

actividad1

January 19, 2023

Curso: Herramientas de Bases de Datos

Estudiante: Santiago Castro Moncada

1 Indice:

1. Diseño de la base de datos,
 - 1.2 Diseño conceptual
 - 1.2 Diseño lógico
 - 1.3 Diseño físico
2. Inserción de los datos a la base de datos
3. Consultas a base de datos
 - 3.1 Consultas simples/básicas
 - 3.2 Consultas multitable (con JOIN)
 - 3.3 Consultas con función de agregación (operaciones sobre conjuntos de filas)
 - 3.4 Subconsultas
4. Actualizaciones a base de datos
5. Borrados a base de datos y
6. Conclusiones.

1.1 1.1 Diseño conceptual

En el archivo HTML adjunto se detalla los atributos de cada tabla

file: beisbol.html

1.2 1.2 Diseño lógico

Tabla 1:

Jugador(jugadorID, nombre, apellidos, nombreDeportivo, anyoNacimiento, mesNacimiento, diaNacimiento, paisNacimiento, estadoNacimiento, ciudadNacimiento, anyoFallecimiento, mesFallecimiento, diaFallecimiento)

imiento, diaFallecimiento,paisFallecimiento, estadoFallecimiento, ciudadFallecimiento, peso, altura, batear, lanzar, fechaDebut, fechaRetirada)

PK: jugadorID

Tabla 2:

Equipo(ano, liga, equipoId, equipoNombre, estadio, publicoEstadio, equipoDivision, clasificacion-Final, G, Ghome, W, L, DivWin, WCWin, LgWin, WSWin, R, AB, H, B2, B3, HR, BB, SO, SB, CS, HBP, SF, RA, ER, ERA, CG, SHO, SV, IPouts, HA, HRA, BBA, SOA, E, DP, FP, BPF, PPF)

PK: equipoId

Tabla 3:

Estadio(estadioID, nombre, alias, ciudad, estado, pais)

PK: estadioID

Tabla 4:

Universidad(universidadID, nombre, ciudad, estado, pais)

PK: universidadID

Tabla 5:

BatearEstadistica(id, jugador, ano, equipo, liga, G, AB, R, H, B2, B3, HR, RBI, SB, CS, BB, SO, IBB, HBP, SH, SF, GIDP)

PK: id

FK1: BatearEstadistica.jugador -> Jugador.jugadorID

FK2: BatearEstadistica.equipo -> Equipo.equipoId

Tabla 6:

LanzarEstadistica(id,jugador, ano, equipo, liga, W, L, G, GS, CG, SHO, SV, IPouts, H, ER, HR, BB, SO, BAOpp, ERA, IBB, WP, HBP, BK, BFP, GF, R, SH, SF, GIDP)

PK: id

FK1: LanzarEstadistica.jugador -> Jugador.jugadorID

FK2: LanzarEstadistica.equipo -> Equipo.equipoId

Tabla 7:

CampoEstadistica(id,jugador, ano, equipo, liga, POS, G, GS, InnOuts, PO, A, E, DP, PB, WP, SB, CS, ZR)

PK: id

FK1: CampoEstadistica.jugador -> Jugador.jugadorID

FK2: CampoEstadistica.equipo -> Equipo.equipoId

Tabla 8:

Sueldo(id, anyo, equipo, liga, jugador, sueldo)

PK: id

FK1: Sueldo.jugador -> Jugador.jugadorID

FK2: Sueldo.equipo -> Equipo.equipoId

Tabla 9:

JugadorPosicionCampo(id, jugador, anyo, Glf, Gcf, Grf)

PK: id

FK1: JugadorPosicionCampo.jugador -> Jugador.jugadorID

Tabla 10:

JugadorPremio(id, jugador, premio, anyo, liga)

PK: id

FK1: JugadorPremio.jugador -> Jugador.jugadorID

Tabla 11:

PremioVotacion(id, premio, anyo, liga, jugador, puntosGanados, puntosPremio, votosPrimer-Puesto)

PK: id

FK1: PremioVotacion.jugador -> Jugador.jugadorID

Tabla 12:

JugadorPosicionDetalle(id, anyo, equipo, liga, jugador, G_all, GS, G_batting, G_defense, G_p, G_c, G_1b, G_2b, G_3b, G_ss, G_lf, G_cf, G_rf, G_of, G_dh, G_ph, G_pr)

