SQL

Flutter

Code Factory

왜 데이터가 초기화되지?

RAM은 빠르지만 장기적으로
데이터를 유지할 수 없다.

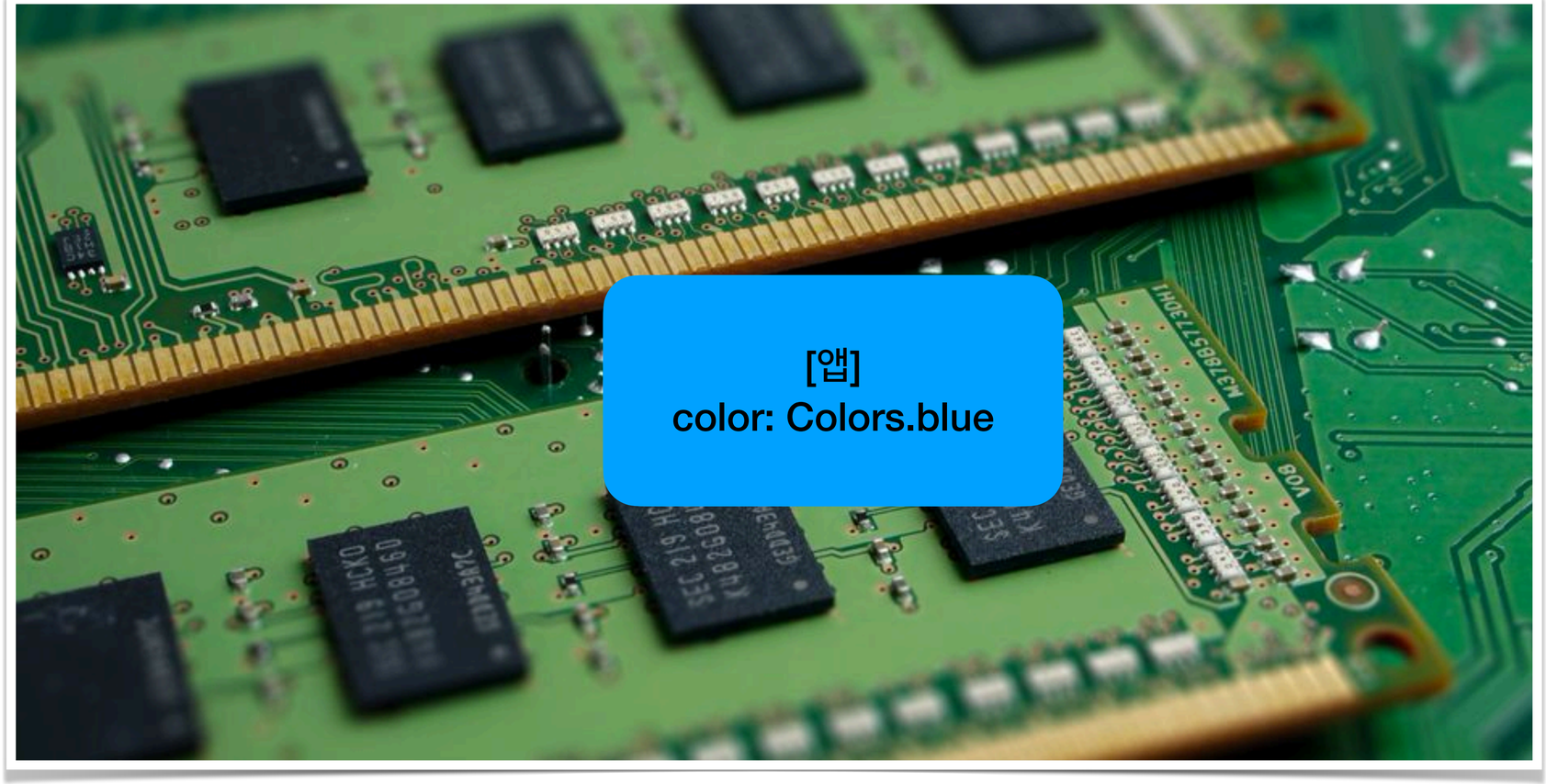
HDD/SSD는 느리지만
장기적으로 데이터를 유지할 수 있다.

