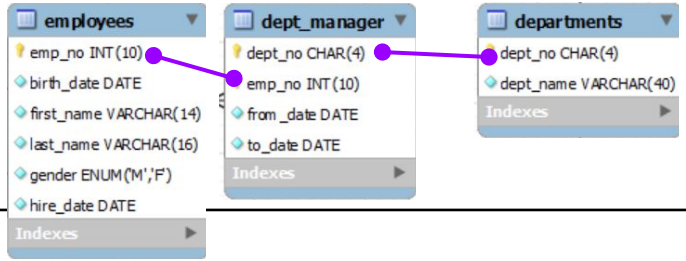
emp_no	dept_no	from_date	to_date	emp_no	birth_date	first_name	last_name	gender	hire_date	dept_no	dept_name
10049	d009	1992-05-04	9999-01-01	10049	1961-04-24	Basil	Tramer	F	1992-05-04	d009	Customer Service
10060	d009	1992-11-11	9999-01-01	10060	1961-10-15	Breannda	Billingsley	M	1987-11-02	d009	Customer Service
10088	d009	1992-03-21	9999-01-01	10088	1954-02-25	Jungsoon	Syrzycki	F	1988-09-02	d009	Customer Service
10098	d009	1989-06-29	1992-12-11	10098	1961-09-23	Sreekrishna	Servieres	F	1985-05-13	d009	Customer Service

```
SELECT * FROM dept_emp AS de
INNER JOIN employees AS e ON de.emp_no = e.emp_no
INNER JOIN department AS d ON de.dept_no = d.dept_no
WHERE de.emp_no=10817;
```

= JOIN




```
SELECT *
  FROM Table1 t1
RIGHT OUTER JOIN Table2 t2
  ON t1.fk = t2.id;
```



6. 부서-매니저의 상세 정보까지 보고 싶다.

```
SELECT * FROM dept_manager as dm
RIGHT OUTER JOIN departments as m ON dm.dept_no = d.dept_no
INNER JOIN employees as e ON dm.emp_no = e.emp_no;
```

```
56 •  select * from dept_manager as dm
57    right outer join departments as d on dm.dept_no=d.dept_no
58    inner join employees as e on dm.emp_no=e.emp_no;
```

emp_no	dept_no	from_date	to_date	dept_no	dept_name	emp_no	birth_date	first_name	last_name	gender	hire_date
111692	d009	1985-01-01	1988-10-17	d009	Customer Service	111692	1954-10-05	Tonny	Butterworth	F	1985-01-01
111784	d009	1988-10-17	1992-09-08	d009	Customer Service	111784	1956-06-14	Marjo	Giarratana	F	1988-02-12
111877	d009	1992-09-08	1996-01-03	d009	Customer Service	111877	1962-10-18	Xiaobin	Spinelli	F	1991-08-17

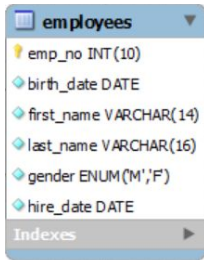
7. 새로 추가한 부서 정보가 빠졌다. 새로 추가한 부서 정보까지 함께 출력하고 싶다.

```
SELECT * FROM dept_manager as dm
RIGHT OUTER JOIN departments as m ON dm.dept_no = d.dept_no
LEFT JOIN employees as e ON dm.emp_no = e.emp_no;
```

```
61 •  select * from dept_manager as dm
62    right outer join departments as d on dm.dept_no=d.dept_no
63    left join employees as e on dm.emp_no=e.emp_no;
```

emp_...	dept_no	from_date	to_date	dept_no	dept_name	emp_no	birth_date	first_name	last_name	gender	hire_date
NULL	NULL	NULL	NULL	d000	Task Force	NULL	NULL	NULL	NULL	NULL	NULL
110022	d001	1985-01-01	1991-10-01	d001	Marketing	110022	1956-09-12	Margareta	Markovitch	M	1985-01-01
110039	d001	1991-10-01	9999-01-01	d001	Marketing	110039	1963-06-21	Vishwani	Minakawa	M	1986-04-12

>< JOIN



start_L	end_L	class
A	D	1
E	H	2
I	L	3
M	P	4
Q	T	5
U	Z	6

8. 직원들에게 나눠줄 물품 정리를 위해 성(last_name) 첫번째 문자를 기준으로 6개의 클래스로 구별하였다.

SELECT * FROM name_class;

position length

직원 이름에서 첫번째 문자만 확인하기.

