

CODINGO x **posco**

K-Digital Training 스 마 트 팩 토 리


JOIN

JOIN

- 두 테이블을 묶어서 하나의 테이블을 만듦, 일종의 집합 개념
- 왜? 두 테이블을 엮어야 원하는 형태가 나오기도 함



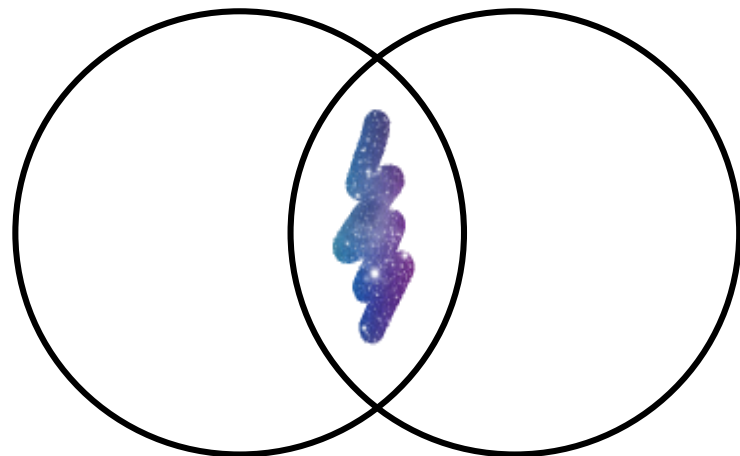
JOIN의 종류

1. Natural Join
 2. Inner Join
 3. Left Outer Join
 4. Right Outer Join
 5. Full Outer Join
- 
- Outer Join

* MySQL에서는 Full outer join을 지원하지 않음

Inner Join

```
SELECT 속성이름, ...  
FROM 테이블A, 테이블B  
WHERE 조인조건 AND 검색조건;
```



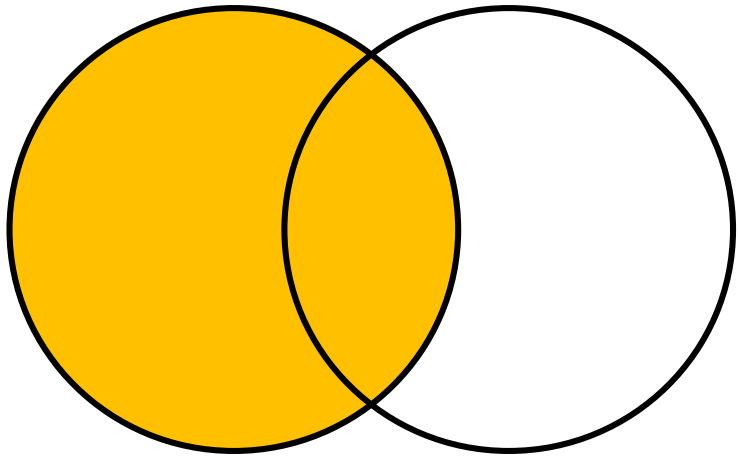
```
SELECT 속성이름, ...  
FROM 테이블A INNER JOIN 테이블B ON 조인조건  
WHERE 검색조건;
```

Natural Join

- Inner Join의 일종
- 두 테이블간 동일한 타입을 가진 동일한 컬럼을 기준으로 자동으로 조인된다.
- 그래서 조인 조건을 따로 명시해주지 않아도 된다.
- 하지만, 동일한 타입을 가진 컬럼이 없거나, 여러 개 있을 경우 원하는 결과를 얻지 못한다.

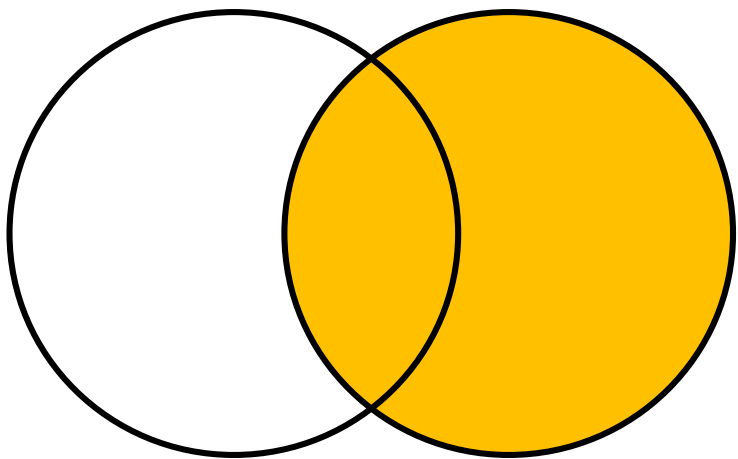
```
SELECT 속성이름, ... FROM 테이블A NATURAL JOIN 테이블B
```

Left Outer Join



```
SELECT 속성이름, ... FROM 테이블A LEFT [OUTER] JOIN 테이블B ON 조인조건
```

Right Outer Join



```
SELECT 속성이름, ... FROM 테이블A RIGHT [OUTER] JOIN 테이블B ON 조인조건
```


UNION

Union

- 두 개 이상의 SELECT문의 결과를 하나의 결과로 결합하는 데 사용된다.
- 이 때, SELECT문의 열 이름과 열의 수가 동일해야 함.
- UNION을 사용하면 **중복된 행이 제거되므로** UNION ALL을 사용하여 중복을 허용할 수 있다.

```
SELECT 속성이름, ... FROM 테이블A UNION SELECT 속성이름, ... FROM 테이블B
```

실습 0. Full Outer Join 구현하기

다음과 같은 두 개의 테이블을 생성

author_id	first_name	last_name	email
3	Stephen	King	sking@hotmail.com
1	J.K.	Rowling	jkrowling@gmail.com
2	George R.R.	Martin	grmartin@yahoo.com
5	아가사	크리스티	agatha.christie@example.com
NULL	NULL	NULL	NULL