현재까지 앱 실행 Flow

HDD

[앱]
color: Colors.black

RAM

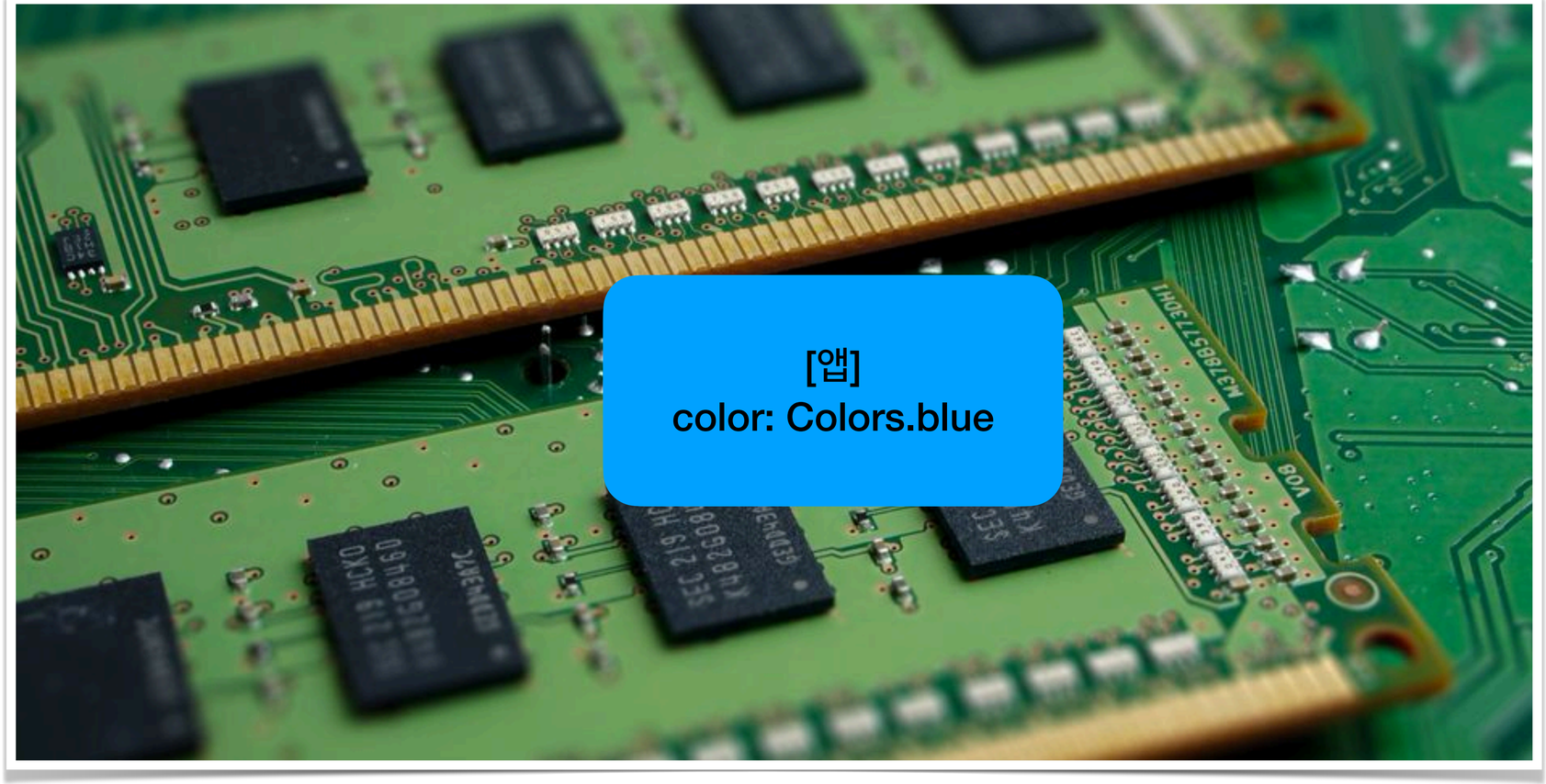


현재까지 앱 실행 Flow

HDD

[앱]
color: Colors.black

RAM

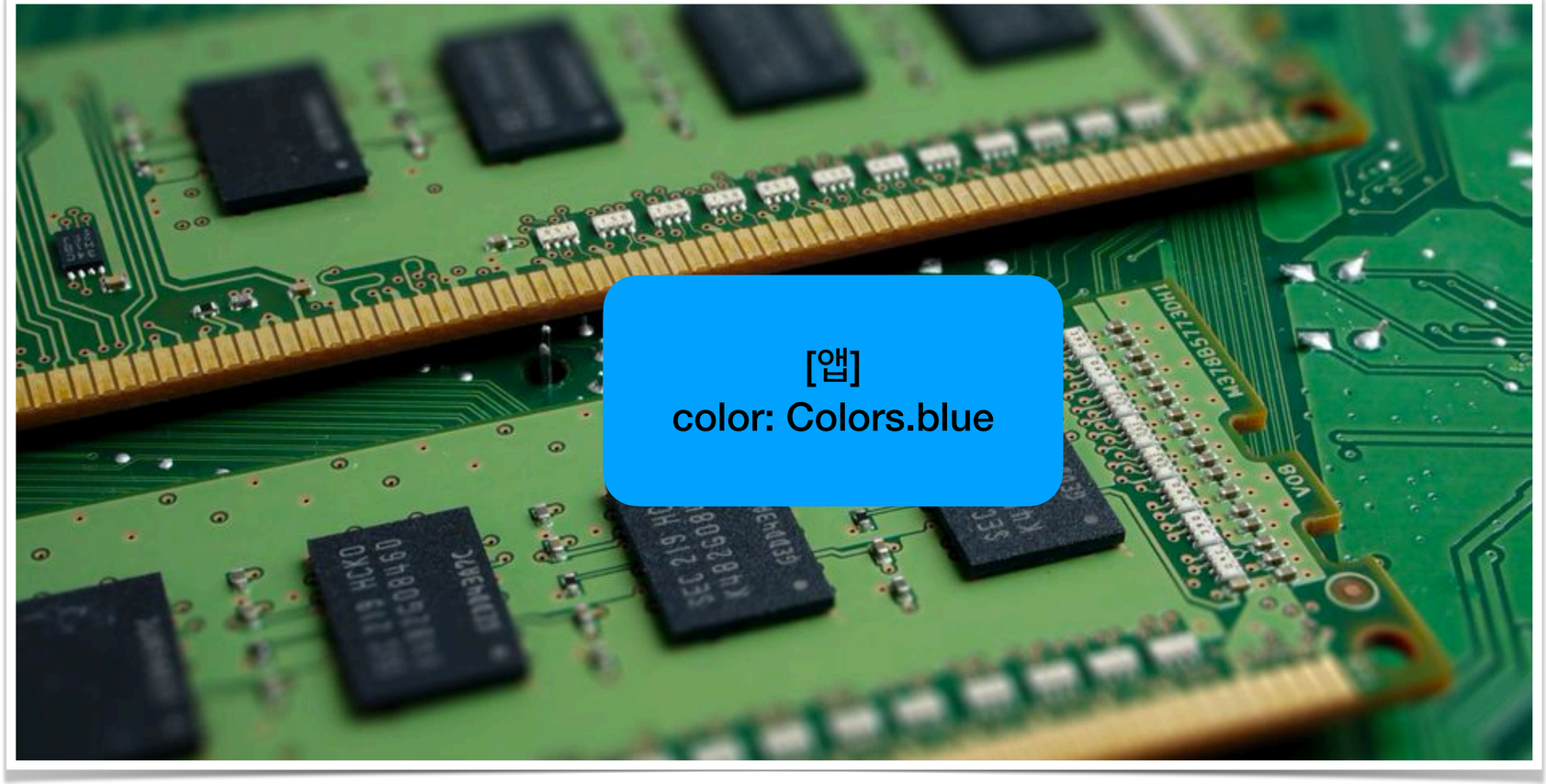


SQL을 사용한 실행 Flow

HDD

[앱]
color: Colors.blue

RAM

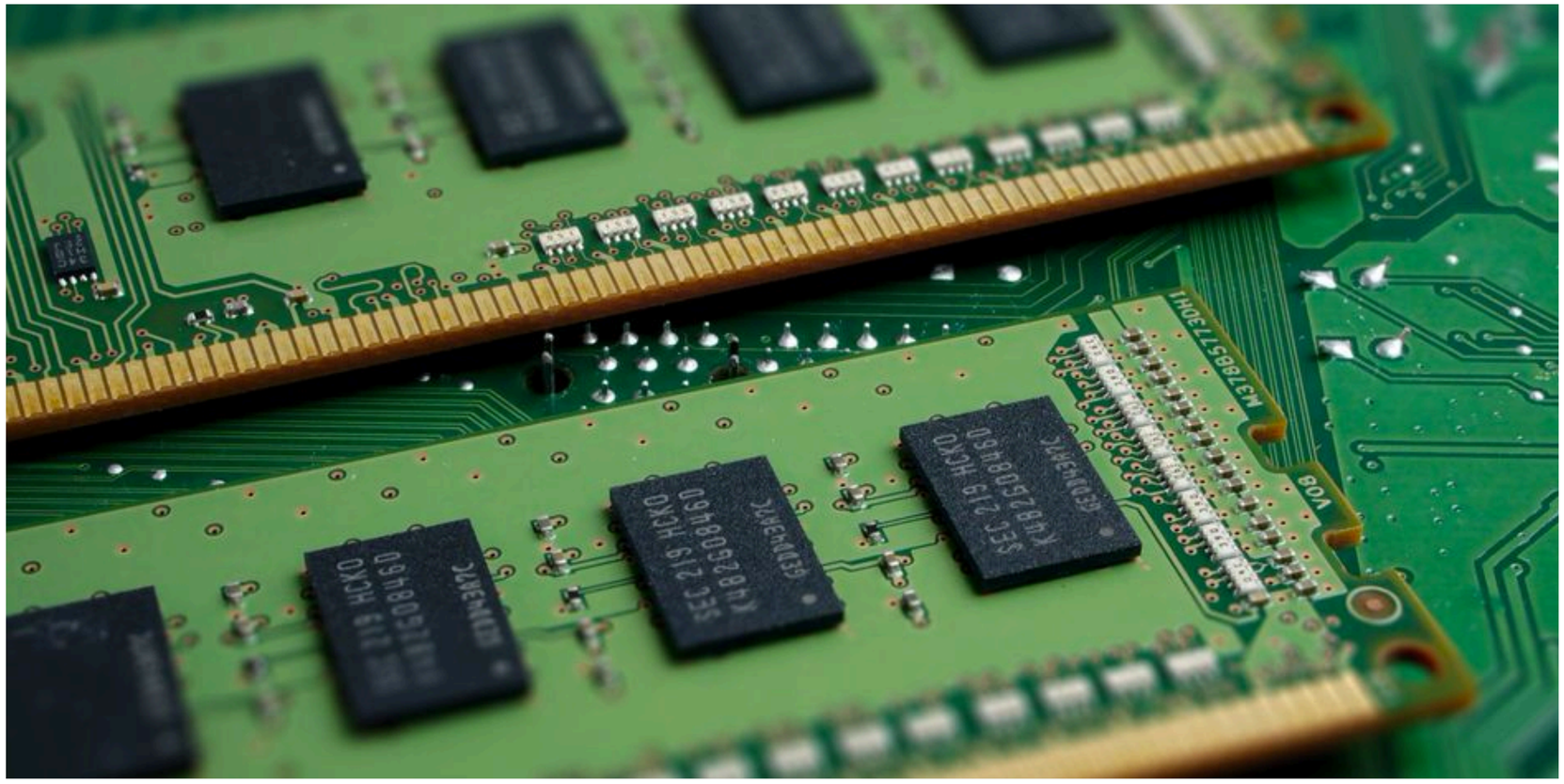


SQL을 사용한 실행 Flow

HDD

[앱]
color: Colors.blue

RAM



Structured Query Language (SQL)

Table - 정보를 담는 구조

Table - 정보를 담는 구조

Column

Column

Column

players table

id	name	gender
1	최민정	F
2	김아랑	F
3	이유빈	F
4	서휘민	F
5	박지윤	F
6	황대헌	M