PK: id

FK1: JugadorPosicionDetalle.jugador -> Jugador.jugadorID

FK2: JugadorPosicionDetalle.equipo -> Equipo.equipoId

Tabla 13:

JugadorUniversidad(id, jugador, universidad, anyo)

PK: id

FK1: JugadorUniversidad.jugador -> Jugador.jugadorID

FK2: JugadorUniversidad.universidad -> Universidad.universidadID

Tabla 14:

EstadioPartidos(id, anyo, liga, equipo, estadio, primerPartido, ultimoPartido, partidosJugados, publico)

PK: id

FK1: EstadioPartidos.equipo -> Equipo.EquipoID

FK2: EstadioPartidos.estadio -> Estadio.estadioID

1.3 1.3 Diseño Fisico

COMANDO PARA EJECUTAR LA BD EN DOCKER `docker run --name postgresql -e POSTGRES_USER=postgres -e POSTGRES_PASSWORD=postgres -p 5432:5432 -v /data:/var/lib/postgresql/data -d postgres`

```
[25]: import psycopg2
```

```
DATABASE="beisbol"
USER="postgres"
PASSWORD="postgres"
conexion1 = psycopg2.connect(
    host='localhost',
    database=DATABASE,
    user=USER,
    password=PASSWORD,
    port='5432'
)
cursor=conexion1.cursor()
conexion1.autocommit=True
```

```
[2]: # CREACIÓN DE TABLAS
```

```
jugador = """
CREATE TABLE IF NOT EXISTS jugador(
    jugadorID VARCHAR(30) PRIMARY KEY,
    nombre VARCHAR(25),
    apellidos VARCHAR(25),
    nombreDeportivo VARCHAR(50),
    anyoNacimiento SMALLINT,
    mesNacimiento SMALLINT,
    diaNacimiento SMALLINT,
    paisNacimiento VARCHAR(40),
    estadoNacimiento VARCHAR(50),
    ciudadNacimiento VARCHAR(40),
    anyoFallecimiento SMALLINT,
    mesFallecimiento SMALLINT,
    diaFallecimiento SMALLINT,
    paisFallecimiento VARCHAR(40),
    estadoFallecimiento VARCHAR(50),
    ciudadFallecimiento VARCHAR(40),
    peso SMALLINT,
    altura SMALLINT,
    batear CHAR(1),
    lanzar CHAR(1),
    fechaDebut DATE,
    fechaRetirada DATE
)
```

```

)""
equipo=""
CREATE TABLE IF NOT EXISTS equipo(
    anyo SMALLINT,
    liga CHAR(5),
    equipoID VARCHAR(30) PRIMARY key NOT NULL,
    equipoNombre VARCHAR(50),
    estadio VARCHAR(60),
    publicoEstadio INTEGER,
    equipoDivision CHAR(10),
    clasificacionFinal SMALLINT,
    G SMALLINT,
    Ghome SMALLINT,
    W SMALLINT,
    L SMALLINT,
    DivWin CHAR(2),
    WCWin CHAR(2),
    LgWin CHAR(2),
    WSWin CHAR(2),
    R SMALLINT,
    AB SMALLINT,
    H SMALLINT,
    B2 SMALLINT,
    B3 SMALLINT,
    HR SMALLINT,
    BB SMALLINT,
    SO SMALLINT,
    SB SMALLINT,
    CS SMALLINT,
    HBP SMALLINT,
    SF SMALLINT,
    RA SMALLINT,
    ER SMALLINT,
    ERA SMALLINT,
    CG SMALLINT,
    SHO SMALLINT,
    SV SMALLINT,
    IPouts SMALLINT,
    HA SMALLINT,
    HRA SMALLINT,
    BBA SMALLINT,
    SOA SMALLINT,
    E SMALLINT,
    DP SMALLINT,
    FP SMALLINT,
    BPF SMALLINT,
    PPF SMALLINT

```

```

)"""
estadio = """
CREATE TABLE IF NOT EXISTS estadio(
    estadioID VARCHAR(10) PRIMARY key NOT NULL,
    nombre VARCHAR(50),
    alias VARCHAR(70),
    ciudad VARCHAR(40),
    estado VARCHAR(50),
    pais VARCHAR(30)
)"""
universidad = """
CREATE TABLE IF NOT EXISTS universidad(
    universidadID VARCHAR(20) PRIMARY key NOT NULL,
    nombre VARCHAR(70),
    ciudad VARCHAR(70),
    estado VARCHAR(30),
    pais VARCHAR(30)
)"""
cursor.execute(universidad)
cursor.execute(estadio)
cursor.execute(equipo)
cursor.execute(jugador)