SELECT emp_no, SUBSTR(last_name, 1, 1), last_name FROM employees;

```
70 •  select emp_no, substr(last_name, 1, 1), last_name from employees;
```

emp_no	substr(last_name, 1, 1)	last_name
10001	F	Facello
10002	S	Simmel
10003	B	Bamford

>< JOIN

start_L	end_L	class
A	D	1
E	H	2
I	L	3
M	P	4
Q	T	5
U	Z	6

9. 직원 이름에서 첫번째 문자를 기준으로 class 지정하기

```
SELECT start_L, last_name, first_name, emp_no
FROM employees AS e
INNER JOIN name_class AS n
ON SUBSTR(last_name, 1, 1) >= n.start_L
AND SUBSTR(last_name, 1, 1) <= n.end_L
ORDER BY start_L, last_name;
```

```
73 •  select start_L, last_name, first_name, emp_no from employees as e
74    inner join name_class as n
75    on substr(last_name, 1, 1) >= n.start_L and substr(last_name, 1, 1) <= n.end_L
76    order by start_L, last_name;
77
```

100% 29:76

Result Grid Filter Rows: Search Export:

start_L	last_name	first_name	emp_no
A	Aingworth	Eben	10106
A	Akazan	Gennady	11474
A	Akazan	Matk	10939
A	Alencar	Aksel	10310
A	Alencar	Yonghoan	10519
A	Aloisi	Nigel	10157

>< JOIN

start_L	end_L	class
A	D	1
E	H	2
I	L	3
M	P	4
Q	T	5
U	Z	6

9. 직원 이름에서 첫번째 문자를 기준으로 class 지정하기 -- BETWEEN 사용

```
SELECT start_L, last_name, first_name, emp_no
FROM employees AS e
INNER JOIN name_class AS n
ON SUBSTR(last_name, 1, 1) BETWEEN n.start_L AND n.end_L
ORDER BY start_L, last_name;
```

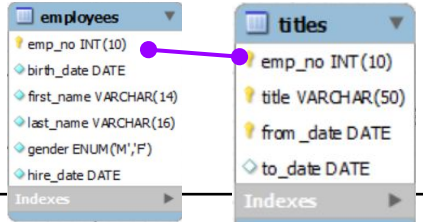
```
79 •  select start_L, last_name, first_name, emp_no from employees as e
80    inner join name_class as n
81    on substr(last_name, 1, 1) between n.start_L and n.end_L
82    order by start_L, last_name;
83
```

100% 29:82

Result Grid Filter Rows: Search Export:

start_L	last_name	first_name	emp_no
A	Aingworth	Eben	10106
A	Akazan	Gennady	11474
A	Akazan	Matk	10939
A	Alencar	Aksel	10310
A	Alencar	Yonghoan	10519
A	Aloisi	Nigel	10157

= JOIN



10. 직원의 직함을 같이 보고 싶다

```
SELECT *
FROM employees AS e
LEFT JOIN titles AS t
ON e.emp_no = t.emp_no;
```

```
91 •  select * from employees as e
92    LEFT JOIN titles as t
93    ON e.emp_no=t.emp_no;
```

emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	title	from_date	to_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26	10001	Senior Engineer	1986-06-26	9999-01-01
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21	10002	Staff	1996-08-03	9999-01-01
10003	1959-12-03	Parto	Bamford	M	1986-08-28	10003	Senior Engineer	1995-12-03	9999-01-01
10004	1954-05-01	Chirstian	Koblick	M	1986-12-01	10004	Engineer	1986-12-01	1995-12-01
10004	1954-05-01	Chirstian	Koblick	M	1986-12-01	10004	Senior Engineer	1995-12-01	9999-01-01

11. 현재의 직함만을 보고 싶다.

```
SELECT *
FROM employees AS e
LEFT JOIN titles AS t
ON e.emp_no = t.emp_no
WHERE t.to_date >= CURRENT_DATE();
```

```
96 •  SELECT *
97    FROM employees AS e
98    LEFT JOIN titles AS t
99    ON e.emp_no = t.emp_no
100   WHERE t.to_date >= CURRENT_DATE();
```

emp_no	birth_date	first_name	last_name	gender	hire_date	emp_no	title	from_date	to_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26	10001	Senior Engineer	1986-06-26	9999-01-01
10002	1964-06-02	Bezalel	Simmel	F	1985-11-21	10002	Staff	1996-08-03	9999-01-01
10003	1959-12-03	Parto	Bamford	M	1986-08-28	10003	Senior Engineer	1995-12-03	9999-01-01
10004	1954-05-01	Chirstian	Koblick	M	1986-12-01	10004	Senior Engineer	1995-12-01	9999-01-01
10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12	10005	Senior Staff	1996-09-12	9999-01-01