

Class 3. SQLite3 심화

SQL 문법 순서

SELECT 컬럼명

FROM 테이블명

WHERE 조건식

GROUP BY 컬럼명

HAVING 조건식

ORDER BY 컬럼명

ORDER BY가 WHERE 앞에 오면 안됨

```
sqlite> SELECT playerID, HR FROM batting ORDER BY HR WHERE yearID = 2019;  
Error: near "WHERE": syntax error
```

에러 발생

조인(JOIN)

지금까지 어떤 선수의 playerId 확인은 people 테이블에서 해야 했고,
그 playerId를 기반으로 batting 테이블이나 pitching 테이블에서 그 선수의 데이터를 조회했음.

꽤 귀찮고 번거로움.

한 번에 해결할 수 없을까?

JOIN을 활용하면 가능.

JOIN은 두 개 이상의 테이블을 조합할 때 사용되는 문법

SQLite3에는 세 가지 유형의 JOIN을 지원

유형	설명
INNER JOIN	조인 중 가장 많이 사용됨 간단히 JOIN이라고 부르기도 함
LEFT OUTER JOIN	간단히 LEFT JOIN이라고 부르기도 함
CROSS JOIN	

내부 조인(INNER JOIN)

```
SELECT <컬럼 목록>  
FROM <첫번째 테이블명>  
      INNER JOIN <두번째 테이블명>  
      ON <조인될 조건>  
[WHERE 검색 조건]
```

```
SELECT <컬럼 목록>  
FROM batting  
      INNER JOIN people  
      ON batting.playerID = people.playerID  
[WHERE 검색 조건]
```

batting 테이블

playerID	yearID	H	...	HR
choosh01	2019	149		24
cruzne02	2019	141		41
...				
alonspe01	2020	155		53
yelicch01	2020	161		44

people 테이블

playerID	...	nameFirst	nameLast
aardsda01		David	Aardsma
abadijo01		John	Abadie
...			
choosh01		Shin-soo	Choo
parkch01		Chan-ho	Park

두 행
결합

내부 조인(INNER JOIN)

```
SELECT <컬럼 목록>  
FROM <첫번째 테이블명>  
      INNER JOIN <두번째 테이블명>  
      ON <조인될 조건>  
[WHERE 검색 조건]
```

```
SELECT <컬럼 목록>  
FROM batting  
      INNER JOIN people  
      ON batting.playerID = people.playerID  
[WHERE 검색 조건]
```

playerID	yearID	H	...	HR	playerID	...	nameFirst	nameLast
choosh01	2019	149		24	choosh01		Shin-soo	Choo
cruzne02	2019	141		41	cruzne02		Nelson	Cruz
...								
alonspe01	2020	155		53	alonspe01		Pete	Alonso
yelicch01	2020	161		44	yelicch01		Christian	Yelich

결합된 결과

내부 조인(INNER JOIN)

1) 2019년 MLB 한국 선수 홈런 개수를 알아보자(playerID 대신에 선수 이름이 보이도록)

```
SELECT batting.*, people.* FROM batting INNER JOIN people ON batting.playerID = people.playerID WHERE batting.yearID = 2019 and people.birthCountry = 'South Korea';
```

sqlite> SELECT batting.*, people.* FROM batting INNER JOIN people ON batting.playerID = people.playerID WHERE batting.yearID = 2019 and people.birthCountry = 'South Korea';																											
ID	playerID	yearID	stint	teamID	team_ID	lgID	G	G_batting	AB	R	H	2B	3B	HR	RBI	SB	CS	BB	SO	IBB	HBP	SH	SF	GIDP	play		
erID	birthYear	birthMonth	birthDay	birthCountry	birthState	birthCity	deathYear	deathMonth	deathDay	deathCountry	deathState	deathCity	nameFirst	nameLast	nameGiven	weight	height	bats	throws	debut	finalGame	retroID	bbrefID	birth_date	debu		
t_date	finalgame_date	death_date																									

