



Base de Datos de Mundiales de Fútbol

Universidad Católica "Nuestra Señora de la Asunción"

Facultad de Ciencias y Tecnologías

Ingeniería Informática

Materia: Base de Datos

Integrantes:

Y14153 - Christian Arce

Y11237 - Camila Do Rego Barros

I. ESPECIFICACIÓN DEL TRABAJO

Diseñar una base de datos con toda la información relevante de un mundial de fútbol. Este sistema permite acceder a los datos de *todos los partidos*: los equipos que participaron en él y el resultado; y de *los jugadores*: la estadística en cada partido en el que participó (la posición en la que jugó, los goles y faltas que tuvo, etc).

II. DESCRIPCIÓN DE LOS CHALLENGES

II-A. Challenge I:

El objetivo de este challenge fue realizar el modelado de la Base de Datos para un Mundial de Fútbol. Para esto, identificamos las entidades principales, que son: País, Estadio, Mundial, Grupo, Equipo, Jugador, Partido, Estadística, y sus principales atributos, además se establecieron las relaciones entre estas entidades. También se definieron y especificaron los Reportes. Nuestra entrega del [Challenge1](#).

II-B. Challenge II:

En este challenge se identificaron las tablas, atributos y relaciones en base a las especificaciones presentadas en el challenge anterior. Como resultado se crearon las siguientes tablas:

- País, tiene los siguientes atributos: id_país, nombre y cantidad de estadios. Un país tiene muchos estadios y puede ser sede de varios Mundiales.
- Estadio, tiene los siguientes atributos: id estadio, id país, nombre del estadio, ciudad donde se encuentra y

capacidad de personas. Un Estadio está ubicado en un País y en él se pueden jugar varios Partidos.

- Mundial tiene los siguientes atributos: id mundial, año, id país donde se juega. Un Mundial se juega en un País, tiene ocho Grupos y tiene muchos Partidos.
- Un Grupo tiene los siguientes atributos: id grupo, id mundial y nombre del grupo. Un Grupo tiene 4 equipos y pertenece a un Mundial específico.
- Un Equipo tiene los siguientes atributos: id equipo, país, id grupo. Un Equipo puede pertenecer sólo a un Grupo, puede jugar varios Partidos y debe tener como mínimo 14 jugadores y máximo 26.
- Un Partido tiene los siguientes atributos: id partido, id mundial, id estadio, fecha, id equipo local, id equipo visitante, valor fase, resultado, ganador. En un partido solamente pueden jugar dos equipos y 22 jugadores.
- Un Jugador tiene los siguientes atributos: id jugador, nombre, numero camiseta, fecha nacimiento. Un Jugador tiene una Estadística por Partido y puede pertenecer a varios equipos (diferentes años).
- Jugador X Equipo tiene los siguientes atributos: id equipo, id jugador
- La Estadística de un jugador tendrá los siguientes atributos: id estadística, id jugador, id partido, id mundial, posición en la que jugó, goles, atajadas (si es arquero), penales atajados (si es arquero), asistencias, faltas, tarjetas amarillas y rojas.

Nuestra entrega del [Challenge2](#).

II-C. Challenge III:

En este challenge se presentaron los diagramas lógicos y físicos de acuerdo a las identificaciones de las tablas, atributos y relaciones presentados en el Challenge anterior. También se realizaron modificaciones en algunas relaciones y se agregaron las siguientes entidades: Persona: id persona, nombre, CI, fecha nacimiento, nacionalidad Árbitro: id persona Entrenador: id persona La entidad Persona está relacionada con Jugador, Árbitro y Entrenador. La entidad Árbitro está relacionada con Partido. Un árbitro puede arbitrar varios partidos y un Partido tiene un solo Árbitro. Y por último, Entrenador se relaciona con Equipo. Un equipo tiene un solo entrenador. También se presentó la declaración de privacidad, con el objetivo de informar sobre los datos que serán públicos en la base de datos.

Nuestra entrega del [Challenge3](#).

II-D. Challenge IV:

Para este challenge nos enfocamos principalmente en la normalización de la base de datos, nos aseguramos de que esta cumpla con los principios de normalización hasta la tercera forma, eliminando redundancias y optimizando la estructura de las tablas. Algunos cambios que se realizaron fueron:

- La eliminación de la relación entre Estadística y Mundial.
- Se agregó la relación entre Equipo y País.
- Se agregó la entidad Fase y su relación con Partido. Fase tiene los atributos: id fase, nombre fase.
- Se agregó la entidad Resultado y su relación con Partido. Resultado tiene los atributos: id resultado, id partido y id equipo ganador.
- Se modificó la relación Entrenador-Equipo a una relación de muchos a muchos y se creó la tabla Entrenador X Equipo.

Nuestra entrega del [Challenge4](#).

