

# Charla de Proyecto (Grupo 36)

## Powerlifting-Database Application

Francisco Maldonado, Camilo Urzúa, Eduardo Reyes

Departamento de Ciencias de la Computación  
Universidad de Chile

22 de junio de 2022



1 Carga de Datos

2 Consultas

3 Optimización

4 La Aplicación

# Datos usados



DANB · UPDATED 3 YEARS AGO



55

New Notebook

Download (9 MiB)



## powerlifting-database

An unchanging copy of the Powerlifting Database



openpowerlifting.csv (30.24 MiB)



Detail

Compact

Column

10 of 17 columns



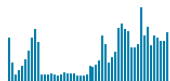
MeetID



Name



Sex



0 8481

136687  
unique values

M

77%

F

23%

### Data Explorer

Version 1 (30.86 MiB)

meets.csv

openpowerlifting.csv

Figura: powerlifting-database de Kaggle [1]

# Estadísticas de los Datos

Los datos correspondían a dos archivos en formato csv: *meets.csv*, *openpowerlifting.csv*. Relacionados por el atributo *MeetID*, presente en ambos.

Inicialmente el archivo *openpowerlifting.csv* contaba con 386414 filas (datos) y 17 columnas (atributos), y el archivo *meets.csv* contaba con 8482 filas (datos) y 8 columnas (atributos). En ambos archivos se pueden encontrar diversos tipos de datos: *string*, *int*, *float*, *date*...

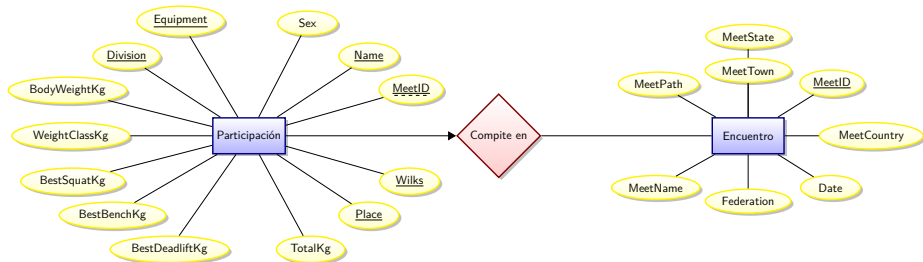
En una primera revisión llama la atención en el archivo *openpowerlifting.csv* la gran cantidad de nulos presentes y la presencia de 545 datos duplicados.

# Proceso de Limpieza y Carga

- Eliminación de columnas problemáticas
- Eliminación de datos corruptos
- Limpieza de nulos
- Carga de datos

Tras la limpieza el archivo *openpowerlifting.csv* cuenta con 269231 filas y 13 columnas.

# Esquema Relacional

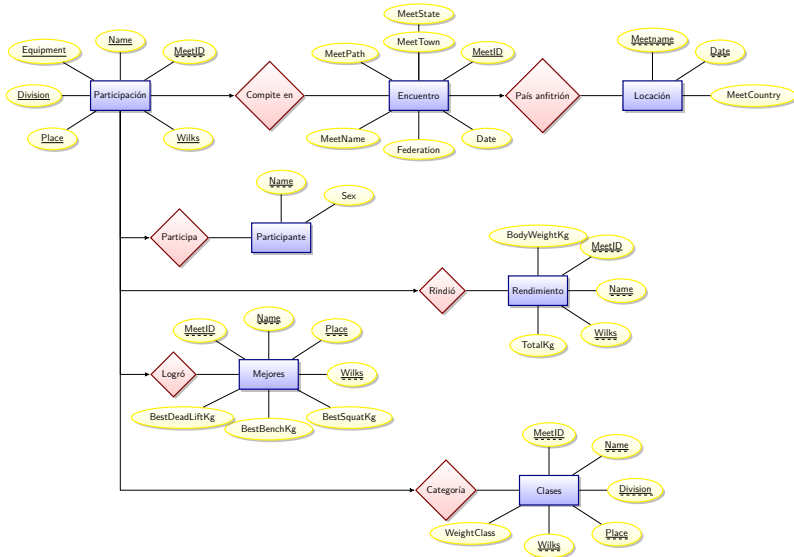


# Proceso de Normalización

## Dependencias Funcionales Problemáticas:

- $\{meetname, date\} \longrightarrow \{meetcountry\}$
- $\{name\} \longrightarrow \{sex\}$
- $\{meetid, name, wilks\} \longrightarrow \{bodyweight, totalkg\}$
- $\{meetid, name, place, wilks\} \longrightarrow \{bestsquatkg, bestbechkg, bestdeadliftkg\}$
- $\{meetid, name, division, place, wilks\} \longrightarrow \{weightclasskg\}$