106135	choijj01	2019	1	TBA	2922	AL	127		410	54	107	20	2	19	63	2	3	64	108	2	6	0	7	7	choi		
j01	1991	5	19	South Korea	Incheon	Incheon							Ji-Man	Choi	Ji-Man	250	73	L	R	2016-04-05	2019-09-29	choijj01	choijj01	1991-05-19	2016		
-04-05	2019-09-29																										
106136	choosh01	2019	1	TEX	2923	AL	151		563	93	149	31	2	24	61	15	1	78	165	3	18	0	1	6	choo		
sh01	1982	7	13	South Korea	Busan	Busan							Shin-Soo	Choo	Shin-Soo	210	71	L	L	2005-04-21	2019-09-29	choos01	choosh01	1982-07-13	2005		
-04-21	2019-09-29																										
106506	kangju01	2019	1	PIT	2917	NL	65		172	15	29	7	1	10	24	0	0	11	60	0	1	0	1	4	kang		
ju01	1987	4	5	South Korea	Gwangju	Gwangju							Jung Ho	Kang	Jung Ho	210	72	R	R	2015-04-08	2019-07-29	kangj01	kangju01	1987-04-05	2015		
-04-08	2019-07-29																										
106076	ohse01	2019	1	COL	2904	NL	21		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ohse		
01	1982	7	15	South Korea	North Jeol	Jeongeup							Seunghwan	Oh	Seunghwan	205	70	R	R	2016-04-03	2019-05-30	oh--s01	ohse01	1982-07-15	2016		
-04-03	2019-05-30																										
107087	ryuhy01	2019	1	LAN	2909	NL	30		51	3	8	1	0	1	3	0	0	4	23	0	0	12	0	0	ryuh		
y01	1987	3	25	South Korea	Incheon	Incheon							Hyun-Jin	Ryu	Hyun-Jin	255	75	R	L	2013-04-02	2019-09-28	ryu-h01	ryuhy01	1987-03-25	2013		
-04-02	2019-09-28																										

```
SELECT batting.yearID, people.nameLast, people.nameFirst, batting.HR FROM batting INNER JOIN people ON batting.playerID = people.playerID WHERE batting.yearID = 2019 and people.birthCountry = 'South Korea';
```

yearID	nameLast	nameFirst	HR
2019	Choi	Ji-Man	19
2019	Choo	Shin-Soo	24
2019	Kang	Jung Ho	10
2019	Oh	Seunghwan	0
2019	Ryu	Hyun-Jin	1

내부 조인(INNER JOIN)

2) 2015년 “오른손” 타자 중에 안타를 가장 많이 친 선수 5명은 누구일까?

```
SELECT batting.yearID, people.nameFirst, people.nameLast, people.bats, batting.H FROM batting INNER JOIN people ON batting.playerID = people.playerID WHERE batting.yearID = 2015 and people.bats = 'R' ORDER BY batting.H DESC LIMIT 5;
```

yearID	nameFirst	nameLast	bats	H
2015	Jose	Altuve	R	200
2015	Xander	Bogaerts	R	196
2015	A. J.	Pollock	R	192
2015	Ian	Kinsler	R	185
2015	Josh	Donaldson	R	184

3) 2015년 “왼손” 투수 중에 탈삼진을 가장 많이 잡은 선수 5명은 누구일까?

```
SELECT pitching.yearID, people.nameFirst, people.nameLast, peoplethrows, pitching.SO FROM pitching INNER JOIN people ON pitching.playerID = people.playerID WHERE pitching.yearID = 2015 and peoplethrows = 'L' ORDER BY pitching.SO DESC LIMIT 5;
```

yearID	nameFirst	nameLast	throws	SO
2015	Clayton	Kershaw	L	301
2015	Chris	Sale	L	274
2015	Madison	Bumgarner	L	234
2015	Dallas	Keuchel	L	216
2015	Jon	Lester	L	207

서브쿼리(subquery)

서브쿼리(subquery): 쿼리문 안에 포함되어 있는 또 다른 쿼리문

서브쿼리 사용시 주의사항

- 1. 서브쿼리는 괄호로 감싸줘야 함
- 2. 서브쿼리에서는 ORDER BY 사용 불가

서브쿼리 사용 가능 위치
SELECT 절
FROM 절
WHERE 절
HAVING 절
ORDER BY 절
INSERT 문의 VALUES 절
UPDATE 문의 SET 절