```

```

[3]: batearEstadistica = """
CREATE TABLE IF NOT EXISTS batearEstadistica(
    jugador VARCHAR(30),
    anyo SMALLINT,
    equipo VARCHAR(30),
    liga CHAR(5),
    G SMALLINT,
    AB SMALLINT,
    R SMALLINT,
    H SMALLINT,
    B2 SMALLINT,
    B3 SMALLINT,
    HR SMALLINT,
    RBI SMALLINT,
    SB SMALLINT,
    CS SMALLINT,
    BB SMALLINT,
    SO SMALLINT,
    IBB SMALLINT,
    HBP SMALLINT,
    SH SMALLINT,
    SF smallint,
    GDP smallint,
    foreign key (jugador) references jugador(jugadorID),

```

```

        foreign key (equipo) references equipo(equipoID)
    )""
lanzarEstadistica = ""
CREATE TABLE IF NOT EXISTS lanzarEstadistica(
    jugador VARCHAR(30),
    anyo SMALLINT,
    equipo VARCHAR(30),
    liga CHAR(5),
    W SMALLINT,
    L SMALLINT,
    G SMALLINT,
    GS SMALLINT,
    CG SMALLINT,
    SHO SMALLINT,
    SV SMALLINT,
    IPouts SMALLINT,
    H SMALLINT,
    ER SMALLINT,
    HR SMALLINT,
    BB SMALLINT,
    SO SMALLINT,
    BAOpp float,
    ERA SMALLINT,
    IBB SMALLINT,
    WP SMALLINT,
    HBP SMALLINT,
    BK SMALLINT,
    BFP SMALLINT,
    GF SMALLINT,
    R SMALLINT,
    SH SMALLINT,
    SF SMALLINT,
    GIDP SMALLINT,
    foreign key (jugador) references jugador(jugadorID),
    foreign key (equipo) references equipo(equipoID)
)""
campoEstadistica = ""
CREATE TABLE IF NOT EXISTS campoEstadistica(
    jugador VARCHAR(30),
    anyo SMALLINT,
    equipo VARCHAR(30),
    liga CHAR(5),
    POS CHAR(5),
    G SMALLINT,
    GS SMALLINT,
    InnOuts SMALLINT,
    PO SMALLINT,

```

```

        A SMALLINT,
        E SMALLINT,
        DP SMALLINT,
        PB SMALLINT,
        WP SMALLINT,
        SB SMALLINT,
        CS SMALLINT,
        ZR SMALLINT,
        foreign key (jugador) references jugador(jugadorID),
        foreign key (equipo) references equipo(equipoID)
    )""
sueldo = ""
CREATE TABLE IF NOT EXISTS sueldo(
    anyo SMALLINT,
    equipo VARCHAR(30),
    liga CHAR(5),
    jugador VARCHAR(30),
    sueldo INT,
    foreign key (equipo) references equipo(equipoID),
    foreign key (jugador) references jugador(jugadorID)
)""
jugadorPosicionCampo = ""
CREATE TABLE IF NOT EXISTS jugadorPosicionCampo(
    jugador VARCHAR(30),
    anyo SMALLINT,
    Glf SMALLINT,
    Gcf SMALLINT,
    Grf SMALLINT,
    foreign key (jugador) references jugador(jugadorID)
)""
jugadorPremio = ""
CREATE TABLE IF NOT EXISTS jugadorPremio(
    jugador VARCHAR(30),
    premio VARCHAR(50),
    anyo SMALLINT,
    liga CHAR(5),
    foreign key (jugador) references jugador(jugadorID)
)""
premioVotacion = ""
CREATE TABLE IF NOT EXISTS premioVotacion(
    premio VARCHAR(50),
    anyo SMALLINT,
    liga CHAR(5),
    jugador VARCHAR(30),
    puntosGanados SMALLINT,
    puntosPremio SMALLINT,
    votosPrimerPuesto SMALLINT,