7	곽윤기	M
8	박장혁	M
9	이준서	M
10	김동욱	M

Row

Row

Row

Row

▪

▪

▪

Select - 데이터 선택하기

Select - 데이터 선택하기

id	name	gender
1	최민정	F
2	김아랑	F
3	이유빈	F
4	서휘민	F
5	박지윤	F
6	황대헌	M
7	곽윤기	M
8	박장혁	M
9	이준서	M
10	김동욱	M

SELECT {column} FROM {table}

SELECT id, name, gender FROM players

Select - 데이터 선택하기

id	name	gender
1	최민정	F
2	김아랑	F
3	이유빈	F
4	서휘민	F
5	박지윤	F
6	황대헌	M
7	곽윤기	M
8	박장혁	M
9	이준서	M
10	김동욱	M

```
[
  {
    'id': 1,
    'name': '최민정',
    'gender': 'F',
  },
  {
    'id': 2,
    'name': '김아랑',
    'gender': 'F'
  },
  ...
  {
    'id': 10,
    'name': '김동욱',
    'gender': 'M',
  }
]
```

Update - 데이터 업데이트하기

Update - 데이터 업데이트하기

id	name	gender
1	최민정	F
2	김아랑	F
3	이유빈	F
4	서휘민	F
5	박지윤	F
6	황대헌	M
7	곽윤기	F
8	박장혁	M
9	이준서	M
10	김동욱	M

UPDATE {table} SET {column} WHERE {condition}

UPDATE players SET gender = ‘F' WHERE name = ‘곽윤기’

Delete - 데이터 삭제하기

Delete - 데이터 삭제하기

id	name	gender
6	황대헌	M
7	곽윤기	M
8	박장혁	M
9	이준서	M
10	김동욱	M

DELETE FROM {table} WHERE {condition}

DELETE FROM players WHERE gender = ‘F’

Insert - 새로운 데이터 추가하기

Insert - 새로운 데이터 추가하기

id	name	gender
1	최민정	F
2	김아랑	F
3	이유빈	F
4	서휘민	F
5	박지윤	F
6	황대헌	M
7	곽윤기	M
8	박장혁	M
9	이준서	M
10	김동욱	M
11	진선유	F

INSERT INTO {table} {column1, column2 …}
VALUES {value1, value2…}

INSERT INTO players (id, name, gender)
VALUES (11, ‘진선유’, ‘F’)

종목을 추가하고 싶다면?

종목을 추가하고 싶다면?

id		name		gender	track
1	1	최민정	최민정	F	500
2	2	김아랑	김아랑	F	1000
3	3	이유빈	이유빈	F	1500
4	4	서휘민	서휘민	F	1000
5	5	박지윤	박지윤	F	1500
6	6	황대헌	황대헌	M	500
7	7	곽윤기	곽윤기	M	1500
8	8	박장혁	박장혁	M	1500
9	9	이준서	이준서	M	1000
10	10	김동욱	김동욱	M	1000

만약에 종목별 정보도 추가 하고싶다면?

만약에 종목별 정보도 추가 하고싶다면?

id	name	gender	track	best_score
1	최민정	F	500	300
2	김아랑	F	1000	650
3	이유빈	F	1500	1000
4	서휘민	F	1000	650
5	박지윤	F	1500	1000
6	황대헌	M	500	250
7	곽윤기	M	1500	900
8	박장혁	M	1500	900
9	이준서	M	1000	550
10	김동욱	M	1000	550

중복 맞아?

테이블간 연동

테이블간 연동

player table

id	name	gender
1	최민정	F
2	김아랑	F
3	이유빈	F
4	서휘민	F
5	박지윤	F
6	황대헌	M
7	곽윤기	M
8	박장혁	M
9	이준서	M
10	김동욱	M

track table

id	length	gender	best_score
1	500	F	300
2	1000	F	650
3	1500	F	1000
4	500	M	250
5	1000	M	550
6	1500	M	900

테이블간 연동

player table

id	name	gender	track_id
1	최민정	F	1
2	김아랑	F	2
3	이유빈	F	3
4	서휘민	F	2
5	박지윤	F	3
6	황대헌	M	4
7	곽윤기	M	6
8	박장혁	M	6
9	이준서	M	5
10	김동욱	M	5

track table

id	length	gender	best_score
1	500	F	300
2	1000	F	650
3	1500	F	1000
4	500	M	250
5	1000	M	550
6	1500	M	900

Normalization (정규화)

Join - 여러 테이블 합치기

Many to One Relationship

player table

id	name	gender	track_id
1	최민정	F	1
2	김아랑	F	2
3	이유빈	F	3
4	서휘민	F	2
5	박지윤	F	3
6	황대헌	M	4
7	곽윤기	M	6
8	박장혁	M	6
9	이준서	M	5
10	김동욱	M	5

track table

id	length	gender	best_score
1	500	F	300
2	1000	F	650
3	1500	F	1000
4	500	M	250
5	1000	M	550
6	1500	M	900

```
SELECT {column} FROM {table}
INNER JOIN {other_table} ON {condition}
```

```
SELECT player.id, player.name, player.gender, track.length, track.best_score
FROM player INNER JOIN track ON player.track_id = track.id
```

Many to One Relationship

joined table

id	name	gender	length	best_score
1	최민정	F	500	300
2	김아랑	F	1000	650
3	이유빈	F	1500	1000
4	서휘민	F	1000	650
5	박지윤	F	1500	1000
6	황대헌	M	500	250
7	곽윤기	M	1500	900
8	박장혁	M	1500	900
9	이준서	M	1000	550
10	김동욱	M	1000	550


```
[
  {
    'id': 1,
    'name': '최민정',
    'gender': 'F',
    'length': 500,
    'best_score': 300
  },
  {
    'id': 2,
    'name': '김아랑',
    'gender': 'F',
    'length': 1000,
    'best_score': 650,
  },
  ...
  {
    'id': 10,
    'name': '김동욱',
    'gender': 'M',
    'length': 1000,
    'best_score': 550
  }
]
```

만약에 한 선수가 여러 종목에 출전한다면?

Many to Many Relationship

player table

id	name	gender
1	최민정	F
2	김아랑	F
3	이유빈	F
4	서휘민	F
5	박지윤	F
6	황대헌	M
7	곽윤기	M
8	박장혁	M
9	이준서	M
10	김동욱	M

player_track table

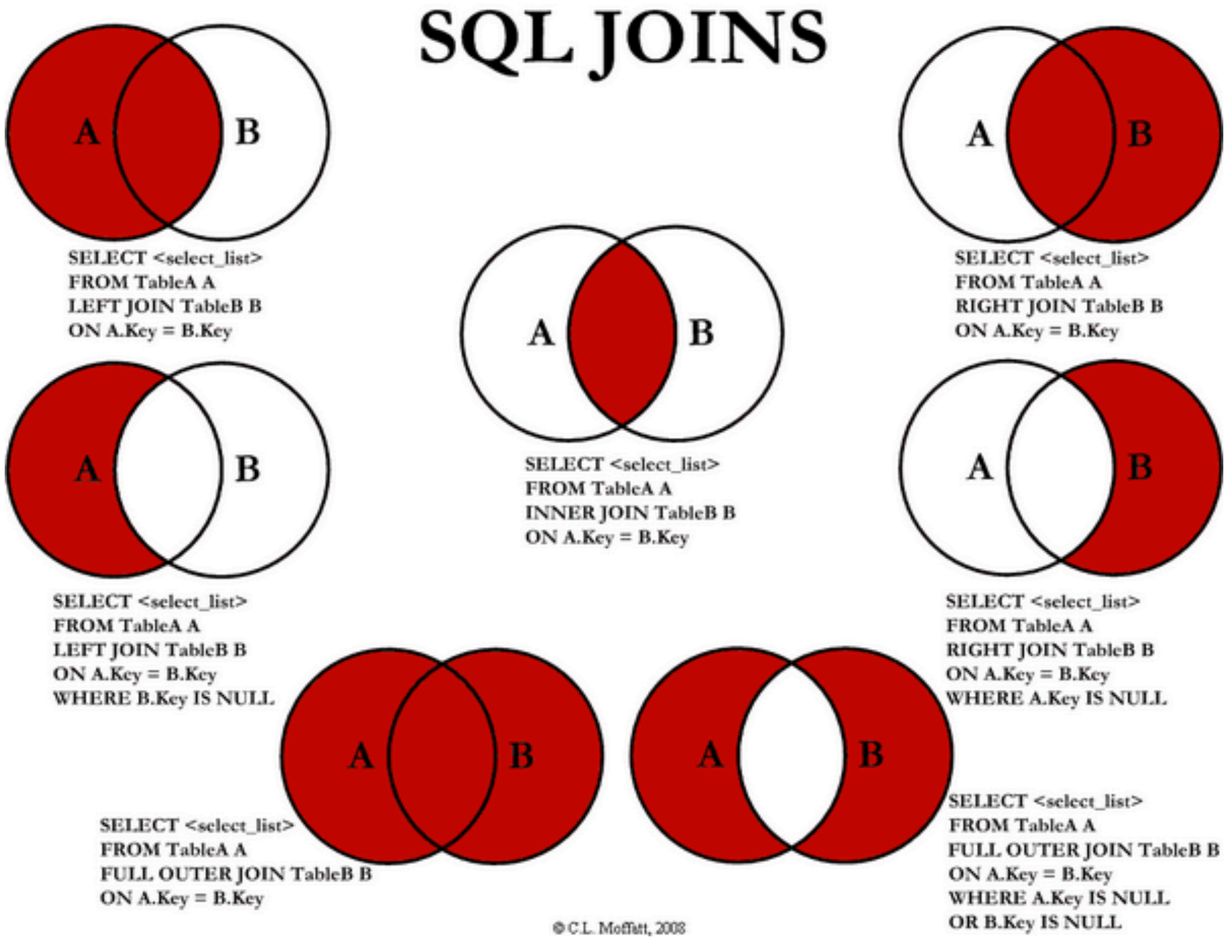
player_id	track_id
1	1
1	2
1	3
2	2
3	2
3	3
6	4
6	5
6	6
7	5

track table

id	length	gender	best_score
1	500	F	300
2	1000	F	650
3	1500	F	1000
4	500	M	250
5	1000	M	550
6	1500	M	900

```
SELECT player.name, track.length  FROM player_track INNER JOIN player
ON player_track.player_id = player.id INNER JOIN track ON
player_track.track_id = track.id
```

Join의 종류


SQL JOINS
A
B
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
© C.L. Moffatt, 2008