데이터 추가시 사용

데이터 수정시 사용

1. FROM 절에 서브쿼리가 사용된 경우(추신수 2019 OPS)

```
SELECT playerID, (SLG + OBP) AS OPS FROM (SELECT playerID, CAST((((H - "2B" - "3B" - HR) + 2*"2B" + 3*"3B" + 4*HR) AS REAL)/AB AS SLG, CAST((H + BB + HBP) AS REAL)/(AB + BB + HBP + SF) AS OBP FROM batting WHERE yearID = 2019 and playerID = 'choosh01');
```

playerID	OPS
choosh01	0.825919048387965

서브쿼리(subquery)

2. SELECT 절에 서브쿼리가 사용된 경우

```
SELECT COUNT(*), (SELECT AVG(salary) FROM salaries WHERE playerID = 'choosh01') FROM batting WHERE playerID = 'choosh01';
```

COUNT(*)	(SELECT AVG(salary) FROM salaries WHERE playerID = 'choosh01')
16	7280200.0

3. WHERE 절에 서브쿼리가 사용된 경우

1) 2019년에 커쇼가 기록한 탈삼진보다 더 많은 탈삼진을 기록한 2019년 투수 기록은?

```
SELECT playerID, SO FROM pitching  
WHERE  
SO > (SELECT SO FROM pitching WHERE playerID = 'kershcl01' and yearID = 2019)  
and yearID = 2019;
```

189

playerID	SO
berrijo01	195
biebesh01	259
boydma01	238
buehlwa01	215
bungama01	203
castilu02	226
college01	326
corbipa01	238
darviyu01	229
degroja01	255
flaheja01	231
giolilu01	228
grayso01	205
lynnla01	246
minormi01	200
mortoch02	240
nolaaa01	229
rayro02	235
rodried05	213
salech01	218
scherna01	243
strasst01	251
syndeno01	202
verlaju01	300
wheelza01	195

서브쿼리(subquery)

2) 일본 출생 선수들의 타자 기록은?

```
SELECT * FROM batting WHERE playerID IN (SELECT playerID FROM people WHERE birthCountry = 'Japan');
```