```



```

        foreign key (jugador) references jugador(jugadorID)
    )""
jugadorPosicionDetalle = ""
CREATE TABLE IF NOT EXISTS jugadorPosicionDetalle(
    anyo SMALLINT,
    equipo VARCHAR(30),
    liga CHAR(5),
    jugador VARCHAR(30),
    G_all SMALLINT,
    GS SMALLINT,
    G_batting SMALLINT,
    G_defense SMALLINT,
    G_p SMALLINT,
    G_c SMALLINT,
    G_1b SMALLINT,
    G_2b SMALLINT,
    G_3b SMALLINT,
    G_ss SMALLINT,
    G_lf SMALLINT,
    G_cf SMALLINT,
    G_rf SMALLINT,
    G_of SMALLINT,
    G_dh SMALLINT,
    G_ph SMALLINT,
    G_pr SMALLINT,
    foreign key (equipo) references equipo(equipoID),
    foreign key (jugador) references jugador(jugadorID)
)""
jugadorUniversidad = ""
CREATE TABLE IF NOT EXISTS jugadorUniversidad(
    jugador VARCHAR(30),
    universidad VARCHAR(20),
    anyo SMALLINT,
    foreign key (jugador) references jugador(jugadorID),
    foreign key (universidad) references universidad(universidadID)
)""
estadioPartidos = ""
CREATE TABLE IF NOT EXISTS estadioPartidos(
    anyo SMALLINT,
    liga CHAR(5),
    equipo VARCHAR(30),
    estadio VARCHAR(10),
    primerPartido VARCHAR(20),
    ultimoPartido VARCHAR(20),
    partidosJugados SMALLINT,
    publico INT,
    foreign key (equipo) references equipo(equipoID),

```

```
foreign key (estadio) references estadio(estadioID)
)""
cursor.execute(estadioPartidos)
cursor.execute(jugadorUniversidad)
cursor.execute(jugadorPosicionDetalle)
cursor.execute(premioVotacion)
cursor.execute(jugadorPremio)
cursor.execute(jugadorPosicionCampo)
cursor.execute(sueldo)
cursor.execute(campoEstadistica)
cursor.execute(lanzarEstadistica)
cursor.execute(batearEstadistica)
```

2 2. Inserción de datos

[4]:

[illegible]

```

cursor.execute(insert, values1)
cursor.execute(insert, values2)

## Tabla Estadio
insert = "INSERT INTO estadio VALUES (%s,%s,%s,%s,%s,%s)"
values1 = ("PAL01","Angel Stadium of Anaheim","Edison Field; Anaheim_
↳Stadium","Anaheim","CA","US")
values2 = ("ATP02","Turner Field","", "Atlanta","GA","US")

cursor.execute(insert, values1)
cursor.execute(insert, values2)

## Tabla Universidad
insert = "INSERT INTO universidad VALUES (%s,%s,%s,%s,%s)"
values1 = ("chirstee","Abilene Christian University","Abilene","TX","USA")
values2 = ("udelphi","Adelphi University","Garden City","NY","USA")

cursor.execute(insert, values1)
cursor.execute(insert, values2)

```

```

[5]: ##Tabla batearEstadistica
insert="INSERT INTO batearEstadistica VALUES_
↳(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
values1=("apashaml34",1967,"ATT","NL",158,603,117,168,23,1,44,127,21,3,76,96,15,1,0,8,14)
values2=("apashaml34",1979,"APO","NL",155,600,113,184,37,3,39,109,17,6,63,97,19,0,0,6,11)

cursor.execute(insert,values1)
cursor.execute(insert,values2)

## Tabla campoEstadistica
insert="INSERT INTO campoEstadistica VALUES_
↳(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
values1=("apashaml34",1966,"APO","NL","OF",158,157,4066,15,12,4,5,0,0,0,0,0)
values2=("jmerliee12",1968,"APO","NL","OF",151,143,3874,330,13,3,2,0,0,0,0,0,0)

cursor.execute(insert,values1)
cursor.execute(insert,values2)

## Tabla 6 lanzarEstadistica
insert="INSERT INTO lanzarEstadistica VALUES_
↳(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
values1=("jmerliee12",2023,"ATT","NL",2,3,8,8,
0,0,0,101,45,22,7,12,23,0.328,588,2,1,1,1,153,0,23,1,2,3)
values2=("apashaml34",2020,"ATT","NL",4,0,
35,0,0,0,0,148,69,36,10,5,37,0.327,657,0,0,1,1,218,6,36,0,1,4)

cursor.execute(insert, values1)

