

Class 5. 레먼데이터베이스

레먼데이터베이스 다운로드 받기

DBMS를 설치했으니 테스트 해볼 DB가 필요.
MLB DB 중 하나인 레먼 데이터베이스로 실습 진행.

레먼 데이터베이스:

손 레먼 만듦.

1871년부터 2021년까지의 MLB 데이터 보유.
매 시즌 후 갱신.

다운로드 링크: <http://www.seanlahman.com/baseball-archive/statistics/>

2019

2019 - MS Access version

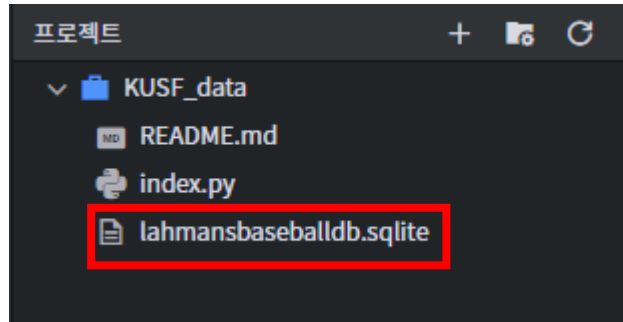
2019 - comma-delimited version

2019 - R Package

2019 - MySQL version

2019 - SQL Lite

2019-SQL Lite 클릭!



다운로드 받은 파일을
현재 작업 디렉토리에 넣어주세요

레먼데이터베이스 테이블 명세서

레먼 데이터베이스 설명서:

<http://www.seanlahman.com/files/database/readme2017.txt>

주요 테이블: People, Batting, Pitching, Fielding

나머지 테이블:

- AllStarFull - All-Star appearances
- HallofFame - Hall of Fame voting data
- Managers - managerial statistics
- Teams - yearly stats and standings
- BattingPost - post-season batting statistics
- PitchingPost - post-season pitching statistics
- TeamFranchises - franchise information
- FieldingOF - outfield position data
- FieldingPost - post-season fielding data
- FieldingOFsplit - LF/CF/RF splits
- ManagersHalf - split season data for managers
- TeamsHalf - split season data for teams
- Salaries - player salary data
- SeriesPost - post-season series information
- AwardsManagers - awards won by managers
- AwardsPlayers - awards won by players
- AwardsShareManagers - award voting for manager awards
- AwardsSharePlayers - award voting for player awards
- Appearances - details on the positions a player appeared at
- Schools - list of colleges that players attended
- CollegePlaying - list of players and the colleges they attended
- Parks - list of major league ballparks
- HomeGames - Number of homegames played by each team in each ballpark

레먼데이터베이스 테이블 명세서

batting 테이블

playerID	Player ID code 선수 ID 코드
yearID	Year 연도
stint	player's stint (order of appearances within a season) 1, 2, 3, 4, 5
teamID	Team 팀 ID
lgID	League 리그 ID: AA, AL(아메리칸리그), FL, NL(내셔널리그), PL, UA
G	Games 경기수
AB	At Bats 타수
R	Runs 득점
H	Hits 안타
2B	Doubles 2루타
3B	Triples 3루타
HR	Homeruns 홈런
RBI	Runs Batted In 타점
SB	Stolen Bases 도루
CS	Caught Stealing 도루실패
BB	Base on Balls 볼넷
SO	Strikeouts 삼진
IBB	Intentional walks 고의4구
HBP	Hit by pitch 사구
SH	Sacrifice hits 희생번트
SF	Sacrifice flies 희생플라이
GIDP	Grounded into double plays 땅볼병살타