Field	Type	Null	Key	Default	Extra
author_id	int	NO	PRI	NULL	auto_increment
first_name	varchar(50)	NO		NULL	
last_name	varchar(50)	NO		NULL	
email	varchar(50)	YES		NULL	

book_id	author_id	title	publication_date
1	1	Harry Potter and the ...	1997-06-26
2	2	A Game of Thrones	1996-08-06
3	3	The Shining	1977-01-28
4	3	스탠드	1978-01-01
5	4	Lucky Day	2023-03-01
NULL	NULL	NULL	NULL

Field	Type	Null	Key	Default	Extra
book_id	int	NO	PRI	NULL	auto_increment
author_id	int	NO		NULL	
title	varchar(100)	NO		NULL	
publication_date	date	NO		NULL	

실습 0. Full Outer Join 구현하기

다음과 같은 결과가 나올 수 있도록 Full Outer Join을 구현해보세요.

	author_id	title
▶	1	Harry Potter and the Philosopher's Stone
	2	A Game of Thrones
	3	스탠드
	3	The Shining
	5	NULL
	NULL	Lucky Day

- Full outer join : left outer join 과 right outer join 을 합친 조인 방법
- Hint, UNION 사용

Foreign Key 연결

```
DROP TABLE IF EXISTS authors;
```

```
CREATE TABLE authors (  
    author_id INT PRIMARY KEY auto_increment,  
    first_name VARCHAR(50),  
    last_name VARCHAR(50),  
    email VARCHAR(50)  
);
```

```
DROP TABLE IF EXISTS books;
```

```
CREATE TABLE books (  
    author_id INT,  
    book_id INT PRIMARY KEY auto_increment,  
    title VARCHAR(100),  
    publication_date DATE,  
    FOREIGN KEY (author_id) REFERENCES authors(author_id) ON UPDATE CASCADE ON DELETE CASCADE  
);
```

데이터 제어어

(DCL, Data Control Language)

데이터 제어어 (DCL)

- Data Control Language
- 데이터베이스에 접근해 읽거나 쓰는 것에 대한 권한 부여/박탈

GRANT

- 특정 데이터베이스 사용자에게 특정 작업에 대한 수행 **권한 부여**

```
GRANT permission_type ON db_name.table_name  
    TO username@host IDENTIFIED BY 'pw' [WITH GRANT OPTION];
```

REVOKE

- 특정 데이터베이스 사용자에게 특정 작업에 대한 **권한 박탈**

```
REVOKE permission_type ON db_name.table_name FROM 'username'@'host';
```

SQL 복습

실습 1. 테이블 생성하기

```
[mysql> desc customers;
```

Field	Type	Null	Key	Default	Extra
id	varchar(50)	NO	PRI	NULL	
name	varchar(50)	NO		NULL	
email	varchar(255)	NO		NULL	
address	varchar(255)	NO		NULL	

```
4 rows in set (0.00 sec)
```

외래키 설정은 update, delete 둘 다 cascade로

orders.customer_id 외래키

```
[mysql> desc orders;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
customer_id	varchar(50)	YES	MUL	NULL	
date	date	NO		NULL	

```
3 rows in set (0.01 sec)
```

order_detail.order_id 외래키

```
mysql> desc order_detail;
```

Field	Type	Null	Key	Default	Extra
order_id	int	YES	MUL	NULL	
item	varchar(50)	NO		NULL	
quantity	int	NO		NULL	

```
3 rows in set (0.00 sec)
```

실습 2. 테이블에 데이터 삽입하기

```
[mysql> select * from customers;
```

id	name	email	address
alice789	Alice Johnson	alice.johnson@hotmail.com	789 Elm St, Nowhere
jane456	Jane Doe	jane.doe@yahoo.com	456 Oak Ave, Somewhere
john123	John Smith	john.smith@gmail.com	123 Main St, Anytown, USA

```
3 rows in set (0.00 sec)
```

```
[mysql> select * from orders;
```

id	customer_id	date
1	john123	2022-01-01
2	john123	2022-02-01
3	jane456	2022-03-01

```
3 rows in set (0.00 sec)
```

```
[mysql> select * from order_detail;
```

order_id	item	quantity
1	pen	10
1	toy	2
2	clothes	5
3	socks	100

```
4 rows in set (0.00 sec)
```

실습 3. DML, DDL 복습

1. orders 테이블에 총 금액(total) 컬럼 추가하기
2. 총 금액을 아래와 같이 업데이트 하기

	id	customer_id	date	total
▶	1	john123	2022-01-01	10000
	2	john123	2022-02-01	15000
	3	jane456	2022-03-01	20000
✱	NULL	NULL	NULL	NULL

3. id가 john123인 고객 John Park으로 이름 수정하기

실습 4. SELECT 복습

1. 2022-01-01 ~ 2022-02-01 기간동안 주문한 고객의 아이디를 중복 없이 가져오기
2. 2022-01-01 ~ 2022-02-01 기간동안 주문한 고객의 이름 중복 없이 가져오기 (서브쿼리)
3. 총 금액이 가장 높은 데이터 가져오기 (MAX 함수 사용하지 않고)
4. 총 금액이 가장 높은 데이터 가져오기(MAX 함수 사용해서)
5. 각각의 customer가 몇 개의 주문을 했는지 검색하기
6. 21000원 이상 결제한 고객의 총 금액 total_price로 가져오기
7. 각 주문 정보에 대해 고객 정보와 주문 정보를 함께 가져오기
8. 주문을 하지 않은 고객 정보도 가져오기