```

```

cursor.execute(insert, values2)

## Tabla sueldo
insert="INSERT INTO sueldo VALUES (%s,%s,%s,%s,%s)"
values1=(1985,"ATT","NL","jmerliee12",870000)
values2=(1985,"APO","NL","jmerliee12",550000)

cursor.execute(insert, values1)
cursor.execute(insert, values2)

## Tabla jugadorPosicionCampo
insert = "INSERT INTO jugadorPosicionCampo VALUES (%s,%s,%s,%s,%s)"
values1 = ("apashaml34",1964,105,0,11)
values2 = ("apashaml34",1975,26,0,105)

cursor.execute(insert, values1)
cursor.execute(insert, values2)

## Tabla jugadorPremio
insert="INSERT INTO jugadorPremio VALUES (%s,%s,%s,%s)"
values1=("apashaml34","TSN All-Star",1956,"ML")
values2=("apashaml34","TSN Player of the Year",1956,"NL")

cursor.execute(insert, values1)
cursor.execute(insert, values2)

## Tabla premioVotacion
insert = "INSERT INTO premioVotacion VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s)"
values1=("MVP",1955,"NL","jmerliee12",36,336,0)
values2=("MVP",1956,"NL","apashaml34",146,336,0)
cursor.execute(insert, values1)
cursor.execute(insert, values2)

## Tabla jugadorPosicionDetalle
insert="INSERT INTO jugadorPosicionDetalle VALUES_
↳(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
values1=(1966,"APO","NL","apashaml34",158,157,158,158,0,0,0,2,0,0,0,5,158,158,0,1,0)
values2=(1968,"ATT","NL","apashaml34",160,157,160,157,0,0,14,0,0,0,0,0,151,151,0,3,0)

cursor.execute(insert,values1)
cursor.execute(insert,values2)

## Tabla jugadorUniversidad
insert="INSERT INTO jugadoruniversidad VALUES (%s,%s,%s)"
values1=("jmerliee12","chirstee",2000)

```

```

values2=("apashaml34","chirstee",1997)

cursor.execute(insert,values1)
cursor.execute(insert,values2)

## Tabla estadioPartidos
insert="INSERT INTO estadioPartidos VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s)"
values1=(1966,"NL","APO","PAL01","12/04/1966","12/09/1966",82,1539796)
values2=(1967,"NL","APO","PAL01","02/04/1967","10/10/1967",81,1379214)

cursor.execute(insert,values1)
cursor.execute(insert,values2)

```

```

[6]: #Funcion para la inserción de datos
import csv
RUTA_LOCAL='/mnt/c/Users/santiagoc/Downloads/ficheros/'

def funcion_de_inserción(ruta:str,
                           nombre_tabla:str,
                           numero_columnas:int):
    """Esta función accede al fichero correspondiente y hace la inserción de
    ↪ los registros en la base de datos

    Parámetros:
        ruta (str): Ruta local de la carpeta con los ficheros
        nombre_tabla (str): Nombre de la tabla en la base de datos para ese
    ↪ archivo
        numero_columnas (int): Número de atributos de la tabla

    """
    RUTA = f'{RUTA_LOCAL}{ruta}'

    with open(RUTA, newline='') as file:
        reader = csv.DictReader(file, delimiter=';')
        string_formart = f'{"%s,"*(numero_columnas-1) + "%s"}'
        string = f'INSERT INTO {nombre_tabla} VALUES ({string_formart})'
        values = []
        for row in reader:
            valores = tuple(row.values())
            values.append(tuple(map(lambda x: None if x=="" else x, valores)))
        cursor.executemany(string, values)

```

```

[7]: ## Tablas iniciales sin llaves foraneas
funcion_de_inserción('1-Jugador.csv', 'jugador', 22)
funcion_de_inserción('5-Equipo.csv', 'equipo', 44)
funcion_de_inserción('12-Universidad.csv', 'universidad', 5)
funcion_de_inserción('10-Estadio.csv', 'estadio', 6)
conexion1.commit()