레먼데이터베이스 테이블 명세서

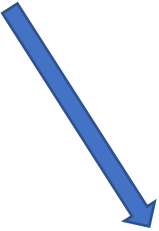
pitching 테이블

playerID	Player ID code
yearID	Year
stint	player's stint (order of appearances within a season)
teamID	Team
lgID	League
W	Wins 승
L	Losses 패
G	Games 경기수
GS	Games Started 선발 횟수
CG	Complete Games 완투
SHO	Shutouts 완봉
SV	Saves 세이브
IP0uts	Outs Pitched (innings pitched x 3) 아웃카운트(이닝*3)
H	Hits 피안타
ER	Earned Runs 자책점
HR	Homeruns 홈런
BB	Walks 볼넷
SO	Strikeouts 삼진
BAOpp	Opponent's Batting Average 상대타자 타율
ERA	Earned Run Average 평균자책점
IBB	Intentional Walks 고의4구
WP	Wild Pitches 폭투
HBP	Batters Hit By Pitch 사구
BK	Balks 보크
BFP	Batters faced by Pitcher 타석, 상대한 타자 수
GF	Games Finished 마무리 경기수
R	Runs Allowed 실점(자책 + 비자책)
SH	Sacrifices by opposing batters 희생번트
SF	Sacrifice flies by opposing batters 희생플라이
GIDP	Grounded into double plays by opposing batter 병살타

레먼데이터베이스 테이블 명세서

fielding 테이블

playerID	Player ID code
yearID	Year
stint	player's stint (order of appearances within a season)
teamID	Team
lgID	League
Pos	Position 포지션
G	Games 경기수
GS	Games Started 선발횟수
InnOuts	Time played in the field expressed as outs 플레이시간(아웃카운트)
PO	Putouts 자살
A	Assists 보살
E	Errors 실책
DP	Double Plays 더블플레이
PB	Passed Balls (by catchers) 패스트볼(포일): 포수가 충분히 잡을 수 있는 공을 놓친 경우
WP	Wild Pitches (by catchers) 포수, 와일드 피치?? 2020년 이후에는 기록이 모두 0
SB	Opponent Stolen Bases (by catchers) 포수 도루 허용
CS	Opponents Caught Stealing (by catchers) 포수 도루 저지
ZR	Zone Rating ZR = 자기 구역에서 아웃 처리한 공 / 자기 구역에 떨어진 공



```
sqlite> SELECT COUNT(*) FROM fielding WHERE WP is not null and yearID between 2000 and 2019;  
0
```

레먼 데이터베이스 열기

```
root@goorm:/workspace/KUSF_data# sqlite3 lahmanbaseballdb.sqlite
SQLite version 3.22.0 2018-01-22 18:45:57
Enter ".help" for usage hints.
sqlite> |
```

```
sqlite> .tables
allstaruttt      fielding          people
appearances     fieldingof       pitching
awardsmanagers  fieldingofsplit  pitchingpost
awardsplayers   fieldingpost     salaries
awardssharemanagers halloffame       schools
awardsshareplayers homegames        seriespost
batting          leagues          teams
battingpost     managers         teamsfranchises
collegeplaying  managershalf    teamshalf
divisions        parks
sqlite> |
```

1. 데이터베이스 열기

```
sqlite3 lahmanbaseballdb.sqlite
```

2. 데이터베이스 내 테이블 확인

```
.tables
```

레먼 데이터베이스는 people, pitching, batting 등 다양한 테이블로 구성되어 있음

레먼 데이터베이스 살펴보기

레먼 데이터베이스

people 테이블: 선수 데이터

pitching 테이블: 투수 데이터

salaries 테이블: 연봉 데이터

fielding 테이블: 수비 데이터

batting 테이블: 타자 데이터

실습1: 2016년 MLB 연봉 TOP 10 확인하기

1. 연봉 테이블 구조 확인

.schema 테이블명

```
sqlite> .schema salaries
CREATE TABLE IF NOT EXISTS "salaries" (
  "ID" INTEGER NOT NULL,
  "yearID" SMALLINT NOT NULL,
  "teamID" CHARACTER(3) NOT NULL,
  "team_ID" INTEGER NULL,
  "lgID" CHARACTER(2) NOT NULL,
  "playerID" VARCHAR(9) NOT NULL,
  "salary" DOUBLE NULL,
  PRIMARY KEY ("ID"),
  FOREIGN KEY("lgID") REFERENCES "leagues" ("lgID") ON UPDATE NO ACTION ON DELETE NO ACTION,
  FOREIGN KEY("team_ID") REFERENCES "teams" ("ID") ON UPDATE NO ACTION ON DELETE NO ACTION,
  FOREIGN KEY("playerID") REFERENCES "people" ("playerID") ON UPDATE NO ACTION ON DELETE NO ACTION
);
CREATE INDEX "salaries_lgID" ON "salaries" ("lgID");
CREATE INDEX "salaries_playerID" ON "salaries" ("playerID");
CREATE INDEX "salaries_team_ID" ON "salaries" ("team_ID");
CREATE UNIQUE INDEX "salaries_yearID" ON "salaries" ("yearID", "teamID", "lgID", "playerID");
```

필드명 또는 컬럼명



salaries 테이블 구조

ID	yearID	teamID	team_ID	lgID	playerID	salary

실습1: 2016년 MLB 연봉 TOP 10 확인하기

2. 연봉 데이터 조회

```
SELECT * FROM salaries;
```

```
26400|2016|TOR|2834|AL|tholejo01|800000.0
26401|2016|TOR|2834|AL|travide01|511200.0
26402|2016|TOR|2834|AL|tulowtr01|20000000.0
26403|2016|WAS|2835|NL|barreaa01|519400.0
26404|2016|WAS|2835|NL|belisma01|1250000.0
26405|2016|WAS|2835|NL|drewst01|3000000.0
26406|2016|WAS|2835|NL|espinda01|2875000.0
26407|2016|WAS|2835|NL|gonzagi01|12100000.0
26408|2016|WAS|2835|NL|harpebr03|5000000.0
26409|2016|WAS|2835|NL|heisech01|1250000.0
26410|2016|WAS|2835|NL|kelllesh01|4000000.0
26411|2016|WAS|2835|NL|lobatjo01|1387500.0
26412|2016|WAS|2835|NL|murphda08|8000000.0
26413|2016|WAS|2835|NL|papeljo01|10936574.0
26414|2016|WAS|2835|NL|perezol01|3000000.0
26415|2016|WAS|2835|NL|petityu01|2500000.0
26416|2016|WAS|2835|NL|ramoswi01|5350000.0
26417|2016|WAS|2835|NL|rendoan01|2800000.0
26418|2016|WAS|2835|NL|reverbe01|6250000.0
26419|2016|WAS|2835|NL|riverfe01|516100.0
26420|2016|WAS|2835|NL|roarkta01|543400.0
26421|2016|WAS|2835|NL|robincl01|534900.0
26422|2016|WAS|2835|NL|rossjo01|514400.0
26423|2016|WAS|2835|NL|scherma01|22142857.0
26424|2016|WAS|2835|NL|strasst01|10400000.0
26425|2016|WAS|2835|NL|taylomi02|524000.0
26426|2016|WAS|2835|NL|treinbl01|524900.0
26427|2016|WAS|2835|NL|werthja01|21733615.0
26428|2016|WAS|2835|NL|zimmer01|14000000.0
```

DB에 누적되어 있는 연봉 데이터가 모두 조회됨.

3. 2016년 연봉 데이터 조회

```
SELECT * FROM salaries WHERE yearID = 2016;
```

```
26397|2016|TOR|2834|AL|smoakju01|3900000.0
26398|2016|TOR|2834|AL|storedr01|8375000.0
26399|2016|TOR|2834|AL|stromma01|515900.0
26400|2016|TOR|2834|AL|tholejo01|800000.0
26401|2016|TOR|2834|AL|travide01|511200.0
26402|2016|TOR|2834|AL|tulowtr01|20000000.0
26403|2016|WAS|2835|NL|barreaa01|519400.0
26404|2016|WAS|2835|NL|belisma01|1250000.0
26405|2016|WAS|2835|NL|drewst01|3000000.0
26406|2016|WAS|2835|NL|espinda01|2875000.0
26407|2016|WAS|2835|NL|gonzagi01|12100000.0
26408|2016|WAS|2835|NL|harpebr03|5000000.0
26409|2016|WAS|2835|NL|heisech01|1250000.0
26410|2016|WAS|2835|NL|kelllesh01|4000000.0
26411|2016|WAS|2835|NL|lobatjo01|1387500.0
26412|2016|WAS|2835|NL|murphda08|8000000.0
26413|2016|WAS|2835|NL|papeljo01|10936574.0
26414|2016|WAS|2835|NL|perezol01|3000000.0
26415|2016|WAS|2835|NL|petityu01|2500000.0
26416|2016|WAS|2835|NL|ramoswi01|5350000.0
26417|2016|WAS|2835|NL|rendoan01|2800000.0
26418|2016|WAS|2835|NL|reverbe01|6250000.0
26419|2016|WAS|2835|NL|riverfe01|516100.0
26420|2016|WAS|2835|NL|roarkta01|543400.0
26421|2016|WAS|2835|NL|robincl01|534900.0
26422|2016|WAS|2835|NL|rossjo01|514400.0
26423|2016|WAS|2835|NL|scherma01|22142857.0
26424|2016|WAS|2835|NL|strasst01|10400000.0
26425|2016|WAS|2835|NL|taylomi02|524000.0
26426|2016|WAS|2835|NL|treinbl01|524900.0
26427|2016|WAS|2835|NL|werthja01|21733615.0
26428|2016|WAS|2835|NL|zimmer01|14000000.0
```