# Esquema Relacional BCNF



<sup>1</sup>Forma Normal de Boyce-Codd (BCNF): Satisface 1NF y para cada:  $X \rightarrow Y$ ,  $X$  es una súper llave o  $Y \subseteq X$ .



1 Carga de Datos

2 Consultas

3 Optimización

4 La Aplicación

# Consulta 1

- *Todos los encuentros en los que una persona dada haya participado*

```
SELECT federation, mee.date, mee.meetState, mee.meetName,  
↪ mee.meetTown, venue.meetcountry  
FROM venue, (  
    SELECT federation, date, meetState, meetName, meetTown  
    FROM meet, (  
        SELECT DISTINCT meetID  
        FROM participations  
        WHERE name=:input  
    ) AS inpMeets  
    WHERE meet.meetID = inpMeets.meetID  
    ) AS mee  
WHERE venue.date = mee.date  
AND venue.meetname = mee.meetName
```

## Consulta 2

- *Número de personas que han levantado más de una cantidad dada (en Kg) de peso muerto*

```
SELECT COUNT(DISTINCT name)
FROM powerlift.bests
WHERE bestDeadliftKg > :input
```

## Consulta 3

- *Promedio peso corporal según sexo en Kg (M o F)*

```
SELECT AVG(bodyweightKg)
FROM participant, performance
WHERE participant.name = performance.name
AND participant.sexo=:input
```

## Consulta 4

- *Cantidad de clases de peso por equipamiento en una división dada*

```
SELECT equipment, COUNT(DISTINCT weightClassKg)
FROM (
    SELECT equipment, class.division, weightClassKg
    FROM participations AS par, class
    WHERE par.meetID = class.meetID
    AND par.name = class.name
    AND par.division = class.division
    AND par.place = class.place
    AND par.wilks = class.wilks
) AS edw
WHERE division=:input
GROUP BY equipment
```

1 Carga de Datos

2 Consultas

3 Optimización

4 La Aplicación

# Optimización: Índices

- **Consulta 1:** Dado que en **venue** los atributos *date* y *meetname* son llave, ya tienen índices asociados, lo mismo para *meetid* en la tabla **meet**. Asociamos índices *HASH* (útiles para búsquedas con igualdades) a *date* y *meetname* en la tabla **meet**, para cuando se hagan consultas que la relacionen con la tabla **venue**.
- **Consulta 2:** Se asoció un índice *ÁRBOL B+* (útiles para búsquedas entre rangos) al atributo *bestdeadweightlift* en la tabla **bests**, para mejorar el tiempo de esta consulta.
- **Consultas 3 y 4:** Dado que en estas consultas sólo se referencian atributos llave, no hace falta crear más índices.

# Optimización: Vistas

Se crearon las siguientes vistas que permitieron no sólo optimizar las consultas, sino también simplificar su código.

- 1 Vista 1. `cpltMeetView`
- 2 Vista 2. `avgBWbyGView`
- 3 Vista 3. `edwView`



# Vista 1

- Une **meetCountry** de **venue**, **name** de **participations** y **federation**, **date**, **meetName**, **meetState** y **meetTown** de **meet**, para en la consulta solo tener que buscar por **name**.

```
CREATE VIEW cpltMeetView AS
SELECT parMeet.name, federation, meet.date, meet.meetName,
↪ meetState, meetTown, meetCountry
FROM meet, venue, (
    SELECT DISTINCT meetID, name
    FROM participations
) AS parMeet
WHERE venue.date = meet.date
AND venue.meetname = meet.meetName
AND parMeet.meetID = meet.meetID
```

## Vista 2

- Calcula el peso promedio de participantes según su sexo.

```
CREATE VIEW avgBWbyGView AS
SELECT sexo, AVG(promedio) AS promByGender
FROM (
    SELECT participant.name, sexo, AVG(bodyweightKg) AS
    ↪ promedio
    FROM participant, performance
    WHERE participant.name = performance.name
    GROUP BY participant.name
) AS connect
GROUP BY sexo
```