```

```
[8]: # Insertar de tablas de estadísticas
    ## NOTA: estas tablas pueden tardar mas tiempo en insertar ya que tienen
    ↪ muchos registros
    # se recomienda ejecutar de a una tabla a la vez
    funcion_de_inserción('2-BatearEstadistica.csv', 'batearEstadistica', 21)
    funcion_de_inserción('3-LanzarEstadistica.csv', 'lanzarEstadistica', 29)
    funcion_de_inserción('4-CampoEstadistica.csv', 'campoEstadistica', 17)
    conexion1.commit()

[9]: funcion_de_inserción('6-Sueldo.csv', 'sueldo', 5)
    funcion_de_inserción('7-JugadorPosicionCampo.csv', 'jugadorPosicionCampo', 5)
    funcion_de_inserción('8-JugadorPremio.csv', 'jugadorPremio', 4)
    funcion_de_inserción('9-PremioVotacion.csv', 'premioVotacion', 7)
    funcion_de_inserción('11-JugadorPosicionDetalle.csv', 'jugadorPosicionDetalle',
    ↪ 21)
    funcion_de_inserción('13-JugadorUniversidad.csv', 'jugadorUniversidad', 3)
    funcion_de_inserción('14-EstadioPartidos.csv', 'estadioPartidos', 8)
    conexion1.commit()
```

3. Consultas a base de datos

3.1 Consulas simples

Mostrar los primeros 10 registros de la tabla equipos

```
[10]: consulta="SELECT equipoid, equiponombre, anyo, liga, estadio FROM equipo LIMIT
    ↪ 10"
    cursor.execute(consulta)
    datos=cursor.fetchall()
    for equipo in datos:
        print(equipo)
```

```
('APO', 'Melbourn Diamondbacks', 1997, 'NL', 'Chase Field')
('ATT', 'Atlanta Braves', 2002, 'NL', 'Turner Field')
('ALT', 'Altoona Mountain City', 1884, 'UA', None)
('ANA', 'Anaheim Angels', 2004, 'AL', None)
('ARI', 'Arizona Diamondbacks', 2021, 'NL', 'Chase Field')
('ATL', 'Atlanta Braves', 2021, 'NL', 'SunTrust Park')
('BAL', 'Baltimore Orioles', 2021, 'AL', 'Oriole Park at Camden Yards')
('BFN', 'Buffalo Bisons', 1885, 'NL', 'Olympic Park I')
('BFP', 'Buffalo Bisons', 1890, 'PL', None)
('BL1', 'Baltimore Canaries', 1874, 'NA', 'Newington Park')
```

Seleccionar los 10 primeros estadios insertados de un estado en específico *Se debe ingresar el indicativo del estado*

```
[11]: ## cambiar este dato por el estado a consultar
ESTADO = 'NY'

query = f"SELECT nombre FROM estadio WHERE estado='{ESTADO}' AND pais = 'US'␣
↳LIMIT 10"
cursor.execute(query)
datos=cursor.fetchall()

for estadio in datos:
    print(estadio)
```

```
('Riverside Park',)
('Riverside Grounds',)
('Olympic Park I',)
('Olympic Park II',)
('International Fair Association Grounds',)
('Maple Avenue Driving Park',)
('Windsor Beach',)
('Long Island Grounds',)
('Union Grounds',)
('Capitoline Grounds',)
```

Seleccionar todos los premios ganados por un jugador *Se debe ingresar el ID del jugador*

```
[12]: ## cambiar este dato por el jugador a consultar
JUGADOR = 'youngcy01'

query = f"SELECT * FROM jugadorpremio WHERE jugador='{JUGADOR}'"
cursor.execute(query)
datos=cursor.fetchall()

for premio in datos:
    print(premio)
```

```
('youngcy01', 'Pitching Triple Crown', 1901, 'AL  ')
('youngcy01', 'Baseball Magazine All-Star', 1908, 'AL  ')
('youngcy01', 'Baseball Magazine All-Star', 1908, 'ML  ')
```

3.2 Consultas con JOIN

Universidades en las cuales ha estado presente un jugador y su año correspondiente

```
[13]: query="""
SELECT
jugadorid, universidad, anyo
FROM jugador
JOIN jugadoruniversidad ON jugador.jugadorid=jugadoruniversidad.jugador
LIMIT 15
```

```

"""
cursor.execute(query)
datos=cursor.fetchall()

for data in datos:
    print(data)

```

```

('jmerliee12', 'chirstee', 2000)
('apashaml34', 'chirstee', 1997)
('aardsda01', 'pennst', 2001)
('aardsda01', 'rice', 2002)
('aardsda01', 'rice', 2003)
('abadan01', 'gamiddl', 1992)
('abadan01', 'gamiddl', 1993)
('abbeybe01', 'vermont', 1889)
('abbeybe01', 'vermont', 1890)
('abbeybe01', 'vermont', 1891)
('abbeybe01', 'vermont', 1892)
('abbotje01', 'kentucky', 1991)
('abbotje01', 'kentucky', 1992)
('abbotje01', 'kentucky', 1994)
('abbotji01', 'michigan', 1986)