ID	playerID	yearID	stint	teamID	team_ID	lgID	G	G_batting	AB	R	H	2B	3B	HR	RBI	SB	CS	BB	SO	IBB	HBP	SH	SF	GIDP
95646	aokino01	2012	1	MIL	2701	NL	151		520	81	150	37	4	10	50	30	8	43	55	1	13	7	5	6
97051	aokino01	2013	1	MIL	2731	NL	155		597	80	171	20	3	8	37	20	12	55	40	1	11	8	3	9
98469	aokino01	2014	1	KCA	2757	AL	132		491	63	140	22	6	1	43	17	8	43	49	0	6	8	1	5
99905	aokino01	2015	1	SFN	2800	NL	93		355	42	102	12	3	5	26	14	5	30	25	0	6	1	0	8
101392	aokino01	2016	1	SEA	2829	AL	118		417	63	118	24	4	4	28	7	9	34	45	0	9	5	1	9
102883	aokino01	2017	1	HOU	2846	AL	71		202	28	55	12	1	2	19	5	2	15	29	1	2	1	4	11
102884	aokino01	2017	2	TOR	2864	AL	12		32	4	9	1	0	3	8	0	0	1	5	0	0	0	1	0
102885	aokino01	2017	3	NYN	2854	NL	27		102	16	29	7	1	0	8	5	0	13	10	0	1	0	0	1
71972	bowiej01	1994	1	OAK	2173	AL	6		14	0	3	0	0	0	0	0	0	0	2	0	0	1	0	1
67607	chitrst01	1990	1	OAK	2065	AL	8		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
68727	chitrst01	1991	1	OAK	2091	AL	56		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
95910	darviyu01	2012	1	TEX	2713	AL	29		3	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
97327	darviyu01	2013	1	TEX	2743	AL	32		3	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
98729	darviyu01	2014	1	TEX	2773	AL	22		5	0	1	1	0	0	0	0	0	0	2	0	0	0	0	0
101651	darviyu01	2016	1	TEX	2833	AL	17		3	1	1	0	0	1	1	0	0	0	1	0	0	0	0	1
103135	darviyu01	2017	1	TEX	2863	AL	22		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
103136	darviyu01	2017	2	LAN	2849	AL	9		17	0	1	0	0	0	0	0	0	1	14	0	0	2	0	0
104639	darviyu01	2018	1	CHN	2871	NL	8		12	0	1	1	0	0	0	0	0	0	7	0	0	2	0	0
106188	darviyu01	2019	1	CHN	2901	NL	31		56	1	5	0	0	0	3	0	0	2	31	0	0	3	1	1
50224	fenwibo01	1972	1	HOU	1599	NL	36		50	7	9	3	0	0	4	0	1	3	13	1	0	0	0	1
51123	fenwibo01	1973	1	SLN	1636	AL	5		6	0	1	0	0	0	1	0	0	0	2	0	0	0	0	0
97462	fujikky01	2013	1	CHN	2721	NL	12		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
98874	fujikky01	2014	1	CHN	2751	NL	15		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100320	fujikky01	2015	1	TEX	2803	AL	2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90514	fukudko01	2008	1	CHN	2571	NL	150		501	79	129	25	3	10	58	12	4	81	104	9	1	2	5	7
91898	fukudko01	2009	1	CHN	2601	NL	146		499	79	129	38	5	11	54	6	10	93	112	3	3	3	5	15
93279	fukudko01	2010	1	CHN	2631	NL	130		358	45	94	20	2	13	44	7	8	64	67	1	0	3	4	5
94641	fukudko01	2011	1	CHN	2661	NL	87		293	33	80	15	2	3	13	2	2	46	57	1	1	5	0	2
94642	fukudko01	2011	2	CLE	2663	AL	59		237	26	59	12	1	5	22	2	4	15	53	3	3	1	2	6
90047	fukudko01	2012	1	CHA	2690	AL	24		41	2	7	1	0	0	4	0	1	8	9	0	0	0	2	0
90515	fukumka01	2008	1	TEX	2593	AL	4		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75877	hasegsh01	1997	1	ANA	2238	AL	50		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77127	hasegsh01	1998	1	ANA	2266	AL	61		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78400	hasegsh01	1999	1	ANA	2296	AL	64		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79767	hasegsh01	2000	1	ANA	2326	AL	66		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
81146	hasegsh01	2001	1	ANA	2356	AL	46		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82476	hasegsh01	2002	1	SEA	2410	AL	53		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
83008	hasegsh01	2003	1	SEA	2440	AL	63		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85213	hasegsh01	2004	1	SEA	2470	AL	68		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
86533	hasegsh01	2005	1	SEA	2499	AL	46		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
104975	hiranyo01	2018	1	ARI	2866	NL	75		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
106531	hiranyo01	2019	1	ARI	2896	NL	62		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79819	housecr01	2000	1	COL	2335	NL	16		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
93444	igarary01	2010	1	NYN	2644	NL	34		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
94820	igarary01	2011	1	NYN	2674	NL	45		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
96222	igarary01	2012	1	TOR	2714	AL	2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
96223	igarary01	2012	2	NYA	2703	AL	2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
89294	igawake01	2007	1	NYA	2553	AL	14		0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
90696	igawake01	2008	1	NYA	2583	AL	2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
86592	iguchta01	2005	1	CHA	2480	AL	135		511	74	142	25	6	15	71	15	5	47	114	0	6	11	6	16
87921	iguchta01	2006	1	CHA	2510	AL	138		555	97	156	24	0	18	67	11	5	59	110	0	3	8	2	7
89295	iguchta01	2007	1	CHA	2540	AL	90		327	45	82	17	4	6	31	8	1	44	65	1	2	1	3	5
89296	iguchta01	2007	2	PHI	2556	NL	45		138	22	42	10	0	3	12	6	1	13	23	0	1	1	3	1
90697	iguchta01	2008	1	SDN	2588	NL	81		303	29	70	14	1	2	24	8	1	26	75	0	0	1	0	11

IN의 활용

playerID가 parkch01 또는 ryuhy01 또는 kimby01 중에 하나인 선수의 ERA를 조회하라

```
SELECT playerID, yearID, ERA FROM pitching
WHERE playerID IN ('parkch01', 'ryuhy01', 'kimby01');
```



```
SELECT playerID, yearID, ERA FROM pitching
WHERE playerID = 'parkch01' OR playerID = 'ryuhy01' OR playerID = 'kimby01';
```

```
SELECT playerID, yearID, ERA FROM pitching
WHERE playerID NOT IN ('parkch01', 'ryuhy01', 'kimby01');
```