## Vista 3

- Cuenta cuántas clases de peso distintas hay por equipamiento en cada división.

```
CREATE VIEW edwView AS
SELECT class.division, equipment, COUNT(DISTINCT
    ↪ weightclasskg) AS conteoWeight
FROM participations AS par, class
WHERE par.meetID = class.meetID
AND par.name = class.name
AND par.division = class.division
AND par.place = class.place
AND par.wilks = class.wilks
GROUP BY (class.division, equipment)
```

# Optimización: Nuevas Consultas (1, 3 y 4)

Ya creadas las vistas se modifican y simplifican las consultas 1, 3 y 4.

- Nueva Consulta 1

```
SELECT federation, date, meetName, meetState, meetTown,  
↪ meetCountry  
FROM powerlift.cpltMeetView  
WHERE name=:input
```

- Nueva Consulta 3

```
SELECT promByGender  
FROM powerlift.avgBWbyGView  
WHERE sexo=:input
```

- Nueva Consulta 4

```
SELECT equipment, conteoWeight  
FROM powerlift.edwView  
WHERE division=:input
```

1 Carga de Datos

2 Consultas

3 Optimización

4 La Aplicación

# Implementación de la aplicación

## Recursos

Gran parte de los recursos de software e instrucciones disponibles en la Wiki de proyectos [2]



Figura: CSS



Figura: HTML



Figura: PHP



Figura: Apache



Figura: sqlmap



Figura: Python

# Medidas de seguridad

## ① Consultas precompiladas

- `$pdo = new PDO(...);`
- `$stmt = $pdo->prepare(...);`
- `$stmt->execute(...);`

Gracias a *PDO*<sup>2</sup> y las consultas precompiladas, no se exponen datos ni se permiten inyecciones así como tampoco se permiten rellenos de credenciales.

## ② *webuser*

- `CREATE USER webuser WITH PASSWORD 'contrasena';`
- `GRANT USAGE ON SCHEMA proyecto TO webuser;`
- ...
- `GRANT SELECT ON proyecto.tabla TO webuser;`

---

<sup>2</sup><https://phpdelusions.net/pdo>

# Medidas de seguridad: *sqlmap*

Gracias a *python* y *sqlmap* podemos hacers tests para comprobar la seguridad exitosa contra inyecciones.

```
[22:30:27] [WARNING] heuristic (basic) test shows that GET parameter 'input1' might not be injectable
[22:30:27] [INFO] testing for SQL injection on GET parameter 'input1'
[22:30:27] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[22:30:27] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[22:30:27] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[22:30:28] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[22:30:28] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[22:30:28] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[22:30:29] [INFO] testing 'Generic inline queries'
[22:30:29] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[22:30:29] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[22:30:29] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[22:30:30] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[22:30:30] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[22:30:30] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[22:30:31] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] Y
[22:30:33] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[22:30:34] [WARNING] GET parameter 'input1' does not seem to be injectable
[22:30:34] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'

[*] ending @ 22:30:34 /2022-06-21/
cc3201@cc3201-36:~$
```

Figura: *sqlmap*



# Breve demostración

Acá ponemos un video

# Lecciones aprendidas

## ① Limpieza de datos

- La relevancia de investigar y limpiar los datos de la fuente para obtener un *dataset* coherente y utilizable.
- A veces es necesario perder información para poder desarrollar una base funcional.

## ② Seguridad

- La importancia de la seguridad de la aplicación web contra inyecciones.
- Uso de *sqlmap* <sup>3</sup>

## ③ Dificultad

- El proyecto puede ser considerado difícil desde un punto de vista de cantidad de conocimientos requeridos. Sin embargo, estos conocimientos se pueden aprender durante el desarrollo.
- Aprendizaje de sintaxis de diferentes lenguajes.



## ④ Desarrollo

- Desarrollo de una aplicación web simple.
- Trabajo en un proyecto grupal y su presentación.

---

<sup>3</sup><https://sqlmap.org/>

# References

-  D. BECKER – “Powerlifting-database. an unchanging copy of the powerlifting database”, 2022, Última vez visitado 12 de junio de 2022.
-  A. HOGAN – “Wiki de proyectos”, 2022.