```

Esta consulta devuelve la información del estadio que más publico ha albergado para un equipo en un lapso de tiempo

```

[14]: query="""
select estadioid, nombre,
estadiopartidos.anyo, estadiopartidos.publico,
estadiopartidos.primerpartido, estadiopartidos.ultimopartido
from estadio
join estadiopartidos on estadiopartidos.estadio = estadio.estadioid
where publico = (select MAX(publico) from estadiopartidos)
"""

cursor.execute(query)
datos=cursor.fetchall()

print('EstadioID', 'Nombre', 'año', 'publico', 'primerPartido',
      ↪ 'ultimoPartido', sep=' - ')
for data in datos:
    print(data)

```

```

EstadioID - Nombre - año - publico - primerPartido - ultimoPartido
('DEN01', 'Mile High Stadium', 1993, 4483203, '1993-04-09', '1993-09-26')

```

3.3 Consultas con función de agregación

En esta consulta se utiliza la clausula group by y sum para retornar la suma de todos los sueldos de los jugadores del equipo en un año en específico.

```
[15]: # este parametro se puede cambiar
      anyo=1995

      query=f"""
      select equipo, anyo, sum(sueldo) from sueldo
      where anyo={anyo}
      group by equipo, anyo ;
      """

      cursor.execute(query)
      datos=cursor.fetchall()
      for data in datos:
          print(data)
```

```
('TOR', 1995, 50590000)
('SEA', 1995, 36481311)
('CHA', 1995, 46961282)
('CLE', 1995, 37937835)
('SDN', 1995, 26382334)
('COL', 1995, 34154717)
('BOS', 1995, 32455518)
('PHI', 1995, 30555945)
('CIN', 1995, 43144670)
('MIN', 1995, 25410500)
('NYN', 1995, 27674992)
('CHN', 1995, 29505834)
('SLN', 1995, 37101000)
('LAN', 1995, 39273201)
('DET', 1995, 37044168)
('OAK', 1995, 37739225)
('FLO', 1995, 24515781)
('KCA', 1995, 29532834)
('NYA', 1995, 48874851)
('CAL', 1995, 31223171)
('HOU', 1995, 34169834)
('MON', 1995, 12364000)
('ML4', 1995, 17798825)
('BAL', 1995, 43942521)
('ATL', 1995, 47235445)
('SFN', 1995, 36462777)
('TEX', 1995, 34581451)
('PIT', 1995, 18355345)
```

mostrar el promedio de los sueldos que ha tenido un jugador en todos los equipos en los que ha participado

```
[17]: query=f"""
select jugador, equipo, AVG(sueldo)
from sueldo
group by jugador, equipo
order by jugador
LIMIT 10
"""

cursor.execute(query)
datos=cursor.fetchall()
print('jugador', 'equipo', 'promedio sueldo', sep=' - ')
for data in datos:
    print(data)
```

```
jugador - equipo - promedio sueldo
('aardsda01', 'SFN', Decimal('300000.000000000000'))
('aardsda01', 'CHA', Decimal('387500.000000000000'))
('aardsda01', 'BOS', Decimal('403250.000000000000'))
('aardsda01', 'SEA', Decimal('2556333.333333333333'))
('aardsda01', 'NYA', Decimal('500000.000000000000'))
('aasedo01', 'BAL', Decimal('633333.333333333333'))
('aasedo01', 'NYN', Decimal('400000.000000000000'))
('abadan01', 'CIN', Decimal('327000.000000000000'))
('abadfe01', 'HOU', Decimal('451500.000000000000'))
('abadfe01', 'OAK', Decimal('806700.000000000000'))
```

3.4 3.4 Subconsultas

Esta subconsulta devuelve todos los equipos en los que han estado los jugadores, adicional a eso se puede filtrar por un jugador en específico si se desea.

```
[70]: # ingresar el id del jugador si se desea consultar individual EJ: 'abbated01'
jugadorID=None

query = f"""
select distinct t.jugadorid, t.nombredeportivo, t.equipo
from (
    select
        jugador.jugadorid, jugador.nombredeportivo,
        jugadorposiciondetalle.equipo
    from jugador
    join jugadorposiciondetalle on jugadorposiciondetalle.jugador = jugador.
    ↪jugadorid
    {f"where jugador.jugadorid='{jugadorID}'" if jugadorID else ''}
) as t order by t.jugadorid
"""

cursor.execute(query)
datos=cursor.fetchall()
```

```
for data in datos:
    print(data)
```

```
('abbated01', 'Edward James', 'BSN')
('abbated01', 'Edward James', 'PHI')
('abbated01', 'Edward James', 'PIT')
```

Esta subconsulta devuelve la información de los estadios en el que más partidos se han realizado

```
[96]: query = """
SELECT estadio.estadioid, estadio.nombre, anyo, partidosjugados
FROM estadiopartidos
JOIN estadio on estadio.estadioid = estadiopartidos.estadio
WHERE partidosjugados = (select MAX(partidosjugados) from estadiopartidos);
"""

cursor.execute(query)
datos=cursor.fetchall()

print('estadioID', 'nombre', 'año', 'partido jugados', sep=' - ')
for data in datos:
    print(data)
```

```
estadioID - nombre - año - partido jugados
('PIT05', 'Exposition Park III', 1892, 89)
('CHI08', 'West Side Grounds', 1898, 89)
('CIN05', 'League Park II', 1898, 89)
```

3.5 4. Actualizaciones a base de datos

Actualizaciones basicas de registros

```
[18]: # Actualización del alias de un estadio
update = """UPDATE estadio SET ciudad = 'Valencia', alias = 'Stadium Columbia'
↳WHERE estadioid = 'ALT01'"""

cursor.execute(update)
print('Actualización exitosa del estadio')

# Actualizar las universidades que pertenezcan al estado de OH ya que este se
↳fucionó con CA
update = """UPDATE universidad SET estado = 'CA' WHERE estado = 'OH'"""

cursor.execute(update)
print('Actualización exitosa de las universidades')
```

Actualización exitosa del estadio

Actualización exitosa de las universidades

Este código sql cambia el nombre de una columna de la tabla

```
[19]: update = """ALTER TABLE batearestadistica RENAME COLUMN G TO num_golpes;"""

cursor.execute(update)
print('Actualización exitosa de la columna de la tabla batearestadistica')
```

Actualización exitosa de la columna de la tabla batearestadistica

Cambiar el tipo de dato de una columna de la tabla

```
[20]: # ALTERAR UN TIPO DE DATO DE UNA TABLA
update = """ALTER TABLE sueldo ALTER COLUMN liga SET DATA TYPE VARCHAR(2);"""

cursor.execute(update)
print('Actualización exitosa de la tabla sueldo')
```

Actualización exitosa de la tabla sueldo

3.6 5. Borrados a base de datos

Borrar todas las estadísticas de bateo de un jugador en específico

```
[27]: jugador = 'oberkke01'

delete = f"""delete from campoestadistica where jugador = '{jugador}'"""

cursor.execute(delete)
print(f'Borrado exitoso de las estadísticas de campo del jugador {jugador}')
```

Borrado exitoso de las estadísticas de campo del jugador oberkke01

Borrado de una columna de la tabla de estadísticas de bateo

```
[22]: delete = f"""ALTER TABLE batearestadistica DROP COLUMN sh"""

cursor.execute(delete)
print(f'Borrado exitoso de la columna sh de la tabla batearestadistica')
```

Borrado exitoso de la columna sh de la tabla batearestadistica

Eliminar la tabla lanzarestadistica

```
[28]: delete = f"""DROP TABLE lanzarestadistica"""

cursor.execute(delete)
print(f'Borrado exitoso de la tabla lanzarestadistica')
```

Borrado exitoso de la tabla lanzarestadistica

3.7 CONCLUSIONES

1. En base a la realización de este ejercicio, se pudo comprobar que el rol de un analista de datos es muy importante en una organización, ya que es éste quien con sus conocimientos y en base a los datos proporcionados, puede mostrar análisis detallados sobre la compañía, que le ayudarán a la organización a administrar de mejor manera todos sus recursos.
2. Es muy común encontrar en la mayoría de bases de datos *data sucia*, mucha de esta se debe a errores humanos, cambios de requisitos entre otros. De este ejercicio identificamos que un buen diseño de la base de datos (tablas, tipos de datos, normalización, etc ...) son esenciales para disminuir la tasa de inserción de *data sucia* que no brinda información veraz sobre el proceso que esta representa.