II-E. Challenge V:

Para este challenge se debía implementar la parte física de la base de datos, realizando una carga de datos ficticios y desarrollando tres funciones de los reportes definidos en el Challenge I, y tres triggers que apliquen alguna restricción en la carga de datos o que ayuden a sincronizar los datos. Reportes:

- REPORTE PODIUM: se obtiene el equipo ganador, subcampeón y tercer puesto de un mundial específico.
- REPORTE TOP JUGADORES: se obtiene un ranking de los 10 jugadores que más mundiales jugaron.
- REPORTE MVP, MEJOR ARQUERO, MEJOR DEFENSA: se obtienen los jugadores que salieron como MVP, Mejor Arquero y Mejor Defensa en un mundial específico. Para ello, en cada partido, se le asigna un 'score' a cada jugador de acuerdo a su desempeño.

Triggers:

- TRIGGER QUE LIMITA LA CANTIDAD DE JUGADORES POR EQUIPO: un equipo puede tener como máximo 26 jugadores.
- TRIGGER PARA CALCULAR EL GANADOR DE UN PARTIDO: la entidad Partido tiene un atributo cantidad jugadores que son todos los jugadores que participaron en ese partido (incluyendo los cambios). Una vez que se cargaron las Estadísticas de todos los jugadores, se calcula la cantidad de goles anotados por cada equipo para definir el equipo ganador y cargarlo en la tabla Resultado.
- TRIGGER QUE CALCULA EL SCORE DE UN JUGADOR EN UN PARTIDO ESPECÍFICO: cuando se carga las estadísticas de un jugador, automáticamente se calcula su score de acuerdo a su desempeño y se carga en la tabla Score.

Nuestra entrega del [Challenge5](#).

II-F. Challenge VI:

El objetivo de este Challenge era extender la muestra de datos a un mínimo de 500.000 registros, realizar un script que inserta datos aleatorios automáticamente. Nosotros realizamos este script en Python, utilizando la librería Faker. Además, debíamos desarrollar índices que mejoren la eficiencia de las consultas realizadas por los Reportes. También, debíamos demostrar esta eficiencia, mediante el EXPLAIN ANALYZE de PostgreSQL que permite analizar y comparar el rendimiento de las consultas antes y después de la implementación de los índices, generando cuadros comparativos. Los índices implementados son:

- CREATE INDEX idx_estadistica_id_partido ON estadistica (id_partido);
- CREATE INDEX idx_estadistica_penal_desempate ON estadistica (penal_desempate);
- CREATE INDEX idx_resultado_id_partido ON resultado (id_partido);
- CREATE INDEX idx_jugadorxequipo_id_equipo ON jugador_x_equipo(id_equipo);
- CREATE INDEX idx_partido_id_mundial ON partido(id_mundial);
- CREATE INDEX idx_estadistica_id_jugador ON estadistica (id_jugador);

Nuestra entrega del [Challenge6](#).

II-G. Challenge VII:

Para el Challenge final, debíamos ampliar la muestra de datos a 7 millones de registros en total y una de las tablas debía tener 5 millones de registros. Se debían terminar todos los reportes propuestos en el challenge 1 y demostrar la eficiencia de los índices definidos en el challenge 6. Pruebas realizadas:

- **Funcion obtener_mvp(id_mundial):** devuelve el reporte del mejor jugador, el mejor arquero y el mejor defensa de los últimos 5 mundiales a partir del id_mundial que se pasó como parámetro. Revisa las

tablas Resultado, Score, Estadística, Jugador_x_equipo
Antes de indexar:

```
in analyze select * from obtener_mvp(17) | Enter a SQL expression to filter results
QUERY PLAN
Function Scan on obtener_mvp (cost=0.25..10.25 rows=1000 width=108)
Planning Time: 0.043 ms
Execution Time: 29286.106 ms
```

Luego de indexar:

```
in analyze select * from obtener_mvp(17) | Enter a SQL expression to filter results
QUERY PLAN
Function Scan on obtener_mvp (cost=0.25..10.25 rows=1000 width=108)
Planning Time: 0.034 ms
Execution Time: 17252.106 ms
```

- **Función reporte_podium(id_mundial):** Devuelve el reporte de los equipos ganadores del primer puesto, el segundo puesto y el tercer puesto del mundial con la id_mundial que se paso como parámetro. Revisa las tablas Resultado, Partido y Equipo (para imprimir los nombres de los equipos)
Antes de indexar:

```
in analyze select * from reporte_podium | Enter a SQL expression to filter results
QUERY PLAN
Function Scan on reporte_podium (cost=0.25..10.25 rows=1000 width=68)
Planning Time: 0.031 ms
Execution Time: 346.643 ms
```