위 playerID인 선수들의 데이터를 제외한 나머지 선수들의 데이터 조회하고 싶을 때

playerID	yearID	ERA
kimby01	1999	4.61
kimby01	2000	4.46
kimby01	2001	2.94
kimby01	2002	2.04
kimby01	2003	3.56
kimby01	2003	3.18
kimby01	2004	6.23
kimby01	2005	4.86
kimby01	2006	5.57
kimby01	2007	10.5
kimby01	2007	4.16
kimby01	2007	23.63
kimby01	2007	8.21
parkch01	1994	11.25
parkch01	1995	4.5
parkch01	1996	3.64
parkch01	1997	3.38
parkch01	1998	3.71
parkch01	1999	5.23
parkch01	2000	3.27
parkch01	2001	3.5
parkch01	2002	5.75
parkch01	2003	7.58
parkch01	2004	5.46
parkch01	2005	5.66
parkch01	2005	5.91
parkch01	2006	4.81
parkch01	2007	15.75
parkch01	2008	3.4
parkch01	2009	4.43
parkch01	2010	5.6
parkch01	2010	3.49
ryuhy01	2013	3.0
ryuhy01	2014	3.38
ryuhy01	2016	11.57
ryuhy01	2017	3.77
ryuhy01	2018	1.97
ryuhy01	2019	2.32

BETWEEN ... AND ... 의 활용

2019년 세이브 개수가 20이상 30이하인 선수의 세이브 수를 조회하라

```
SELECT playerID, yearID, SV FROM pitching WHERE (SV BETWEEN 20 AND 30) AND yearID = 2019;
```

playerID	yearID	SV
colomal01	2019	30
diazed04	2019	26
doolise01	2019	29
gileske01	2019	23
greensh02	2019	22
hendrli01	2019	25
kenneia01	2019	30
martica04	2019	24
nerishe01	2019	28
paganem01	2019	20
riverfe01	2019	28
robleha01	2019	23
rogerta01	2019	30

GROUP BY의 활용

GROUP BY는 그룹으로 묶어주는 역할을 함
GROUP BY는 보통 SUM, COUNT 등의 집계함수와 함께 사용됨

batting 테이블을 활용하여 2019 시즌 각 팀의 총 안타 개수를 조회하라

```
SELECT playerId, teamID, H FROM batting WHERE yearID = 2019 ORDER BY teamID;
```

```
SELECT teamID, SUM(H) FROM batting WHERE yearID = 2019 GROUP BY teamID;
```

batting 테이블을 활용하여 2019 시즌 각 팀 총 안타 개수가 1500개 초과인 팀의 총 안타 개수를 조회하라

```
SELECT teamID, SUM(H) FROM batting WHERE yearID = 2019 GROUP BY teamID HAVING SUM(H) > 1500;
```

GROUP BY 절 내에서 집계 함수에 대해서 조건을 주고 싶을 때는 HAVING을 사용

teamID	SUM(H)
ARI	1419
ATL	1432
BAL	1379
BOS	1554
CHA	1443
CHN	1378
CIN	1328
CLE	1354
COL	1502
DET	1333
HOU	1538
KCA	1356
LAA	1368
LAN	1414
MIA	1326
MIL	1366
MIN	1547
NYA	1493
NYN	1445
OAK	1384
PHI	1369
PIT	1497
SDN	1281
SEA	1305
SFN	1332
SLN	1336
TBA	1427
TEX	1374
TOR	1299
WAS	1460

teamID	SUM(H)
BOS	1554
COL	1502
HOU	1538
MIN	1547

LIKE의 활용

MLB에서 뛴 한국에서 태어난 선수들을 검색하고 싶음.

그런데 한국이 Korea로 들어가 있을 지,
Republic of Korea로 들어가 있을 지,
South Korea로 들어가 있을 지는 모르는 상황.

어쨌든 Korea는 들어가 있을 것으로 예상됨.

이런 상황에 LIKE를 사용할 수 있음.

```
SELECT * FROM people WHERE birthCountry LIKE '%Korea%';
```

여기서 %는 무엇이든 허용한다는 뜻

playerID	birthYear	birthMonth	birthDay	birthCountry	birthState	birthCity	deathYear	deathMonth	deathDay	deathCountry	deathState	deathCity	nameFirst	nameLast	nameGiven	weight	height	base
ts	throws	debut	finalGame	retroID	bbrefID	birth_date	debut_date	finalgame_date	death_date									
baekch01	R	1980	5	29	South Korea	Busan							Cha-Seung	Baek	Cha-Seung	190	76	R
bongju01	L	1980	7	15	South Korea	Seoul							Jung	Bong	Jung Keun	175	75	L
choihe01	L	1979	3	16	South Korea	South Jeol Hwasun							Hee-Seop	Choi	Hee-Seop	235	77	L
choiji01	R	1991	5	19	South Korea	Incheon							Ji-Man	Choi	Ji-Man	250	73	L
choji01	R	1975	8	16	South Korea	North Jeol Jeonju							Jin Ho	Cho	Jin Ho	175	72	R
choosh01		1982	7	13	South Korea	Busan							Shin-Soo	Choo	Shin-Soo	210	71	L

LIKE의 활용

레먼데이터베이스에서 teamID는 3글자로 구성되어 있음.
BAL, DET, WAS, HOU, TOR 등.
teamID가 T로 시작하는 팀들을 조회하고 싶을 때는
teamID LIKE 'T_'를 사용할 수 있음.

```
SELECT yearID, teamID, name FROM teams WHERE yearID = 2019 and teamID LIKE 'T_';
```

언더바(_) 두번

yearID	teamID	name
2019	TBA	Tampa Bay Rays
2019	TEX	Texas Rangers
2019	TOR	Toronto Blue J

DISTINCT의 활용

중복된 것은 1개씩만 보여주고 싶을 때

MLB에서 역대 댄 선수들의 출생국가를 조회해보자

```
SELECT birthCountry FROM people;
```

이렇게 하면 중복된 국가가 엄청나게 많이 조회됨

```
SELECT DISTINCT birthCountry FROM people;
```

```
SELECT COUNT(DISTINCT birthCountry) FROM people;
```

```
sqlite> SELECT COUNT(DISTINCT birthCountry) FROM people;
```

57

57개 국가에서 태어난 선수들이 MLB에 입성했음

Russia
Curacao
Colombia
Nicaragua
Germany
Norway
Japan
Ireland
Italy
Bahamas
United Kingdom
South Korea
Australia
Czech Republic
V.I.
Netherlands
France
Aruba

Sweden
Hong Kong
Afghanistan
Spain
Greece
Taiwan
Philippines
Jamaica
Poland
Honduras
Brazil
Viet Nam
Guam
Denmark
Switzerland
Austria
Singapore
China
Belgium
Peru
Belize
Indonesia
Finland
Lithuania
South Africa
At Sea
Slovakia
American Samoa
Saudi Arabia
Portugal
Latvia

Quiz

2019년 teamID가 A로 끝나는 팀 선수 중 BABIP이 좋은 선수 3명의 데이터를 조회하시오.

- 1) INNER JOIN 활용할 것
- 2) LIKE 활용할 것
- 3) 규정타석 적용할 것
- 4) 다음과 똑같이 출력되게 할 것.

playerID	teamID	nameFirst	nameLast	BABIP
moncayo01	CHA	Yoan	Moncada	0.405970149253731
anderti01	CHA	Tim	Anderson	0.399463806970509
garcile02	CHA	Leury	Garcia	0.353348729792148
merriwh01	KCA	Whit	Merrifield	0.349907918968692
lemahdj01	NYA	DJ	LeMahieu	0.348979591836735

참고로 지금까지의 내용을 완벽히 이해하지 않았다면
어렵습니다...ㅠㅠ

과제#3

내부 조인, 서브쿼리, IN, BETWEEN ... AND ..., GROUP BY 각각 하나씩 활용해서 데이터 조회하기(총 5개)

과제 수행 내용 word로 정리해서 kyohoonsim@gmail.com 으로 보내주세요~!

문서 제목 양식:

KUSF데이터분석_과제3_이름.docx

ex) KUSF데이터분석_과제3_심교훈.docx