2016년 연봉 데이터만 조회됨.

실습1: 2016년 MLB 연봉 TOP 10 확인하기

4. 2016년 연봉이 높은 사람부터 낮은 사람 순으로 정렬

```
SELECT * FROM salaries WHERE yearID = 2016 ORDER BY salary DESC;
```

5. 2016년 연봉 TOP 10 조회

조회한 결과 중 처음 10개 행만

```
SELECT * FROM salaries WHERE yearID = 2016 ORDER BY salary DESC LIMIT 10;
```

```
sqlite> SELECT * FROM salaries WHERE yearID = 2016 ORDER BY salary DESC LIMIT 10;
25966|2016|LAN|2819|NL|kershcl01|33000000.0
25589|2016|ARI|2806|NL|greinza01|31799030.0
25674|2016|BOS|2809|AL|priceda01|30000000.0
25833|2016|DET|2815|AL|cabremi01|28000000.0
25859|2016|DET|2815|AL|verlaju01|28000000.0
26067|2016|NYN|2824|NL|cespeyo01|27328046.0
26242|2016|SEA|2829|AL|hernafe02|25857143.0
25702|2016|CHN|2811|NL|lestejo01|25000000.0
25933|2016|LAA|2818|AL|pujolal01|25000000.0
26111|2016|NYA|2823|AL|sabatcc01|25000000.0
```

실습1: 2016년 MLB 연봉 TOP 10 확인하기

6. 컬럼별로 정리되어 보여지게 하고, 컬럼명도 함께 보여지게 하기

.mode column

.header on

```
sqlite> .mode column
sqlite> SELECT * FROM salaries WHERE yearID = 2016 ORDER BY salary DESC LIMIT 10;
```

25966	2016	LAN	2819	NL	kershcl01	33000000.0
25589	2016	ARI	2806	NL	greinza01	31799030.0
25674	2016	BOS	2809	AL	priceda01	30000000.0
25833	2016	DET	2815	AL	cabremi01	28000000.0
25859	2016	DET	2815	AL	verlaju01	28000000.0
26067	2016	NYN	2824	NL	cespeyo01	27328046.0
26242	2016	SEA	2829	AL	hernafe02	25857143.0
25702	2016	CHN	2811	NL	lestejo01	25000000.0
25933	2016	LAA	2818	AL	pujolal01	25000000.0
26111	2016	NYA	2823	AL	sabatcc01	25000000.0

컬럼별로 정리됨

```
sqlite> .header on
sqlite> SELECT * FROM salaries WHERE yearID = 2016 ORDER BY salary DESC LIMIT 10;
```

ID	yearID	teamID	team_ID	lgID	playerID	salary
25966	2016	LAN	2819	NL	kershcl01	33000000.0
25589	2016	ARI	2806	NL	greinza01	31799030.0
25674	2016	BOS	2809	AL	priceda01	30000000.0
25833	2016	DET	2815	AL	cabremi01	28000000.0
25859	2016	DET	2815	AL	verlaju01	28000000.0
26067	2016	NYN	2824	NL	cespeyo01	27328046.0
26242	2016	SEA	2829	AL	hernafe02	25857143.0
25702	2016	CHN	2811	NL	lestejo01	25000000.0
25933	2016	LAA	2818	AL	pujolal01	25000000.0
26111	2016	NYA	2823	AL	sabatcc01	25000000.0

컬럼명도 보임

그런데 이 선수들은 대체 누구?

실습1: 2016년 MLB 연봉 TOP 10 확인하기

7. people 테이블의 구조 확인

.schema people

```
sqlite> .schema people
CREATE TABLE IF NOT EXISTS "people" (
  "playerID" VARCHAR(9) NOT NULL,
  "birthYear" INTEGER NULL,
  "birthMonth" INTEGER NULL,
  "birthDay" INTEGER NULL,
  "birthCountry" VARCHAR(255) NULL,
  "birthState" VARCHAR(255) NULL,
  "birthCity" VARCHAR(255) NULL,
  "deathYear" INTEGER NULL,
  "deathMonth" INTEGER NULL,
  "deathDay" INTEGER NULL,
  "deathCountry" VARCHAR(255) NULL,
  "deathState" VARCHAR(255) NULL,
  "deathCity" VARCHAR(255) NULL,
  "nameFirst" VARCHAR(255) NULL,
  "nameLast" VARCHAR(255) NULL,
  "nameGiven" VARCHAR(255) NULL,
  "weight" INTEGER NULL,
  "height" INTEGER NULL,
  "bats" VARCHAR(255) NULL,
  "throws" VARCHAR(255) NULL,
  "debut" VARCHAR(255) NULL,
  "finalGame" VARCHAR(255) NULL,
  "retroID" VARCHAR(255) NULL,
  "bbrefID" VARCHAR(255) NULL,
  "birth_date" DATE NULL,
  "debut_date" DATE NULL,
  "finalgame_date" DATE NULL,
  "death_date" DATE NULL,
  PRIMARY KEY ("playerID")
);
```

실습1: 2016년 MLB 연봉 TOP 10 확인하기

8. playerId로 어떤 선수인지 확인하기

```
SELECT * FROM people WHERE playerId = 'kershcl01';
```

```
sqlite> SELECT * FROM people WHERE playerId = 'kershcl01';
```

playerID	birthYear	birthMonth	birthDay	birthCountry	birthState	birthCity	deathYear	deathMonth	deathDay	deathCountry	deathState	deathCity	nameFirst	nameLast	nameGiven	weight	height
ght	bats	throws	debut	finalGame	retroID	bbrefID	birth_date	debut_date	finalgame_date	death_date							
kershcl01	1988	3	19	USA	TX	Dallas							Clayton	Kershaw	Clayton Edward	226	76
	L	L	2008-05-25	2019-09-29	kersc001	kershcl01	1988-03-19	2008-05-25	2019-09-29								

```
sqlite> SELECT * FROM salaries WHERE yearID = 2016 ORDER BY salary DESC LIMIT 10;
```

ID	yearID	teamID	team_ID	lgID	playerID	salary
25966	2016	LAN	2819	NL	kershcl01	33000000.0
25589	2016	ARI	2806	NL	greinza01	31799030.0
25674	2016	BOS	2809	AL	priceda01	30000000.0
25833	2016	DET	2815	AL	cabremi01	28000000.0
25859	2016	DET	2815	AL	verlaju01	28000000.0
26067	2016	NYN	2824	NL	cespeyo01	27328046.0
26242	2016	SEA	2829	AL	hernafe02	25857143.0
25702	2016	CHN	2811	NL	lestejo01	25000000.0
25933	2016	LAA	2818	AL	pujolal01	25000000.0
26111	2016	NYA	2823	AL	sabatcc01	25000000.0

- 클레이튼 커쇼 (약 395억)
- 잭 그레인키
- 데이빗 프라이스
- 미겔 카브레라
- 저스틴 벌랜더
- 요에니스 세스페데스
- 펠릭스 에르난데스
- 존 레스터
- 알버트 푸홀스
- CC 사바시아

실습2: 류현진 연봉 확인하기

1. 류현진의 성을 이용해서 류현진의 playerId 찾기

```
SELECT playerId, nameFirst, nameLast FROM people WHERE nameLast = 'Ryu';
```

특정 컬럼들의 데이터만 보고 싶을 때
*은 모든 컬럼을 보고 싶을 때

```
sqlite> SELECT playerId, nameFirst, nameLast FROM people WHERE nameLast = 'Ryu';
playerID    nameFirst  nameLast
-----
ryuhy01     Hyun-Jin   Ryu
ryuja01     Jae Kuk    Ryu
```

실습2: 류현진 연봉 확인하기

2. 류현진의 playerId를 활용하여 류현진의 연봉 조회

```
SELECT * FROM salaries WHERE playerId = 'ryuhy01';
```

```
sqlite> SELECT * FROM salaries WHERE playerId = 'ryuhy01';
ID      yearID  teamID  team_ID  lgID  playerId  salary
-----
23513   2013    LAN     2729     NL     ryuhy01   3333333.0
24321   2014    LAN     2759     NL     ryuhy01   4333000.0
25973   2016    LAN     2819     NL     ryuhy01   7833333.0
```

3. 다른 데이터 말고 연봉만 보고 싶다면?

```
SELECT salary FROM salaries WHERE playerId = 'ryuhy01';
```

```
salary
-----
3333333.0
4333000.0
7833333.0
```

집계함수(aggregate)

집계함수	설명
AVG	평균
COUNT	개수
MAX	최대값
MIN	최소값
SUM	합계(정수들의 합계라면 정수로)
TOTAL	합계(항상 실수로)
GROUP_CONCAT	문자열들 연결

4. 류현진의 연봉을 모두 합하면?

```
SELECT SUM(salary) FROM salaries WHERE playerId = 'ryuhy01';
```

```
15499666.0
```

약 185억

실습3: MLB 역대 한국 선수들 조회하기

1. people 테이블에서 birthCountry가 South Korea인 선수 조회

```
SELECT * FROM people WHERE birthCountry = 'South Korea';
```

2. 데뷔순으로 조회

```
SELECT * FROM people WHERE birthCountry = 'South Korea' ORDER BY debut;
```

playerID	deathDay	deathCountry	deathState	deathCity	nameFirst	nameLast	nameGiven	weight	height	bats	throws
debut	finalGame	retroID	bbrefID	birth_date	debut_date	finalgame_date	death_date				
parkch01	1994-04-08	2010-10-01	parkc002	parkch01	1973-06-30	1994-04-08	2010-10-01	210	74	R	R
choji01	1998-07-04	1999-07-30	cho-j001	choji01	1975-08-16	1998-07-04	1999-07-30	175	72	R	R
kimby01	1999-05-29	2007-09-28	kim-b001	kimby01	1979-01-19	1999-05-29	2007-09-28	176	71	R	R
leesa01	2000-06-29	2000-10-01	lee-s001	leesa01	1971-03-11	2000-06-29	2000-10-01	190	73	L	L
kimsu01	2001-06-15	2006-09-17	kim-s001	kimsu01	1977-09-04	2001-06-15	2006-09-17	180	74	R	R
bongju01	2002-04-23	2004-06-20	bongj001	bongju01	1980-07-15	2002-04-23	2004-06-20	175	75	L	L
seoja01	2002-07-21	2007-05-29	seo-j001	seoja01	1977-05-24	2002-07-21	2007-05-29	215	73	R	R
choihe01								235	77	L	L

3. 총 몇 명인지?

```
SELECT COUNT(*) FROM people WHERE birthCountry = 'South Korea';
```

23

룸 레프스나يدر(김정태): 한국에서 태어났지만 생후 5개월만에 미국으로 입양됨.
토미 펠프스: 아버지가 주한미군으로 근무할 때 출생.

실습4: MLB 2019시즌 홈런왕은 누구?

1. batting 테이블 구조 확인

```
.schema batting
```

2. 2019년도 타자기록 홈런 개수 역순으로 정렬한 후 첫번째 행만 조회

```
SELECT * FROM batting WHERE yearID = 2019 ORDER BY HR DESC LIMIT 1;
```

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											
105902	alonspe01	2019	1	NYN	2914	NL	161		597	103	155	30	2	53	120	1	0
72	163	6	21	0	3	13											

피트 알론소
2021 시즌에도 홈런 37개로 3위

3. 다른 데이터 말고 playerID와 홈런 개수만 보고 싶다면?

```
SELECT playerID, HR FROM batting WHERE yearID = 2019 ORDER BY HR DESC;
```

playerID	HR
alonspe01	53

실습5: MLB 2019시즌 탈삼진 TOP 5는 누구?

1. pitching 테이블 구조 확인

```
.schema pitching
```

2. 2019년도 투수 기록 탈삼진 개수 내림차순으로 정렬한 후 처음 5개 행만 조회

```
SELECT playerID, SO FROM pitching WHERE yearID = 2019 ORDER BY SO DESC LIMIT 5;
```

playerID	SO
college01	326
verlaju01	300
biebesh01	259
degroja01	255
strasst01	251

모두 굉장히 유명한 선수들

게릿 콜
저스틴 벌랜더
세인 비버
제이콥 디그롬
스티븐 스트라스버그

* 삼진을 잘 잡는 투수의 몸값이 대체로 높은 편. 그 이유는? 이후 강의에서 설명 예정^^

실습6: MLB 2018시즌 도루 저지를 가장 많이 한 포수는 누구?

1. fielding 테이블 구조 확인

```
.schema fielding
```

2. 2018년도 도루 저지 개수 내림차순으로 정렬한 후 처음 5개 행만 조회

```
SELECT playerID, CS FROM fielding WHERE yearID = 2018 ORDER BY CS DESC LIMIT 5;
```

<u>playerID</u>	CS	
lucrojo01	31	조너선 루크로이
contrwi01	27	윌슨 콘트레라스
mccanja02	27	제임스 매캔
perezsa02	25	살바도르 페레스
cervefr01	23	프란시스코 세르벨리

Sqlite3 쿼리 문법 요약

목적	쿼리 문법	예시
데이터베이스 열기	sqlite3 데이터베이스 파일명	sqlite3 lahmanbaseballdb.sqlite
존재 테이블 확인	.tables	
컬럼명과 함께 데이터 조회	.mode column .header on	
테이블 구조 확인	.schema 테이블명	.schema batting
데이터 조회	SELECT * FROM 테이블명; (모든 필드 조회시) SELECT 필드명1, 필드명2,... FROM 테이블명; (특정 필드 조회시)	SELECT * FROM batting; SELECT playerID, HR FROM batting;
조건을 가지고 데이터 조회	SELECT * FROM 테이블명 WHERE 조건;	SELECT * FROM batting WHERE yearID = 2019;
지정 컬럼을 기준으로 데이터 정렬	SELECT * FROM 테이블명 ORDER BY 필드명;	SELECT * FROM batting ORDER BY HR; SELECT * FROM batting ORDER BY HR DESC; (내림차순 정렬)
설정한 개수의 데이터만 조회	SELECT * FROM 테이블명 LIMIT 개수;	SELECT * FROM batting LIMIT 5;
조회한 행 개수 확인	SELECT COUNT(*) FROM 테이블명;	SELECT COUNT(*) FROM batting;
조회한 데이터 합계	SELECT SUM(필드명) FROM 테이블명;	SELECT SUM(HR) FROM batting;

중요한 것은 필요에 따라 이 문법들을 잘 조합해서 사용해야 한다는 점!

과제

레먼데이터베이스에서 흥미로운 데이터 5건 조회하기

예시1. 2018 정규시즌에 3루타를 가장 많이 친 타자 3명은?

예시2. 추신수 선수의 연봉이 가장 높았던 연도는?

예시3. 2019 정규시즌에 평균자책점이 가장 좋았던 투수 10명은?

예시4. 2019 포스트시즌에 가장 많은 홈런을 친 타자는?

예시5. 2019 정규시즌에 가장 많은 실책을 범한 선수 5명은?