Luego de indexar:

```
QUERY PLAN
Function Scan on reporte_podium (cost=0.25..10.25 rows=1000 width=68)
Planning Time: 0.028 ms
Execution Time: 5.103 ms
```

- **Función reporte_top_jugadores():** Devuelve el reporte de los top 10 jugadores que más mundiales jugaron y la cantidad de posiciones en las que jugó. Revisa las tablas Partido y Estadística
3 Antes de indexar:

```
QUERY PLAN
Function Scan on reporte_top_jugadores (cost=0.25..10.25 rows=1000 width=60)
Planning Time: 0.031 ms
Execution Time: 12201.743 ms
```

Luego de indexar:

```
in analyze select * from reporte_top_jugadores | Enter a SQL expression to filter results
QUERY PLAN
Function Scan on reporte_top_jugadores (cost=0.25..10.25 rows=1000 width=60)
Planning Time: 0.031 ms
Execution Time: 11766.559 ms
```

- **Función reporte_tabla_grupos(id_mundial_var):** Devuelve la tabla de puntuación de los equipos de cada grupo de un mundial específico. Revisa las tablas Grupo, Equipo, Pais, Mundial y Resultado
Antes de indexar:

```
in analyze select * from reporte_tabla_grupo | Enter a SQL expression to filter results
QUERY PLAN
Function Scan on reporte_tabla_grupos (cost=0.25..10.25 rows=1000 width=80)
Planning Time: 0.032 ms
Execution Time: 256.417 ms
```

Luego de indexar:

```
in analyze select * from reporte_tabla_grupo | Enter a SQL expression to filter results
QUERY PLAN
Function Scan on reporte_tabla_grupos (cost=0.25..10.25 rows=1000 width=80)
Planning Time: 0.029 ms
Execution Time: 189.816 ms
```

- **Función reporte_partido(id_partido_N):** Devuelve la estadística de un partido específico. Revisa las tablas Estadística, Jugador_x_Equipo y Equipo
Antes de indexar:

```
in analyze select * from reporte_partido | Enter a SQL expression to filter results
QUERY PLAN
Function Scan on reporte_partido (cost=0.25..10.25 rows=1000 width=256)
Planning Time: 0.051 ms
Execution Time: 7755.704 ms
```

Luego de indexar:

```
in analyze select * from reporte_partido | Enter a SQL expression to filter results
QUERY PLAN
Function Scan on reporte_partido (cost=0.25..10.25 rows=1000 width=256)
Planning Time: 0.035 ms
Execution Time: 5.710 ms
```

- **Función estadísticas_jugador_mundial(id_mundial_var, id_jugador_var):** Devuelve la estadística de un jugador específico en un mundial específico, considerando todos los partidos en los que participó en dicho mundial. Revisa las tablas Estadística, Partido, Jugador y Persona
Antes de indexar:

```
in analyze select * from estadisticas_jugador_mundial | Enter a SQL expression to filter results (use C)
QUERY PLAN
Function Scan on estadisticas_jugador_mundial (cost=0.25..10.25 rows=1000 width=120)
Planning Time: 0.033 ms
Execution Time: 535.628 ms
```

Luego de indexar:

```
QUERY PLAN
Function Scan on estadisticas_jugador_mundial (cost=0.25..10.25 rows=1000 width=120)
Planning Time: 0.033 ms
Execution Time: 1.167 ms
```

- **Función estadísticas_jugador_total(id_jugador_var):**

Devuelve la estadística de un jugador específico de todos los mundiales en los que participó. Revisa las tablas Estadística, Partido, Jugador, Persona y Mundial. Antes de indexar:

Enter a SQL expression to filter results (use `WHERE`)

SQL QUERY PLAN

Function Scan on estadísticas_jugador_total (cost=0.25..10.25 rows=1000 width=124)

Planning Time: 0.043 ms

Execution Time: 537.241 ms

Luego de indexar:

Enter a SQL expression to filter results (use `SELECT`)

QUERY PLAN

Function Scan on estadísticas_jugador_total (cost=0.25..10.25 rows=1000 width=124)

Planning Time: 0.042 ms

Execution Time: 2.508 ms

- **Función reporte_goleador(id_mundial N):** Devuelve

la información del jugador que más goles logró en un mundial específico. Revisa las tablas Jugador, Persona, Partido y Estadística.

Antes de indexar:

```

Enter a SQL expression to filter rows
In analyze select * from reporte_goleador
QUERY PLAN
Function Scan on reporte_goleador (cost=0.25..10.25 rows=1000 width=52)
Planning Time: 0.041 ms
Execution Time: 637.029 ms

```

Luego de indexar:

Function Scan on reporte_goleador (cost=0.25..10.25 rows=1000 width=52)
Planning Time: 0.031 ms
Execution Time: 13.708 ms

El diseño final de nuestra base de datos es el siguiente:

